伟联科技 www.witlinc.cn



WiSCADA 2.0 工业组态软件



北京伟联科技有限公司

本手册中的信息可能在不事先声明的情况下被修改,对此北京伟联科技有限公司不承担责任。本手册所描述的软件是在授权或不扩散协议下完成的。本软件只能按合同规定的条款使用 或拷贝。在没有北京伟联科技有限公司书面许可的前提下,除购买者自己使用外,不得为任何 目的、使用任何方法(包括复印和录制在内的电子或机械手段)对本手册的任何部分进行复制 或传播。

WiSCADA 2.0 工业组态软件 使用手册

Copyright 2016-2017 北京伟联科技有限公司。著作权所有。保留所有权利。

本文档可能涉及北京伟联科技有限公司的相关知识产权,除非得到北京伟联科技有限公司 的明确书面许可协议,本文档不得复制、转售或者用于其他商业用途。

技术支持热线: 010-58449900

E-mail: support@witlinc.cn

网址: http://www.witlinc.cn

目 录

_,	安装和使用入门	19
	1.1 关于 WiSCADA	19
	1.2 软件安装	19
	1.3 使用入门	20
	1.3.1 组态开发环境	20
<u> </u>	组态工程开发	23
	2.1 创建工程	23
	2.2 创建窗口	24
	2.2.1 工程画面浏览窗口创建	24
	2.2.2 命令创建	24
	2.3 创建 IO 设备通信	25
	2.3.1 工程数据库浏览窗口创建	25
	2.3.2 命令创建	26
	2.4 增加IO 通信变量	27
	2.4 增加IO 通信变量	27 27
	 2.4 增加IO 通信变量 2.4.1 增加变量 2.4.2 批量增加 	27
	 2.4 增加IO 通信变量 2.4.1 增加变量 2.4.2 批量增加 2.5 建立动画链接 	27 27 27 27
	 2.4 增加IO 通信变量 2.4.1 增加变量 2.4.2 批量增加 2.5 建立动画链接 2.6 调试与运行 	27 27 27 27 28 28
Ξ,	 2.4 增加IO 通信变量 2.4.1 增加变量 2.4.2 批量增加 2.5 建立动画链接 2.6 调试与运行 工程管理 	27 27 27 27 28 28 29 31
Ξ,	 2.4 增加IO 通信变量	27 27 27 28 29 31 31
三、	 2.4 增加IO 通信变量	27 27 27 28 29 31 31 31
==,	 2.4 增加IO 通信变量 2.4.1 增加变量 2.4.2 批量增加 2.5 建立动画链接 2.5 建立动画链接 2.6 调试与运行 工程管理 3.1 概述 3.2 新建工程 3.3 删除工程 	27 27 27 28 29 31 31 31 31
Ξ,	 2.4 增加IO 通信变量	27 27 27 28 29 31 31 31 31 31 33
	2.4 增加IO 通信变量 2.4.1 增加变量 2.4.2 批量增加 2.5 建立动画链接 2.5 建立动画链接 2.6 调试与运行 工程管理 3.1 概述 3.2 新建工程 3.3 删除工程 3.4 打开/关闭工程 3.4.1 起始背景面板中打开工程	27 27 28 29 31 31 31 31 33 33
Ξ	 2.4 增加IO 通信变量	27 27 27 28 29 31 31 31 31 31 33 34 34



	3.4.4 关闭工程	34
	3.5 当前默认工程	35
	3.6 工程打包	35
	3.7 工程加密	36
	3.8 下载工程	36
四、	工程浏览	38
	4.1 概述	38
	4.2 画面端	39
	4.3 数据库端	40
五、	画面布局与外观	41
	5.1 画面布局	41
	5.2 画面外观	42
六、	IO 设备通信	43
	6.1 概述	43
	6.2 新建 IO 设备通信	43
	6.2.1 从工程数据库浏览窗口创建	43
	6.2.2 通过命令创建	44
	6.3 配置IO 设备通信	44
	6.3.1 IO 通信工具栏	44
	6.3.2 IO 基本配置选项	45
	6.4 IO 通信通道	46
	6.4.1 串口通道	46
	6.4.2 TCP 通道	46
	6.4.3 UDP 通道	46
	6.4.4 GPRS_CDMA 通道	46
	6.4.5 动态修改通道	47
	6.5 DTU 与GPRS-CDMA 通道	47



	6.5.1 概述	. 47
	6.5.2 DTU 可以自定义心跳包	. 48
	6.5.3 DTU 不可自定义心跳包	. 48
6.	6 增加IO 通信变量	. 49
	6.6.1 增加变量	. 49
	6.6.2 批量增加	. 49
6.	7 IO 变量相关操作	. 50
6.	8 IO 通信变量选项	. 52
6.	9 IO 设备通信采集说明	. 53
	6.9.1 IO 采集包	. 53
	6.9.2 优先级	. 53
	6.9.3 诊断	. 54
	6.9.4 优化采集	. 54
6.	10 调试IO 设备通信	. 54
6.	11 Modbus 驱动	. 55
	6.11.1 概述	. 55
	6.11.2 Modbus 多站	. 56
	6.11.3 特殊函数	. 57
	6.11.4 特殊设备	. 57
6.	12 Modbus 从站	. 58
6.	13 OPC 驱动	. 58
	6.13.1 新建 OPC 驱动	. 58
	6.13.2 配置服务	. 59
	6.13.3 增加变量	. 59
	6.13.4 批量增加	. 60
	6.13.5 别名	. 61
	6.13.6 主要问题	. 61



	5.14 OPC 服务器6	1
	6.14.1 概述	1
	6.14.2 配置6	1
	6.14.3 服务器名和 CLSID 6	2
	6.14.4 主要问题 6	2
	6.15 万能协议驱动	3
	6.15.1 万能驱动原理6	3
	6.15.2 包及包域6	3
	6.15.3 常规包6	4
	6.15.4 写值包6	5
	6.15.5 配置驱动及协议 6	6
	6.15.6 脚本模式6	8
	6.15.7 其它说明	0
	5.16 SNMP 驱动	0
	5.17 IO 通信驱动开发包	1
七、	窗口与图形7	2
	7.1 概述	2
	7.2 新建窗口7	2
	7.2.1 从工程画面浏览窗口创建 7	2
	7.2.2 通过命令创建 7	3
	7.3 窗口分组7	4
	7.3.1 新增窗口组	4
	7.3.2 窗口组的窗口文件7	4
	7.4 窗口组态开发7	4
	7.4.1 属性设置7	5
	7.4.2 事件配置7	6
	7.4.3 其它操作 7	6



7.5 图形组态设计7	77
7.5.1 概述	77
7.5.2 新建图形	78
7.5.3 鼠标键盘操作7	78
7.5.4 旋转	79
7.5.5 倾斜	79
7.5.6 对齐	79
7.5.7 等大小	30
7.5.8 置前、置后	31
7.5.9 等间距 8	31
7.5.10 居中	32
7.5.11 镜像	33
7.5.12 锁住	33
7.5.13 组合 8	34
7.5.14 子对象	34
7.5.15 特有操作	35
7.6 图形属性	37
7.6.1 概述	37
7.6.2 通用属性	37
7.7 图形动画	38
7.7.1 概述	38
7.7.2 闪烁	38
7.7.3 填充	39
7.7.4 显隐) 0
7.7.5 移动) 0
7.7.6 缩放	€1
7.7.7 旋转	€2



7.7.8 倾斜
7.7.9 图像切换
7.8 图形事件
7.8.1 概述
7.8.2 响应动作
7.8.3 事件脚本
7.8.4 拖动特有响应
7.9 主要图形、控件 101
7.9.1 编辑框101
7.9.2 标签
7.9.3 按钮102
7.9.4 开关按钮103
7.9.5 单选框103
7.9.6 复选框104
7.9.7 列表框104
7.9.8 下拉框 105
7.9.9 棒图105
7.9.10 滑块
7.9.11 日期时间106
7.9.12 图像
7.9.13 GIF 动画108
7.9.14 流动
7.9.15 仪表
7.9.16 触发器
7.9.17 定时器110
7.9.18 报表浏览器111
7.9.19 趋势图



	7.9.20 百度地图11	1
	7.9.20.1 说明	1
	7.9.21 海康威视11	2
	7.10 ActiveX 控件11	4
	7.10.1 选择一个ActiveX 控件对象类11	5
	7.10.2 选择 ActiveX 窗口图形11	5
	7.10.3 配置 ActiveX	6
	7.10.4 通过JavaScript 调用11	6
	7.10.5 主要问题11	6
	7.11 定制图形(图形模板) 11	7
	7.11.1 说明11	7
	7.11.2 与"组合"图形的区别11	7
	7.11.3 创建11	7
	7.11.4 增加定制属性、动画、事件11	7
	7.11.5 引用定制图形11	8
	7.12 图形开发包	8
八、	三维窗口与模型11	9
	8.1 概述	9
	8.2 新建三维窗口 11	9
	8.2.1 从工程画面浏览窗口创建12	0
	8.2.2 通过命令创建 12	0
	8.3 三维窗口分组	1
	8.4 三维窗口组态开发 12	1
	8.4.1 属性设置12	2
	8.4.2 事件配置12	3
	8.4.3 其它操作12	3
	8.5 3D 工具栏12	3



8.6 三维场景漫游125
8.6.1 手动漫游 125
8.6.2 自动漫游126
8.7 三维模型组态设计 127
8.7.1 概述
8.7.2 新建模型127
8.7.3 鼠标、键盘操作129
8.7.4 移动
8.7.5 旋转
8.7.6 缩放
8.7.7 锁住132
8.7.8 组合
8.7.9 子对象134
8.7.10 特有操作135
8.7.11 方法命令
8.8 三维模型属性136
8.8.1 概述
8.8.2 通用属性136
8.9 三维模型动画
8.10 三维模型事件
8.11 3DS 模型138
8.11.1 导入3DS
8.11.2 调整为原来比例138
8.11.3 一些注意事项 139
8.11.4 方法命令
8.11.5 3DS 模型站点140
8.12 光源



	8.12.1 方向光源	141
	8.12.2 点光源	142
	8.12.3 聚光光源	142
九、	历史记录	143
	9.1 概述	143
	9.1.1 简介	143
	9.1.2 注意	143
	9.2 基本配置	143
	9.2.1 基本配置	143
	9.2.2 记录块	144
	9.2.3 强行写盘间隔	145
	9.2.4 延时开始记录	145
	9.2.5 按时开始记录	145
	9.3 历史库备份	146
	9.3.1 说明	146
	9.3.2 配置	146
	9.3.3 备份项目	146
	9.4 历史变量	148
	9.5 读取显示	149
	9.5.1 历史趋势图	149
	9.5.2 实时趋势图	149
	9.5.3 历史表格	150
	9.5.4 历史报表	150
	9.5.5 历史数据读取接口	151
+、	报警	152
	10.1 概述	152
	10.2 基本配置	152



10.3 报警变量153
10.3.1 配置
10.4 短信报警154
10.5 微信报警155
10.6 邮件报警158
10.7 读取显示
10.7.1 报警表格159
10.7.2 显示选项与条件159
10.8 报警触发
10.9 运行时动态设置 161
10.9.1 动态修改报警限值、类型等属性 161
十一、事件163
11.1 概述
11.2 配置
11.3 操作事件变量 163
11.4 读取显示
11.4.1 事件表格164
11.4.2 显示选项与内容165
十二、用户与安全区165
12.1 概述
12.2 安全区配置166
12.3 用户管理166
12.3.1 组态时用户管理166
12.3.2 运行时编辑用户167
12.4 用户登录 167
十三、配方
13.1 概述



13.2 创建与编辑	169
13.3 配方表格	170
十四、时间表	171
14.1 概述	171
14.2 创建与编辑	171
14.3 时间表表格	171
十五、网络与Web	173
15.1 概述	173
15.2 配置	174
15.3 远程浏览	176
15.3.1 远程浏览Windows 工程	176
15.3.2 完全版组态浏览器与控件	176
15.3.3 64 位机器	176
15.4 智能移动web	177
15.4.1 概述	177
15.4.2 配置	177
15.4.3 功能说明	178
15.4.4 如何在 PC 上调试	178
15.4.5 如何在智能移动上监控	178
15.4.6 一些说明	179
15.5 手机客户端监控	179
15.5.1 简介	179
15.5.2 配置	179
15.5.3 安卓APP 操作	180
15.5.4 苹果APP 操作	180
15.5.5 一些说明	181
15.6 P2P <u>국</u>	182



	15.6.1 基本概念	182
	15.6.2 总结	182
	15.6.3 云屏、云盒子	182
	15.6.4 配置	183
	15.7 互联网云监控	185
	15.7.1 互联网监控的方式	185
	15.7.2 互联网监控要点——提高互联网的通讯效率	185
	15.8 浏览指定窗口	187
	15.8.1 浏览指定窗口	187
	15.8.2 浏览指定三维窗口	187
	15.8.3 保留原来画面布局	188
	15.8.4 传入参数	188
	15.9 主要问题	188
	15.10 组态开发要点	189
	15.10.1 概述	189
	15.10.2 画面端与数据库端	190
	15.10.3 智能移动WEB 画面端	191
	15.10.4 画面端变量和数据库端变量	191
	15.10.5 尽量减少网络通信包	192
	15.10.6 数据库事务的处理	194
	15.10.7 数据库端触发画面端事件	194
	15.10.8 严禁画面端定时执行数据库端事务	195
	15.10.9 严重警告	195
十六	六、多语言	196
	16.1 概述	196
	16.2 设置	196
	16.3 多语言切换	197



16.3.1 组态开发时切换	
16.3.2 运行时切换	197
十七、JvavaScript 程序	
17.1 概述	
17.1.1 简介	
17.1.2 学习JavaScript	
17.1.3 文档对象模型(DOM)	
17.1.4 一些说明	199
17.2 全局脚本	200
17.3 JavaScript 编辑器	201
17.3.1 智能感知	
17.3.2 工具箱	202
17.3.3JavaScript 编辑器选项设置	202
17.4 窗口javascript	203
17.4.1 概述	203
17.4.2 对象模型	
17.4.3 连接变量	204
17.5 数据库jvavascript	206
17.5.1 概述	206
17.5.2 新建	207
17.5.3 分组	
17.5.4 全局变量与内存JS	209
17.5.5 脚本参数	
17.6 调用com 组件	
17.6.1 概述	212
17.6.2 调用Excel 组件	213
17.6.3 调用SQL 数据库组件	



17.6.3.3 连接字符串参数 218
十八、表达式
18.1 概述
18.1.1 语法 220
18.1.2 表达式对话框 220
18.2 函数
18.3 操作符 221
十九、报表
19.1 概述
19.2 创建报表
19.2.1 从工程数据库浏览窗口中创建 223
19.2.2 通过命令创建 223
19.3 报表分组 224
19.4 报表设计 225
19.4.1 报表工具栏 225
19.4.2 设置报表属性 225
19.4.3 设置单元格属性 226
19.5 单元格内容 227
19.5.1 文本
19.5.2 实时变量
19.5.3 历史时间、历史变量 227
19.5.4 变量历史值
19.5.5 SQL 字段/条件源字段 228
19.5.6 统计计算 228
19.5.7 表达式
19.5.8 序号 230
19.5.9 定制多行值 230



	19.6 历史报表	230
	19.6.1 手动历史时间	230
	19.6.2 自动历史时间	231
	19.6.3 历史变量	232
	19.6.4 历史时间与历史变量的对应	232
	19.7 数据库报表	232
	19.7.1 数据源	233
	19.7.2 SQL 字段	233
	19.7.3 条件源字段	234
	19.7.4 显示说明	234
	19.8 报表显示	234
	19.8.1 报表浏览器	234
	19.9 报表打印	235
	19.9.1 从报表浏览器中打印	235
	19.9.2 通过命令打印	235
	19.10 JS 定制数值	235
<u> </u>	-、数据库访问	237
	20.1 概述	237
	20.2 新建	237
	20.3 配置	238
	20.4 访问access	239
	20.5 访问SQL Server	240
	20.6 访问MySQL	241
	20.7 操作使用	242
	20.7.1 数据库报表	242
	20.7.2 数据库表格	242
	20.7.3 属性说明	243



	20.7.4 其它使用 2	43
二十一、	其他辅助模块2	44
21.1	概述2	44
21.2	定制变量2	44
	21.2.1 新建	44
	21.2.2 编辑	45
21.3	动作集2	45
	21.3.1 新建	46
	21.3.2 编辑	46
21.4	定时器2	47
	21.4.1 新建 2	47
	21.4.2 编辑 2	47
21.5	触发器2	48
	21.5.1 新建	48
	21.5.2 编辑 2	48
	21.5.3 触发方式	49
21.6	动态库2	49
	21.6.1 新建	50
	21.6.2 编辑 2	50
	21.6.3 指针类型说明 2	51
	21.6.4 严重警告	52
21.7	系统启动	52
21.8	模块开发包2	53
二十二、	很重要!常用的优化、技巧、问题 2	54
22.1	模块平板拖动	54
22.2	HMI 注意事项	54
	22.2.1 旋转、倾斜	54



22.2.2 图像	. 255
22.3 窗口刷新、定时器	. 255
22.3.1 窗口定时器	. 255
22.3.2 严重警告	. 256
22.3.3 数据库端定时器	. 256
22.3.4 其他说明	. 256
22.3.5 补充说明	. 256
22.4 自定义图库	. 257
22.4.1 SVG 图库	. 257
22.4.2 复制到图库中	. 257
22.5 网络Web 组态开发说明及要点	. 258
22.5.1 概述	. 258
22.5.2 画面端与数据库端	. 258
22.5.3 智能移动WEB 画面端	. 259
22.5.4 画面端变量和数据库端变量	. 259
22.5.5 尽量减少网络通信包	. 260
22.5.6 数据库事务的处理	. 262
22.5.7 数据库端触发画面端事件	. 262
22.5.8 严禁画面端定时执行数据库端事务	. 263
22.5.9 严重警告	. 263
22.6 互联网云监控要点	. 263
22.6.1 互联网监控的方式	. 263
22.6.2 互联网监控要点——提高互联网的通讯效率	. 264
22.7 IO 设备通信采集说明	. 265
22.7.1 IO 采集包	. 265
22.7.2 优先级	. 266
22.7.3 诊断	. 266



22.7.4 优化采集 266
2.2.8 变量位读写操作
22.8.1 通过"位变量"辅助功能模块 267
22.8.2 通过数据库定制变量的表达式 267
2.9 变量替换(窗口模板)267
22.9.1 简介
22.9.2 用法
22.9.3 使用要点
22.9.4 其它说明
22.9.5 严重警告
2.2.10 定制图形(图形模板)
22.10.1 说明
22.10.2 与"组合"图形的区别270
22.10.3 创建 270
22.10.4 增加定制属性、动画、事件 270
2.11 SQL 关系数据库



一、安装和使用入门

1.1 关于 WiSCADA

WiSCADA 是一个应用最新技术开发出来的一款技术领先的组态软件。 WiSCADA 使用全新的调度管理程序,超强实时数据库,网络访问一点即达的快捷。

运行 JavaScript 语言,纯WEB 发布,手机 APP,整合百度地图,兼容 GIS 组件。

最新一代的 WiSCADA 2.0 软件平台基于微内核、可伸缩、全插件、跨平台的 系统架构,提供国际顶级的、完全可定制的图形化界面,支持全景高清三维动态交 互,支持完全沉浸式的虚拟现实漫游。系统支持各类智能手机和平板电脑,支持 海量数据的深度挖掘和分析。系统提供从设备管理到大型指挥控制中心构建了完 整的物联网软件生态系统。

1.2 软件安装

下载安装

从北京伟联科技有限公司处获取安装文件。

1、下载 WiSCADA_NET. zip 文件;

2、解压;

- 3、双击 WiSCADA_NET. exe
- 4、下一步一一下一步一一下一步一一完成。
- 5、桌面上多出三个图标,安装完成。

注意:默认安装到 c 盘,如需更改安装目录,在安装过程中选择!



1.3 使用入门

1.3.1 组态开发环境

1.3.1.1 概述

组态开发环境是一个可视化、"傻瓜"式的集成开发环境。下图所示是开发 环境启动时的初始界面。

◎ ☆ 供 (P) 编辑 (R) 视图 (V)	田雅 (141)日安 (171)星丁	108) 菱苗	组态开发环境 与工 目 栏		_ = X
10日月日日 10日日 10日日 10日日 10日日 10日日 10日日 10日	(϶ϫϫϝ Ϳϭʹϼʹ≣ʹϷ᠂Ϳϣʹ <mark>ͺͺͺ</mark> ;ʹϞͺϨϡͿϣͿͺ ;;ͺͺ ͺͺϥϗϗͿͺϥ	● 中文 <mark> </mark>	<mark>≫</mark> ⊒ 223, ४४६२२२,
数据库		2	97 · · · · · · ·	4	工具箱 平 ×
工程的画面和 数据库浏览树	新建工程 删除工程 工程名称 • 机房监控 • 煤矿监控 • 新仪表测试 • 胶带机MII	日 ガ ガ ガ 北 程 う 辨率 1024x768 1280x800 1024x768 800x600	添加工程 运行工程 默认工程 工: 工程路径 C:\Documents and Settings\Adminis	程打包 工程描述 工程列表 旅帯机:mu	此組没用可用的控件。
23 回面 温数据库	工程调度系统	1280x768	В//用户测试\工程调度系统-最终\工		
属性栏 ♀ ×					
	<			>	
对象的属性、事 件、动画配置			工作区		
就绪					

图 1-1 组态开发环境

1.3.1.2 开发环境说明

菜单与工具栏:开发环境主要的命令操作。

工程的画面和数据库浏览:对整个打开工程的一个总体概貌入口。

工程管理:对所有组态编辑过的工程的统一管理。

工作区: 画面窗口、主要模块操作的区域。

工具箱: 窗口图形、控件的工具箱。

属性栏: 窗口图形、控件对象的属性、动画、事件的配置窗口



下图是打开一个组态工程后的界面。



图 1-2 组态示意图

1.3.1.3 显示选项

可以通过显示选项对话框来修改开发环境的显示选项。通过系统"工具"菜单或者系统工具栏,可以弹出"显示选项"对话框。如下图所示:

系统		-
系统主题	Office 2007	*
3 窗口		
背景网格	直线	
网格间距	50	
网格颜色	c0c0c0	
鼠标摄取间距	5	
三维窗口(未	完成)	
网格间距	30	
网格颜色	000000	
JavaScript	肩楫器	
字体	宋体	
字号	12	
关键字颜色	0000ff	
字符串颜色	a31515	
注释行颜色	008000	

图 1-3 显示选项

在该对话框中,可以修改开发环境的显示主题、窗口/三维窗口/JavaScript 编辑器等的显示选项。



1.3.2 运行组态工程

系统提供了多种方法运行某个特定组态工程。

方法1:从开发环境起始背景面板处运行。先选择要运行的工程,然后点 击起始背景面板工具栏的"运行工程" 图标即可运行。如下图所示:

[] 新建工程	× 册除工程	(1) 打开工程	添加工程	运行工程	》 默认工程	令 工程打包
工程名称		分辨率	工程路径	▶ 运行1	程	工程描述
🔪 简单演示		1024x768	E:\简单演;	运行选	择的组态工程	
●HMI演示		640x480	E:\HMI演示	、HMI 循示. t	pj	
eee		1280x800	E:\eee\eee	e. tpj		演示工程

方法 2: 打开该工程后,点击系统工具栏的运行图标运行。此法适合开发工 程时调试使用。如下图所示:



方法 3: 将工程设置为当前默认工程后(如何设置默认工程,请参考"工 程管理"的"当前默认工程"一节),直接启动组态运行环境即可。 这里也分 成两种情况:桌面 PC 网络版本和非桌面 PC 网络版本。

桌面 PC 网络版本:先启动组态运行数据库,然后启动组态浏览器或者 IE 浏览器。

非桌面 PC 网络版本:直接启动运行环境(桌面 PC 版本)或者启动。



二、组态工程开发

2.1 创建工程

点击"文件→新建工程"菜单项,或者点击起始背景面板工具栏的"新 建工程"图标,将创建一个新工程,如下图所示:

工程名称:	eee		
工程描述:	演示工程		
分辨率:	1280x800	一桌面PC版本	
了程路径:	E:\		6

图 2-1 新建工程

输入工程名称、工程描述、分辨率、工程路径后,点击确定后,将显示画面布 局设计对话框。

<u>注:创建新工程时,开发环境自动会在选择的"工程路径"下新建一个以工</u> 程名称命名的子目录作为工程存放目录。

画面分辨	率自适	Ŵ,	🗹 画面水平居中岛	記示	
区域	显示	宽或高	起始窗口	记录窗口	1
主区域	4	-	窗口 - 登录	V	
上栏框	V	90	窗口 - 主栏框	1	
左栏框	E	126		1	G
右栏框	E	100		10	
下栏框	100	100		10	
安照栏框的 示顺序,位 在前的栏框 先占用边界 或。)	显置优区		主区域		

图 2-2 画面布局

如果此时需要设计画面布局,则在设计之后点击确认,系统会自动根据栏框尺



寸生成所选中的栏框窗口。未来也可以通过工程画面浏 览窗口的"画面配置"下的 "画面布局"节点重新设计。有关画面布局 的使用,请参考画面布局相关章节。 设计好画面布局之后,一个新工程就生成了。下面可以进行工程的设计开发工作 了。

2.2 创建窗口

在创建新的工程时,系统已经默认创建了主窗口。还可以采用如下方法创建 新的窗口。

2.2.1 工程画面浏览窗口创建

在工程画面浏览窗口中右键点击"窗口"节点弹出菜单后,点击"新增 窗 口",弹出新增窗口对话框。

在新增窗口对话框中,输入窗口名称和窗口大小。根据窗口坐落区域设置窗口 大小:即根据工程设置的画面分辨率以及栏框的宽度或高度来计算各个区域的大 小。

画面	Д Х	新建窗口	×
	画面配置 窗口 新增窗口 Ctrl+N	窗口名称: 窗口1 确定	
	 調整窗口大小 増加窗口組 添加画面文件 	窗口大小□ 取消○ 根据窗口坐落区域设置窗口大小	
	 ゴ	主区域 - 1180x600 ● 手动设置窗口大小 宽: 800 高: 600	
	☆ 设置主窗口		

图 2-3 浏览创建窗口

2.2.2 命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。



新建文件	
 类别: ● 画面 承统模块 ● 漫 数据库 ● 承 系统模块 ● 通 系统模块 ● 通 通用设备 ● 通 PLC 	文件类型:
二维平面窗口。 2 将新建的文件添加到当前工程 新建文件名: 确定	取消

图 2-4 新建文件

这种方法创建的窗口无法指定窗口大小,不建议使用。一般这种方法用来创建 临时窗口文件。

2.3 创建 IO 设备通信

通过如下方法创建新的 I0 设备通信。

2.3.1 工程数据库浏览窗口创建

在工程数据库浏览窗口中右键点击"IO 设备通信"节点弹出菜单后, 点击

数据库 ■ ■ 月 ■ 1 ■ 月 ■ 月 ■ 月	系统 服表 TavaS 系统材	C置 Cript 夏块	φ ×	E
	b	新增I0设备通信	C	trl+N
		添加数据库文件		
	3	打开	0	Ctrl+O
		重命名		
	×	 		

"新增 IO 设备通信",弹出新增 IO 设备通信对话框。



在新增 I0 设备通信对话框中,先选择"**类别**",然后选择"**文件类型**",最 后输入**新建文件名**。

w.	MODDUSKIU	Modbusitr
X		
虚拟设备		

2.3.2 命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。

新建文件			D
类别:	文件类型:		
 □ □ □ □ □ □ □ □ □ □ □ □ □	Opc opc	Nodbus ModbusRTV	Nodbus ModbusTCP
□ · · · · · · · · · · · · · · · · · · ·	虚拟设备		
实现OPC功能的驱动。	>		
新建文件名:		1	
・			

在新增文件对话框中,先选择"**类别**",然后选择"**文件类型**",选中将文件 加入工程,最后输入要创建文件名。



图 2-5 命令创建 IO 通信

2.4 增加IO 通信变量

创建 I0 设备通信后,就可以增加该通信变量了。为了演示方便,下面以 系统内置的虚拟设备为例,讲述如何增加变量。

下图是驱动的工具栏。



2.4.1 增加变量

点击驱动工具栏的"增加变量"图标,将在驱动表格增加一个变量,如下 图。增加的位置为当前焦点变量的位置,如果无当前焦点变量,则增加到表格 尾部。

变量	类型	值变模式	初始值	读写方式	报警	历史记录	操作记录	保存值	数据转换	描述
演示变量	INT	随机	0	读写					1	

修改变量名为"演示变量",类型为"INT",值变模式为"随机"。

2.4.2 批量增加

也可以点击"批量增加"按钮一次性地增加大量变量,如下图所示:

文里心・	量名: *		(如:abc*de)	
通配符个数:	1		(1~5)	
起始地址:	0			
变量个数 :	100		(1~5000)	
☑ 增加到新组	新组名:	批量测	l 试组	
内容		值	1	
类型	类型		INT	
值变模式		随机 0 读写		
初始值				
读写方式				
报警				
历史记录				
操作记录				
1070 million halos			100	
保存值				
保存值数据转换				

- **变量名:** 变量名必须包含"*****",表示地址。当批量增加变量时,地址值将替换 *"**"。
- 通配符个数:地址名位数将根据通配符个数来生成,地址名位数如果比通配符个数少,前面将自动增加"0"。
- 起始地址:要生成批量变量的起始地址。



- **变量个数:**要生成批量变量的个数。
- **增加到新组**:是否增加一个新组来存放这些批量生成的变量。
- 初始值:对话框中的表格为变量其它属性选项的初始值。

最后保存该文件。

2.5 建立动画链接

下面介绍一下如何制作一个简单的动画。

1.打开画面主窗口。

2. 点击工具箱的"基本图形"中的"矩形"图形,在主窗口中拖拽创建一个矩形。

3. 在属性栏的"动画"选项中,点击"填充"动画。

如下图 2-6 所示:



4. 在弹出的"填充动画配置"对话框中,按下图增加颜色填充列表。然后点击"选择变量"按钮。如图:



表达式	选择变量	表达式
数据库 v1. 演示变量		
<u>C</u>		2
填充列表		
	The second se	
表达式值〈=	颜色	Hàthn
表达式值<=	颜色	— ^{增加}
表达式值<= 10000.000000 20000.000000	颜色 c84479 055188	· · · · · · · · · · · · · · · · · · ·
表达式值<= 10000.000000 20000.000000 30000.000000 30000.000000	颜色 c84479 055188 68654d	·····································
表达式值<= 10000.000000 20000.000000 30000.000000 10000.000000	颜色 c84d79 055188 6865dd 3205cf	増加
表达式值<= 10000.000000 20000.000000 30000.000000 10000.000000	颜色 c84d79 055188 6865dd 3205cf	増加

5. 在连接变量的对话框中,选择前面增加的 IO 通信变量。

表达式	选择变量	表达式
数据库.v1. 演示变量	\sim	~
C		3
填充列表		
表达式值〈=	颜色	增加
10000.000000	C84d79	
20000.000000	055188	
30000.000000	6865dd	
40000.000000	3205cf	

6. 保存主画面,并运行工程,将看到矩形的背景填充颜色随着变量值的改变而不断 变化。

2.6 调试与运行

系统提供了多种方法运行某个特定组态工程。



方法1:从开发环境起始背景面板处运行。先选择要运行的工程,然后 点击起始背景面板工具栏的"运行工程"图标即可运行。如下图所 示:

新建工程	★ 册除工程	同 打开工程	添加工程	运行工程	》 默认工程	工程打包
工程名称		分辨率	工程路径	▶ 运行1	程	工程描述
🔪 简单演示		1024x768	E:\简单演习	运行选	择的组态工程	
●IMI演示		640x480	E:\HMI演示	、HMI 循示. t	pj	-
oee 🕘		1280x800	E:\eee\eee	. tpj		演示工程

图 2-9

方法2:打开该工程后,点击系统工具栏的运行图标运行。此法适合开发工程时调试使用。如下图所示:

<u>注: 桌面 PC 网络版本与其它版本略有不同,因为网络版本的运行包括组态浏</u> <u>览器、IE 浏览器、组态数据库等多个程序的运行。网络版本下如果直接点击运行</u> 图标,默认将先后启动组态数据库和组态浏览器。



- 方法3:将工程设置为当前默认工程后(如何设置默认工程,请参考"工程管理"的"当前默认工程"一节),直接启动组态运行环境即可。这里也分成两种情况:桌面PC网络版本和非桌面PC网络版本。
- **桌面 PC 网络版本**:先启动组态运行数据库,然后启动组态浏览器或 者 IE 浏览器。
- 非桌面 PC 网络版本: 直接启动运行环境(桌面 PC 版本)或者启动。



三、工程管理

3.1 概述

每个工程对应一个实际的应用案例。

用户有可能会开发多个工程。系统提供方便的工程管理,实现了对多个组态工程的集中管理。用户可以通过"文件"菜单或开发环境启动时的背景面板实现对组态工程的集中管理。如下图所示:

注: 桌面 PC 版本与 HMI 版本略有不同,桌面 PC 版本最后一个为"工程打包", 而 HMI 版本为"下载工程",参见图中蓝色部分。



工程管理具有新建工程、删除工程、打开工程、关闭工程、添加工 程、下载 工程、运行工程、设置默认工程、工程打包、工程加密等 功能。

3.2 新建工程

点击"文件"菜单上的"新建工程"菜单项,或者点击起始背景面板工具栏 的"新建工程"图标,将创建一个新工程,如下图所示:

<u>注:HMI 开发环境与 PC 桌面版本的分辨率提示略有不同,但实质一样,如下</u> 图红线圈住的位置。



新建工程	
工程名称:	eee
工程描述:	演示工程
分辨率:	1280x800 - 桌面PC版本
工程路径:	E:\
将在 E:\ee	e 处创建工程。
	确定取消
新建工程	
新建工程 工程名称:	eee
新建工程 工程名称: 工程描述:	eee 演示工程
新建工程 工程名称: 工程描述:	eee 演示工程
新建工程 工程名称: 工程描述:	eee 演示工程 TSHMI005 - 800x600
新建工程 工程名称: 工程描述: MMI型号: 工程路径:	eee 濱示工程 TSHMI005 - 800x600 王:\ E:\
新建工程 工程名称: 工程描述: 「MMI型号: 工程路径:	eee 濱示工程 TSHMI005 - 800x600 → HⅢI版本 E:\ E:\
新建工程 工程名称: 工程描述:	eee 濱示工程 TSHMI005 - 800x600 → HII版本 E:\ @

图 3-2

输入工程名称、工程描述、分辨率、工程路径后,点击确定后,将显示画面布 局设计对话框。

<u>注: 创建新工程时,开发环境自动会在选择的"工程路径"下新建一个以工程名称</u> 命名的子目录作为工程存放目录。



画面分辨	率自适.	Ň.	☑ 画面水平居中显	示	
区域	显示	宽或高	起始窗口	记录窗口	
主区域	4	-	窗口 - 登录	V	-
上栏框	V	90	窗口 - 主栏框	10	LŤ
左栏框	E	126		(E)	
右栏框		100		173	
下栏框		100		17	
(布局原则 安照栏框的 天顺序,位 正前的栏框 上占用边界 或。)	:显置忧区		主区域		

如果此时需要设计画面布局,则在设计之后点击确认,系统会自动根据栏框 尺寸生成所选中的栏框窗口。未来也可以通过工程画面浏 览窗口的"画面配置" 下的"画面布局"节点重新设计。有关画面布局 的使用,请参考画面布局相关章 节。

设计好画面布局之后,一个新工程就生成了。下面可以进行工程的设计开发工 作了。

3.3 删除工程

点击起始背景面板工具栏的"删除工程"图标,将删除一个旧工程,如下图所示:





3.4 打开/关闭工程

3.4.1 起始背景面板中打开工程

在起始背景面板的工程列表中,选择要打开的工程,双击该工程,或者点击 背景面板工具栏的"打开工程"图标,将打开该工程。

3.4.2 在"文件"菜单或系统工具栏中打开工程

点击"文件"菜单上的"打开工程"菜单项,或者系统工具栏的"打开"的 图标,可以从磁盘上选择要打开工程。

3.4.3 最近工程列表打开工程

点击系统工具栏的"打开"的图标旁边的下拉菜单,出现最近工程列表,点击要 打开的工程即可,如下图所示。



图 3-4

3.4.4 关闭工程

当前如果有打开的工程,点击"文件"菜单上的"关闭工程"菜单项,即 可关闭当前工程。


3.5 当前默认工程

默认工程是指: 当组态运行环境启动时, 默认运行的工程。

在起始背景面板的工程列表中,选择要置为默认的工程,然后点击背景面板 工具栏的"默认工程"图标,将该工程置为默认工程。

当工程被置为默认工程后,该工程在工程列表中的图标被置为默认工程的图标,这样与其它工程区分开来,如下图所示:



3.6 工程打包

点击"文件"菜单上的"工程打包"菜单项,或者点击起始背景面板工 具栏的"工程打包"图标,将打包一个工程,如下图所示:

打包选项	1	
口打包开	发环境	
打包名称	: 简单演示	
打包路径	G:/	
将打包为	G:\简单循示.exe。	

- **要打包的工程:** 需要打包工程的完整工程文件路径。
- **打包开发环境**:如果希望解包后能继续组态开发,则将该选项打勾。
- 打包名称和打包路径:打包后包裹的完整文件路径由打包名称和打包路径 组合而成。



3.7 工程加密

工程加密是为了保护工程不被修改、复制。

打开工程后,点击"文件"菜单上的"工程加密"菜单项,将加密当前 打开工程,如下图所示:

程加密后,一旦密码ì 将无法打开工程 !	贵
位长度的密码: 長示解密)	
***]
***	1
运行 ,只有当前指定的加密 行!该功能只有在工利)	狗」
取消	
	程加密后,一旦密码; 将无法打开工程? 位长度的密码: 反示解密) **** **** **** 云行 ,只有当前指定的加密 行? 该功能只有在工程) 取消

- 工程密码: 输入工程密码进行加密。如果密码为空,表示该工程不需要加 密了(解密该工程)。
- 捆绑加密狗:为了更好的保护您的知识产权,系统提供了捆绑当前加密 狗功能。这种情况下,只有该加密狗才能使工程正常运行,即使其它的 授权狗也无法运行。

注: 工程加密后,工程的主要文件都将被加密,所有文件都只能在 该工程中才能 被打开,不方便文件复制,尤其是 3DS 模型文件不能在其它 3D 模型编辑器打开 编辑。所以建议在组态开发状态下不要加 密工程,而正式提供给用户使用时再加 密工程。

3.8 下载工程

下载工程是提供给 HMI 版本使用的,目的是将工程下载到 HMI 设备上运行。

系统能自动检测 HMI 所需要运行的组件,并且根据工程文件是否被修改来下载。 所以一般第一次下载新工程时间较长一些(得下载所有工程文件),以后下载该工



程速度较快(只要下载新的或者修改过的工程文件)。

目前系统提供了 IP、U 盘、USB 数据线形式的下载。

点击"文件"菜单上的"下载工程"菜单项,或者点击起始背景面板工具栏的 "下载工程"图标,将下载一个工程,如下图所示:

下载工程	
要下载的工程	
\组态(HMI版)\project\HMI演示	\HMI演示.tpj 🔽
□ 下载智能移动Web支持组件	
下载方式	
⊙使用IP下载,远程主机IP:	10. 5. 192. 36
◯使用VSB盘下载,V盘符:	×
○ 使用USB数据线下载	
下载	关闭
2	₹ 3-8

- 要下载的工程:本地要下载工程的完整的工程文件路径。
- 智能移动 Web 支持组件: 是否下载能支持智能移动 Web 浏览监控 的组件。
- 远程主机 IP: 远程要下载目标 HMI 设备的 IP 地址。下载之前得保证与该设备的网络相通。
- USB 盘符:如果选择 U 盘方式下载,则系统先将工程下载到 U 盘上,然后将 U 盘插入 WinCE 设备,则 WinCE 上的 DLServer 程序会自动将新工程复制到设备,并重新运行新工程。此处输入要下载的 U 盘盘符。
- USB 数据线下载: 与 ActiveSync 原理相同,得安装 ActiveSync (但是可以不必运行 ActiveSync), ActiveSync 能通信,则就能通过 USB 数据线下载。



四、工程浏览

4.1 概述

组态开发环境为当前打开的组态工程提供了一个总体浏览功能,通过该总体概貌,用户就可以快捷地定位到需要组态编辑的部分。 为了与系统先天的 B/S、C/S 架构相对应,工程浏览分为画面端(客 户端)浏览和数据库端(服 务器)浏览部分。如下图所示:





4.2 画面端



画面端浏览包括了所有画面端运行时相关的功能模块。如下图所示:

画面端浏览包括如下部分:

- **画面配置**:包括画面布局、画面外观、画面端的多语言设置。
- **窗口**: 画面中最主要的部分。请参考窗口与图形一节。
- 三维窗口:三维立体监控画面中最主要的部分。请参考三维窗口与图形 一节。
- 定制图形:用户自定义的图形模板。请参考窗口与图形一节中的定制
 图形。
- 系统模块: 画面端的辅助功能模块,目前包括定制变量和动作集。请
 参考画面和数据库系统模块一节。



4.3 数据库端



数据库端浏览包括了所有实时数据库端运行时相关的功能模块。如下图所示:

数据库端浏览包括如下部分:

- ◆ 系统配置:包括报警、事件、历史记录、安全区、用户、WEB 等配置。
- ◆ 报表:请参考报表一节。
- ◆ JavaScript: 系统内置脚本语言。请参考 JavaScript 一节。
- ◆ 系统模块:数据库端的辅助功能模块,目前包括定制变量、定时器、 触发器、配方、SQL 访问、网络共享等。请参考相关部分。
- ◆ I0 设备通信: 数据库端最主要部分。请参考 I0 设备通信一节。



五、画面布局与外观

5.1 画面布局

画面布局是指画面上除了主要的显示区域外,还可以有上下左右栏 框,可以 用这些栏框来显示菜单、状态栏、窗口管理等。这样即使 主区域的窗口不断切 换,栏框的内容也不变。 在画面浏览窗口中双击"画面配置"节点下的"画面布 局"子节点,打开画面布局配置对话框。

画面	ά×	画面布局					
□ 💿 画面配置	D		率自适	应	☑ 画面水平居中显	显示	
一 回 面外观		区域	显示	宽或高	起始窗口	记录窗口	
● 多语言も	置	主区域	~	=	窗口 - 登录	V	
田 🔄 留口		上栏框	V	90	窗口 - 主栏框		LT I
三维窗口		左栏框		126			\square
- 🦉 定制图形		石栏框		100			
🗉 🛒 系统模块		r 4≘fi±		100		183	
		布局预览: (布局原则 按照栏框的 示顺序,位	: 聖 王		上栏框		
		在前的栏框 先占用边界 域。)	区		主区域		
國面				備定	取消		

图 5-1

- 画面分辨率自适应:根据工程分辨率与屏幕窗口分辨率的比例关系, 将画面自动缩放显示到屏幕窗口上。
- 画面水平居中显示:当屏幕窗口分辨率的宽比例大于工程分辨率的宽比
 例时,能将画面水平居中显示在屏幕窗口上。

画面布局的表格列说明:

- ◆ 区域:包括主区域、上下左右等栏框。
- ◆ 显示: 表示是否显示该区域。
- ◆ 宽或高:如果区域是上下栏框,则该列内容表示上下栏框的高;如果区域是左右栏框,则该列内容表示左右栏框的宽。
- ◆ **起始窗口**:系统画面启动时,该区域的初始窗口。
- ◆ 记录窗口:记录下区域显示窗口的历史,供调用"画面.上一窗口



()"、"画面.下一窗口()"时使用。

调整栏框的显示范围:

通过"↑"、"↓"按钮调整上下栏框显示顺序,位置在前的栏框优先占用边 界区域。

5.2 画面外观

画面外观仅仅对Windows 单机版本有效。该配置就是设置Windows 单机版本运行时的界面外观。在画面浏览窗口中双击"画面配置"节点下的"画面外观"子节点,打开画面外观对话框后即可进行配置。

画面	🖌 百面外观配置 🛛 🔀
 ■面部員 ■面部局 ■面部局 ● 画介拠 ● 多语言设置 ● 第四 ● 三維窗口 ● 三維窗口 ● 定制图形 ● 系统模块 	注:画面外观配置仅仅对Windows单机版本有效。 标题: 滴示工程 □显示标题栏 □显示系统莱单 □显示关闭框 □显示最小框 确定 □显示最大框 取消
一國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國國	

该对话框可以设置标题、是否显示标题栏、是否显示系统菜单、是否显示关闭 框、是否显示最小/大框、边框是否可变大小等。



六、IO 设备通信

6.1 概述

I0 设备通信是工控监控中非常重要的环节。系统就是通过 I0 设备通 信驱动 与现场设备进行通信采集。由于前期一直注重新一代组态软件内核架构开发,所以 目前 I0 设备 通信驱动较少,未来会加大 I0 通信驱动的开发力度,逐步支持更多 的驱动。

目前系统已经支持了一些标准的、常见的驱动,包括 Modbus RTU、Modbus TCP、OPC、西门子、三菱、SNMP 等。

6.2 新建 IO 设备通信

通过如下方法创建新的 I0 设备通信。

6.2.1 从工程数据库浏览窗口创建

在工程数据库浏览窗口中右键点击"I0 设备通信"节点弹出菜单后, 点击"新增 I0 设备通信", 弹出新增 I0 设备通信对话框。

新增10设备通信		
类别:	文件类型 :	
■ ■ ■ ■ ■ ■ ■ ■ ■ ■	OPC Nodbus Nodbus OPC ModbusRTV ModbusTCP	0
	虚拟设备	
实现OPC功能的驱动。		
新建文件名:	确定取消	
	图 6-1 图 6-2	



在新增 I0 设备通信对话框中,先选择"类别",然后选择"文件类型", 最后输入要创建文件名。

6.2.2 通过命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令创建。

■面 承統模块 ● 過 数据库 ● 近 近设备通信 ● 近 PLC ● 近 四门子 ● 欧姆龙		文件类型:
□ <u>1</u> ID设备通信 □ <u>1</u>	 □ 3 画面 ● 3 系统模块 ● 3 数据库 ● 5 条统模块 	OPC Nødbus OPC ModbusRTV ModbusTCP
虚拟设备	日 5 10设备通信 5 通用设备 日 5 PLC 平 5 PLC	16-10120.25
	● 西门子 ● 欧姆龙	虚拟设备
实现OPC功能的驱动。	实现OPC功能的驱动。	
新建文件名:	新建文件名:	

图 6-3

在新增文件对话框中,先选择"类别",然后选择"文件类型",选中将文件 加入工程,最后输入要创建文件名。

6.3 配置IO 设备通信

新建 I0 设备通信后,必须在默认配置基础上作些配置才可以与 I0 设备正常通信。

点击 IO 通信驱动的工具栏的"基本配置"弹出配置对话框,或者直接 在属性 栏进行配置。

6.3.1 IO 通信工具栏





6.3.2 IO 基本配置选项

- 通道类型:表示系统采用哪种通信介质与 I0 设备通信。目前已经支持的有串口、Tcp、Udp。当选择其中一种通道时,其它的通道选项配置将变成只读(变 灰)。
- **设备地址:** 设备的地址 ID 号。
- 读包字节数: 部分 IO 通信具有该选项。不同 IO 设备的处理包的能 力不同,但 是它们都有一个最大发包字节数。一般而言,发包字 × 属性栏 □ 普通配置 节数越大的设备,处理包的能力越强,通信效率也越 通道类型 串口 ٧ 高。用户须将这个选项设置为设备的最大发送包的字 设备地址 1 读包字节数 100 节数,这样通信效率高。 双字顺序 3412
- 强行写值:即使写入值与设备的内存值相同,也强 行将值写入设备。
- 优化采集: 仅采集当前打开窗口和历史保存的所使用
 到的 I0 变量,以提高 I0 采集效率。
- 写值有效期:如果写 I0 变量值时通信失败了,允许 延期尝试的有效时间(秒),当超过这个时间后,就 放弃该 I0 写操作。
- 通信超时:当发送一个请求包之后,I0 设备必须在规定的时间内应答,否则认为是本次通信失败。这个时间就是通信超时时间,单位为毫秒。
- 采集周期:每次发送请求包之间的间隔时间,单位为 毫秒。在设备处理能力允许的情况下,一般可以将采 集周期设短,而将通信超时设长。

通信错误控制

- 失败次数:每一次通信超时,则认为一次通信失败, 当失败次数达到指定的数目时,系统认为通信出错 了,将重置该驱动,重连设备。这个选项就是指定 多少次失败后,重置通信。
- 浮点数顺序 3412 强行写值 是 忧化采集 是 通信超时 300 采集周期 300 写值有效期 10 描述 □ 串口通道 串口 COM9 波特率 19200 数据位 8 校验 无 停止位 1 接口 RS232 是 DTR 是 RTS □ Tep通道 主机 80 端口 🖸 Udp通道 主机 80 端口 □ 通信错误控制 失败次数 3 尝试间隔 10 最长时限 60 重连写保存值 否 □ 文件属性 文件类型 驱动模块 驱动类型 ModbusRTU 通道类型 驱动通信的IO通道类型
- 尝试间隔:当通信错误发生后,每次尝试重新连接设备的间隔时间,单位为秒。
- 最长时限:当通信错误发生后,不断尝试连接的最长时间限制,当超过这个时限后,将不在尝试连接,单位为分钟。0表示不限制,即不断尝试连接,直到连接成功。

● **重连写保存值:**当通信重新连接成功后,是否将要保存的变量值写入设备。

6.4 IO 通信通道

IO 通信得通过指定通道才能与 IO 设备通信,目前支持的 IO 通道有 串口、 TCP、UDP 等通道。另外,系统也支持运行时动态修改通道。

6.4.1 串口通道

- **串口:** 串口通信 I0 口。 波特率: 串口通信波特率。 数据位: 串口通信数据 位。
- 校验:串口通信校验类型,包括无校验、奇校验、偶校验、标记校 验、空格 校验。
- **停止位:**串口通信停止位,包括1位停止位、1.5位停止位、2位停止位。
- 接口: 串口通信接口,包括 RS232、RS422、RS485。
- DTR: 串口初始化连接时是否设置数据终端就绪(data-terminal-ready) 信 号。
- RTS: 串口初始化连接时是否设置请求发送(request-to-send)信号。

6.4.2 TCP 通道

主机: TCP 通信远程主机 IP 或主机名, 空值表示本机。

端口: TCP 通信远程服务器端口。

6.4.3 UDP 通道

主机: UDP 通信远程主机 IP 或主机名, 空值表示本机。

端口: UDP 通信远程服务器端口。

6.4.4 GPRS_CDMA 通道

GPRS_CDMA 通道可以实现对 DTU 终端的支持。 该节有点复杂,请参考专门介绍 DTU 与 GPRS_CDMA 通道的章节。



6.4.5 动态修改通道

通过 IO 驱动的"_设置通道"命令函数可以动态修改通信通道。

按照"项目名=项目值"的格式设置,要修改的项目以";"隔开。如修改 串口通道: "串口=COM3;波特率=9600",修改 TCP 通道: "端口=666; 主机 =192.168.3.3"。

下图是系统启动时通过调用 m1 驱动的"_设置通道"命令函数来设置新的通道。

mbl夜宜通迫(中口=CUM3)	増加
	删除
	编辑
	上移
	下移
统初始化后执行的动作	
运行JavaScript("系统初始化")	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	删除
	编辑
	上移

图 6-4

6.5 DTU 与GPRS-CDMA 通道

6.5.1 概述

GPRS_CDMA 通道/DTU 终端使用较为广泛,而且比较复杂,所以本节专门介绍它们的使用。

数据中心: 在使用 GPRS_CDMA 通道之前,得先配置数据中心,然 后选择某



Web:http://www.witlinc.cn E-mail:support@witlinc.cn

个数据中心进行通信。当前数据中心仅支持 TCP 协议以及 DTU 的透明传输模式。数据中心可以在工程管理栏的"数据库"选项 卡的"DTU 数据中心"进行 配置,也可以直接点击该属性进行配置。如下图。两种配置略有不同,因为后 一种多了选择数据中心的按钮。

敷据中心配	置				DTU数据中心配	置			
数据中心名	协议	端口	DTV厂商	增加	数据中心名	协议	端口	DTU厂商	增加
中心1	TCP	8000	透明传输		中心1	TCP	8000	透明传输	田底全
中心2	TCP	8001	透明传输		中心2	TCP	8001	透明传输	101124
									选择 保存 取消

图 6-5

目前本系统 DTU 的应用包括两种情况: DTU 可以自定义心跳包、DTU 不可自定 义心跳包。

6.5.2 DTU 可以自定义心跳包

终端 ID: 该驱动对应的 DTU 终端 ID,终端 ID 可以是除空格以外的 任何 ASCII 字符组合、长度为不少于 3 位的字符串。

DTU 终端的心跳/登录注册包必须如下定义: HELLO 终端 ID。

例如:终端 ID 定义 "ABCDE",自定义心跳包则为 "HELLO ABCDE",十 六进制格 式为 "48 45 4C 4C 4F 20 41 42 43 44 45"。如下图所示。

B GPRS CDMA	通道
数据中心	数据中心1
终端ID	ABCDE
终端IP	
终端端口	0

这种情况下,因为 DTU 可以通过终端 ID 识别,所以一个数据中心可以连接多 个不同的 DTU。另外终端 IP 为空、终端端口为0 即可。

6.5.3 DTU 不可自定义心跳包

这种情况下,因为DTU 无法通过终端 ID 识别,所以一个数据中心只 能连接一个 DTU。



也正是因为DTU 无法通过终端 ID 识别,所以此法不太安全,为了提 高安全,如果DTU 能固定 IP 或端口,则可以输入终端 IP 或终端端 口来提高安全性。

数据中心	数据中心2
终端ID	1
终端IP	
终端端口	5000

由于无法自定义心跳包,终端 ID 必须配置为空。

6.6 增加IO 通信变量

创建 I0 设备通信后,就可以增加该通信变量了。为了演示方便,下面 以系统内置的虚拟设备为例,讲述如何增加变量。

下图是驱动的工具栏。



6.6.1 增加变量

点击驱动工具栏的"增加变量"图标,将在驱动表格增加一个变量,如下 图。增加的位置为当前焦点变量的位置,如果无当前焦点变量,则增加到表格尾 部。



修改变量名为"演示变量",类型为"INT",值变模式为"随机"。

6.6.2 批量增加

也可以点击"批量增加"按钮一次性地增加大量变量,如下图所示:



变量名:	*		(如:abc*de)
通配符个数:	1		(1~5)
起始地址:	0]
变量个数:	100		(1~5000)
☑ 增加到新组	新组名:	批量	则试组
内容		值	1
类型		INT	
值变模式		随机	
初始值		0	
读写方式		读写	
报警			
历史记录			
操作记录			10-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2-2
保存值			
数据转换			
描述			

- **变量名:** 变量名必须包含"*",表示地址。当批量增加变量时,地址值将替换 *"**"。
- 通配符个数:地址名位数将根据通配符个数来生成,地址名位数如果比通配符
 个数少,前面将自动增加"0"。
- **起始地址**:要生成批量变量的起始地址。
- **变量个数:**要生成批量变量的个数。
- **增加到新组**:是否增加一个新组来存放这些批量生成的变量。
- 初始值:对话框中的表格为变量其它属性选项的初始值。

最后保存该文件。

6.7 IO 变量相关操作

除了上节介绍的变量增加外,变量相关操作包括编辑、删除、复制/粘贴/ 剪切、分组、导入/导出等。 变量的大部分操作是在驱动工具栏中。



编辑

- 1. 单个编辑: 直接点中要编辑变量的属性列编辑。
- 批量编辑:如果要批量编辑某属性列值,选择这些变量的所有列值 后,可以按住 "Ctr1"键,修改其中一个变量的属性值,会导致所有 选中的变量属性列值改变。

删除

点中要删除的变量(可以多个变量),点击驱动工具栏的"删除"按钮。

复制/粘贴/剪切

开发环境允许相同类型的驱动间的变量复制/剪切/粘贴。要想使用 不同类型 的驱动变量,可以采用导入/导出功能。 点中要复制/剪切的变量(可以多个变 量),点击复制/剪切命令(或者快捷键Ctr1+C/Ctr1+X),然后点击要粘贴的目标位 置,最后点击粘 贴命令(或快捷键Ctr1+V),将粘贴到当前焦点变量位置,如果没 有 选择粘贴的目标位置,将粘贴到驱动的尾部。

分组 当驱动的变量较多时,为了更好地管理变量,系统提供了变量管理功能。而且,在组态选择变量和调试驱动时,界面会出现分组信息,方便查找与调试。

- 增加分组:点击驱动工具栏的"增加组"按钮,将新增一个组。如果原来驱动没有组,则将原来所有变量归入新组;否则增加了一 个带有"新变量"的新组。
- **组更名:**点中要更名分组的行,点击驱动工具栏的"组更名"按 钮, 弹出更名对话框,输入新组名即可。
- 3. 删除组: 当该组下所有变量被删除了,则该组自动被删除。

导入/导出

系统提供了导入/导出 CVS 文件(逗号分隔的文件)的功能。点击驱动工 具栏的"导入/导出"按钮,然后选择要导入/导出的文件即可。 该功能可以用 来导入/导出变量词典列表文件,或者不同驱动类型之 间变量的复制。



6.8 IO 通信变量选项

所有 IO 通信变量都有一些共同的属性选项,如下图红色圈住的部分:

◎ 基本配量	★ 1000 1000 1000 1000 1000 1000 1000 10	加制除变量	1000 1月 1月	11 平久	→ 最入	■ → 単出	□ → → ☆新 运	(7			
变量	寄存器类型	寄存器地址	值类型	读写方式	报警	历史记录	操作记录	保存值	优先级	数据转换	描述
新变量	Holding Register	0	WORD	读写				1	100		

值类型:变量值类型,有的驱动叫数据类型,包括:BIT/BOOL(位/布尔 类型,即开关量)、CHAR(字符类型,1字节)、BYTE(1字节无符号整数)、 SHORT(2字节符号整数)、WORD(字类型,2字节无符号整数)、INT(4字节 符号整数)、DWORD(双字类型,4字节无符号整数)、FLOAT(4字节单精度浮 点数)、DOUBLE(8字节双精度浮点数)。

读写方式:包括读写、只读、只写,分别表示变量可以读写、变量只能读不能 写、变量只能写不能读。

报警:变量是否需要报警,请参考报警章节。

历史记录:变量是否需要历史记录,请参考历史记录章节。

操作记录:变量是否需要记录操作,请参考事件章节。

保存值: 变量是否需要保存值,这样在下一次系统启动时,能以保存的 值写入设备。

优先级: 0~100,优先级越大,采集频率越高,0 表示不采集(手 动调 用驱动方法"_采集变量"时触发采集)。采集频率按优先级比例 分配,举例 如下:变量1 优先级为1,变量2 优先级为10,变量3 优先级为100,则当总 共采集次数为111 次时,变量1 采集了1 次,变量2 采集了10 次,变量 3 采集了100 次。但是实际上,变量采集的优先级是变量所属10 包的优先 级,具体请参考《10 设备通信变量 采集说明》。

描述: 该变量的说明,可以在选择变量时提示用户。 数据转换: 转换类型目前包括线性转换和开方转换,并且可以转换 为其它类型的数据。

-线性转换:按照"a*采集数据 + b = 转换数据(其中 a、b 为系数)"计算。可以点击右边的"配置"按钮进行设置。



一开方转换:即转换后的值为采集值的开方值。

变量数据转换配置 🛛 🔀	变量线性转换配置	
 转换类型: 30/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2 31/2	采集数据最小值: 〕最大值: 0	转换数据 最小值: 0 最大值:
确定取消	确定	取消

6.9 IO 设备通信采集说明

6.9.1 IO 采集包

当采集 I0 设备的数据时,组态软件会根据用户所配置的变量,将它 们组合成 I0 采集包。为了提高 I0 通信效率,得尽量减少 I0 采集包。 因为包越少,那么 每个包被调度请求的频率就越高,这样采集效率 就高了。一般 I0 通信协议都是规 定采集某个寄存器类型的地址段,所以组态软件会将连续的地址组合成地址段。所 以在相同数量的变 量个数情况下,如果寄存器类型越少、连续的地址段越多,则组 合 成的包就越少了,这样采集效率就高了。 另外,组合包个数与设备通信能力相 关,即设备每次通信允许的最 大字节数。当组合的包超过这个限制后,系统会自动 新建一个包, 然后再对剩余变量进行组合。

注:系统会自动对变量的地址排序,然后组合包。所以即使组态时变量地址是 无序的,系统也会排序组合。排序与分组无关,即使不同组的变量,如果地址是连 续的,也会组合在一起。

6.9.2 优先级

变量具有优先级 (0~100),优先级越大,采集频率越高,0 表示不 采集(手动调用驱动方法"_采集变量"时触发采集)。采集频率按优 先级比例分配,举例如下:变量1 优先级为1,变量2 优先级为10, 变量3 优先级为100,则当总共采集次数为111 次时,变量1 采集了1 次,变量2 采集了10 次,变量3 采集了100 次。而实际上,变 量组合成10 采集包后,变量的采集频率就由这个采集包的



优先级决 定了。而采集包优先级取自所有组合变量的最高优先级。上述的例 子中,如果变量地址是连续的,并且是同种寄存器,则三个变量有 可能组合成一个包,这种情况下,包的优先级为三个变量中的最高 优先级(即100),三个变量的优先级都变成 100 了。

6.9.3 诊断

未来计划用户点击 IO 驱动"诊断"按钮后,列出 IO 采集包的打包情况,包括:共打成的包个数、每个包由哪些变量组合成、包的优先级等。

6.9.4 优化采集

为了提高采集速度, I0 驱动提供了"优化采集"选项, 只有窗口的图 形 有连接或者历史记录需要保存的变量, 系统才会采集。

注:优化采集时,脚本中调用"_取变量值"的变量不会被采集,而脚本中 调用"_连接变量"的变量会被采集。

6.10 调试IO 设备通信

在驱动工具栏点击"运行"命令,可以启动驱动测试程序进行调试。双击 变量的值列,可以写变量值。如下图所示:



:\Documents and	. Settin;	gs\Administ:	rstor\桌面\OPC2.tdv 🗸	
	i	इंस्ट्री 🚺	停止	
动变量列表:	(
变量	ſ	1	描述	^
连接状态	1	2	表示与设备连接状	
Float, fl	0.	271889		
Float. f2	-0	0. 470698		
Float. f4	0.	982971		
Float. f6	0.0	663045		
Float. f8	0.	739189		
Float. fO	0.	567278		
Float. f3	50)		
Float. f5	1			
Float. f7	0.	19599		V
	-	BALAIF		
2的运行消息:				
时间		消息		^
2011-08-01 04	4:44:54	AddItems "	Float.x6"失败!错误号	
Ä		A 3 3T 4 44	87	

图 6-8

6.11 Modbus 驱动

6.11.1 概述

Modbus 通信是目前最常用的 IO 通信方式,也是国际标准的通信协议。目前系 统支持 Modbus RTU 和 Modbus TCP 协议,除了通道略有不同外,其它使用基本相 同。RTU 当前支持串口、TCP、UDP 三种通道方式;TCP 由于用户主要还是使用 TCP 通道,所以仅仅支持了 TCP 通道,但是如果用户需要,增加其它通道支持还是很简 单的。

新建 modbus 驱动时, modbus 驱动位于"IO 通信设备"的"通用设备"节点下。有关驱动、变量的创建、组态等请参考前面相关章节。本节主要介绍 Modbus 驱动变量的两个主要属性选项:寄存器类型和 寄存器地址。如下图所示:

寄存器类型	寄存器地址
CS (1)	1
CS(1)	2
IS (2)	3
IR(4)	4



寄存器类型:目前包括类型如下表所示:

类型	类型说明	读功能码	写功能码	变量数据类型	备注
CS(1)	Coil Status(线圈状 态)	1号功能码	5号功能码	位操作,类型为BOOL,即开关量	
IS(2)	Input Status(输入状 态)	2号功能码	只读,不可写	位操作,类型为BOOL,即开关量	
HR(3)	Holding Register(保 持寄存器)	3号功能码	6(单字)、16 (多 字) 号功能码	模拟量,可以是WORD、SHORT、INT、 DWORD、FLOAT、DOUBLE	DOUBLE类型只有支持8个字 节的设备才有效
IR(4)	Input Register(输入 寄存器)	4号功能码	只读,不可写	模拟量,可以是WORD、SHORT、INT、 DWORD、FLOAT、DOUBLE	DOUBLE类型只有支持8个字 节的设备才有效

图 6-9

寄存器地址:变量所要读写的对应的设备寄存器地址,地址前不必带功能 码。

6.11.2 Modbus 多站

Modbus 多站允许一个串口下连接多个 Modbus 设备。Modbus 多站的每个变量 多了"设备地址"属性列,用来设置该变量对应的站地址, 而原来 Modbus 单站 驱动属性栏"设备地址"属性则被取消。另外, modbus 多站除了"连接状态"的 内置变量外,还具有每个站连接状态的内置变量。只有所有的站的通信中断时, 才认为设备通信中断, 否则只要有一个站的通信是正常的,则设备的"连接状态"为True。

			1	变量	类型	描述
			8	动内置变量		
				连接状态	BOOL	表示与设备连接状
语名地址	安方與米刑	客方架地址		站1连接状态	BOOL	表示与站1连接状
COM HRAT	可打破大空	017064646		站2连接状态	BOOL	表示与站2连接状
1 🗘	Coil Status	0		社会在培州太	BOOT	事会与社会在接出
	G 13 G1 1		4	3H J/EIST/V/JES	DOOL	NUM-DAD DEDUCT

由于多个设备共享同个串口,当某个设备站通信异常时的超时等待会影响其它 设备站。为了尽可能降低影响,属性栏增加了"站首次等待、站增加等待、站最 长等待"属性。其原理就是降低异常站的通信频率,而让通信机会留给其它站。

站首次等待:当检测首次某站通信异常时,该站尝试下一次通信的 等待间 隔,单位为秒。

站增加等待:如果站连续通信异常,则不断增加尝试通信的间隔时间。该值为每次增加的幅度,单位为秒。



站最长等待:如果站连续通信异常,不断增加尝试通信间隔时间后 的最大限 值,单位为秒。

如果上述属性值越大,则异常站的尝试通信频率越低,正常站的频率就高了。 但是由于检测异常的频率降低,异常站恢复正常站的时间就变长了。可以根据具体 情况进行配置。

例如下表的设置,当第一次连接失败后,等待10秒进行连接,如果再次失败,则等待15(10+5)秒,如果连续失败则20、25、
30、...不断递增等待时间,直到100秒。

6.11.3 特殊函数

批量写值: 批量写 *I*/*0* 值。批量写值(起始地址,终止地址,寄存器类型,值数组, 值类型)。

寄存器类型:目前支持"HR(3)"、"CS(1)"。

值类型:目前支持"WORD"(寄存器类型为"HR(3)"时)、"BOOL"(寄存器 类型 为"CS(1)"时)。当采用"HR(3)"类型时,功能码为10(十六进制)/16(十进制); 当采用"CS(1)"类型时,功能码为F(十六进制)/15(十进制)。

如:

arr1 = new Array(1, 2, 3, 4, 5, 6, 7, 8, 9, 10); _批量写值(6, 15, "HR(3)", arr1, "WORD"); arr2 = new Array(1, 0, 0, 1, 0, 1, 1, 1, 0, 0); _批量写值(3, 12, "CS(1)", arr2, "BOOL");

6.11.4 特殊设备

一些特殊 Modbus 设备(如:台达)连接时会产生问题,可尝试如 下方法:

1. 部分设备可能采用 "ASCII" 方式通讯,可设置驱动的 "ASCII" 选项为 "是"。

2. 一些设备的文档描述或者用其它组态软件通讯时,寄存器地址如果为 4XXXX、40XXXX、3XXXX、30XXXX 等,则此处的 4、3 指的是 4 区、3 区地



址,不是功能码!!!一般来说,4 区需要 3 功能码读,而3 区需要 4 功能码读。前文提到,本系统的 HR(3)、IR(4)后面的 3、4 是功能码。因此,如果地址是 4XXXX,则选择 HR(3),地址是 3XXXX,则选择 IR(4)、同时寄存器地址输入 XXXX,不要前面的 4 或 3!!!

3. 部分设备的寄存器地址比 Modbus 标准多1,可以将驱动"基地址"选项设置为1。

6.12 Modbus 从站

Modbus Slave (Modbus 从站)是将当前工程做为从站,从而数据库 端变量 能通过 Modbus 协议提供给外部程序访问,外部程序是一个 Modbus 主站。

新建 Modbus 从站驱动时, modbus 从站驱动位于"IO 通信设备"的"通 用设备"节点下。目前已经实现了 Modbus RTU 从站和 Modbus TCP 从站,可以 通过串口或者 TCP 通道来访问。

新建 ModbusSlave 变量后,选择要供外部访问的数据库变量,并设置变量对 应的寄存器类型、地址、值类型,外部 Modbus 主站程序 就可以通过这些寄存器 类型、地址、值类型来读写系统的数据库变量了。

变量的寄存器类型、地址说明请参考上一节《Modbus 驱动》中的说明。

6.13 OPC 驱动

OPC 通信包括早期的 OPC 标准(OPC DA)通信、新的 OPC 标准(OPC UA)通信。由于 OPC UA 是新的标准,所以目前支持该设备的厂家并不多。 系统目前也是仅支持了旧的 OPC 标准,即 OPC DA。

下面所有提到的 OPC 技术,如果没有特别声明,都是指 OPC DA。 OPC 通信与其它协议通信有很大的不同,它是基于微软的 COM/DCOM 技术制定 的,因此它严重依赖于 Windows 平台和 COM/DCOM 技术。另外, OPC 的组态 开发与其它通信驱动区别也较大,下面介绍 OPC 驱动的组态开发。

6.13.1 新建 OPC 驱动

新建 OPC 驱动与其它新建驱动类似, OPC 驱动位于"IO 通信设备"的"通用



设备"节点下。

6.13.2 配置服务

与其它驱动不同的是, OPC 驱动得先配置 OPC 服务器, 才能完成接 下来的 配置。点击工具栏"配置服务", 弹出 OPC 服务器对话框, 如下图所示:

配置OPC用	务番	X
主机: 服务器: 服务器cls 这种情况	ICONICS.SimulatorOPCDA.2 id: G远程访问时,本机不一定知道远 F组态时就应该设置好Clsid,如Wince	(为空表示本机) 程服务器的Clsid, 访问远程服务器)
(A879768) OPC服务器 ■ OPC II GR BR BR M	<pre>k-7387-11D4-B0D8-009027242C59} 列表: NA 2.0 raybox. Simulator. 1 sijer. ElectronicsOPCServer. 1 :ONICS. SimulatorOPCDA. 2 strikon. OPC. Simulation. 1</pre>	確定取消

图 6-11

- **主机**:远程 OPC 服务器主机 IP 或者主机名,空表示本机。
- **服务器:** OPC 服务器名称,如果 OPC 服务器在本地也安装了,则会显示在下面的列表中,可以从列表中选择。
- CLSID: 如果 OPC 服务器在本地也安装了,则选中列表中某服务器时, 会自动检测出该 OPC 服务器的 CLSID。

6.13.3 增加变量

点击工具栏"增加变量"按钮,增加变量后,点击该变量行的"..."变量选择 按钮,弹出变量选择对话框,选择要监控的变量即可,如下图:





图 6-12

6.13.4 批量增加

点击工具栏"批量增加"按钮,弹出如下对话框,选择要监控的变量 后,点击"=>"按钮,批量增加变量。

选择OPC受量 PC服务器中的变量: 曰 Gravbox.Simulator.1	选中要监控的变量: numeric.square.int16	□ 增加到新组: 变量其它项初始	 í值:
bool double float int16 uint12 double float int16 uint12 double float int16 uint12 double float double float double float double float double float double float double float double float double float	<pre>numeric square.int32 numeric square.uint32 numeric triangle.int3 numeric triangle.int8 numeric triangle.uint</pre>	内容 类型 读写方式 报警 历史记录 操作记录 保存值 数据转换 描述	值 INT 读写

图 6-13



6.13.5 别名

默认情况下驱动的变量名为 OPC 服务器的变量名,但是某些特殊情况下(例 如,OPC 服务器的某个内置变量名带"/",这个变量又需要 参与表达式计算,而 在表达式中,符号"/"为除号,这样就会导致表 达式计算出现错误),用户不希 望变量名为 OPC 服务器内置变量名,这时就可以使用别名来替代了。使用别名 后,别名就成了驱动的变 量名了,这样一来,变量选择对话框、表达式、脚本等 模块就可以 采用别名来引用该变量了。

6.13.6 主要问题

单机版本下ActiveX 与 OPC Client/Server 冲突。默认情况下, 启用 ActiveX 功能, 禁用 OPC 功能。在安装目录下, 生成一个名为"mtcom"的 空文件, 则启用 OPC, 禁用 ActiveX。

网络Web 版本没有该问题。

6.14 OPC 服务器

6.14.1 概述

系统提供了 OPC 服务器供其它 OPC 客户端访问数据库端内核变量。 与大部分组态软件不同的是,本系统无需输出所有变量,用户通过 配置,可以仅仅输出外部 OPC 客户端需要用到的变量。

6.14.2 配置

在工程数据库浏览窗口中双击"系统配置"节点下的"OPC 服务"子节点,打 开 OPC 服务配置对话框后即可进行配置。对话框中设置需要输出 为 OPC 客户端访 问的变量。如下图,在需要输出的变量项目的复选 框打勾即可。



	确定
 ● ● 二年 ● 二日 ● □ □<th>取消</th>	取消
 京統模块 定制变量 家統模块 家証明变量 家証 家証 家証 家証 家証 家証 家証 承証 承証 承証 承証 承証 承証 承証 金属示变量 予約 予約 予約 	

图 6-14

6.14.3 服务器名和 CLSID

目前 OPC 服务器尚未支持 WINCE。Windows 单机版和网络 Web 版 的 Opc 服务器的服务器名和 CLSID 分别如下:

单机版: TsRun. OpcServer. 1、{ADC4203A-ABBF-4B28-926C-8AF8C09F7104}。

网络版: TsDatabase.OpcServer.1、{DFFCC3D7-5289-425C-9296- FCAB9336A54E}。

6.14.4 主要问题

单机版本下ActiveX 与 OPC Client/Server 冲突。默认情况下, 启用 ActiveX 功能, 禁用 OPC 功能。在安装目录下, 生成一个名为"mtcom"的空文件, 则启用 OPC, 禁用 ActiveX。

网络Web 版本没有该问题。

由于目前公司网络 Web 版本的市场前景远超单机版本,就销售额而言,相对 于网络 Web 版本,单机版几乎可以忽略不计。所以目前没有解决这个问题的计 划。



6.15 万能协议驱动

万能协议驱动即自定义协议驱动。不少驱动的协议不复杂,如果专门为这些 协议去开发一个驱动,则成本过高,而且更为麻烦。万能协议驱动就是让用户通 过自定义配置协议,不必编程开发,即可实现某个协议的驱动功能。但是万能驱 动也要求用户对要配置的协议比较了解。

另外,还要说明的是,由于市面上的驱动实在太多了,万能驱动不可能真正 的"万能",但是它是一个可扩展的驱动,会逐步支持越来越多的协议。

6.15.1 万能驱动原理

众所周知,一般协议驱动是通过发送请求包,而 I0 设备通过应答该 请求 包,来完成 I0 设备内存读写的。在系统的驱动架构体系中,驱 动后台不断读取 I0 设备内存值,一旦用户修改了某个 I0 内存对应 的变量值,则驱动启动写 I0 值请求。

与之相对应,万能驱动目前主要分成两种协议包:

常规包:即后台不断读取 I0 设备内存值的包;

写值包:将 I0 内存对应的变量值写入 I0 内存。

两种协议包都包括发送请求包和接收应答包两种格式定义。

下面将分别介绍这几种包。在介绍这几种包之前,先简单介绍一下万能协议驱动包的定义。

6.15.2 包及包域

每个万能协议驱动包都是由一些域组成,这些域有固定的字节数,并都有 一定类型,例如: ASCII、字符串、读出变量的值、写入变量的值、变量地址、 变量名、校验码等类型。万能驱动会根据这些域类型,对域内容进行相应的处 理。



6.15.3 常规包

万能协议驱动运行时会不断循环调用常规包队列,根据包的处理"模式"采 取对应的策略。目前有三种"模式":发送、接收、脚本。本小节主要介绍发送、接收,脚本模式请参考下面"脚本模式"小节。



发送请求包: 发送请求包比较简单,根据具体协议将包顺序对应到 域即可。 例如,要利用 modbus 读取设备 ID 为1,地址从 0~3 的4 个 WORD 保持寄存器 (Holding Register)值。那么发送包可以分成 5 个域:

1. 设备 ID 域 (字节数-1、类型-ASCII、内容-1);

- 2. 功能码域 (字节数-1、类型-ASCII、内容-3);
- 3. 起始地址域 (字节数-2、类型-ASCII、内容-00 00);
- 4. 寄存器数域 (字节数-2、类型-ASCII、内容-00 04);
- 5. 校验域(字节数-2、类型-校验、内容-CRC16)。

因为前4 个域都是ASCII 类型,实际上它们可以合并成一个域,但是 这样可 读性不好,所以分成4 个域描述。

接收应答包:接收应答包相对发送请求包复杂一些,主要是接收回来的数据, 必须读入到对应的变量中。万能驱动提供了按照地址和 按照变量名的对应方式, 可以通过域类型来设置对应方式。如果返 回批量连续地址则一般采用地址映射方 式(这种情况下,也可以采 用变量名对应,但是配置较繁琐);如果返回包无地址 信息(部分 I0 设备通信协议很简单,无地址信息),则必须采用变量名映射方 式。

一般说来,接收包主要作用有两个:一是将 IO 指读入变量中;另一个就是对接收到的包与预先定义好的域值进行比较,如果不相等,则认为接收错误,对于一些不必要比较核对的域,可以将该域的类型设置为"任意值"。

在上述发送请求包的例子中,假如万能驱动已经创建了4个变量,其变量名为 W0、W1、W2、W3,其地址为0、1、2、3,如果按照地址对应,则各个域配置如下:

1. 设备 ID 域 (字节数-1、类型-ASCII、内容-1);



2. 功能码域 (字节数-1、类型-ASCII、内容-3);

- 3. 字节数域 (字节数-1、类型-ASCII、内容-8);
- 4. 变量值域(字节数-8、类型-按地址读值、内容-0[~]3);
- 5. 校验域(字节数-2、类型-校验、内容-CRC16)。

如果按照变量名对应,则各个域配置如下:

1. 设备 ID 域 (字节数-1、类型-ASCII、内容-1);

- 2. 功能码域 (字节数-1、类型-ASCII、内容-3);
- 3. 字节数域 (字节数-1、类型-ASCII、内容-8);
- 4. 变量值域0(字节数-2、类型-按名读值、内容-W0);
- 5. 变量值域1(字节数-2、类型-按名读值、内容-W1);
- 6. 变量值域2(字节数-2、类型-按名读值、内容-₩2);
- 7. 变量值域3(字节数-2、类型-按名读值、内容-W3);
- 8. 校验域(字节数-2、类型-校验、内容-CRC16)。

6.15.4 写值包

写 I0 值包与常规包类似,就是多了一个"类别"的概念。为了灵活地 实现写 I0 值功能,万能驱动引入了一个变量的重要属性——"类别"。 因为不同的变 量其写 I0 设备的协议格式、形式可能不同,这个"类别"就是用来定义这个不同 的。把相同协议格式、形式的写 I0 值的 变量统一成某个"类别",当系统检测到 需要往 I0 设备写 I0 值时,万能驱动就调用该变量对应的"类别"协议进行通 信。因此,可以把"类别"理解成包模板。

通俗来说,就是每个写值包都有类别,每个变量也可以设置成某类别。当 该变量值改变时,万能协议驱动会自动调用变量对应的类别包发送。

一般而言,需要把不同设备 ID、寄存器类型、数据类型的变量归成不同类别。例如,对于上述的地址从 0~3 的4 个 WORD 保持寄存器,可以设置成一个类别,如: "WORD 保持寄存器"。如果要写 FLOAT 保持寄存器,则得另外设置成"FLOAT 保持寄存器"类别。

写值接收包与常规接收包类似,如果不希望对接收包某些域进行核对比较,可以将这些域的类型设置为"任意值"。

下面以"WORD 保持寄存器"类别为例说明如何配置各个域。



发送包:

- 1. 设备 ID 域 (字节数-1、类型-ASCII、内容-1);
- 2. 功能码域 (字节数-1、类型-ASCII、内容-6);
- 3. 地址域 (字节数-2、类型-地址);
- 4. 值域(字节数-2、类型-写入值);
- 5 校验域(字节数-2、类型-校验、内容-CRC16)。

接收包:

- 1. 设备 ID 域 (字节数-1、类型-ASCII、内容-1);
- 2. 功能码域 (字节数-1、类型-ASCII、内容-6);
- 3. 地址域 (字节数-2、类型-任意值);
- 4. 值域 (字节数-2、类型-任意值);
- 5 校验域(字节数-2、类型-校验、内容-CRC16)。

6.15.5 配置驱动及协议

理解了上述万能协议的原理和概念后,就可以进行配置了。

新建万能协议驱动时,万能协议驱动位于"I0 通信设备"的"通用设备" 节 点下。有关驱动、变量的创建、组态等请参考前面相关章节。万能协议驱动变量有 一个重要属性选项: 类别,其概念和使用方法上面已经介绍了。

点击万能协议驱动工具栏的"协议配置"命令按钮,就可以进行协议 的自定义 配置了。如下图所示:

植人包	開除し		リート移 isto				見名 tate
白柳	1英355	現1	19%2 1+# クロー	パン (1)	現4	8%5	<i>1</i> 5%0
自我1-3-0	友达	現名(設	戦名(功)	现名(地	现名(奇	现名(LKL	
⊻告1-3-0	授収	现名(股	戦名(切)	现名(子	现名 受…	现名(150	
育马(1-3-1	友达	现名(版	戦名(功)	现名[地	现名(苛	现名(LKL	
		1.000				and the second	



图 6-15

名称	模式	域1	域2	域3	域4	域5	域6
WORD保持寄存	24						
请求1-3-0	发送	域名(设	域名(功	域名(地	域名(值)	域名(校	
应答1-3-0	接收	域名(设	域名(功	域名(地	域名(值)	域名(校	
FLOAT保持寄存	器						
请求2-3-0	发送	域名(设	域名(功	域名(地	域名(值)	域名(校	
应答2-3-0	接收	域名(设	域名(功	域名(地	域名(值)	域名(校	

图 6-16

对话框中可以对常规包和写变量包进行配置。写变量包比常规包多了一个配置 步骤,即:得先增加写变量包的类别,然后对该类别的 写变量包进行配置。

点击域配置,弹出如下域配置对话框:

城名:	设备ID	确定
■节数:	1	取消
类型及对	拉 应内容	
类型:	ASCII	~
类型: ASCII常	ASCII 量值,输入内容以 S核分隔开,加:1	十六进制表示,

图 6-17

不同包的域类型略有区别。下面4个列表框分别是常规发送包、常规接收包、 写值发送包、写值接收包的域类型。





所有包都具有 ASCII、字符串、校验、表达式四种类型。

接收包有任意值类型,表示接收到的包的域不需要核对检查。

常规接收包还具有读出值类型,可以将收到的包读出到变量中,可以根据变 量名或者变量地址来读。

写值发送包还具有地址和写入值类型,这两个类型都不需要填写内容,万能 协议驱动自动会填入当前要写 I0 的变量地址和变量值。

6.15.6 脚本模式

在某些情况下,包或者域内容有可能是不定长的,这种情况下,无 法用填充固定长度域的方式处理包。脚本的主要目的就是为了解决 这个问题。脚本允许用户采用 JavaScript 对包自行进行填充、解析,从而大大提高了万能协议驱动的灵活性。

假如一个包的名称为"包A",它采取了"脚本"的方式。则当驱动调度 到 该包时,包括常规和写值两种情况。如果是常规包则会自动调用 名为"**常规_包** *A"*函数;如果是写值包则调用 "写值_包 A(变量,地址,值)" 的函数。这两个函 数类似事件触发函数。用户接着可在这个函数中 实现对"包A"的填充、解析。 如果该包对应函数返回"false",则万能 驱动认为通信错误,并进行错误控制处 理。注:上述二个函数声明 当点击域中的按钮时会自动生成,用户无须手动增 加,如下图所示。



另外,得说明一下写值包的触发函数的三个参数:变量,地址,值。 这 三个参数分别表示当前要写值变量的名称、地址和值。当发送写 值包时,可 以根据"变量、地址、值"的情况进行发送。

为了有效地实现脚本功能,系统还提供了如下几个函数供调用。

- 1._接收字符串(接收长度,接收所有):接收指定长度的字符串。
- **接收长度**:要接收字符串的长度。
- 接收所有:是否等待接收指定长度字符串后才返回,如果为"true"
 或"1",则该函数一 直等待接收参数1所指定的长度后才返回,否



则只要有收到字符则立即返回。

● **返回值**:成功则返回接收到的字符串,失败则返回 null。如果"接收 所有"为"true",则只有接收到所有字符才认为是成功。例如:

```
接收串 = _接收字符串(300, false);
if(接收串 != null)
{
//进行分析
}
```

2._接收字节数组(接收长度,接收所有):接收指定长度的字节数组。与函数 "接收字符串"类似。如果对方主机返回的是字符串则调用"_接收字符串"函数,否则调用该函数。

- **接收长度:**要接收字节数组的长度。
- 接收所有:是否等待接收指定长度字节数组后才返回,如果为 "true"或"1",则该函数一直等待接收参数1所指定的长度后才 返回,否则只要有收到字节则立即返回。
- **返回值**: 成功则返回接收到的字节数组,失败则返回 null。如果"接 收所有"为"true",则只有接收到所有字节才认为是成功。例如:

```
返回内容 = _接收字节数组(300, false);
if(返回内容 != null)
{
接收数组 = 返回内容.toArray();
//对"接收数组"进行分析
}
```

3. _发送字符串(字符串, 0 结束[可选, 默认 false]): 发送指定的字符串。 如果 "0 结束"参数为 true,则自动在字符串结尾添加串结束符 "0" (不是字符 0)后发送。默认不填写则为 "false",即不添加 "0"发送。

4. _发送字节数组(字节数组):发送指定的字节数组。

如:

var byteArray = new Array(10, 20, 30, 40, 50, 60); _发送字节数组(byteArray);

5._调试输出(内容): 在列表框显示输出调试内容。

6._取变量值(变量名):设置指定变量的值,返回变量值。



7._设变量值(变量名,变量值):设置指定变量的值。

关于接收函数:

1. 如果通道类型是"TCP/UDP",则"接收所有"可以填"false","接收 长度"填写一个足够大的数字即可,因为TCP/UDP的数据一般是整批同时到达的。

2. 如果通道类型是"串口",则"接收所有"填"true","接收长度"必须 填写准确的数字,因为串口的数据是一个字节一个字节陆续到达的,如果一次性无法估算准确应答字节数,可以先接收部分数据,然后 逐步分析,并接收后面的数据。

3. 如果接收的数据为 null 或者格式不正确,需立即返回"false",用 来通知万能驱动本次接收超时或者数据有误,驱动会采取相应措施 进行错误 控制处理。

6.15.7 其它说明

组:常规包可以分组,除了方便管理外,还可以与串口的单片机校验配合使用。此外,组还有一个重要功能:组里面的所有收发操作不会被写 IO 值包中断。由于写 IO 值包具有高优先权,在非组情况下,如果某个读值操作由 3 个或者 3 个以上的包构成,第 3 个包或 者第 3 个以后的包有可能被写 IO 值包中断。

单片机校验:万能协议驱动的串口通道拥有特殊的"单片机"校验, 组 (常规包)或者类别(写值包)的第一个包是标记(MARK,1) 校验,后面包 是空格(SPACE,0)校验。

6.16 SNMP 驱动

SNMP(Simple Network Management Protocol,简单网络管理协议)的 前身是 简单网关监控协议(SGMP),用来对通信线路进行管理。随后,人们对 SGMP 进行 了很大的修改,特别是加入了符合 Internet 定义的 SMI 和 MIB:体系结构,改进 后的协议就是著名的 SNMP。SNMP 的目标是管理互联网 Internet 上众多厂家生产 的软硬件平台,因此 SNMP 受 Internet 标准网络管理框架的影响也很大。现在 SNMP 已经 出到第三个版本的协议,其功能较以前已经大大地加强和改进了。 新 建 SNMP 驱动时, SNMP 驱动位于"IO 通信设备"的"通用设备"节点 下。有关


驱动、变量的创建、组态等请参考前面相关章节。

SNMP 变量主要有个"OID"(对象标识符)属性,标识是设备中用来代表一个对象的数字,每个标识都是整个设备中唯一的,也就是说,标识不会、也不允许重复。SNMP 驱动主要就是根据OID 通过发送 SNMP 协议请求从设备中读写值。如:"1.3.6.1.2.1.1.1"。

6.17 IO 通信驱动开发包

目前系统 IO 通信驱动较少,但是提供了驱动开发包,用户可以根据 实际 应用自行增加驱动,详情请参考《SDK 开发包》章节。



七、窗口与图形

7.1 概述

窗口是画面的主要部分。工程监控的操作、动画主要就是通过窗口来完成 的。 除了提供方便、易用的可视化组态开发外,系统的窗口和图形还具有如 下特点:

自由缩放、拖拽:窗口可以自由缩放,当窗口放大后,也可以自由拖动漫游。目前具备该功能的组态软件很少见。

高级界面技术:包括透明、过渡、旋转/倾斜、反锯齿等高级界面技术,这些技术在 Windows 上利用 GDI+实现并不困难,然而在 Wince、 嵌入式 Linux 上实现则较为少见。而且经测试,本系统所采用图形技术的效率要高于 GDI+的效率。

动画效率超高:目前大部分组态软件的画面刷新频率得到秒级,然而本系 统刷新频率为10 毫秒级,但是 CPU 仍然可以保持非常低的 水平 (接近 0), 画面操作依然非常平滑。

超大图优化: 当窗口长宽达到几千甚至上万的分辨率、并且图形数量达到 几百甚至上千时, 通过窗口优化技术,仍然可以保持超高的动画效率。

7.2 新建窗口

在创建新的工程时,系统已经默认创建了主窗口。还可以采用如下方法创 建新的窗口。

7.2.1 从工程画面浏览窗口创建

在工程画面浏览窗口中右键点击"窗口"节 点弹出菜单后,点击"新增 窗口",弹出新增窗 口对话框。





建窗口	
印名称: 窗口1	确定
窗口大小	取消
	and the second se
 根据窗口坐落区域设置窗口大小 	<u> </u>
 ● 根据窗口坐落区域设置窗口大小 主区域 - 1180x600 	
 ●根据窗口坐落区域设置窗口大小 主区域 - 1180x600 ●手动设置窗口大小 	

图 7-1

在新增窗口对话框中,输入窗口名称和窗口大小。

根据窗口坐落区域设置窗口大小:即根据工程设置的画面分辨率以及栏框 的宽度或高度来计算各个区域的大小。

7.2.2 通过命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。

 类别: ○ 画面 斎 系统模块 ○ 認 数据库 ◎ 系统模块 ○ 10设备通信 ○ 通用设备 ● PLC 	新建文件				X
	类别: ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	文件类型:	三维窗口	定制图形	
二维平面窗口。 図 将新建的文件添加到当前工程 新建文件名:	二维平面窗口。 2 将新建的文件添加到当前工程 新建文件名: 确定		i		

这种方法创建的窗口无法指定窗口大小,不建议使用。一般这种方法用来创建临时窗口文件。



7.3 窗口分组

当画面窗口较多时,就可以采用分组来管理了。

7.3.1 新增窗口组

在工程画面浏览窗口中右键点击"窗口"节点弹出菜单后,点击"增加窗口 组",将新增一个窗口组,输入窗口组名称。



7.3.2 窗口组的窗口文件

右键点击窗口组,弹出菜单后,点击"新增窗口",将在该窗口组下新增窗 口文件。 也可以通过拖拽,在窗口组之间拖动窗口文件,使得窗口放置到其 它 窗口组中。如下图所示:



7.4 窗口组态开发

新建窗口后,就可以开始窗口的设计组态工作了。



实际上,窗口本身的组态工作不多,主要还是窗口上大量的图形组态开发,包括图形拖放位置、图形属性配置等。

7.4.1 属性设置

窗口所有属性都显示在属性栏中,对于每个属性的描述,点中该属 性后,会提 示在属性的下端。如下图所示:

-	基本属性	
	名称	窗口1
	层显示	OxFFFFFFFF
	忧化	否
	描述	
Ξ	大小	
	宽	1180
	高	600
	缩放最小值	0.500000
	缩放最大值	2.000000
Ξ	外观	
	背景色	ffffff
	过渡	无过渡
	过渡色	ffffff
	背景图	
	背景图显示	按背景图大小
	透明色	📘 无颜色
	过渡色 背景图 背景图显示 透明色	

下面主要解释一下几个不易理解的属性:

层显示:窗口拥有 32 个层,每个图形都隶属某个层,为了方便窗口图形开发,有时需要隐藏一些不需要编辑的图形,这时就可以通过 隐藏层的方法来实现。点击层显示属性的"..."按钮,弹出层显示对话框后,设置要显示的层即可。如果设置了隐藏某些层,则隶属于这些层的图形在组态开发时就看不见了。

缩放最大/小值:窗口可以自由缩放,但是在运行状态下,有时用户不希望随意缩放,这时就可以通过设置缩放最大/小值来限制。

优化:默认情况下,当窗口检测到某个区域改变时,会检测与该区 域相关的图形,然后进行重画。如果窗口拥有成百上千的图形时,检测区域的相关图形时间



较长,这时就必须通过优化来实现了。在 窗口设置了优化属性的情况下,所有图 形默认不重画,除非该图形 设置了重画属性。所以仅仅将那些运行时有可能会变 化的图形设置 为重画,这样就大大地提高了窗口刷新效率。

<u>注: 该优化属性仅仅 对拥有大量图形、但是需要重画的图形仅仅占到整个窗口图</u> 形的 30%以下的窗口才比较有效。

7.4.2 事件配置

窗口所有事件都显示在属性事件栏中,对于每个事件的描述,点中 该事件 后,会提示在事件的下端。如下图所示:

雇性栏	×
國田	~
含属性 🔡 动词	画 多 事件
日 基本事件	
打开	1 50
关闭	
定时器	
按下键盘	
弹起键盘	
输入字符	
打开 事件 当窗口打开时触	- <mark>描述</mark> _{发该事件。}

7.4.3 其它操作

如下图系统工具栏部分窗口相关操作,可以组态时对窗口进行自由 缩放、 设置为主窗口(房子图标)、编写 JavaScript 脚本(J 字图标) 等。

)⊃ 100% ▼) ÷	
-----------	-----	--

编写脚本也可以通过右键点击窗口空白处,弹出菜单后,选择"编写 JavaScript"。





7.5 图形组态设计

7.5.1 概述

窗口仅仅提供了图形的容器,真正要实现复杂的监控,还必须通过图形来实现。系统目前提供的图形较少,但是未来会逐步增加。对于系统尚未提供的复杂图形,也可以采用组合、定制图形等方法实现。

7.5.1.1 工具箱

系统所有图形/控件都位于工具箱中。当打开窗口时,自动会列出这些**图形**/ **控件**。如下图所示:





7.5.1.2 布局工具栏

图形/控件的部分操作(包括对齐、同大小、置前/后、居中、等间 距、镜 像等)得通过布局工具栏实现,如下图所示:

布局		▼ X
┣╣╦╩ ╦	•11 භ ங 🛍 🏎 දු	₽₽₽₽

7.5.2 新建图形

所有图形创建都得先点击工具箱中要创建的图形/控件,然后按住鼠标进行拖拽。

7.5.2.1 普通形状图形创建

大部分图形/控件的创建,只要根据图形大小在窗口中拖拽出橡皮筋 矩形 即可,也可以直接点击生成默认大小的图形。如图:



7.5.2.2 特殊形状图形创建

直线创建:先点击窗口为直线的起始点,按住鼠标进行拖拽释放后,为直线的终点。

折线、多边形、样条曲线的创建:这些图形都有一个共同特征,就是它们都是 由多个基点生成。这些图形创建过程中,鼠标可以不断点击产生一系列基点,最 后鼠标双击生成最后一个基点后,结束图形创建。

7.5.3 鼠标键盘操作

图形的选择、位置调整等可以通过鼠标、键盘操作来完成。

图形选择 可以直接用鼠标点中某个图形来选择单个图形,或者用鼠标拖出



Web:http://www.witlinc.cn E-mail:support@witlinc.cn

橡皮筋矩形来选择多个图形。 如果选择时,按下"Ctrl"键,则会切换图形的 选择状态。即原来选中 的图形,就会从选中图形列表中移走;原来没选中的图 形,会增加到图形选择列表中。

图形位置调整 对选中的图形用鼠标进行拖拽,会改变图形位置。

对选中的图形用"↑、↓、←、→"键,可以对选中的图形的位置进行微调。

7.5.4 旋转

当鼠标移动到图形的旋转操作位置时,会变成旋转图标,点击拖拽就可以旋转 图形了。如图:



7.5.5 倾斜

倾斜分成垂直倾销和水平倾斜,其操作位置分别位于图形的左边和 顶边。 当鼠标移动到图形的倾斜操作位置时,会变成倾斜图标,点击拖拽 就可以倾斜 图形了。如图:



7.5.6 对齐

对齐包括左对齐、右对齐、上对齐、下对齐,表示以某图形的左/右/上/下边 为基准对齐其它图形的左/右/上/下边。在布局工具栏中,如图:





这里我们以左对齐为例,说明如何对齐图形的。 图形对齐是以第一个选中的 图形为基准对齐的,下图的"按钮1"就是 第一个选中的图形,以实心选择框突 出显示。



点击左对齐后,"按钮2"和"按钮3"的左边就与"按钮1"一致了。



7.5.7 等大小

等宽、等高、等大小:表示将选择的图形设置为基准图形的相同宽 度、高度、大小。在布局工具栏中,如图:



这里我们以等宽为例,说明如何设置相同宽度的。

图形等宽是以第一个选中的图形为基准的,下图的"按钮1"就是第一个选中的图形,以实心选择框突出显示。



点击等宽后,"按钮2"和"按钮3"的宽度就与"按钮1"一致了。





7.5.8 置前、置后

置前、置后:调整选择图形的显示顺序。在布局工具栏中,



如图: 下图的"按钮1"原先是位于"按钮3"之后显示。



点击"置前"后,"按钮1"就在"按钮3"之前显示了。



再点击"置后",又恢复原来的显示顺序了。



7.5.9 等间距

水平等间距、垂直等间距:将选择的图形在水平/垂直方向上设置为相同间 距。在布局工具栏中,如图:

布局 X [월 리 파 프] 글 킨 昭] 💁 💁 [👓 흥] 🕀 🗗 🗛 考

这里我们以水平等距为例,说明如何设置等间距的。 下图的三个按钮原来的间距不同。



按钮1 按钮2 按钮3

选中它们,并点击水平等距后,它们的间距就相同了。



7.5.10 居中

水平居中(中心分布)、垂直居中(中心分布):将选择的图形放置 在窗口 的水平/垂直方向上的中间位置。在布局工具栏中,如图:

布局		▼ X
時 引 迎 平 二	희 표 💁 🗞 🚥 흫	✐₽▲◀

这里我们以水平居中为例,说明如何设置图形居中的。 下图的三个图形原来 位于窗口偏左的位置。



选中它们,并点击水平居中后,它们就位于窗口水平方向上的中间位置了。





7.5.11 镜像

水平镜像、垂直镜像:将选择图形的形状在水平/垂直方向上的进行翻转。 一般只有非矩形形状的图形才能被镜像。在布局工具栏中,如图:

布局 [월] 朝 帝 幽 | 宗 朝 段 | 월 월 | 四 홍 | 단 臣 [(A

这里我们以水平镜像为例,说明如何设置图形镜像的。 下图为镜像前的图形 形状。



选中它们,并点击水平镜像后,所有图形沿总图形的中心线翻转了一遍。



7.5.12 锁住

锁住:将选择的图形设置为组态时不允许鼠标、键盘编辑操作。主 要包括:拖拽、修改位置、大小等。可以用"解锁"操作解除。在系 统工具栏、图形属性或鼠标右击图形设置,如图:



雇性栏 ×	
@ 按钮1 ✔	
一番 一番 小画 / 事件	
安全区	🔒 锁住
 锁住 否	- 『 解锁
· 房0	Affity

当图形被锁住后,鼠标移到上面时会出现锁住的图标,如图:



7.5.13 组合

可以将多个图形组合成一个图形对象,以方便组态编辑。组合后的做为一个 完整对象可以有自己的属性、动画、事件,也可以"取消组合",回到原先的状态。

选择要组合/取消组合的图形后,右键点击弹出菜单。如图:



组合后,原来的图形就成了这个新的组合图形的子对象,也可以对子对象直接 编辑组态,请参考"子对象"一节。

7.5.14 子对象

部分图形具有子对象,如趋势图的趋势线、饼图的子饼图、组合的图形、折 线、表格、菜单等。

7.5.14.1 选择子对象

有时候得选择某子对象编辑,可以通过下面方法:



 1. 通过鼠标点击选择:必须先选中该图形,然后再点击该图形中的子对 象,才能选中子对象。比如要选择趋势图的趋势线,就得先选中趋势图,然后 选择趋势线。

2. 也可以通过右键点击该图形,弹出菜单后,点击"子对象"菜单选项,弹 出子对象选择对话框后,选择要编辑的子对象。如图:



7.5.14.2 子对象属性、动画、事件

子对象属性、动画、事件的组态查看与编辑也分别位于属性栏的属 性、 动画、事件选项页中。

7.5.15 特有操作

部分图形具有特有操作,如趋势图增删趋势线、菜单增删子菜单项、表格 增删列、图像操作等。

当某图形拥有这些特有操作时,点击选中图形后,就会在图形左上角出现一个 小工具栏。

下图是菜单和趋势图被选中后,在左上角出现小工具栏的例图:





特有操作也可以通过右键点击图形,弹出菜单后点击操作选项,如图:





7.6 图形属性

7.6.1 概述

可以在属性栏中修改图形属性,每个属性的说明描述位于属性栏的下方,如 图:

7.6.2 通用属性

所有图形都有共有的属性,包括:

- 名称 用来标识该图形对象。首字符必须是字母或下划线,后面的字符
 必 须是字母、数字或下划线。
- **可见** 该图形运行时是否可见,闪烁、显 隐动画会用到该属性。
- 安全区结合当前登录用户的安全区来确定该图形的安全操作权限。如果该对象的安全区为空,则任何用户都可以操作;否则,当前用户要操作该对象时,只有该用户拥有该对象安全区列表之一的安全区时,才允许操作。请参考"用户与安全区"章节。
- 锁住 将选择的图形设置为组态时不允许
 鼠标、键盘编辑操作。请参考前 面"锁
 住"一节。
- **层** 与窗口的"层显示"配合完成。
- 窗口拥有 32 个层,每个图形都隶属某 个层,为了方便窗口图形开发,有时需 要隐藏一些不需要编辑的图形,这时就



可以通过隐藏层的方 法来实现。如果设置了隐藏某些层,则隶属于这 些层的图形在组态 开发时就看不见了。

 重画 与窗口的"优化"属性配合,只有设置了窗口的"优化"属性, 图形的 "重画"属性才有效。

默认情况下,当窗口检测到某个区域改变时,会检测与该区域相关的图形,



然后进行重画。如果窗口拥有成百上千的图形时,检测区域的相关图形时间较长, 这时就必须通过优化来实现了。在窗口设置了优化属性的情况下,所有图形默认不 重画,除非该图形设置了重画属性。所以仅仅将那些运行时有可能会变化的图形 设置为重画,这样就大大地提高了窗口刷新效率。

7.7 图形动画

7.7.1 概述

图形属性是静态的,可以通过脚本运行时动态修改属性,从而产生动画效 果。但是脚本开发较为麻烦,而图形的动画设置只要简单的配置就可以产生动 画效果了。

可以在属性动画栏中配置图形动画,点击对应动画的"..."即可进入某动 画选项的配置。每个图形的动画描述位于属性动画栏的下方,如 图:

1		(
		ΨIT
	2014年1月1日	
	内际	
	並認	0
20	項允	<u></u>
Ξ	移动	
	水半移动	
	垂直移动	
Ξ	缩放	
	水平缩放	
	垂直缩放	
Ξ	旋转	
	旋转	
Ξ	倾斜	
	水平倾斜	
	垂直倾斜	
C	ますす。	•
揮	济 列则 伊比	2

7.7.2 闪烁

大部分图形都有闪烁动画。当闪烁的条件表达式值为真时,图形会以指定的



闪烁间隔,不断的显示隐藏,从而产生闪烁效果。下图中 当变量"数据 库.vl.lr1"的值大于10时,表达式的值就为真,此时图形就以1000毫秒的间 隔产生闪烁动画了。

闪烁动画配置		
表达式	选择变量	表达式
数据库.v1.lr1 > 10		~
		×
闪烁间隔: 1000 毫秒 确定] [取消]]

7.7.3 填充

有背景颜色的图形一般都有填充动画。填充动画:图形的背景色会随着表达式 的值改变而设置为对应颜色。

下图中,当变量"画面.系统变量._秒"的值<=20时,表达式值<=1,则显示 红色;当变量"画面.系统变量._秒"的值<=40时,表达式值<=2,则显示蓝色; 当变量"画面.系统变量._秒"的值<=60时,表达式值<=3,则显示黄色。

面.系统变量秒	/ 20	
		>
充列表		
表达式值	颜色	- Hititur
1.000000	ff0080	
2.000000	00c0ff	一册除
3.000000	ffff57	

伟联科技
www.witlinc.cn

7.7.4 显隐

大部分图形都有显隐动画。当显隐的条件表达式值为真/假时,图形会条件设置显示/隐藏。下图中当变量"数据库.v2.br1"的值大于或等于10时,表达式的值就为假,此时图形就隐藏起来看不见了。

表达式	选择变量	表达式
数据库.v2.br1 < 10		<u>^</u>
<u><</u>		3
表达式为真时	-	
	ξ	

7.7.5 移动

大部分图形都有移动动画。移动包括水平移动和垂直移动。如下图。

水平移动动画			
表达式	C	选择变量	表达式
画面.系统变:	 量秒		<u>^</u>
<			×
最小值		最大值	
表达式:	0	表达式:	100 🔶
图形X座标:	0	图形X座标:	500 😂
		The bits	
	備定	- 取消	



表达式		选择变量	表达式.	
画面.系统变	量秒			^
				2
ζ.				>
最小值 ———		最大值		
表达式:	0	表达式:	100	**
图形Y座标:	0	图形Y座标:	800	*

- 最小值:表示表达式的最小值对应图形 X/Y 座标的最小值。
- 最大值:表示表达式的最大值对应图形 X/Y 座标的最大值。

7.7.6 缩放

大部分图形都有缩放动画。缩放包括水平缩放和垂直缩放。如下图:

- 最小值:表示表达式的最小值对应缩放百分比的最小值。
- **最大值**:表示表达式的最大值对应缩放百分比的最大值。
- **水平缩放方向:** 向左一右边不变, 左边变化; 向右一左边不变, 右边 变化; 左右一中间不变, 左右变化。
- 垂直缩放方向:向下一上边不变,下边变化;向上一下边不变,上边 变化;上下一中间不变,上下变化;



表达式 画面.系统变量	t秒	选择变量	表达式
٤.			2
最小值 表达式: 缩放百分比:	0	最大值 表达式: 缩放百分比:	100 🗘
缩放方向 ④ 向左 (
直缩放动画			
表达式 画面.系统变量	t秒	选择变量	表达式
表达式 画面.系统变量	t秒	选择变量	表达式
表达式 画面.系统变量 量小值 表达式: 缩放百分比:	t秒 0.89 10.78 📚	选择变量 最大值 表达式: 缩放百分比:	表达式
表达式 画面.系统变量 量小值 表达式: 缩放百分比: 缩放方向 ⊙向下	t秒 0.89 10.78 〇向上 (选择变量 (最大值 表达式: 缩放百分比:	表达式

7.7.7 旋转

大部分图形都有旋转动画。旋转动画会修改图形的"旋转角度"属性。

● **最小值:**表示表达式的最小值对应旋转角度的最小值。



● **最大值:** 表示表达式的最大值对应旋转角度的最大值。

能转动画				
表达式	1	选择变量	表达式.	
画面.系统变				
				3
3				2
最小值		最大值	100	
表达式:		表达式:	100	×
ルビキ57月1月 ・		加強に対応	100	
	UHLE J	PK(FI		

7.7.8 倾斜

大部分图形都有倾斜动画。倾斜包括水平倾斜和垂直倾斜,分别与 图形的水平、垂直倾斜角度相对应。

- **最小值**:表示表达式的最小值对应水平、垂直倾斜角度的最小值。
- **最大值**:表示表达式的最大值对应水平、垂直倾斜角度的最大值。

表达式		1	选择变量	表达式.)
画面. 系统变:	量秒				~
C					>
最小值			最大值		
表达式:	0	\$	表达式:	100	*
倾斜角度:	0	*	倾斜角度:	100	*
	0.2		-		



長达式		C	选择变量	表达式.	
画面.系统变:	量秒				^
<u>(</u>					>
最小值 ————————————————————————————————————			最大值		
表达式:	0	-	表达式:	100	*
倾斜角度:	0	*	倾斜角度:	100	*

7.7.9 图像切换

7.7.9.1 概要

图像切换动画是专门提供给"图像"图形使用的。图像切换分成两种: 单 个图像的各个部分之间的切换(图像切换)、多个图像之间的切换(多图像切换)。

7.7.9.2 图像切换

图像切换动画是单个图像的各个部分之间的切换,根据表达式值来 显示图像中的某部分。 首先把图像沿水平方向分成相等的几部分,将表达式值设置为对应的图像部分。当表达式值改变时,图像显示对应那部分图像,从而就产生了图像动画效果。

下图是某机械推杆运动的分解图, 共4 个子图。



所以在图像切换动画的配置中,将图像沿水平方向平均分成4 幅图像,并根据 表达式值对应了4 个图幅序号。当表达式值不断周而复始地变化时,就产生了推杆 运动的动画。



	选择变量	表达式
面. 演示变量. 图像	象切换	
		>
) 日像沿水平方向平 回知志・	均分成 4 幅。	
】]像沿水平方向平 副列表: 表达式值<=	均分成 4 幅。	- těitu
3 像沿水平方向平 3列表: 表达式值<= 1.000000	均分成 4 幅。 显示图幅序号	
日像沿水平方向平 副列表: 表达式值<= 1.000000 2.000000	均分成 4 幅。 显示图幅序号 1 2	増加
	均分成 4 幅。 显示图幅序号 1 2 3	增加
]像沿水平方向平 列表: 表达式值<= 1.000000 2.000000	均分成 4 幅。 显示图幅序号 1 2	 一 册除

7.7.9.3 多图像切换

多图像切换动画是多个图像之间的切换,根据表达式值来显示某指定图像。如图:

	选择变量	表达式
画面、系统变量、_和	些 / 10	1
5		2
幻像刘丰		
	日一周後	Http
	亚尔图象	
表达式值(=	1 Dmm	田底全
表达式组<= 1.000000	1. omp	1001525
表达式值<= 1.000000 ▶ 2.000000	2. bmp	
表达式值<= 1.000000 2.000000 3.000000	2. bmp 3. bmp	
表达式/且<= 1.000000 ▶ 2.000000 3.000000 4.000000	2. bmp 3. bmp 4. bmp	
表达式14<= 1.000000 ≥ 2.000000 3.000000 4.000000 5.000000	2. bmp 3. bmp 4. bmp 5. bmp	
表达之小目 <= 1.000000 2.000000 3.000000 4.000000 5.000000 6.000000	2. bmp 3. bmp 4. bmp 5. bmp 6. bmp	

上图所示的表达式对应值由图幅索引号变成了要显示的图像。



7.8 图形事件

7.8.1 概述

当用户操作图形时将触发事件,可以在事件中增加操作响应处理。

可以在属性事件栏中配置图形事件,点击对应事件的"..."即可进入某事件选项的配置。每个图形的事件描述位于属性事件栏的下方,如图:

了席任 日初世	
コーズモージョン	
道記	
按住	
1 拖动事件	
水平拖动	6.71
垂直拖动	C. H. L.
旋转拖动	

非拖动事件只有动作和脚本两种响应方式, **2** , 左边图标是脚本的,右边图标是动作的。

而拖动事件除了有动作和脚本两种响应方式,还有专门的拖动响应处理, ,比非拖动事件多了一个拖动图标。

7.8.2 响应动作

对于简单的图形事件响应可以用动作来完成,对于复杂的响应,必须用脚本 来完成。 点击事件的动作响应图标,弹出动作配置对话框后,可以有两种选 择。

1. 执行对象方法: 在左边的对象树中选择对象,对象包括当前窗口 所有图形 对象、画面/数据库的所有驱动、模块对象。右边为该对象 所有方法(即函数)。 如果该方法有参数,则点击确定之后,还会弹 出参数配置对话框,在对话框下面



会提示当前参数的说明。

动作: 画面.显示窗口		设置动作参数	
对象:	成员:	设置"画面,弹出智	弱口"参数:
 ● 矩形3 短形5 短形6 ② 圓面 ③ 頁面 ③ 系统变量 ④ 系统模块 ④ 数据库 ③ 系统变量 	 ·	参数名 窗口名 模式 × y 透明度	参数值 "报表演示" 1 -1 -1 255
田 🗊 系统模块 田 🔂 IO设备通信	—— =◆退出	模式 0-其它窗口可以	操作;1-只能在此窗口操作。

2. 变量或对象属性赋值操作:包括两部分,目标赋值对象(可以是变量或者是当前窗口的某个图形的属性)和源表达式(由事件参数、对象属性、变量结合表达式函数/操作符)。该操作是计算出源表达式的值,然后赋值给目标赋值对象。 对象属性、变量的概念前面已经介绍了,大家应该也比较熟悉。表达式请参考"表达式"章节。这里主要一下介绍"事件参数"。

事件参数:一些事件在触发时,会传入事件参数,动作可能需要根据该参数进行对应的处理。例如:按键事件,会传入键值参数,表示当前哪个键按下了。事件参数对话框中还可以对事件参数值设置最大值、最小值限制(当参数值大于最大值时,取最大限值;小于最小值时,取最小限值)。

(选择对象属性	为赋值对象	选择变量	是为赋值对象	
窗口.编辑	框2. 文本				_
源表达式					
事件参	数	属性 造	^释 变量	表达式	
事件参数1	.键 + 窗口,矩	形1.左 + 数据	库.v1.演示	变量*6	1
					~
<		IIII		>	



上图的例子,就是把"键盘键值"、"矩形左边"、"数据库.vl.演示变量"6 倍的总和显示在"编辑框 2"中。

ID	参数名	参数说明
1	一键	按下键盘的键值
限值	范围	
昆	小店・「	
-42	UNE ·	取八直・

7.8.3 事件脚本

对于简单的图形事件响应可以用动作来完成,对于复杂的响应,必须用脚本 来完成。有关 JavaScript 脚本程序的详细介绍,请参考 "JavaScript 程序"章 节。

点击事件的脚本响应图标,将打开当前窗口的脚本处理程序编辑器。

如果该图形对象事件的触发程序已经存在,则系统会自动定位到触发程序行; 如果没有存在,系统会自动生成事件触发程序框架。包括版本 v0.9.9.92 之前和 之后的两种框架。如下图所示,上面两个图是 v0.9.9.92 之后更为简洁、通用的 做法,下面两个图是 v0.9.9.92 之前的做法。

2 □ 按钮1.按住 = function() 3	2
2 日 function 按钮1_按住() 3 {	2 2 □ function 按钮1_按住() 3 {
5 ↓ 6 按钮1.按住 = 按钮1_按住;	□□□. 個小文型 (文型10(文型3 , □□□. 個小文型 w文型10(文型3) + 1); 5 _ } 6 按钮1. 按住 = 按钮1_按住;

先介绍旧版本。旧版本的左图是脚本编辑器自动生成的事件触发程序框架。 "function 按钮1 按住(){}"声明了一个函数, "按钮1.按住=按钮1 按住;" 表示"按钮1"的"按住"事件触发时,将调用"按钮1 按住" 事件处理函数。



实际上"按钮 1 按住"可以改为其它名称,框架是为了代码可读性,才把事件触发 函数名称自动生成为"对象名_事件名"的形式。例如,你可以把框架改为:

function 变量3 递加() ł } 按钮1.按住= 变量3 递加;

右图是在框架中增加了事件处理程序。

v0.9.9.92 之后的版本直接将函数体赋值给对象事件,而不必通过函数名。当 然用户也可以自行修改为旧版本的处理模式。

如果你修改了对象名称,则框架中相关对象名称都必须修改。在上 面的例子 中,如果你把"按钮1"的对象名改为"按钮2",则框架中三 处"按钮1"的都 要改为"按钮2"。

最后强调一点:必须理解了上面所介绍的事件触发程序框架的原理,才可以修改框架!

7.8.4 拖动特有响应

当鼠标按住图形拖拽,并产生位置或者角度改变时,将触发拖动事件。

为了方便拖动事件的处理,专门增加了拖动事件的特有响应功能。



拖动事件包括水平拖动事件、垂直拖动事件、旋转拖动事件,它们分别响应 水平位置("左"属性)、垂直位置("顶"属性)、"旋转角度"属性的改变。 下图是这3个事件特有的配置对话框。

水平拖动赋值	重直拖动赋值
目标赋值对象:变量或对象属性 选择对象属性为赋值对象 选择变量为赋值对象 画面. 演示变量. 水平拖动	目标赋值对象:变量或对象属性 选择对象属性为赋值对象 选择变量为赋值对象 画面. 演示变量. 垂直拖动
左边	顶边
位置: 0 🔷 位置: 100	☆ 位置: 0 ◇ 位置: 100 ◇
対应値: 0 🜍 対应値: 100	対应値: 0 対应値: 100
備定 取消 旋转拖动赋值 目标赋值对象:变量或对象属性 选择对象属性为赋值对象 选择变量 画面: 演示变量: 旋转拖动	
)並时针	
角度: 0 🛟 角度:	100
对应值: 0 动应值:	100
确定 取消	

拖动特有事件处理主要功能就是:将拖动后产生的水平位移/垂直位移/旋转角 度按照一定的对应关系赋值给目标对象,目标赋值对象(可以是变量或者是当前窗 口的某个图形的属性),对应关系是通过对话框中的4 个数值来映射的。

水平拖动对应关系

- 左边:表示水平拖动图形左边的位置对应目标赋值对象的值。
- 右边:表示水平拖动图形左边的位置对应目标赋值对象的值。

垂直拖动对应关系

- 顶边:表示水平拖动图形顶边的位置对应目标赋值对象的值。
- 底边:表示水平拖动图形底边的位置对应目标赋值对象的值。

旋转拖动对应关系

- 逆时针:表示旋转拖动图形逆时针方向的旋转角度对应目标赋值对象的值。
- 顺时针:表示水平拖动图形顺时针方向的旋转角度对应目标赋值对象的值。

7.9 主要图形、控件

7.9.1 编辑框

7.9.1.1 说明

编辑框可供用户输入、显示、编辑文本内容,还可以连接输入、输 出变量。

7.9.1.2 主要属性

- 文本:当前显示的内容。
- 输入变量: 文本内容会随该变量值改变。
- 输出变量: 文本输入值会改变该变量值。

7.9.1.3 智能移动WEB(HTML5 Web)

已支持的属性:可见、左、顶、宽、高、文本、文本色、背景色、边框色、输入变量、输出变量、只读、小数位、字号、字体、进制。

已支持的事件:按下、弹起。

7.9.2 标签

7.9.2.1 说明

标签可显示静态的文本内容,或者根据指定表达式显示定制内容。

7.9.2.2 主要属性

● **标题:**当前显示的内容。



 数据显示:非常有用的一个属性,根据指定表达式的类型(模拟量/ 开关量/时间/字符串等)、格式来显示定制内容。

7.9.2.3 智能移动 WEB(HTML5 Web)

已支持的属性:可见、左、顶、宽、高、对齐、标题、字号、字体、 文本 色、背景色、边框色。

已支持的事件:按下、弹起。

7.9.3 按钮

7.9.3.1 说明

按钮主要供用户点击触发事件使用。

7.9.3.2 主要属性

- 标题:当前显示的内容。
- **图像**:正常状态时按钮显示的图像。
- **按下图像**:按下状态时按钮显示的图像。
- **图像位置**:图像所显示的位置,包括上/下/左/右/填充。

7.9.3.3 主要事件

- 按下:按下鼠标左键时触发该事件。
- 弹起: 弹起鼠标左键时触发该事件。

7.9.3.4 智能移动WEB(HTML5 Web)

已支持的属性:可见、左、顶、宽、高、标题、背景色、边框色、 字号、 字体。

已支持的事件:按下、弹起。



7.9.4 开关按钮

7.9.4.1 说明

类似按钮,但是连接一个开关量进行读写。

7.9.4.2 主要属性

- **标题**:当前显示的内容。
- **开状态图像**:状态打开时按钮显示的图像。
- **关状态图像**:状态关闭时按钮显示的图像。
- **图像位置**:图像所显示的位置,包括上/下/左/右/填充。
- **变量:**要进行读写的开关量。

主要事件

- 按下:按下鼠标左键时触发该事件。
- 弹起: 弹起鼠标左键时触发该事件。

<u>注: 智能移动 WEB(HTML5 Web) 尚未支持。</u>

7.9.5 单选框

7.9.5.1 说明

单选框可在多个选项中选择某个项目。

7.9.5.2 主要属性

- 数据:所有项目内容,用分号分隔每个项目。
- 选中项目:当前选择的项目。

<u>注: 智能移动 WEB(HTML5 Web)尚未支持。</u>



7.9.6 复选框

7.9.6.1 说明

复选框可在选择/取消选择某个项目,也可以连接一个开关量进行读 写,当 选中状态改变后,还可以触发 "Check 改变"事件。

7.9.6.2 主要属性

- 变量:要进行读写的开关量。
- 选中:当前选中的状态。 主要事件
- Check 改变: 当Check 状态改变时触发该事件。

注: 智能移动 WEB (HTML5 Web) 尚未支持。

7.9.7 列表框

7.9.7.1 说明

列表框可在列表中选择某个项目,并触发项目"选择改变"事件。

7.9.7.2 主要属性

- 数据:所有项目内容,用分号分隔每个项目。
- 选中项目:当前选择的项目。

7.9.7.3 主要事件

选择改变: 当选中项目改变时触发该事件。

7.9.7.4 主要函数

用户在脚本中可调用"增加项目"、"删除项目"等方法,实现动态增 删项目功能。

<u>注: 智能移动 WEB(HTML5 Web)尚未支持。</u>



7.9.8 下拉框

7.9.8.1 说明

下拉框可在下拉列表中选择某个项目,并触发项目"选择改变"事件。

7.9.8.2 主要属性

- 数据:所有项目内容,用分号分隔每个项目。
- 选中项目:当前选择的项目。

7.9.8.3 主要事件

选择改变: 当选中项目改变时触发该事件。

7.9.8.4 主要函数

用户在脚本中可调用"增加项目"、"删除项目"等方法,实现动态增 删项目功能。

注: 智能移动 WEB (HTML5 Web) 尚未支持。

7.9.9 棒图

7.9.9.1 说明

可以根据所连接变量的值以及指定的最小值/最大值显示棒图位置。

7.9.9.2 主要属性

- 变量:要连接的变量。
- 最小值:棒图无填充时的值。
- 最大值:棒图填充满时的值。 方向:棒图伸展方向,上/下/左/右。
- 形状:平面/立方体/圆柱体。



7.9.9.3 智能移动WEB(HTML5 Web)

已支持的属性:可见、左、顶、宽、高、变量、边框色、颜色、最大值、 最小值。

7.9.10 滑块

7.9.10.1 说明

可以根据所连接变量的值以及指定的最小值/最大值显示滑块位置。

7.9.10.2 主要属性

- 变量:要连接的变量。
- 值:当前值。
- 最小值: 滑块位于最左边时的值。
- 最大值: 滑块位于最右边时的值。

<u>注: 智能移动 WEB(HTML5 Web) 尚未支持。</u>

7.9.11 日期时间

7.9.11.1 说明

可以根据所连接日期时间变量以及指定的格式显示/设置时间串。

7.9.11.2 主要属性

- 变量:要连接的日期时间变量。
- 格式: 自定义显示格式。

<u>注: 智能移动 WEB(HTML5 Web) 尚未支持。</u>


7.9.12 图像

7.9.12.1 说明

图像可以引入一个图片文件显示,也可以实现多个图片的切换动画。

7.9.12.2 小工具栏

🗟 🗇 💋

组态时点击图像,会在图像上方显示一个小工具栏。工具栏包括三 个图标,从左到右,分别为:

1. 选择图像: 与"图像"属性设置一致。

2. **实际大小:** 将图像显示大小设置为实际图像大小(即修改宽、高属性)。

3. **设置透明色**:通过鼠标点击选择屏幕颜色做为透明色(修改透明 色属性)。

7.9.12.3 主要属性

- 图像:要显示的图像文件。
- 透明色:图像中与该颜色相同的部分以透明显示。可以用小工具栏选择。

7.9.12.4 主要动画

- 图像切换:请参考"动画"的"图像切换"章节。
- 多图像切换:请参考"动画"的"多图像切换"章节。

7.9.12.5 主要事件

- 按下:按下鼠标左键时触发该事件。
- 弹起: 弹起鼠标左键时触发该事件。



7.9.12.6 智能移动WEB(HTML5 Web)

- 已支持的属性:可见、左、顶、宽、高、图像。
- 已支持的事件:按下、弹起。

7.9.13 GIF 动画

7.9.13.1 说明

可引入一个 GIF 文件显示 GIF 动画。

7.9.13.2 小工具栏

10

组态时点击图像,会在图像上方显示一个小工具栏。工具栏包括两个图标, 从左到右,分别为:

1. 选择图像: 与"图像"属性设置一致。

2. 实际大小:将图像显示大小设置为实际图像大小(即修改宽、高属性)。

3. 设置透明色:通过鼠标点击选择屏幕颜色做为透明色(修改透明色属性)。

7.9.13.3 主要属性

- 图像:要显示的图像文件。
- 透明色:图像中与该颜色相同的部分以透明显示。可以用小工具栏选择。

7.9.13.4 主要事件

- 按下:按下鼠标左键时触发该事件。
- 弹起: 弹起鼠标左键时触发该事件。

注:智能移动 WEB(HTML5 Web) 尚未支持。



7.9.14 流动

7.9.14.1 说明

可以根据所连接变量的值以及指定的最小值/最大值改变流速。

7.9.14.2 主要属性

- **变量:**要连接的变量。
- **最小值**:流动不动时的值。
- **最大值**:流动最快时的值。
- 方向:流体移动方向,上/下/左/右。
- 形状:流体形状。

注: 智能移动 WEB (HTML5 Web) 尚未支持。

7.9.15 仪表

7.9.15.1 说明

可以根据所连接变量的值显示指针位置。如果需要自定义的表盘背景,可以 将背景色等颜色设置为透明色。

主要属性

- 变量:要连接的变量。
- 最小值:指针位于起始角度时的变量值。
- 最大值:指针位于起始角度+扫描角度时的变量值。
- 起始角度:变量值<=最小值时指针位置。
- 扫描角度:指针允许旋转的最大角度,大于0表示顺时针旋转,反之逆时针。

注: 智能移动 WEB (HTML5 Web) 尚未支持。



7.9.16 触发器

7.9.16.1 说明

触发器: 当连接的表达式值改变时,将触发事件。该组件类似数据 库端/画面端的触发器模块,所不同的是,因为该组件位于窗口内,所以可以结合该窗口以 及该窗口内的图形实现功能。

7.9.16.2 主要属性

表达式:根据该表达式值触发事件。

7.9.16.3 主要事件

触发: 表达式值改变时触发该事件。

注: 智能移动 WEB (HTML5 Web) 尚未支持。

7.9.17 定时器

7.9.17.1 说明

定时器:根据一定的时间间隔定时触发事件。该组件类似数据库端/画面端 的定时器模块,所不同的是,因为该组件位于窗口内,所以可以结合该窗口以及 该窗口内的图形实现功能。

7.9.17.2 主要属性

间隔: 定时触发的间隔时间(单位: 毫秒)。

首次触发:打开窗口时是否触发。

7.9.17.3 主要事件

定时: 定时间隔到达时触发该事件。



注:智能移动 WEB(HTML5 Web) 尚未支持。

7.9.18 报表浏览器

7.9.18.1 说明

可用来查看、打印报表。详情请参考报表相关章节。

_注: 智能移动 WEB(HTML5 Web)尚未支持。___

7.9.19 趋势图

7.9.19.1 说明

包括历史趋势图和实时趋势图。详情请参考历史记录相关章节。

注: 智能移动 WEB (HTML5 Web) 尚未支持。

7.9.20 百度地图

7.9.20.1 说明

通过百度地图控件,画面根据经纬度显示百度地图,尤其可通过与本系统完美 结合实现地图上的监控功能。

7.9.20.2 主要概念

- 标记:标记位于百度地图某个经纬度位置,通过标记将窗口中的图 形控
 件增加为标记子对象,百度地图就可以与窗口其它图形融为一体了。
- 路径:路径可以连接经纬度历史变量,实现百度地图上的经纬度轨 迹曲线。(该功能暂未实现。)

7.9.20.3 主要属性

● 经度:地图中心点经度位置。



- 纬度: 地图中心点纬度位置。
- 缩放级别:范围[3, 19],每升高一级,精度加倍。

7.9.20.4 工具栏

Li Xi 🎭 🍃 🐮 X. Q. Q. 🖛 🔿 🛊 🦊

点中百度地图后,会显示工具栏。百度地图工具栏提供了标记(包括增删、 增加子对象等)、路径(包括增删)、缩放、移动等操作方法。

7.9.20.5 标记整合图形控件

先增加标记和图形控件,选中标记后,点击工具栏的"加入子对象"按钮, 弹出如下对话框,选择需要加入该标记的图形控件,加入后该图形控件与此标记 融为一体,可以随着地图、标记一起移动。

增加子对象	X
矩形1 圆1	确定
GIF动画1 棒图2 饼图2	

<u>注:</u>智能移动 WEB (HTML5 Web) 尚未支持。

7.9.21 海康威视

7.9.21.1 说明

系统内置了海康威视组件,可以无缝连接海康威视视频设备。 云台控制可以通过"云台控制"函数实现对海康视频设备的云台控制。对云台实施的每一个动作都



需要调用该函数两次,分别是开始和停止控制。 下面主要介绍云台控制的命令码。

命令码含义

- 2 接通灯光电源
- 3 接通雨刷开关
- 4 接通风扇开关
- 5 接通加热器开关
- 6 接通辅助设备开关
- 7 接通辅助设备开关
- 11 焦距变大(倍率变大)
- 12 焦距变小(倍率变小)
- 13 焦点前调
- 14 焦点后调
- 15 光圈扩大
- 16 光圈缩小
- 21 云台上仰
- 22 云台下俯
- 23 云台左转
- 24 云台右转
- 25 云台上仰和左转
- 26 云台上仰和右转
- 27 云台下俯和左转



- 28 云台下俯和右转
- 29 云台左右自动扫描
- 58 云台下俯和焦距变大(倍率变大)
- 59 云台下俯和焦距变小(倍率变小)
- 60 云台左转和焦距变大(倍率变大)
- 61 云台左转和焦距变小(倍率变小)
- 62 云台右转和焦距变大(倍率变大)
- 63 云台右转和焦距变小(倍率变小)
- 64 云台上仰和左转和焦距变大(倍率变大)
- 65 云台上仰和左转和焦距变小(倍率变小)
- 66 云台上仰和右转和焦距变大(倍率变大)
- 67 云台上仰和右转和焦距变小(倍率变小)
- 68 云台下俯和左转和焦距变大(倍率变大)
- 69 云台下俯和左转和焦距变小(倍率变小)
- 70 云台下俯和右转和焦距变大(倍率变大)
- 71 云台下俯和右转和焦距变小(倍率变小)
- 72 云台上仰和焦距变大(倍率变大)
- 73 云台上仰和焦距变小(倍率变小)

注: 智能移动 WEB(HTML5 Web) 尚未支持。

7.10 ActiveX 控件

ActiveX 插件以前也叫做 OLE 控件或 OCX 控件, 它是一些软件组件或 对



象,可以将其插入到WEB 网页或其它应用程序中。ActiveX 的支 持,大大增强了 系统的二次开发性与扩展性,并大大提高了系统与 其它系统的交互性、集成性。 尤其目前大多数视频厂家都提供了视 频的 ActiveX,所以利用 ActiveX 能方便地 与视频厂家集成。

二维窗口提供了对 ActiveX 控件的支持,可以通过"高级图形"栏目下 的 "ActiveX"窗口图形图片地址:

7.10.1 选择一个ActiveX 控件对象类

创建 "ActiveX" 窗口图形后,点击属性栏的"控件类别"属性,选择需要引入的 ActiveX 控件。

搜索控件 内容:	☑ 从当前位置起搜索 搜索	
控件名	ProgID	-
Microsoft TreeView Control,	MSComctlLib. TreeCtrl. 2	Th
Microsoft UpDown Control 6.0	MSComCt12. UpDown. 2	
Microsoft UpDown Control, ve	ComCtl2. UpDown. 1	
Microsoft Visio 11.0 Drawing	VisOcx. DrawingControl. 1	
Microsoft Visual Studio Tool	VSTO. RuntimeStorage. 1	
Microsoft Web Browser	Shell. Explorer. 2	
Microsoft WinSock Control, v	MSWinsock. Winsock. 1	15
Microsoft Works Imaging Server	MicrosoftWorks.WkImgSrv.Wks	
Microsoft. VisualStudio. Offic	WinForms. Control. Host	
MMC IconControl class	MMC.IconControl.1	
MMCCtrl class	MMCCtrl.MMCCtrl.1	Y

7.10.2 选择 ActiveX 窗口图形

由于选择 ActiveX 控件类别后, ActiveX 可能会在窗口上创建一个子 窗口, 这种情况下,点击 ActiveX 时,是点击在 ActiveX 子窗口上,所以可能无法通过 点击选择 ActiveX,只能通过拖出足够大的橡皮筋 矩形来选择,或者在属性栏的 对象栏选择。



属性栏	ά×	
🛼 ActiveX1	*	
窗口		
🗙 ActiveX1		
ab 按钮1	i i i i i i i i i i i i i i i i i i i	
abl 编辑框1		
A标签1		

7.10.3 配置 ActiveX

选中ActiveX 窗口图形后,通过小工具栏,可以对ActiveX 控件进行 配置。

B F	
配置控件 弹出控件属性配置道	t项窗口配置ActiveX

7.10.4 通过JavaScript 调用

在 JavaScript 中,通过调用 ActiveX 图形对象的"控件对象()"方法,返回 ActiveX 控件对象后,就可以实现对 ActiveX 控件操作访问了。例如, ActiveX1 是一个"IE Web 浏览控件"图形,如下即可实现浏览"www.abc.com" 网站 (Navigate2 是该控件的导航方法)。

ie = ActiveX1. 控件对象(); ie.Navigate2("www.abc.com");

7.10.5 主要问题

单机版本下ActiveX 与 OPC Client/Server 冲突。默认情况下, 启用 ActiveX 功能, 禁用 OPC 功能。在安装目录下, 生成一个名为 "mtcom"的空 文件, 则启用 OPC, 禁用 ActiveX。

网络Web 版本没有该问题。



7.11 定制图形(图形模板)

7.11.1 说明

当窗口中有大量相同的复杂图形时,即可采用定制图形开发,实现一次开发 多处引用,大幅提高组态效率。

7.11.2 与"组合"图形的区别

与"组合"图形相似,定制图形也是由很多图形组合而成。但是它们之间一个 最重要、最直接的区别就是图形维护:当需要修改组合图 形时,需要将所有窗口 的所有组合图形修改一遍;而定制图形只要 修改一次,所有引用到的定制图形会 自动更新!

7.11.3 创建

1. 在工程画面浏览窗口中右键点击"定制图形"节点弹出菜单后,点击"新增 定制图形"。如图:



 也可以通过点击"文件"菜单的"新建文件"或者系统工具栏"新建文件" 命令来创建。与创建窗口类似,请参考"新建窗口"一节。

7.11.4 增加定制属性、动画、事件

通过定制图形工具栏可以增加定制属性、动画事件。如图:





7.11.5 引用定制图形

在窗口工具箱的"高级控件"栏中,点击"定制图形",如图:



然后在定制图形的属性栏的"图形"属性中选择定制的图形。

7.12 图形开发包

目前系统图形较少,但是提供了图形开发包,用户可以根据实际应用自行增加图形,详情请参考《SDK 开发包》章节。



八、三维窗口与模型

8.1 概述

欢迎使用 3D 功能,目前 3D 功能在国内组态界是首创,全球组态领域也非常少见。

下面着重说明一下使用 3D 功能时要注意的一些问题。

1. 本系统侧重的是 3D 监控组态功能,而不是 3D 建模功能。当然也可以利用系统进行一些不复杂的建模。

2. 目前已经支持最流行的 3D 模型文件格式: 3DS,可以通过三维窗口的 "模型 3DS"对象来引入。创建"模型 3DS"对象后,将其"文件"属性设置为要 引入的 3DS 文件即可。

3. 目前市场上已经很多流行的 3D 建模工具,例如: 3DS MAX、 google SketchUp、SolidWorks 等。可以采用这些工具建模,接着转换为 3DS 格式,然 后引入工程。

4. 几乎所有常见的 3D 模型文件格式都可以相互转换,所以非 3DS 格式可以使 用一些 3D 模型转换工具软件来转换为 3DS 格式。目前常见的 3D 转换工具软件有 3D Object Converter、Deep Exploration 等,利用 3DS MAX 的导入导出插件也 可以实现转换。SolidWorks 转 3DS:

http://www.sharewareconnection.com/3ds-export-for-solidworks.htm 。 SketchUp Pro 版本也提供了 3DS 转换

 $\label{eq:http://support.google.com/sketchup/bin/answer.py?hl=en&answer=114325 \ .$

5. 未来会逐步支持其它常见的 3D 模型格式。

6. 引入 3DS 后,利用表达式、动画、脚本等方法与现场 IO 数据相结合,即可实现三维真实立体监控效果!

8.2 新建三维窗口

可以采用如下方法创建新的三维窗口。



注:为了组态方便,新建的三维窗口默认在场景上方创建了两个 方向光源。

8.2.1 从工程画面浏览窗口创建

在工程画面浏览窗口中右键点击"三维窗口"节点弹出菜单后,点击"新增三 维窗口",弹出新增三维窗口对话框。



在新增三维窗口对话框中,输入三维窗口名称。

8.2.2 通过命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。

新建文件	X
类别: ■ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	文件类型: 窗口 窗口 定制图形
三维立体窗口。 ○ 将新建的文件添加到当前工程 新建文件名:	定 取消



一般这种方法用来创建临时三维窗口文件。

8.3 三维窗口分组

当画面三维窗口较多时,就可以采用分组来管理了。 新增三维窗口组 在工程 画面浏览窗口中右键点击"三维窗口"节点弹出菜单后,点击"增加三维窗口 组",将新增一个三维窗口组,输入三维窗口组名称。



三维窗口组的三维窗口文件

右键点击三维窗口组,弹出菜单后,点击"新增三维窗口",将在该三维窗口组下新增三维窗口文件。

也可以通过鼠标拖拽,在三维窗口组之间拖动三维窗口文件,使得三维窗口 放置到其它三维窗口组中。

8.4 三维窗口组态开发

新建三维窗口后,就可以开始三维窗口的设计组态工作了。

实际上,三维窗口本身的组态工作不多,主要还是三维窗口上大量的模型组态开发等。



8.4.1 属性设置

三维窗口所有属性都显示在属性栏中,对于每个 属性的描述,点中 该属性后,会提示在属性的下端。 如下图所示:

下面主要解释一下几个不易理解的属性:

背景图:一幅采用天空盒原理合成的图,其中宽 与高比为6:1,分别放置天空盒的前、右、后、左、 上、下视图。天空盒是一种让计算机 3D 屏幕背景具 有更广阔视野的一种方法,当使用天空盒的时候,水 平画面被限制在一个立方体当中,背景中遥远的山 川,远处的建筑,还有天空等等都被投射到了立方体 的各个面。通过这样的方式 来创建逼真的三维环境。 很多 3D 工具都可以创建天空盒,请参考相关资料, 这里就不再解释了。

环境光颜色:先解释一下环境光源:与三维空间 内的位置无关的光源,对场景中所有对象都提供了固 定不变的照明,没有方向与位置,只存在颜色和强 度。环境光颜色就是这个环境光源的颜色。 属性栏 × 回三維窗口 ¥ ☞ 属性 🛃 动画 🔰 事件 名称 三维控件 背景颜色 ffc057 背景图 SkyBox_sanl. . 环境光颜色 383838 描述 B 眼睛及所看的目标位置 眼x: 0.000000 眼Y 0.000000 0.000000 看X 0:000000 看Y 0.000000 看Z 0.000000 眼X最小 -1000.000000眼X最大 1000.000000 眼Y最小 -1000.000000眼Y最大 1000.000000 眼Z最小 -1000.000000眼2最大 1000.000000 属性描述 一幅采用天空盒原理合成的图 中宽与高比为6:1,分别放置天 盒的前、右、后、左、上、下衬 图。 I 下视

眼 X/Y/Z:当前眼睛(即当前摄像机、或者人所站的位置)的 X/Y/Z 座标,与 "看 X/Y/Z"一起构成场景视角。

看 X/Y/Z:当前眼睛所注视的目标位置的 X/Y/Z 座标,与"眼 X/Y/Z"一起 构成场景视角。通过不断地修改"眼 X/Y/Z"和"看 X/Y/Z",就可以实现场景漫 游了。 在状态栏中会提示当前三维窗口的眼睛和目标位置。



8.4.2 事件配置

三维窗口所有事件都显示在属性事件栏中,对于每个事件的描述, 点中该 事件后,会提示在事件的下端。如下图所示:

▶二班崮山 肇属性 <mark>署</mark> 武	画(多事件)
] 基本事件	
打开	1 50
关闭	
定时器	
定时器	<u>₩</u>

8.4.3 其它操作

如下图系统工具栏部分窗口相关操作,可以设置为主窗口(房子图标)、编写 JavaScript 脚本(J字图标)等。



编写脚本也可以通过右键点击窗口空白处,弹出菜单后,选择"编写 JavaScript"。

8.5 3D 工具栏

2 编写JavaScript

3D 有专门的工具栏,如图:

3D丁目 1 800 Ø 田氏 * 1 Ø, 2 5 1 3 6 4

1. 鼠标操作模式:包括选择图形/模型模式和操作 3D 场景模式。

▶ **一选择模型模式(箭头图标)**:在该模式下,鼠标在窗口空白处拖拽



时,可以拖出橡皮筋矩形选择模型/图形。

 一操作场景模式(手状图标): 默认模式。在该模式下,鼠标在窗口空 白处拖拽时,会导致 3D 场景手动漫游。有关 3D 场景漫游请参考 "场 景漫游"章节。

2. 模型框架显示: 将当前三维窗口的场景中的 3D 模型设置为框架显示,对平面的 Windows 控件/图形没有影响。下图为长方体和圆柱体的框架显示例子。



3. 多视图显示:为了能在同个三维窗口中同时从多个视角查看场景,开发环境提供了多视图显示。点击多视图图标的下拉按钮,列出了目前支持的多视图布局 (下面左图)。下面右图还显示了"上下左右 视图"的布局。



4. 视图视角:组态开发环境提供了多种视角查看。点击多视角图标的下拉按钮,列出了目前支持的各种视角。除了主视图的视角是可以手动漫游修改的,其它视图分别固定从上下左右前后进行观看的。

一修改视角:点中要修改的视图后,选择要设置的新视角即可。例如,将 上右图的右下角的"前视图"改为"后视图"。





5.投影显示: 3D 场景投影有两种,直角投影(也叫平行投影)、透视投影。直角投影远处与近处的大小一样,因此真实感不强。除主视 图默认是透视投影外,其它视图默认都是直角投影。

一修改投影显示:点中要修改的视图后,选择要设置的投影方式。

6.3D 模型操纵器:移动、旋转、缩放的操作方式与 2D 图形区别较 大,对 3D 模型的移动、旋转、缩放是通过 3D 操纵器来完成。请参 考"三维模型组态 设计"相关章节。

8.6 三维场景漫游

前面已经提到过视角是由眼睛位置与眼睛所看的位置决定的,不断改变视 角,就实现了场景漫游。大家可以想像一下,人在走动时,眼睛和目标观看位置 不断在变化,从而导致所看的场景不断变化,漫游也是一样道理。 漫游可以分成 手动漫游与自动漫游。

8.6.1 手动漫游

手动漫游是利用鼠标、键盘促使场景漫游。上文已经提过开发环境 中三维窗 口只有鼠标设置为"操作场景模式"时,鼠标才可以操作漫 游,否则为"选择模 型模式",此时鼠标只能选择图形/模型。而在运行环境下,鼠标不存在选择图形/ 模型,因此可以随时漫游场景。 鼠标、键盘的操作漫游包括按住"Ctrl"键和没



有按住 "Ctrl"键两种情况。 对于直角投影视图, "Ctrl"键没有效果。按 "Home"键可以回到初始状态。下表列出了鼠标、键盘操作方法:

	鼠标左右拖拽	"←、→"键	鼠标上下拖拽	"↑、↓"键	鼠标滚轮滚动	PageUp/PageDown键
不按"Ctr1"键	漫游A	漫游B	漫游C	漫游D	漫游E	漫游E
按住"Ctr1"键	漫游B	漫游A	漫游D	漫游C	漫游F	漫游F

漫游 A: 眼睛绕目标位置水平旋转,目标位置不变。相对于人绕着一个圆形景点行走观看。

漫游 B: 眼睛、目标位置水平左右移动。相对于人目视前方但往左 右行 走、或者扭头看旁边的景色但是往前行走。

漫游C:眼睛上下移动,但是目标位置不变。

漫游 D: 眼睛、目标位置上下同时移动。

漫游 E: 眼睛沿目标位置移动。相当于人走近/远看物体。

漫游F: 缩放焦距。

8.6.2 自动漫游

自动漫游目前得通过动作或者 JavaScript 脚本改变眼睛及目标位置, 开 发起来比较麻烦。未来考虑开发一个漫游路线模型,当三维窗口设置了某个漫 游路线,就会自动沿着这个漫游路线漫游了,这样就会大大方便了用户组态开 发工作。 除了直接设置眼睛及目标位置外,也可以调用三维窗口方法命令来 实现。这些方法命令有:

眼睛绕点旋转(Px, Py, Pz, 角x, 角y, 角z):绕某个基点旋转眼睛位置。参数分别表示:基点X座标、基点Y座标、基点Z座标、X方向旋转角度、Z方向旋转角度。

看绕点旋转(Px, Py, Pz, 角x, 角y, 角z): 绕某个基点旋转目标位置。参数分别表示:基点X座标、基点Y座标、基点Z座标、X方向旋转角度、Y方向旋转角度、Z方向旋转角度"。

眼睛绕轴旋转(Px, Py, Pz, Vx, Vy, Vz, 角):眼睛位置绕某个轴旋转。 参数分别表示:轴基点X座标、轴基点Y座标、轴基点Z座标、轴X方向



向量、轴Y方向向量、轴Z方向向量、旋转角度。

看绕轴旋转(Px, Py, Pz, Vx, Vy, Vz, 角):所看到的目标位置绕某个轴旋转。参数分别表示:轴基点 X 座标、轴基点 Y 座标、轴基点 Z 座标、轴 X 方向向量、轴 Y 方向向量、轴 Z 方向向量、旋转角度。

8.7 三维模型组态设计

8.7.1 概述

三维窗口仅仅提供了图形/模型的容器,真正要实现复杂的监控,还必须通过 图形/模型来实现。 **工具箱** ×

目前系统提供的图形/模型较少,但是未来会逐步增加。对于系统尚未提供的复杂图形/模型,也可以采用组合、引入 3DS 模型等方法实现。

8.7.1.1 工具箱

系统所有图形/模型都位于工具箱中。当打开三维窗 口时,自动会列出这些图形/模型。如下图所示:

8.7.2 新建模型

所有图形/模型创建都得先点击工具箱中要创建的图 形/模型,然后按住鼠标进行拖拽。其中二维图形的创建 与二维平面窗口的二维图形类似,这里就不在重复了。 本节主要讲述三维模型的创建。

<u>注:所有模型创建时,第一个鼠标点击位置要落在地</u> <u>平面网格的之内,否则位置可能超出三维窗口之外。</u>

8.7.2.1 创建平面

在地平面网格中先点击需要创建的位置,按住鼠标拖拽出橡皮筋矩形,在 需要的位置释放鼠标即可,如图:







8.7.2.2 创建立方体

创建立方体只要在平面的基础上,增加一个 Z 轴(即高度)方向的点即 可。大多数模型都是按照立方体的方式创建,如立方体、三维 文字、模型 3DS、火焰等。如图:



8.7.2.3 创建球体

在地平面网格中先点击需要创建的位置,按住鼠标拖拽出球体半径,在需要的位置释放鼠标即可,如图:



8.7.2.4 创建圆柱体

先在地平面网格创建一个圆,然后在 Z 轴 (高度)方向上点击最后 一点,即形成圆柱体了。如图:



8.7.2.5 创建光源

目前创建光源比较简单,直接在要创建的位置点击拖动即可。





8.7.3 鼠标、键盘操作

图形/模型的选择、位置调整等可以通过鼠标、键盘操作来完成。

8.7.3.1 图形选择

可以直接用鼠标点中某个图形来选择单个图形,或者在鼠标操作模式为选 择模型模式的情况下,用鼠标拖出橡皮筋矩形来选择多个图 形/模型。

如果选择时,按下"Ctrl"键,则会切换图形/模型的选择状态。即原来选中的图形/模型,就会从选中图形/模型列表中移走;原来没选中的图形/模型,会增加到图形/模型选择列表中。

8.7.3.2 图形位置调整

对选中的图形用"↑、↓、←、→"键,可以对选中的图形的位置进行微调。

8.7.4 移动

将当前操纵器设置为移动操纵器,选择模型后,就可以移动模型了。

8.7.4.1 移动

首先点击 3D 工具栏的"移动"工具,接着点击要移动的三维模型,显示如下 图所示的移动操纵器,点击操纵器相应位置,并拖拽,则实现相应移动。





所有相关移动介绍如下。

*: 沿 X 轴方向移动。
 *: 沿 Y 轴方向移动。
 *Z: 沿 Z 轴方向移动。
 *Z: 沿 X 平面方向移动。
 *X: 沿 X 平面方向移动。
 *Z: 沿 X 平面方向移动。
 *Z: 沿 Y 平面方向移动。

8.7.4.2 旋转方向移动

如果对象被旋转后,而用户想沿旋转后的方向移动, 就可以使用"旋转方向移动"命令了。首先点击 3D 工具栏的"旋转方向移动"工具,接着点击要移动的 三维模型,显示如下图所示的移动操纵器,点击操纵器相应位置,并拖拽,则实现 相应移动。





所有相关移动介绍如下。



8.7.5 旋转

旋转: 首先点击 3D 工具栏的"旋转"工具,接着点击要旋转的三维模型,显示如下图所示的旋转操纵器,点击操纵器相应位置,并拖拽,则实现相应旋转。



所有相关旋转介绍如下。





[●]Z:绕Z轴旋转。

8.7.6 缩放

缩放: 首先点击 3D 工具栏的"缩放"工具, 接着点击要缩放的三维 模型, 显示如下图所示的缩放操纵器, 点击操纵器相应位置, 并拖 拽, 则实现 相应缩放。



所有相关缩放介绍如下。

. Ⅰ. Υ 轴方向缩放。

- **Z**:沿Z 轴方向缩放。
- ∃xy:沿XY 面方向缩放。
- ℜ. YZ: 沿 XZ 面方向缩放。
- ·□ ☑: 沿YZ 面方向缩放。
- ➡. 沿 XYZ 方向统一缩放。

8.7.7 锁住

与二维平面窗口的图形一样,也可以在三维窗口中锁住三维模型。

锁住:将选择的图形/模型设置为组态时不允许鼠标、键盘编辑操作。 主要 包括:拖拽、修改位置、旋转、大小等。可以用"解锁"操作解 除。在系统工具



栏、图形属性或鼠标右击图形设置,如图:



当图形被锁住后,鼠标移到上面时会出现锁住的图标,如图:



8.7.8 组合

与二维平面窗口相同,三维窗口的图形/模型也可以组合。但是与二维平面 窗口也有不同,因为三维窗口包括二维图形和三维模型,二维图形和三维模型 不能组合在一起,只能分别各自组合。

可以将多个图形/模型组合成一个图形/模型对象,以方便组态编辑。 组合后的做为一个完整对象可以有自己的属性、动画、事件,也可以"取消组 合",回到原先的状态。

选择要组合/取消组合的图形/模型后,右键点击弹出菜单。如图:



		—— 撤消 Alt+Backspace
	2	重做 Ctrl+Y
		编写JavaScript
	म	组合
\sim		取消组合

组合后,原来的图形/模型就成了这个新的组合图形/模型的子对象,也可以 对子对象直接编辑组态,请参考"子对象"一节。

8.7.9 子对象

与二维平面窗口相同,三维窗口的部分图形/模型也有子对象,如组合的模型、复杂的 3DS 模型等。

8.7.9.1 选择子对象

有时候得选择某子对象编辑,可以通过下面方法:

1. 通过鼠标点击选择: 必须先选中该模型, 然后再点击该模型中的子对 象, 才能选中子对象。比如要选择一个复杂的 3DS 模型的子对 象, 例如下图 挖掘机的铲子, 得先选中挖掘机, 然后再选择铲子。

2. 也可以通过右键点击该图形,弹出菜单后,点击"子对象"菜单选项,弹出 子对象选择对话框后,选择要编辑的子对象。如图:





8.7.9.2 子对象属性、动画、事件

子对象属性、动画、事件的组态查看与编辑也分别位于属性栏的属性、动 画、事件选项页中。

8.7.10 特有操作

与二维平面窗口相同,三维窗口的部分模型具有特有操作,如 3DS 模型等。 但是三维窗口的模型不能通过小工具栏操作,只能通过右 键点击模型,弹出菜单 后点击操作。不能用小工具栏操作的主要原 因是:三维窗口的视角改变可能就会 导致模型看起来有很大的变化,不方便定位小工具栏的位置。

右键点击模型,弹出菜单后点击操作选项。下图是 3DS 模型的特有 操作"实际比例":将模型按照原来的 XYZ 比例设置。



8.7.11 方法命令

除了前面提到的动画, 3DS 模型子对象还有一些方法函数可供调用, 实现更 复杂的动画效果。其中移动的编写比较简单,只要改变它的 X、Y、Z 座标即可。



但是旋转比较复杂,所以系统提供了一些旋转方法函数供调用。

绕点旋转子模型(子模型集,角x,角y,角z):绕本子模型的基准点旋转子 模型集一定的角度。参数分别表示:子模型名称集合(当中用","隔开)、X轴 方向旋转角度、Y轴方向旋转角度、Z轴方向旋转角度。

绕轴旋转子模型(子模型集, Vx, Vy, Vz, 角, 本地): 绕经过本子模型基 准 点的轴旋转子模型集一定的角度。参数分别表示:子模型名称集 合(当中用"," 隔开)、轴 X 方向向量、轴 Y 方向向量、轴 Z 方向向量、旋转角度、是否是子模 型本地坐标系。

旋转主模型(角 x, 角 y, 角 z): 绕本子模型的基准点旋转主模型一定的 角 度。参数分别表示: X 轴方向旋转角度、Y 轴方向旋转角度、Z 轴方向旋转角 度。

8.8 三维模型属性

8.8.1 概述

可以在属性栏中修改模型属性,每个属性的说明描述位于属性栏的下方,如 图:

8.8.2 通用属性

所有模型都有共有的属性,包括:

- 名称 用来标识该图形对象。首字符必须 是字母或下划线,后面的字符必 须是字 母、数字或下划线。
- 可见 该图形运行时是否可见,闪烁、显 隐动画会用到该属性。
- 安全区 结合当前登录用户的安全区来确 定该图形的安全操作权限。如果该对象的 安全区为空,则任何用户都可以操作: 否 则,当前用户要操 作该对象时,只有该用 户拥有该对象安全区列表之一的安全区时, 才允许操作。请参考"用户

-	基本属性	
	名称	圆柱体1
	可见	是
	安全区	
	锁住	否
	ĸ	15.285892
	R1	2.907507
	R2	2.907507
	纵横比	1.000000
	颜色	5d57ff
Ξ	位置	
	X	30.483330
	Y	-48.694176
	Z	7.642946
	X角	0.000000
	Y角	0.000000
	Z角	0.000000



与安全区"章节。

锁住 将选择的图形设置为组态时不允许鼠标、键盘编辑操作。请参考前面"锁住"一节。

8.9 三维模型动画

模型属性是静态的,可以通过脚本运行时动态修改属性,从而产生动画效 果。但是脚本开发较为麻烦,而模型的动画设置只要简单的配置就可以产生动画 效果了。

可以在属性动画栏中配置模型动画,点击对应动画的"..."即可进入某动 画选项的配置。每个模型的动画描述位于属性动

画栏的下方,如图:

- 闪烁/显隐:"闪烁/显隐"动画与平面 窗口图形动画一致,请大家参考"窗口 与图形"的"图形动画"的"闪烁/显 隐"章节。
- X/Y/Z 移动: "X/Y/Z 移动"动画与平面窗口图形的"水平/垂直移动"动画一致,只不过是沿 X/Y/Z 轴移动,请大家参考"窗口与图形"的"图 形动画"的"移动"章节。
- X/Y/Z 旋转: "X/Y/Z 旋转"动画与平面窗口图形的"旋转"动画一致,只不过是绕 X/Y/Z 轴旋转,请大家参考 "窗口与图形"的"图形动画"的 "旋转"章节。



X/Y/Z 缩放: "X/Y/Z 缩放"动画与平面窗口图形的"水平/垂直缩放"动 画一致,只不过是沿 X/Y/Z 轴缩放,请大家参考"窗口与图形"的"图 形动画"的"缩放"章节。

8.10 三维模型事件

当用户操作控件/模型时将触发事件,可以在事件中增加操作响应处理。可以在属性事件栏中配置图形事件,点击对应事件的"..."即可进入某事件选项的



配置。每个图形的事件描述位于属性事件栏的下方,如图:

■ 圆柱体1		*
🔗 属性 📙 动画	多 事件	
日 基本事件 按下	1	1-0
弹起		9

事件有动作和脚本两种响应方式, **Z** , 左边图标是脚本的,右边图标是动 作的。

除了没有拖动事件,模型的事件响应处理与平面窗口图形的处理一致,请大 家参考"窗口与图形"的"图形事件"章节。

8.11 3DS 模型

系统侧重组态功能,所以建模功能较弱,但是可以引入最流行的 3D 模型格式 ——3DS 模型。

8.11.1 导入3DS

新建一个"模型 3DS"模型,然后将其"文件"属性设置为要导入的 3DS 文件。当导入 3DS 模型时,模型所附带的贴图文件也会相应导入。

属性栏	×
🖤 挖掘机	~
一一 属性	动画 🍠 事件
安全区	<u>不</u>
文件	industry)
X	-13. 450380

8.11.2 调整为原来比例

导入后,有可能不是原来模型的 X/Y/Z 比例,看起来会有变形,可以"模型



3DS"的命令调整为原来比例。右键点击模型,弹出菜单后点 击"实际比例",将 模型按照原来的 XYZ 比例设置。如果已经是原来比例,"实际比例"是灰色的。如 图:



8.11.3 一些注意事项

一导入 3DS 时,贴图文件得在 3DS 文件的相同目录下。

一建模时,每个 3DS 子对象都有一个默认基点,旋转、放大都是以这个基点进行的,该基点是建模时确定的。如果不想绕该基点旋转,那么得使用下面将要介绍的方法函数旋转。

一如果对模型、模型子对象涉及到 JavaScript 编程,尤其是旋转,得 仔 细确定建模时的基准点和变换矩阵。

一目前市场上已经很多流行的 3D 建模工具,例如: 3DS MAX、 google SketchUp、SolidWorks 等。可以采用这些工具建模,接着转换为 3DS 格式,然后引入工程。

一几乎所有常见的 3D 模型文件格式都可以相互转换,所以非 3DS 格式可以使



用一些 3D 模型转换工具软件来转换为 3DS 格式。目前 常见的 3D 转换工具软件 有 3D Object Converter、Deep Exploration 等。 SolidWorks 转 3DS: http://www.sharewareconnection.com/3ds-export-for-solidworks.htm 。 SketchUp Pro 版本也提供了 3DS 转换:

http://support.google.com/sketchup/bin/answer.py?hl=en&answer=114325 。

8.11.4 方法命令

除了前面提到的动画, 3DS 模型子对象还有一些方法函数可供调用, 实现 更复杂的动画效果。

绕点旋转子模型(子模型集,角x,角y,角z): 绕本子模型的基准点旋 转子模型集一定的角度。参数分别表示:子模型名称集合(当中用","隔开)、X 轴方向旋转角度、Y 轴方向旋转角度、Z 轴方向旋转角度。

绕轴旋转子模型(子模型集, Vx, Vy, Vz, 角, 本地): 绕经过本子模型基 准 点的轴旋转子模型集一定的角度。参数分别表示:子模型名称集合(当中用"," 隔开)、轴 X 方向向量、轴 Y 方向向量、轴 Z 方向向量、旋转角度、是否是子模 型本地坐标系。

旋转主模型(角 x, 角 y, 角 z): 绕本子模型的基准点旋转主模型一定的角度。 参数分别表示: X 轴方向旋转角度、Y 轴方向旋转角度、Z 轴 方向旋转角度。

8.11.5 3DS 模型站点

下面列出一些 3D 模型站点供参考。

3d 仓库模型下载

<u>王国 3D 模型库</u>

3D 模型库

all3dmodel.com

Free 3D Models

Klicker free 3D Models



51Render.com

3D Model Download

3dsmodels.com

FREE 3D MODELS

3DXtras.com

Amazing 3D graphics

3D Model Libraries Free Download

Free 3D Objects at Creative-3D.net

<u>3D World Club - 3D Models - Home</u>

Free 3D models download

Baumgarten Enterprises-Free 3D Model Downloads

60 Excellent Free 3D Model Websites

8.12 光源

三维场景得有光源,场景才能明亮。为了组态方便,新建的三维窗口默认在 场景上方创建了两个方向光源,所以大部分情况下不必再创建光源了。

除了在三维窗口提到的环境光以外,三维窗口工具栏提供了三种光源:方向 光源、点光源、聚光光源。

8.12.1 方向光源

向一个方向无限延伸的光,和光源位置无关,其相似的例子就是太阳光,光的 强弱程度与距离无关。下面左图为无光源的情况,右图为有方向光源的情况。







8.12.2 点光源

以一个点为中心向四周发出的光,没有方向性,其代表例子就是照 明灯泡。 点光源的光强度随距离平方成反比衰减 (attenuation)。如 下图所示。



8.12.3 聚光光源

对固定的位置和方向进行照射的光源。聚光光源的光强度随距离平 方成反比 衰减 (attenuation)。聚光光源向某方向照射时还具有一定的角度。如下图所示。




九、历史记录

9.1 概述

9.1.1 简介

本历史库是专门针对工业通用实时历史库而精心设计研发的,能满足一万 个变量每秒同时存储。而且本历史库具有一定的压缩性能,并具有很高的访问速度,尤其适合远程网络访问。

9.1.2 注意

为了存储效率,所有变量的数据统一保存在一个文件中。而FAT16 文件系统单个文件最大为2GB,FAT32 文件系统单个文件最大为4GB,NTFS 文件系统单个文件最大为2TB(2048GB)。所以,在历史记录的变量多、采样周期短、保存时间长的情况下,务必采用NTFS 文件系统来保存。

9.2 基本配置

9.2.1 基本配置

在工程数据库浏览窗口中双击"系统配置"节点下的"历史记录"子节 点,打开历史记录配置对话框后即可进行配置。





本电应 历史库留历 保存期限 历史记录有效保存期限: 100	记录块大小 天 历史记录块大小: 512K	↓]
写盘间隔 强行写盘间隔: 分	严重警告: 修改块大小后,所有 的历史数据都将被清除! 有关块 的选择请参考用户手册。	一 原来 大小
保友日录		
保存目录 空目录表示保存在工程默认路 将整个工程移到该磁盘上。 延迟开始记录	径下。不建议设置,若想保存在另外磁盘上,系 按时开始记录 (延识记录时间为0时有效)	建议
保存目录 空目录表示保存在工程默认路 将整个工程移到该磁盘上。 延迟开始记录 延迟开始记录	径下。不建议设置,若想保存在另外磁盘上, 致时开始记录(延迟记录时间为0时有效) ● 不按时 ● 按秒:当秒为 0	建议

图 9-1

历史记录有效保存期限:到达设定的天数后,系统自动将过期的记录标记置为 无效。

保存目录:可以指定历史库保存的目录路径,空表示保存在工程目录下。

9.2.2 记录块

 为了存储效率,每个新建变量需要预先分配记录块大小的磁盘空间,所以 当新建大量历史变量时,第一次运行历史库需要预先分配大量的空间,这会导致 第一次启动组态数据库的速度较慢。当该变量的记录块空间不够时,会重新分配 记录块。记录块大小可以根据实际情况进行设置,具体请参考下文。

2. 修改了记录块大小配置后,目前软件得清除旧的历史记录才能生效,所以在 工程正式实施之前,就得确定好记录块大小。

3. 记录块大小推荐设置:记录块大小推荐根据工程大多数变量的采样周期进行设置。



大多数变量采样周期	记录块大小设置
1~3秒	512K
3~5秒	256K
5~10秒	128K
10~20秒	64K
20~40秒	32K
40~80秒	16K
>80秒	8K

大多数变量采样周期记录块大小设置:

9.2.3 强行写盘间隔

0 为默认,表示系统自动选择合适的时间存入磁盘,这种情况下有可能会 因为系统突然掉电造成历史数据丢失。

非0 情况下,如果写盘间隔短、保存变量多、历史记录周期短,会导致运行 效率低。

9.2.4 延时开始记录

默认情况下,系统启动后,就开始记录历史数据,但是有些驱动此时可能尚 未成功连接设备。这个设置可以延时一段时间后开始记录。

9.2.5 按时开始记录

从字义上理解按照一个指定的时间开始记录,但实际上是根据指定的时间配 合各个变量的采集时间间隔,并结合当前实际时间来计算系统启动后各个变量开 始记录的时间。

如:设置了按13时0分开始记录,某个变量采集周期是2小时,系统启动时间是16点,那么17点的时候就会开始记录,而不是等到第二天的13点开始记录。



9.3 历史库备份

9.3.1 说明

历史库备份能将历史记录按照指定的间隔时间存入关系数据库,目前支持 Access 和 SQL server 备份。

9.3.2 配置

在工程数据库浏览窗口中双击"系统配置"节点下的"历史记录"子节点,打 开历史记录配置对话框后,接着点击"历史库备份"选项卡即可进行备份配置。

	历史记录	×
数据库 × □ □ □ 系统配置 ▲ 系统启动	基本配置 历史库备份 备份时间间隔: 30 分钟 备份项目列表: 增加备份项目	
● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	目标库表《采集周期》时间字段《变量与字段的对应列表	
□ □ □ □ □ □ □ □ □ □ □ □ □ □		
③ 画面 圖数据库	确定 取消 应用(A) 帮助	
	图 9-2	

备份时间间隔:每隔多长时间(单位分钟)备份一次。

9.3.3 备份项目

历史数据库可以配置多个备份项目。每个备份项目由目标库表、采 集周期、 时间字段、要备份的历史变量组成。其目的就是将要备份的历史变量记录按照指 定的采集间隔保存到指定的关系库表中。



目标库表:备份的目标关系数据库表。如下图,选择所要用来备份 的数 据库表。

数据源类型:		Microsoft SQL S	Server 数技	居库 🔹
MsSQL服务器	: [SQLXP\SQLEXPRES	S (✔ [搜索
登录数据库				
用户名	:	sa		
密码:		***		
数据库:	н	istBackup	~	刷新库
数据库表:	5		~	刷新表

图 9-3

采集周期: 必须是所有备份历史变量的实际采集记录周期的公倍数。比如 某备份项目包含三个历史变量,它们的记录周期分别为2 秒、3 秒、4 秒,则项 目采集周期必须为12 秒的倍数(12 秒、24 秒、36 秒、...)。

时间字段: 备份项目的关系表必须包括一个时间字段,用来记录每 条记录对应的采集时间。

变量与字段的对应列表:每个历史变量要保存到的目标字段。如下图,选择每 个历史变量对应的备份目标字段。

hist1	
hist4	T T T T T T T T T T T T T T T T T T T
hi st5	
	hist5

图 9-4



9.4 历史变量

注意:为了存储效率,每个新建变量需要预先分配记录块大小的磁盘空间, 所以当新建大量历史变量时,第一次运行历史库需要预先分配大量的空间,这会 导致第一次启动组态数据库的速度较慢。

通过如下方法来配置要记录的历史变量:

第一步:打开要记录的历史变量所属驱动。

第二步:在驱动表格要记录的历史变量行和对应的"历史记录"列的项目上 点击,弹出历史数据对话框。

	变量	寄存器类型	寄存器地址	值类型	读写方式	报警	历史记录	操作记录	保存值	优先级	数据转换	描述
日知	[1						1					
	0	Holding Register	0	WORD	读写					100		
Þ	1	Holding Register	1	WORD	读写	(定时=1)		100		

第三步: 配置变量历史数据对话框,包括以下几个选项。

不记录:不记录变量历史值。

定时记录: 定时记录在磁盘历史库中,重启系统后仍然存在。记录周期为采 样间隔时间。可供历史趋势图、实时趋势图、历史报表、历史表格等使用。

缓存记录:每隔1 秒将采集值保存在内存中。记录笔数为缓存的变量采集值个数,即保存最近的多个变量值。可供实时趋势图使用。

值变记录:即值改变后才记录。记录在磁盘历史库中。

变量历史数据			×
○不记录			
②定时记录			
记录周期:	1	(单位:秒)	
○緩存记录():(見知):	可供实时起	월势图使用) │	
○ 值变记录(开发中,	」 暂时不要用 !)	
死区: 🛛			
确定		取消	



9.5 读取显示

系统历史数据可以用历史/实时趋势图、历史表格、历史报表、历史 数据读 取接口等读取显示。

9.5.1 历史趋势图

历史趋势图在画面窗口中显示。通过如下步骤创建:

打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"历史 趋势图",在窗口中创建即可。如下图所示:





▶ 增加、删除趋势线: 趋势图的左上角的小工具栏可以增加、删除曲线。

趋势线历史变量:点中某个趋势线后,就可以在属性栏对该趋势线进行编辑了,包括设置趋势线的所连接的历史变量、颜色等。

9.5.2 实时趋势图

与历史趋势图类似,但是显示变量最近值的趋势变化情况。如果变量有缓存记录或历史记录,则实时趋势图在打开时,即可从缓存或历史库中读取并显示最近 多个变量值,否则得重新采样记录变量值显示。





图 9-6

9.5.3 历史表格

历史表格在画面窗口中显示。通过如下步骤创建:

打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"历史表格",在窗口中创建即可。如下图所示:



图 9-7

- **增加、删除列**:表格的左上角的小工具栏可以增加、删除列。
- 列编辑:点中列可以设置该列属性。列有一个重要属性"类型",可以显示历史变量或者日期时间。如果想将日期时间分开显示,则可以设置两列的"类型"属性都为"时间",然后设置对应的"格式"属性即可。

9.5.4 历史报表

有关历史报表的详细介绍请参考报表的相关章节。

9.5.5 历史数据读取接口

数据库提供了"取历史数据"的函数命令,其原型为"历史数据=数据 库. 取历史数据(历史变量,起始时间,时长,间隔)"。

历史变量: 要获取历史数据的历史变量。

起始时间: 要获取的起始日期时间。

时长: 要获取的记录时间跨度。

间隔:获取历史数据的时间间隔(秒)。

返回:历史数据,返回记录的历史数据,为数组类型。



十、报警

10.1 概述

当系统中某些变量的值超过了所规定的界限时,系统自动产生相应警告信息,表明该量的值已经超限,提醒操作人员,这就是报警。本报警系统具有很高的访问速度和存储效率,尤其适合远程网络访问。

10.2 基本配置

在工程数据库浏览窗口中双击"系统配置"节点下的"报警配置"子节 点,打开报警配置对话框后即可进行配置。

 塔库 第 10 10<		~报警组			
新作 新名 建築市 新名 市		增加		播放方式:	安报警类型播放 🛛 🔽
客席 × 家 系统配置 系统启动 事件配置 方史记录 安全区 第月户管理 用户管理 月户登录 Web配置 振表 JavaScript 系统模块 JogA通信	929.	组名	是否记录	1 1 循环播放	无报警声音 所有报警统一声音
家統配置 (低低 () (低電 () () () () () () () () () () () () ()	B库 ×	888	E	类型	安报警组播放
水抗伯宜 ddd 低 Loopyflus 系统启动 jjj 面 Loopyflus 事件前置 一 历史记录 安全区 第用户管理 用户登录 服表 JavaScript 承統模块 ご录最小报警级别: 3 记录保存时间(天): 3 订0设备通信 (保存目录:		cec		低低	安报答类型播放
	🔤 系筑能宜	ddd		低	LoopyHus
● 第件前置 ● 市方史记录 ● 安全区 ● 安全区 ● 用户管理 ● 用户管理 ● 用户管理 ● 水素 ● 小素等級別: 3 ● 記录保存时词(天): 3 ● 法 ● 水 ● 水 ● 小振警級別: 3 ● 記录保存目录: ● ●	今 系统启动	jjj		高	LoopyMus
サ件配置 ● 方史记录 ● 安全区 ● 安全区 ● 用户管理 ● 用户登录 超速 ● 水 ● 安全区 ● 安全区 ● 安全区 ● 日户管理 ● 日户管理 ● 水 ● 水 ● 水 ● 安全区 ● 安全区 ● 日户管理 ● 日户管理 ● 水 ● 水 ● 安全区 ● 安全区 ● 日户管理 ● 日戸管理 ● 水				高高	LoopyMus
★ BuzzingB <	大田田田			开	BuzzingB
□ 历史记录 □ 安全区 ● 安全区 ● 用户管理 Ⅰ 用户登录 Ⅰ 服表 Ⅰ JavaScript Ⅰ 演奏小报警级别: 3 Ⅰ 记录最小报警级别: 3 Ⅰ 记录最小报警级别: 3 Ⅰ 记录最小报警级别: 3 □ 请选择保存目录, 空目录表示保存在工程默认路径下。 (保存目录: (Q)				关	BuzzingB
	 ● 安全区 ● 用户管理 ● 用户登录 ● Web配置 ● Web配置 ● JavaScript ● 系统模块 ● 10设备通信 	报警记录 记录最小报警 请选择保存目: 保存目录: [级别: 3 录,空目录表示保存在	记录保存时间(天)	: 3

报警组: 可以利用报警组对报警变量进行分类,方便查询、保存、声音等报警管理。可以增加、删除报警组,同时设置报警组的组名、 是否记录(是否保存到历史报警中)等属性。

报警声音: 当发生报警时,可以通过声音来通知操作人员。

报警记录:记录最小级别一当当前产生的报警级别大于等于记录最小级别时,



则记录到历史报警中,记录保存时间一到达设定的天数 后,系统将自动删除过期的报警记录。

保存目录:可以指定报警保存的目录路径,空表示保存在工程目录下。

10.3 报警变量

10.3.1 配置

通过如下方法来配置要进行报警的变量:

第一步:打开要报警变量所属驱动。

第二步:在驱动表格要记录的报警变量行和对应的"报警"列的项目上点击,弹出报警配置对话框。

	变量	类型	值变模式	初始值	读写方式	报警	历史记录	操作记录	保存值	数据转换	描述
-) 基	本支量		all from the second	4.1	00/001/00						
-	sr1	SHORT	随机	0	读写		1				
•	lr1	INT	随机	0	读写 🤇		2				

第三步:配置报警变量对话框。根据变量是否是开关量,分别弹出模拟量和 开关量的报警配置框。

模拟量报警配置	开关量报警配置 🛛 🛛 🔀
 报警组: ccc ▼ 报警级别: 1 报警类型 报警限值 报警文本 低低 0 低低 10 低报警 高 高高 面高 面高 面高 取消 	 报警组: ddd ▼ 报警级别: 1 报警类型与文本 ● 开 开报警 □ 关 確定 取消

"_报警状态"子变量 配置了"报警"的变量,会自动生成一个"报警状态"的子变量。可以类似普通变量一样,在动画、表达式、图形等引用该"报警状态"子变量。



状态变量包括如下值: 0-无报警; 1-开报警; 2-关报警; 3-高高报 警; 4-高报警; 5-低报警; 6-低低报警。 下图为引用状态变量的一个例 子。



10.4 短信报警

系统可以通过短信设备(一般是串口短信猫)发送短信报警。

串口: COM1 ₩ 短信中心: +86	空表示不发送 13800100500	波特率: 9600	🗙 数据位: 8 💌
短信格式			
\$报警时间\$时\$变量	\$发生报警		
接收手机 「 増加 (删除		
手机号	报警组	报警类型	报警级别
13588888888	ccc; ddd	高高:低	5
13366666666			5
说明 1. 如果一行中有多 2. 如果报警纽为空 3. 如果报警纽为空 4. 报警纽列·主	个手机号码,则可 ,则表示选择所有; 空,则表示选择所 当据繁练别士于等	以用":"分隔开。 报警组。 有报警类型。 于设置的练到时才发	·¥-

图 10-4



- 发送设备:一般是短信猫,需插入有效的手机 SIM 卡。
- 报警格式:定制报警短信格式。包括如下预定义内容:\$报警时间\$、
 \$变量\$、\$报警类型\$、\$报警组\$、\$报警级别\$、\$报警值\$。如:系统于
 \$报警时间\$发生了一个\$报警类型\$报警,值为\$报警值\$。
- **接收手机**:接收短信通知的手机号码。

10.5 微信报警

系统可以通过微信发送微信报警消息通知。 微信企业号先到微信公众平 台(https://mp.weixin.qq.com/) 注册一个微信企业号。一个企业号可以在 多个工程中共用。当前腾讯允许未认证企业号的成员上限为 200 个,而每天 可发的数量为:帐号上限数*30 人次/天。因此如果需要发送大量报警,最好 进行认证。

进入应用中心创建一个应用用于报警通知。要通知的用户得先关注该企业 号,然后进入通讯录,将需要通知的用户添加到"可见范围"内。



图 10-5



创建应用之后,接着进入报警应用。

AgentId: 必须与下文微信组态界面的 Agent ID 对应。

Secret: 必须与下文微信组态界面的 Secret 对应。

可见范围:要通知的用户,必须在可见范围内。

	报警推送 Z 组态软件及HMI的报警推送通知	日启用
Agentid	100002	编辑
Secret	EoOgyolVlkviaMPmanYvEKcUpFpdn6zZsZ58rRn2SV8	
可见范围	■ 报警组	

图 10-6

点击**我的企业->权限管理**,新建管理组,并设置好管理员、通讯录权限(需要将上述"可见范围"加入)、应用权限(需要将上述的"应用"加入)。

首页	通讯录	企业应用	微信插件	管理工具	我的企业
企业信息	管理员管理组十	管理组详情			
成员申请	▼ 北 超级管理组	编辑 册	削除		
权限管理	4 Admin	22	Admin		
聊天管理					
通讯录管理		管理员	管理员需绑定微信后才能登录管理后台		
安全与保密					
设置		通讯录权限	具有管理权限 副 报警组		
		应用权限	具有管理权限 报警推送		

图 10-7

进入我的企业->企业信息。

CorpID: 必须与下文微信组态界面的 CorpID 对应。



首页	通讯录	企业应用	微信插件	管理工具	我的企业
企业信息	企业信息				
成员申请	企业logo	????{	5		
权限管理		推荐尺寸702*180			
聊天管理	企业简称	?????公司 修改			
通讯录管理	今世中日	1 & # =			
安全与保密	企业部门	2 个部门			
设置					
	主体类型	企业 未认证			进入认证系统
	企业全称	???????????公司			认证后可提高使用人
	已使用/人数上限	1/200 申请扩容			
	创建时间	2017年2月23日			
	域名	绑定企业域名			
	CorpID	wxaf87596055919f7a			

图 10-8

微信组态配置

基本配置 短信报题 微信企业号 CorpID: CorpSecret:	冬 E-mail报警 wxaf875960559;	微信报警	
-微信企业号 CorpID: CorpSecret:	wxaf875960559)	10.57	
CorpID: CorpSecret:	wxaf8759605591	10.67 -	
CorpSecret:		IDITA	AgentID: 1
	Il6zleB5n2bio	oFu8-8r4N-pwotHhyDS	sDmhV-sjOhhSHAdWkXZPJLP ₂
信息格式			
在\$报警时间\$时	收到\$报警组\$\$	报警值\$。	
增加	開除	招数米刑	据数码别
user1;user	100 and 100 an	高:低	3
user3;user	l .		5
说明 1.如果一行中有 2.如果报警组为 3.如果报警类型 4.报警级别:表	多个微信号,贝 空,则表示选择 为空,则表示涉 为空,别表示涉	呵以用";"分隔开。 「奶有报警组。 「择所有报警类型。 「干等于设置的级别时	, 才发送。
	74	定 取省] [应用 (A)] 担 助

图 10-9



- 报警格式:定制报警短信格式。包括如下预定义内容:\$报警时间\$、\$变量
 \$、\$报警类型\$、\$报警组\$、\$报警级别\$、\$报警值\$。如:系统于\$报警时间\$发生了一个\$报警类型\$报警,值为\$报警值\$。
- ----自定义变量:报警格式还可以嵌入自定义变量,格式为:\$#变量#xxx驱动.xxx变量\$。如:\$#变量#定制变量.系统名称\$。
- 微信号:关注了企业号的微信帐号。
- CorpID、AgentID、Secret:分别与微信企业号中的CorpID、应用ID、应用Secret对应。

10.6 邮件报警

系统可以通过互联网发送报警邮件。

发送源		
SMTP服务器: smtp.126.com	登录密码:	****
发送E-mail: helwjh@126.com	确认密码:	*****
信自救武		
1000 (10 mm		
接收E-mail		
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1		
	(40 BP 카드 프네	17 56 / ct Du
增加 删除 E-mail地址 报警组	报警类型	报警级别
增加 删除 E-mail地址 报警组 8213284850 重要	报警类型 高高;高	报警级别 0
增加 删除 E-mail地址 报警组 8213284850 重要 helwjh0126 一般	报警类型 高高;高 低:低低	报警级别 0 3
增加 删除 E-mail地址 报警组 8213284850 重要 helwjh0126 一般	报警类型 高高;高 低:低低	报警级别 0 3
增加 删除 E-mail地址 报警组 8213284850 重要 helwjh@126 一般	报警类型 高高:高 低:低低	报警级别 0 3
增加 删除 E-mail地址 报警组 8213284850 重要 helwjh0126 一般	报警类型 高高:高 低:低低	报警级别 0 3
增加 删除 E-mail地址 报警组 8213284850 重要 helwjh0126 一般	报警类型 高高;高 低;低低	报警级别 0 3
增加 删除 E-mail地址 报警组 8213284850 重要 helwjh@126 一般 说明 1. 如果一行中有多个E-mail地址,则 2. 如果报警组为空,则表示选择所有:	报警类型 高高;高 低:低低 可以用":"分隔开 报警组。	报警级别 0 3

图 10-10

- SMTP 服务器: SMTP 协议的发送邮件务器。如, smtp.126.com、 smtp.qq.com。
- 发送 E-mail: 用户的 E-mail 地址,由于目前大部分邮件服务器的 E-mail 地址与登录用户一致,所以也是邮件服务器的登录用户。



- 登录密码:登录到邮件服务器的密码。
- 确认密码: 与登录密码一致。
- 报警格式:定制报警短信格式。包括如下预定义内容:\$报警时间\$、
 \$变量\$、\$报警类型\$、\$报警组\$、\$报警级别\$、\$报警值\$。如:系统于\$报警时间\$发生了一个\$报警类型\$报警,值为\$报警值\$。
- 接收 E-mail: 接收邮件通知的 E-mail 地址。

10.7 读取显示

报警记录可以用报警表格显示查看。

10.7.1 报警表格

报警表格在画面窗口中显示。通过如下步骤创建:

打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"报警 表格",在窗口中创建即可。报警表格分成历史报警表格和实时报警 表格,也 可以在运行时动态修改,如下图所示:

	f f										
实时	报警 🔽 确认	┃ 显示选项									
	报警时间	事件时间	委皇	事件类型	报警类型	报警组	报警级别	报警值	限值	恢复值	^
1			_		_	_	8 8	(3 7		
3			2	þ					()		1
4											
6	5			3			5		5 <u>7</u>		~
历史	北警 🖌 清除	□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□□		-	0	<u></u>	<u></u>		<u>to - 1</u>		
	报警时间	事件时间	变量	事件类型	报警类型	报警组	报警级别	报警值	限值	恢复值	^
1											
2											2
3											
4	2	53	6			1	8				
5											
6											V

图 10-11

删除报警列:对于不需要的报警列,可以点中该列进行删除。

10.7.2 显示选项与条件

静态设置:在属性栏中的"行为"卡下设置显示选项和条件。如下左图。



	报警显示条件
	报警类型 ✓ 湿示所有报警类型 ✓ 低低 ✓ 高 ✓ 开 ✓ 低 ✓ 高高 ✓ 关
届性栏 × 11报警表格2 ✓ 11届警表格2 ✓ 11日 計画 ✓ 11日 二 ▲ 11日 二 ▲ 11日 二 ▲ 11日 二 ▲	报警组 ✓显示所有报警记录组 ✓ ccc ✓ jjj
显示选项 A11 显示条件	报警级别 最小级别: 0 最大级别: 100 确定 取消

动态设置 运行时点击报警表格的"显示选项"按钮。如下右图。

实时报警	V	确认	【 显示选项 】 ◀ ↓ ▶	导出

报警类型 121显示所有报警类型	报警级别 最小级别: 0 最大级别: 100
 ✓ 低低 ✓ 高高 ✓ 并 ✓ 低 ✓ 高高 ✓ 并 ✓ 基 ✓ 基示所有报警记录组 ✓ ccc ✓ iiii 	 毎页显示方式 ● 显示所有记录 ● 按条数显示 毎页条数: 100 ● 选择日期显示 从: 2012~ 6-22 ↓ 到: 2012~ 6-22 ↓
	确定 取消

通过 js 脚本动态设置

调用表格"设置项目"函数。参数:项目,内容。

- 项目:0——设置类型,用逗号分隔
- 项目:1一一设置组,用逗号分隔



- 项目: 2——设置级别,比如 3[~]10
- 项目: 3——设置显示选项, 0-显示所有, 1-按天显示, 2-按页显示, 3-按日期显示

如:

设置项目(0, "高,低"); 设置项目(1, "ccc, j j j"); 设置项目(2. "5~9"); 设置项目(3, "2");

10.8 报警触发

数据库端的"系统变量"提供了"_报警"的变量,当有新报警或报警状态改 变时,该变量值会改变。可以通过"触发器"或"连接变量"功能连接该变量,从 而触发执行定制动作。该变量是一个"DWORD"(双字)类型值,第一个字包括如 下值:0-无报警、1-恢复报警、2-确认报警、3-新报警;第二个字报警 ID, 为系统内部使用,用来产生变化的值。

数据库端还提供了"当前报警"函数调用,该函数返回一个当前报警对象。当 "系统变量.报警"改变时,可以如下调用。

报警对象=数据库.当前报警();

域报警表格类似,报警对象包括如下域:报警时间、事件时间、报 警变 量、事件类型、报警类型、报警组、报警级别、报警值、报警 限值、恢复 值、报警状态。

例如: "标签1.标题=报警对象.报警变量;",则标签1的标题变成当前报警的变量名。

10.9 运行时动态设置

本节介绍工程运行时如何动态修改报警设置。

10.9.1 动态修改报警限值、类型等属性

根据变量项目 ID 设置变量项目值(含报警限值、报警类型等)。



函数原型:数据库.xxx 驱动._设变量项目(变量名,项目 ID,项目值)。

变量名:要设置项目的变量名。

项目 ID: 一个整数,表示要设置的项目。2-高高限值,3-高限值,4-低限值,5-低 低限值,6-启停报警,7-报警延时,8-开类型,9-关类型,10-高高类型,11-高类 型,12-低类型,13-低低类型。

项目值: 要设置的项目新值。

- 如果项目是报警限值,那么项目值是一个数值,比如:数据库.v1._设变量项目 ("alarm", 3, 100); //把驱动 v1 的变量 alarm 的高限值修改为 100。
- 如果项目是报警类型,那么项目值是一个字符串,比如:数据库.v1._设变量项目("alarm",11,"温度太高"); //把驱动 v1 的变量 alarm 的高类型内容提示 修改为"温度太高"。
- 如果项目是启停报警,那么项目值1表示停用,0表示不停用(即重新启用), 比如:数据库.v1._设变量项目("alarm", 6, 1); //停用驱动 v1 的变量 alarm 的报警。
- 如果项目是报警延时,那么项目值是个整数,表示报警延时的秒数,比如:数据库.v1._设变量项目("alarm", 7, 10); //把驱动 v1 的变量 alarm 的报警 延时改为 10 秒,如果 10 秒内报警恢复,则不触发报警。





十一、事件

11.1 概述

事件是指用户对系统的行为、动作。主要包括系统启动/停止事件,用户登录/注销事件,用户操作变量事件等。

本事件系统具有很高的访问速度和存储效率,尤其适合远程网络访问。

11.2 配置

在工程数据库浏览窗口中双击"系统配置"节点下的"事件配置"子节 点,打开事件配置对话框后即可进行配置。

	事件配置
数据库 ×	要保存的事件 ▼ 茶菇启动 ▼ 系统退出
 □ ● 系统記宣 ● 系统启动 ● 振擎配置 ● 振擎配置 	 ☑用戶登家 ☑用戶注销 ☑ 变量操作
→ → 历史记录 → → → 安全区 → ● 安全区 → ● 田戸管理 → ● 用户管理 → ● 用户登录	保存目录 请选择保存目录,空目录表示保存在工 程默认路径下。
₩eb配置 报表 JavaScript 承 系统模块	
	确定 取消 図 11-1

事件保存时间:到达设定的天数后,系统将自动删除过期的事件记录。 **保存目录**:可以指定事件保存的目录路径,空表示保存在工程目录下。

11.3 操作事件变量

通过如下方法来配置要记录操作的变量:



第一步:打开变量所属驱动。

第二步: 在驱动表格要记录的操作变量行和对应的"操作记录"列的项目上点击,弹出变量操作记录配置对话框。

重变	类型	值变模式	初始值	读写方式	报警	历史记录	操作记录	保存值	数据转换	描述
基本变量										
sr1	SHORT	随机	0	读写						
lr1	INT	随机	0	读写		(
br1	BYTE	B宿村	0	读写				- E		

第三步: 配置变量操作记录对话框。

2 000000	12家内谷 店/-2	_ 增加
7 000000	1日 1-3	册所
10.000000	值7~10	
"值记录内容		

包括如下选项:

- 记录用户操作:是否记录用户对变量的操作事件。
- 记录系统操作:是否记录系统(主要时采集值改变)变量的变化事件。
- 值对应记录内容列表:记录某个区间段对应值的记录内容,如果为空,则
 不记录。
- 其它值记录内容:上述列表以外的其它值的记录内容,如果为空,则不记录。

11.4 读取显示

事件记录可以用事件表格显示查看。

11.4.1 事件表格



事件表格在画面窗口中显示。通过如下步骤创建:

打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"事件 表格",在窗口中创建即可。如下图所示:

		** ***	导出					0
		时间		内容	值	操作员	事件源	^
	1							
4	2							4
	3		þ þ					
	4							
	5							
	6							~

删除事件列:对于不需要的事件列,可以点中该列进行删除。

11.4.2 显示选项与内容

静态设置:在属性栏中的"行为"卡下设置显示选项和内容。如下左图。 动态设置:运行时点击事件表格的"显示内容"按钮。如下右图。

	事件显示内容
届性栏 × □ 事件表格1 ✓ ☞ 属性 ●	要显示的事件 又系统启动 又系统退出 受量操作 可用户登录 可用户注销
 删除线 否 显示选项 A11 显示内容 系统启动: 系 	 毎页显示方式 ● 显示所有记录 ○ 按天显示 ○ 按条数显示 毎页条数: 100 ○ 选择日期显示 从: 2012- 3-29 ▼
显示内容	到: 2012- 3-29 ▼

十二、用户与安全区

12.1 概述

系统通过用户管理和安全区的相互配合来共同保证组态工程运行的安全 性。



每个用户通过密码正确登录系统后,就拥有了该用户所属组的安全权限和 安全区属性。系统根据安全权限和安全区来判断是否允许当前用户执行某个操 作。

画面窗口上每个图形对象都有安全区属性。如果该对象的安全区为 空,则任何用户都可以操作;否则,当前用户要操作该对象时,只有该用户拥有该 对象安全区列表之一的安全区时,才允许操作。

12.2 安全区配置

在工程数据库浏览窗口中双击"系统配置"节点下的"安全区"子节点, 打开安全区配置对话框后即可进行配置。

数据库 ×		
	安全区配置 安全区列表 III ▲ ◆ ◆ asa bbb ccc	孫定 取消
 用户登录 Web配置 报表 JavaScript 承统模块 ① IO设备通信 ③ 画面 圆数据库 		

可以在该安全区配置对话框中增加/删除安全区。

12.3 用户管理

12.3.1 组态时用户管理

在工程数据库浏览窗口中双击"系统配置"节点下的"用户管理"子节点, 打开用户管理配置对话框后即可进行配置。



数据库	× 用户配置				
🖃 🔜 系统配置	用户组列表:				
	组名	安全区	权限		增加组
	gl	aaa;	启停系统		
● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	e2	bbb;			
	用户列表:	田口祖	1 啓己时12		
	用尸名	用尸组	登录时长	密码	
1 抵表	user1	gl 2	1		删除用户
 ✓ JavaScript Ⅲ 承 系统模块 Ⅲ 叠 IO设备通信 	userz				
國國 圖数据库		Đ	航定 I I I I I I I I I I I I I I I I I I I	湖	

- 用户组:可以增加/删除用户组,以及编辑用户组的组名、安全区和权限。
 所有该组的用户都具有这个组的安全区和权限。
- 用户:可以增加/删除用户,以及编辑用户的名称、所属组、登录时长、 密码等属性。一旦用户归属某个用户组,该用户就具有这个组的安全区和 权限。
- 登录时长:指用户登录后如果在指定的时间(单位为分钟)内没有发生任何操作,系统自动注销该用户。输入"0"表示不限制登录时间。

12.3.2 运行时编辑用户

运行时调用画面的"编辑用户"命令,并且当前登录用户具有"编辑用户" 的权限,就可以弹出如下对话框进行运行时动态编辑用户了。

用户名	用户组	登录时长	密码	増加
user1	g1	1	***	
user2	e2	0	***	
用户1	g1	0	***	编辑
u3	g1	100		-
				-
				-
				(
				備定
				The bill
				取消

12.4 用户登录

用户登录主要是配置用户启动或者停止时,是否需要登录确认。 在工程



数据库浏览窗口中双击"系统配置"节点下的"用户登录"子节点,打开用 户登录配置对话框后即可进行配置。如下图所示:

数据库	×	
 □		
用戶管理	用户	登录设置 🔀
● WebBC里 报表 JavaScript ● 承 系统模块 ■ 通 IO设备通信		 □ 系统启动需要用户登录确认 □ 系统停止需要用户登录确认
國國 圖数据库		确定 即消



十三、配方

13.1 概述

配方描述了制造一件产品所用的不同配料成份之间的比例关系。

系统配方由"成份"和"配方组"构成。配方中"成份"对应某个变量, "配方组"就是配方的"成份"所分别对应变量的实际值,一个配方可以有多 个组态时配置好的"配方组",也可以在运行时动态添加和修改。

13.2 创建与编辑

在工程数据库浏览窗口中右键点击"系统模块"节点下的"配方"子节 点 弹出菜单后,点击"新增配方"即可新建配方。



權計产品	品配方.tdv					
	★! 删除成份	名』 増加配方組	★! 删除配方组			
成份	外部变	量	类型	配方组1	配方组2	配方组3
-	-		-	果汁	蜜露	饮料
水	橙汁配:	方.水	WORD	60	20	10
混合物	橙汁配	方.混合物	WORD	20	30	30
糖	橙汁配	方.糖	WORD	6	10	6
香料	橙汁配:	方.香料	WORD	100	60	50

在配方编辑中,可以进行增加/删除成份、增加/删除配方组等操作。上图就



是一个橙汁产品配方的例子。

13.3 配方表格

配方可以用配方表格进行显示查看、动态增加/删除/编辑等操作。

配方表格在画面窗口中显示。通过如下步骤创建:

打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"配方 表格",在窗口中创建即可。如下图所示:



- 选择配方:组态开发时点击配方小图标可以选择初始运行时的配方,运行时也可以通过下拉框来动态选择配方。
- 配方编辑:运行时可以通过工具栏的按钮动态增加/删除/编辑配方,也可以 保存、应用配方。



十四、时间表

14.1 概述

时间表可以计划在每周规定的时间内外执行一系列的动作。

运行时可以通过时间表表格来动态修改预定义的时间段。

14.2 创建与编辑

在工程数据库浏览窗口中右键点击"系统模块"节点下的"时间表"子节点弹 出菜单后,点击"新增时间表"即可新建时间表。



在时间表编辑中,可以进行增加/删除/编辑时间段、增加/删除/编辑动作等操 作。上图就是一个时间表的例子。

14.3 时间表表格

时间表的时间段可以用时间表表格运行时进行显示查看、动态增加/删除/编辑 等操作。



时间表表格在画面窗口中显示。通过如下步骤创建: 打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"时间 表表格",在窗口中创建即可。如下图所示:

起始时间 發圧时间	一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一 一	 	1		删除	编辑		1
				起始时间		终止即	前间	

选择时间表:组态开发时通过属性可以选择初始运行时的时间表。运行时也可以通过下拉框来动态选择时间表。

时间表编辑:运行时可以通过工具栏的按钮动态增加/删除/编辑时间表,修改完 后可以应用生效。



十五、网络与Web

15.1 概述

网络(Web)系统是本公司精心设计开发的,具有全球革命性、领先的Web 技术,与其它组态软件的网络功能相比,具有如下众多特点:

1. 全球首款利用 HTML5 实现无插件、纯 Web 监控的组态软件。

2. 内置微小、高效的网络/WEB 模块,与实时数据库完美结合,因此天生具备 网络/WEB 服务功能。目前大部分组态软件得借助其它 Web 服务器完成 Web 服务 功能。

3. 组态文件无须发布,通过 WEB 浏览器或者组态浏览器即可远程监控该设备工程。目前几乎所有组态都必须发布画面页面才能浏览,每次修改了画面文件又得重新发布,操作非常繁琐。

4. B/S 与 C/S 架构完全一致。目前几乎所有组态软件的 B/S 与 C/S 架构 差别很大。

5. 单机版本上的任何画面操作,网络版本都能完成。目前几乎所有组态软件的网络(尤其是 Web)的画面操作都有很多的功能限制。

 6. 画面刷新采用值变通知机制,即只有变量值改变了才通知画面客户端, 值不变的变量不发送。目前几乎所有组态软件都采用画面不断主动请求值机 制,导致网络数据包增多,通信效率低下。

7. 批量连接、读写数据库变量值:系统可以自动或手动批量处理大量画面发 往数据库的请求。每次画面请求连接、读写数据库变量值时,会发送一个请求 包,然后等待数据库端应答,如果变量数目多,则请求包也相应增加,导致网络 数据包增多,通信效率低下,还增加了数据库服务器端的负担。本系统提供了批量处理功能,只要一个包即可完成大量请求。目前很少软件具备这个功能。

8. 智能改变变量值时间通知间隔:系统能根据网络传输速度自动调整通 知频率,该特点尤其适用于广域网等传送速率较低的网络。



9. 优化网络传输数据包:如:仅下载修改过的画面文件,减少网络传输内容;画面文件经过压缩后下载;对变量值、报警、事件等的数据包通知内容都进行了大量优化。

10. 插件安装非常简单,当系统还没安装插件时,系统会自动提示下载。

11. 插件非常小 (才 200 多 K), 而且优化插件下载, 即需要用到的图 形库 程序文件, 系统才会下载安装。

网络功能其它说明:

1. 由于本系统具备特有的 3D 功能,有的 3D 模型可能达到大几十兆, 拥有众 多 3D 模型的 3D 窗口总共可能达到上百兆,即使经过压缩,总共也会达到几十 兆。如此庞大的文件,在局域网内监控没有任何 问题,但是在互联网、广域网等 传输较慢的网络下,有时候下载得等待较长的时间。针对这种情况,未来计划用多 线程断点续传的技术来解决。

2. 由于本系统具备特有的跨平台架构,所以未来能实现各个平台间的相互 监控,例如:利用 iPhone、Android 监控 Windows、Wince、 Linux 等平台的 实时数据库。

3. 未授权的 Web 网络版本允许三个客户端同时监控。

15.2 配置

相比较其它组态软件,本系统的 Web 配置极其简单。而且配置完,不必发 布,启动组态运行数据库后,在远程 IE 上输入本机 IP 或机器 名即可监控, 在本机的 IE 输入 127.0.0.1 也可以监控。因此,从配置 就可以领略到本系统 方便、易用、傻瓜式的 Web 功能。

在工程数据库浏览窗口中双击"系统配置"节点下的"Web 配置"子节 点,打开Web 配置对话框后即可进行配置。另外,网络版本的组态 运行数据库 的启动界面上,也可以直接配置。但是它们的配置内容略有不同,前者多了智 能移动的"主窗口"的配置(请参考智能移动Web 监控章节)。



数据库 >	×		
□ 🔜 系统配置	■ 组态数据库		
● ● 报警配置	运行消息监控用户		后台设置 Web设置
	时间	源	消息
	10:34:31	系统	系统启动
1 日户管理	10:34:31	系统	启动Web服务(端口=80)成功。
日白香夏	2011-07-24 10:34:32	系统	启动工程: E:\简单演示\简单演示
报表 JavaScript			
 □ ● ● 系统模块 □ ● ● ● ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■ ■			
國國 副数据库			



Web网络设置	
Web网络服务端口 端口: 80 (1~65535, 仅对网络版本有效) 注:如果端口的设置不是80,则浏览器访问时要在IP之后添加"端口"。 比如数据库IP是192.168.0.100,而端口是200,则浏览器需要输入"192.168.0.100:200" 来访问数据库。 Html5Web(智能移动Web) 主窗口: 手机HOME P2P云(穿透内网)监控 这种孩工程	
工程ID: 密码: 确认密码: 服务器: 心跳间隔: 300 确定 取消	▼eb网络端口设置 Web服务端口(1~65535): 10 (仅对网络版本有效) 注:如果端口的设置不是80,则浏览器访问时要在IP之后添加":端口"。 比如数据库IP是192.168.0.100,而端口是200,则浏览器需要输入"192.168.0.100:200"来访问数据库。 确定 取消

Web 服务端口:默认是80。



15.3 远程浏览

15.3.1 远程浏览Windows 工程

远程启动 IE 浏览器后,输入要浏览组态数据库机器的 IP 或机器名或 域名,如果端口不为 80,则在后面加上":端口"。如 IP 为 192.168.0.6,端口为 800,则输入 http://192.168.0.6:800。

远程浏览需要安装一个控件,如果该机器没有安装过本组态软件并且是第一次 浏览,则会提示如下界面:

🏉 简单演示			
如果无法浏览,可能是由于下面原因 1.尚未下载安装组态浏览器控件。下 2.Web浏览器禁止了ActiveX控件的运行 3.没有使用IE或基于IE技术的Web浏览	因造成: <u></u>		
您也可使用监控浏览器程序监控,行	マ具有微小、高	霸效、安全等特点。	下载该程序

根据用户情况可以选择安装控件或组态监控浏览器程序。

注: 组态监控浏览器包含了控件,所以如果是下载安装了组态浏览器,则自动安装了控件,这样 IE 也可以浏览了。

安装了控件或者监控浏览器后,刷新页面或者启动监控浏览器,就可以远程 监控了。

15.3.2 完全版组态浏览器与控件

上述只是一个简易版的组态浏览器或控件,安装运行后,会根据需要从服务 器上下载安装其它需要的组件。如果部分机器无法下载或安装简易版或其他相 关组件,也可以直接安装完全版的组态浏览器与控件。

15.3.3 64 位机器

软件目前是 32 位,在 64 位的机器上必须用 32 位的 IE 才能监控。



15.4 智能移动web

15.4.1 概述

本系统是全球第一款利用 HTML5 技术实现无插件、纯 Web 的 Web 监控组态软件。

HTML5: HTML5 是近十年来 Web 开发标准最巨大的飞跃。和以前的版本不同, HTML5 并非仅仅用来表示 Web 内容,它的新使命是将 Web 带入一个成熟的应用平 台,在 HTML5 平台上,图像、动画、视 频、音频,以及同电脑的交互都被标准 化。

正是 HTML5 标准的诞生,使得组态界多年的无插件、纯 Web 监控梦想变成现 实!

目前主要的Web 浏览器都已经支持HTML5,如: IE、Chrome、Firefox、Safari、Opera 等。

注: IE 需要9以上版本才支持,而 IE9 只能在 Vista、Win7、WP7 上运行。

15.4.2 配置

智能移动 Web 监控需要指定专门的主窗口。在工程数据库浏览窗口中双击"系统配置"节点下的"Web 配置"子节点,打开 Web 配置对话框后即可进行配置。



图 15-1

智能移动 Web 主窗口:请选择一个窗口做为智能移动 Web 监控的主窗口。



15.4.3 功能说明

当前利用 HTML5 实现的 Web 画面客户端功能未完全实现,所以不推荐 PC 利用该功能进行监控。不过功能会逐步增加。

当前 HTML5 画面窗口图形仅仅实现了直线、标签、按钮、编辑框、 矩形、圆、多边形、折线、图像等常见图形,而且仅实现了这些图 形的一些主要属性。

目前图形的事件动作和图形动画尚未实现,但是利用 JavaScript 可完 全代 替实现这些功能。由于 JavaScript 是 Web 浏览器的标准语言,而本系统的内置 语言也是 JavaScript,本系统已通过 JavaScript 与 Web 浏览器完美的结合在一 起了。

15.4.4 如何在 PC 上调试

PC 上调试时,需要在URL 最后增加"/html5"进行浏览调试。例如, 组态数据库运行起来后,在Web 浏览器输入"127.0.0.1/html5"即可(假如端口是80),注意"html5"后不能有"/"!实际智能移动Web监控不必添加"/html5"。

XP/Win2K/Win2003:由于 IE 只有 9+版本才支持 HTML5,而 IE9+不 支持
XP/Win2K/Win2003 操作系统,所以得下载安装 Chrome (推荐)、Firefox、
Safari、Opera 等 Web 浏览器之一方可调试。 Vista/Win7:这些系统上,需要将
IE 升级到 IE9 才能调试,或者下载 安装 Chrome (推荐)、Firefox、Safari、
Opera 等 Web 浏览器之一进 行调试。

<u>强烈建议:先在PC的chrome上调试,因为手机缓存不容易清除,每次</u> 修改画面,有些手机需要较长时间才能更新到新画面,即使刷 新也无济于 <u>事。</u>

15.4.5 如何在智能移动上监控

目前主要的智能移动平台(Android、iPad/iPhone、WP7)等的 Web 浏览 器都支持了 HTML5。

组态数据库运行起来后,在 Web 浏览器输入该组态数据库机器的 URL 或


IP 即可(如果端口不是80,还需输入端口),与 PC 调试不同,不必在后面 增加"/html5"。

15.4.6 一些说明

1. 浏览器一般都有缓存,所以如果修改了画面,浏览器得强行刷新或者清除缓存后,才能看到修改后的内容。

2. 智能移动是无线网络,而数据库服务器运行在 PC 上,如果 PC 有公网 IP (或者映射公网 IP),则智能移动可直接连接浏览监控。否则 得通用 Wi-Fi、 ZigBee 等无线网络连接到 PC 上监控。

15.5 手机客户端监控

15.5.1 简介

WEB 版本当前支持苹果、安卓客户端监控,未来将支持微软的手机监控。

15.5.2 配置

默认情况下, APP 客户端的主窗口、布局与 PC 客户端的主窗口、布局一致。 也可以通过画面->画面配置->手机客户端布局进行配置。如图:



其中大部分配置内容与 PC 画面布局配置一致,可以参考画面布局外观部分。



主要区别就是这里是"手机画面分辨率",而PC 是利用"工程分辨率"进行自适应、居中计算。

15.5.3 安卓APP 操作

安卓手机客户端为一个 APK 包(应用程序包文件-application package file),直接复制到安卓系统上安装即可使用。 客户端程序启动后,弹出如下界面。



点击增加按钮弹出如下界面,在编辑框中输入要监控的数据库服务器 IP 或名称即可,如果端口不是 80,则需要在尾部增加":端口号"(如端口为 6000,则 192.168.1.101:6000)。

日 请	输入数据库服务器
192.168	.1.101
	确定

15.5.4 苹果APP 操作

苹果 APP 请到 App Store 上下载安装。操作与安卓 APP 类似。



15.5.5 一些说明

1. 以竖屏连接: 客户端启动连接时, 建议以竖屏连接。

2. 横屏监控:对于很宽的图形,为了看到其余部分,可以缩小显示,但是 缩小后图形不清晰,这种情况就可以采用横屏监控。当切换到横屏时,为了显示更 大的主区域,会自动隐藏所有栏框,当切换回竖屏时,又恢复原来的栏框。如图:





 多点触控缩放、拖动:可以通过多点触控对当前窗口自由缩放,超过屏幕大小的 窗口可以通过拖动查看隐藏部分。



15.6 P2P **云**

15.6.1 基本概念

P2P: 点对点 (Peer to Peer, 简称 p2p) 对等网络技术。

NAT: 当在专用网内部的一些主机本来已经分配到了本地 IP 地址(即仅在本专 用网内使用的专用地址),但现在又想和因特网上的主机通信(并不需要加密) 时,可使用 NAT 方法。这样,所有使用本地地址的主机在和外界通信时,都要在 NAT 路由器上将其本地地址转换成全球 IP 地址,才能和因特网连接。这种通过使 用少量的公有 IP 地址代表较多的私有 IP 地址的方式,将有助于减缓可用的 IP 地址 空间的枯竭。

穿透内网:实际上就是指穿透 NAT。NAT 技术虽然在一定程度上解决了 IPv4 地址短缺的问题,在构建防火墙、保证网络安全方面都发挥了一定的作用,却破坏了端到端的网络通信。NAT 阻碍主机进行 P2P 通信的主要原因是 NAT 不允许外网 主机主动访问内网主机,但是 P2P 技术却要求通信双方都能主动发起访问,所以 要在 NAT 网络环境中进行有效的 P2P 通信,就必须采用新的解决方案。

15.6.2 总结

上述概念对于用户来说可能不理解。 总之,本系统可以在不需要公网 IP 或 者其它技术(花生壳)等的情况下,在另外一个局域网通过互联网或手机通过 2G/3G/4G 监控到本系统的运行情况。

15.6.3 云屏、云盒子

由于本系统的嵌入式 WINCE 版本与 PC 版本具有同样强大的 web 网络功能, 而极少嵌入式组态系统拥有网络 web 功能,更谈不上移动、云的监控功能。因此, 本公司现已推出高性价比的,以 WINCE 版本为基础的云屏(带云监控功能的 HMI 触摸屏)、云盒子(去除 HMI 屏,保留云监控功能)的嵌入式硬件系统。 借助云 屏、云盒子,用户可以非常方便、高效、廉价地将组态监控系统置于云上。这样只 要有互联网,用户就可以随时随地利用 PC 或手机监控到现场。



15.6.4 配置

在工程数据库浏览窗口中双击"系统配置"节点下的"Web 配置"子节 点,打开Web 配置对话框后即可进行配置。



Web网络设置	×
Web网络服务	5端口
端口: 80	(1~65535,仅对网络版本有效)
注: 如 时要在IP之 比如数 200,则测 来访问数据	1果端口的设置不是80,则浏览器访问 2居添加":端口"。 据库IP是192.168.0.100,而端口是 览器需要输入"192.168.0.100:200" 5库。
Html5Web(潅	g能移动Web)
主窗口: [手机HOME ▼ 显示登录框
P2P云 (穿透	内网)监控
☑ 启用P2P	注册该工程
工程ID:	xds. demo123
密码:	•••••
确认密码:	•••••
服务器:	
心跳间隔:	300
	确定取消



15.6.4.1 配置P2P 云步骤:

1. 启用 P2P: 要使用 P2P 云功能,首先需要将"启用 P2P"打勾。

2. 输入工程 ID、密码。对于某个云服务器来说,工程 ID 是唯一的,用来标识这个工程或嵌入式设备。工程 ID 是英文字母、数字、"."的 组合,如:witlinc.demo3。

3. 服务器:即云服务器 IP,目前设置为空即可。

4. 注册该工程:点击"注册该工程"按钮,软件将以当前输入的密码在当前云服务器注册这个工程 ID。

5. 设置心跳间隔:在繁忙、不稳定的网络环境下,可以将该值置小。

15.6.4.2 进行P2P 云监控:

1.运行工程或嵌入式设备:系统运行后,会以设置的工程 ID、密码到云服务器上登录,并告诉云服务器当前系统机器或设备在互联网中的位置。

2. 运行客户端:包括 PC 的组态浏览器客户端和移动的 APP 客户端,在客户端中输入"p2p:工程 ID",如上图中的配置,则输入"p2p:witlinc.demo1"。 客户端则会以这个工程 ID 到云服务器上查询对应工程机器或设备在互联网中所 在的位置,客户端知道位置后开始尝试 P2P 穿透监控。

15.6.4.3 一些注意点:

1. 避免 ID 重复:如果某工程以某个 ID 运行后,另外一个工程也以此 ID 运行,而且密码都正确,则新工程会代替旧工程,客户端以该 ID 监控时,则看到的是新工程。因此工程 ID 与密码应该足够复杂。尤 其是嵌入式设备,工程可能都一样,但是下载到每个设备的 ID 必须 不一样,否则新设备会顶替掉旧设备。

2. 局域网内先监控:有条件的情况下,可以在局域网内先监控,因为第一次 监控时,会将工程文件自动下载到客户端,再次监控就不 会下载了,目前影响监 控效率的97%以上是工程文件引起的。如果 在客户端监控,系统能检测与工程服 务器同个局域网,会自动进行局域网监控,而无须P2P。众所周知,局域网监控一 般都是百兆速度,再加上丢包等情况,通信效率一般是互联网的几百倍。另外,



在局域网监控时,也必须输"p2p:工程 ID",不能输入工程机器的 IP, 因为这 两种输入方式是分别下载到不同的目录下。也可以参考《互联网云监控要点》一 节。

3. P2P 监控只支持客户端,不支持 WEB 浏览器。PC 上可以用组态浏览器监控,手机上可以用 APP 客户端监控。

4. 在繁忙、不稳定的网络环境下,可以将心跳间隔置小。

15.7 互联网云监控

越来越多的用户需通过互联网监控,本节就着重介绍一下互联网监控的一些 注意点。

15.7.1 互联网监控的方式

公有云 IP:通过租借云上的公网 IP(如:阿里云)等,一般上传运行工程 后,在客户端直接输入云 IP 即可开始监控。

私有云 IP:本地拥有固定公网 IP 时,一般采用端口映射方法来实现(包括路由器映射、利用 porttunnel 等端口映射软件等)。有关如何进行端口映射相关资料较多,这里就不再详述了。

无公网 IP: 目前几乎所有组态软件都没有提供无公网 IP 情况下监控内网的解决方案,但是可以借助"花生壳"等工具实现。

P2P 云:本系统采用独特的 P2P 穿透技术,可以不需要公网 IP 与其他辅助 工具情况下,实现外网穿透内网进行监控。详情请参考 P2P 云一节。

15.7.2 互联网监控要点——提高互联网的通讯效率

无论采用上述何种方式,它们都有共同点,即互联网的网速、稳定性远远低 于局域网(局域网一般都是百兆,而互联网共享一兆都没有,考虑丢包等不稳定 因素,局域网内通讯效率是互联网几百倍),因此需要尽可能地提高互联网通讯 的效率。

尽管本系统对工程文件、监控数据的传输做了大量的优化,但是了解下面的设



计要点将有助于进一步提高通讯效率。互联网监控体验:对于用户来说,影响互联 网体验主要有3 个响应等待时间,而响应等待时间主要是由请求应答包的个数决 定。

A)连接系统的响应时间。

B)窗口切换(窗口显示)的响应时间。

C)监控控制(即与数据库相关操作,主要是读写数据库变量)的响应时间。 其中,A)连接系统时间由路由寻找时间、获取系统初始信息响应时间、初始窗口显 示时间组成,而路由与获取信息是必备的,已经无法优化,只剩窗口显示时间,因 此A)与B)实际上是同个问题。

本节的主要内容就是介绍如何提高本系统在互联网通讯(包括 P2P 云)的效率,即如何提高上述3个响应时间,也就是说如何降低请求应答包的个数。可以 从两大部分来提高通讯效率,数据传输与文件传输。

1. 数据传输:数据传输要解决的就是上述的C)监控控制响应时间。主要在 组态开发时要注意如下2 个问题。

A) 尽量避免在画面定时器中不断执行与数据库相关操作(比如读写 数据库 变量值)。

B)对于需要大量读写数据库变量、连接数据库变量的操作,务必加 上"批 量请求"相关函数。

有关这2个问题的详细解决方法请参考《网络WEB 组态开发说明及要点》一节。

实践证明,只要注意这2个问题,数据传输对互联网监控体验影响几乎可以忽略不计。因为经过优化后,几乎只要1个请求应答包。而且本系统是采用变量值改变后才通知客户端的机制,因此数据传输的效率已经到达极致。

2. 文件传输:从目前本系统大量实际运行情况看,文件传输是影响用户互联网监控体验最主要的因素,也就是上述的 A)连接系统时间 与 B)窗口切换时间。在介绍要点之前,需要先介绍一下本系统的文件传输机制。

当客户端显示某个窗口(包括内部的媒体图像等文件)时,会先检测该窗口是否需要下载。当窗口文件不一致,或者从来没有下载过,则会发送请求



下载。另外,系统是通过文件时间和大小来判断是否与服务器的文件一致。下面的要点就是根据这个机制制定的。

A)局域网内先监控:有条件的情况下,可以在局域网内先监控,因为第一次监控时,会将工程文件自动下载到客户端,再次监控就不会下载了,目前影响监控效率的97%以上是工程文件引起的。尤其是某窗口如果使用了大量的图片,则此窗口第一次在互联网打开时间比较长,而如果在局域网先下载过,则不会出现此问题。

B)上传工程文件:当上传新的工程到服务器上时,尤其要注意,有些上传 工具会把工程文件(包括媒体图片等文件)时间修改为上传的时间,而实际上 这些文件没有修改过,但是客户端判断到这些文件时间与服务器的不一致,则 又会重新下载!建议采用 rar、zip 等压缩工具把整个工程压缩打包上传,然 后解压,这样会保留实际的文件时间。附带说一句,WINCE 设备下载工程不会 有这个问题。

15.8 浏览指定窗口

远程监控时,默认画面监控方式与单机版本一致,即首先是主窗口以及画面整体布局,然后经过操作进入某个指定窗口。但是,在特殊情况下,例如与其它http服务器(如:IIS、apache)等结合使用时,某个URL链接可能需要监控指定的窗口,那么可以按照如下方法实现。

15.8.1 浏览指定窗口

在 URL 链接后面增加 "wnd=窗口名称"。例如: 组态运行数据库服务器 IP 为 192.168.0.6,指定的窗口为 "锅炉",则 Url 链接为 "http://192.168.0.6/wnd=锅炉"。

15.8.2 浏览指定三维窗口

在 URL 链接后面增加 "3dw=三维窗口名称"。例如:组态运行数据库服务器 IP 为192.168.0.6,指定的窗口为"三维锅炉",则 Ur1 链接为 "http://192.168.0.6/3dw=三维锅炉"。



15.8.3 保留原来画面布局

如果浏览时要保留画面布局,即保留上下左右等栏框,则在后面增加 "?layout=1"。上述两个例子分别是: "http://192.168.0.6/wnd=锅 炉?layout=1"和"http://192.168.0.6/3dw=三维锅炉?layout=1"

15.8.4 传入参数

指定 URL 时,也可以传入自定义的参数,在 URL 后面加上"?参数名称=参数值"即可,而后通过"画面.取输入参数("参数名称")"来获取参数值。

如: "http://192.168.0.6/wnd=锅炉?aaa=23?bbb=xyz", 然后 画面. 取输入 参数("aaa")将返回"23", 而 画面. 取输入参数("bbb")将返回"xyz"。

15.9 主要问题

问题描述: 在某些 WIN7 系统上运行时,会提示权限问题(如下图), 主要原因是 WIN7 的安全性比较高, Win7 自带的防火墙限制了端口 80 的访问,而本软件的 web 默认端口也是 80。



解决方法 1: 将本软件的 web 端口修改称其它端口即可,如何修改端口 请参考前面章节。

解决方法 2: 重新设置 Win7 防火墙规则。

点击 Win7 系统桌面左下方的圆形开始按钮,依次打开 Win7 系统的 "控制面板→系统和安全→Windows 防火墙",点击左侧菜单中的"高级设置"。



🚱 🕞 🖉 ▶ 控制面板 ▶ 系统	和安全 🕨 Windows 防火墙	and an arrive the
控制面板主页 允许程序或功能通过 Windows 防火墙 受更改通知设置 受打开或关闭 Windows 防火墙 受还原默认设置 受 <u>高级设置</u> 对网络进行疑难解答	使用 Windows 防火墙来帮助锅 Windows 防火墙有助于防止黑客或恶意等 防火墙如何帮助保护计算机? 什么是网络位置? ② 家庭或工作(专用)网络(如 您知道且信任的用户和设备所在的家庭或 Windows 防火墙状态: 传入连接: 活动的家庭或工作(专用)网络:	科护您的计算机 次件通过 Internet 或网络访问您的讨 O) 乾工作网络

在"高级安全 Windows 防火墙"设置面板中,鼠标右键点击"入站规则",从 弹出菜单中点击"新建规则"。

 ▲ 注接安 按配置文件筛选(P) ▲ 监视 按状态筛选(S) 按组筛选(G) 查看(V) 		新建规则(N)	
	■	按配置文件筛选(P) 按状态筛选(S) 按组筛选(G) 查看(V)	+ + +

在"要创建的规则类型"中,点选"端口",下一步,选择"TCP",点选 "特定本地端口"后填入端口号 80。 点击"下一步",点选"允许连接",再点 "下一步",根据用户的实际情况和需求勾选域、专用或共用等选项,最后为这 个规则命名,允许连接。

15.10 组态开发要点

15.10.1 概述

区别与传统组态软件,本组态是一款真正意义、完全的 B/S、C/S 架构的组态软件。 有关本组态的特点与配置使用,请参考手册的《网络与 Web》相关章节,本节



主要介绍网 络 Web 组态开发(尤其是 JavaScript 开发)的主要概念、原理、要 点。掌握了这些要点,网络 Web 功能就能发挥更强大的作用。

15.10.2 画面端与数据库端

系统的网络 Web 分成画面端和数据库端,即画面监控客户端(包括 Web 浏览 器监 控)和数据库运行服务器端。画面端的组件包括窗口、三维窗口、画面模 块、画面变量 等,数据库端的组件包括驱动、数据库模块、数据库变量等。

有关画面和数据库具体模块可以通过工程浏览查看,如下图所示。详情可参 考"工程 浏览"章节。



一台机器只能运行一个数据库程序,其上只能运行一个组态工程,而可以有多个 机器启动画面监控端连接到这个数据库端进行监控。

画面端可以调用数据库端组件,而数据库端不能调用画面端组件。试想一下, 当多个 画面端连接到数据库时,数据库应该调用哪个画面的组件呢?显然是做不 到的。所以,在 进行画面的组态开发时,会看到"画面.XXX"和"数据 库.XXX",就是表示当前是调用画面 或数据库的组件/变量。而进行数据库的组态 开发时,只能看到"XXX",这是因为,数据 库端只能调用数据库组件/变量,所 以"XXX"默认就是数据库端的组件/变量。

虽然数据库端不能调用画面端组件,但是画面端当前的监控窗口连接了数据库端的某 些变量时,当这些变量的值改变了,就会通知画面端,从而触发画面执行 某些操作。例 如:窗口的编辑框的"输入变量"为数据库变量、某图形的动画的 表达式用到了数据库变 量、某窗口的 JS 脚本用了"数据库.xxx._连接变量("xxx 变量")"等,都表明画面端连 接了数据库变量。

画面端调用数据库端组件原理:当画面端调用数据库端的方法时,实际上是通 过网络发送了一个数据包,数据库收到包后,执行对应的方法,然后将结果通过



网络包返回给画 面端。因此,实际上执行的机器还是数据库端的机器。所以,类似"画面.打印报表"将会 在画面所在的机器打印,而调用"数据库.打印报表" 将在数据库所在的机器打印,等等。

15.10.3 智能移动WEB 画面端

本系统是完全 B/S、C/S 架构, PC 上的浏览器客户端(包括 IE 浏览器)完全 具备单机版的画面浏览监控功能。虽然智能移动 WEB 画面端目前仅仅实现了画面端 部分监控浏览功能,但是本系统是按照真正 B/S 架构设计的,所以未来会逐步实现 其它画面监控功能。

已经实现的智能移动 Web 画面监控模块、图形会在用户手册相关章节专门说明,请参考手册相关模块、图形的说明。

15.10.4 画面端变量和数据库端变量

画面端变量供当前的画面监控端使用,而数据库端的变量供所有当前监控的画面端共享使用。只有正确地理解这个,才能正确地使用画面端变量和数据库端变量! 千万不能混 淆画面端变量和数据库端变量的应用,要在适当的场合选择合适的变量使用! 下面举个例子说明它们的应用。

某个工程有三个监控站,某个窗口需要通过一个变量来选择其中一个站进行监控,根据这个变量值来监控对应的站,当值为1、2、3时分别监控站1、站2、站3,窗口上有个编辑框连接该变量。当前有3个用户通过不同机器连接该数据库,并同时打开这个窗口监控这些站。如果该变量为数据库端变量,其中一个用户想监控站2,于是将连接该变量的编辑框输入2,由于数据库端变量是所有画面端共享使用,数据库会将该变量值改变通知到所有当前连接的画面端,那么此时将出现如下情况:另外两个监控的用户将被迫也变成监控站2!也就是说:只要其中一个用户监控某个站,其他用户也随之改变!如果该变量为画面端变量,由于画面端变量仅供当前画面端使用,它的改变不会影响其他画面端的用户,那么每个用户可以同时监控不同的站。

另外,画面端和数据库端都具有"_年"、"_月"、"_日"、"_时"、"_ 分"、"_秒"、"_运行时间"、"_日期时间"等系统变量。它们分别表示画 面端机器的时间和数据库端机器的时间。



15.10.5 尽量减少网络通信包

画面端每调用一次"数据库.XXX",就会发送一个网络请求包,如果网络通讯包太多 了,会导致系统效率下降,增加数据库服务器的负担,还可能引起数据同步的问题。因 此,应该尽量降低网络通信包。为了降低网络通信次数,组态开发时应该注意以下几点。

1) 中间变量的使用

在进行画面端的组态时中会用到一些临时、中间变量,应该采用画面端中间变量,因为如果采用数据库端中间变量,除了增加网络通信外,还可能引起数据库变量在各画面端 共享、同步的问题。

2) 画面端批量调用数据库端方法、组件

如果画面端某个功能涉及到批量调用数据库端方法、组件,而这些调用中没有 画面端 的方法、组件,则可以在数据库端增加一个 JavaScript,在这个 JavaScript 中调用 这些数据库端方法、组件,然后画面端调用这个 JavaScript 即 可。这样的话,本来需要 多次通讯完成的功能,一次通讯即可实现。例如,本来 是如下的一个画面端调用:

- 数据库.组件1.方法1();
- 数据库.组件 1. 方法 2();
- 数据库.组件2.方法1();
- 数据库.组件2.方法2();
- 数据库. 方法1();
- 数据库. 方法 2();

在数据库端增加一个"功能 1" JavaScript,在这个 JavaScript 实现上述功能。

- 组件 1. 方法 1();
- 组件 1. 方法 2();
- 组件 2. 方法 1();
- 组件 2. 方法 2();
- 方法1();
- 方法2();



然后在画面端原来的位置用"数据库.运行 JavaScript ("功能 1");"代替原来的语句。

3) 画面端批量读或写数据库端变量

有时画面必须读写大量不同数据库变量,那么可以采用如下的方法进行批量读 写。

取批量值: "画面.取批量变量值(变量列表)"返回一个数组,对应每个变量 值,参数"变量列表"是要获取值的变量集合,当中用空格分隔开。

例如:

"arr = 画面. 取批量变量值("v1. a06 v1. a07 v1. a08"). toArray();",

则返回三个变量"v1.a06 v1.a07 v1.a08"的值分别存在 arr[0]、arr[1]、arr[2]中。

设批量值: 与"取批量值"有较大不同。得在写变量值前增加"画面. 开始批量请求(2);",结束时增加"画面. 结束批量请求();"。(注:此处的"画面. 开始批量请求"与4)中的一样,只不过参数是2,表示设变量值)。例如:批量往数据库的 v1 驱动的"a0~ a99"100个变量写入值,这样的话只要一个往返包即可,否则得100个往返包。

<i>画面.开始批量请求(2);</i>
for(i=0; i<100; i++)
{
数据库. v1设变量值("a" + i,i);
}
画面.结束批量请求();

4) 画面端动态批量连接数据库端变量

在窗口打开/关闭时系统能自动批量连接/断开变量,但是有些工程要求动态地 连接变量。这种情况下,由于有画面端的参数、对象参与执行,所以无法用 2)中 所提到的方法来实现。得采用如下方法:在这些语句的前面增加"画面.开始批量 请求(1);",在这些语句结束时增加"画面.结束批量请求();"。(注:此处的

"画面.开始批量请求"与3)中的一样,只不过参数是1,表示连接/断开变量)。 例如在点击一个按钮时,执行如下语句:



画面. 开始批量请求(1); 编辑框 1. 输入变量 = "数据库. XXX1. 变量 1"; 编辑框 2. 输入变量 = "数据库. XXX1. 变量 2"; 开关按钮 1. 变量 = "数据库. XXX2. 变量 1"; 复选框 1. 变量 = "数据库. XXX2. 变量 2"; 数据库. XXX3._连接变量("变量 a", 触发函数 a); 替换所有变量("XXX1", "XXX2");

画面.结束批量请求();

.

<u>注:上述批量调用功能在窗口打开/关闭时不能使用,因为窗口打开时,系统</u> 默认启用了批量调用功能。

15.10.6 数据库事务的处理

关于数据库事务的处理,切记住一条原则: "会改变数据库端组件状态的、且 与画面端 无关"的事务一定要放在数据库端。下面的数据库端触发画面端事件和严 禁画面端定时执行数据库端事务就是该原则的应用演示。

15.10.7 数据库端触发画面端事件

画面端利用触发器或"数据库. XXX._连接变量("变量 X″,触发函数);"连接 变量,当该变量值改变时,会画面端会调用"触发动作"、"触发函数"执行。在 这些"触发动作"、"触发函数"函数中,一般而言,不能调用"会改变数据库端 组件状态的、且与画面端无关函数"(例如写值、打印、保存变量值、运行外部程 序、...等)。但是可以调用读值函数,因为读值函数需要返回给客户端。也可以调 用与画面端相关的函数,例如某函数有个参数是画面端变量或图形当前状态值。

试想一下以下这种情况:如果当前有3个用户连接到数据库,并且浏览同个窗口,当这个变量值改变时,同时触发了3个用户画面窗口的"触发函数",那么数据库端将执行3遍的"写值、打印、保存变量值、运行外部程序、...",这显然不符合常理。

那么如何解决这个问题呢?可以在数据库端增加一个"触发器",触发器的 触发条件可以是这个值是否改变,然后将上述的"会改变数据库端组件状态的、 且与画面端无关函 数"移到这个触发器的触发执行动作中执行。如果触发动作 有些函数无法执行(因为有些函数可能需要特殊参数),也可以增加一个 "JavaScript",然后将函数移到这个"JavaScript"中,然后触发动作调用



这个"JavaScript"。

15.10.8 严禁画面端定时执行数据库端事务

举个简单例子:需要对某个变量定时加1,如果利用数据库的定时器对该变量定时加1,则运行正常。

但是如果利用画面端的定时器对该变量定时加1,那会出现什么情况呢?假如5 个用户同时打开5个客户端,那么5个画面端都同时对该变量定时加1!而且为了从 数据库读写值,还生成了大量的网络通讯包。

15.10.9 严重警告

目前有很多用户在网络 web 版本中利用窗口定时器,不断从数据库端取设变 量值来对图形控件设置属性,而且不加批量命令,从而造成网络故障。

为什么会出现网络故障呢?窗口定时器 10 毫秒触发一次,假如定时器中有 10 条"数据库.XXX._取变量值"语句,而且不加批量命令,那么一秒钟内将产生 1000 次的发包请求并等待应答,这在互联网上就是致命的!!! 解决这个问题请参 考《窗口刷新、定时器》中描述的 4 个办法。



十六、多语言

16.1 概述

为了满足全球化业务的需要,系统提供了非常方便、易用、强大的 多语言 功能。利用多语言系统,您可以非常方便地在组态或者运行时自由地切换 编辑 语言。并在组态编辑时可以做到所见即所得。

16.2 设置

在工程画面浏览窗口中双击"画面配置"节点下的"多语言设置"子节 点,或者点击系统工具栏多语言图标,打开多语言设置对话框后即 可进行配 置。





用	中文	日文	韩文	英语	繁体中文	俄语	^
🐻 标签替换		1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1. 1				1	1
▶ 标签组			10				2
▶ 表格			10				
▶ 侧栏框			10				
🛃 超大图忧化			6				
🐻 窗口控件	1		0				
□◇♀说明							
11 标题	本窗口演示	20012	이 창에서	This windo	本窗口演示	Dieses F	e
표 √ 组合1		1	1	1			
田 → 开关按钮1			S				=
⊞ 🖋 按钮1			(i				
⊞ 🖋 按钮2			10 million (10 million)				
田 → 菜单1			10				
⊞ 📌 按钮3			10				
⊞ 🖋 按钮4			(i				
표 → 复选框1							
🖽 🔗 下拉框1			N				
田 📌 标签1							
▶ 弾出窗口							
🛃 定制图形							2
🐻 动画演示						1	1
🐻 方趋势图			N			1	1
📕 脚本动画			N				1
▶ 连接变量演示			14			1	V

对话框中可以增加、删除、编辑语言。

对话框中也可以对所有窗口多语言的相关文本进行统一编辑。

16.3 多语言切换

16.3.1 组态开发时切换

点击系统工具栏多语言下拉框,选择需要切换的语言,如下图所示:

0	中文 🔽
	中文 日文 韩文
(<u></u>	英语 繁体中文 俄语

16.3.2 运行时切换

运行时只要将"画面.系统变量._当前语言"变量设置为需要切换的语言即可。

假如要设置为"英语": 如果在动作中执行只要"画面.系统变量._当前语 言= "英语""即可。 如果在 JavaScript 中执行只要"画面.系统变量.设变量值 ("当前语言", " 英语");"即可。



十七、JvavaScript 程序

17.1 概述

17.1.1 简介

在某些逻辑复杂的动画、事务无法用普通的组态方法处理时,就可以使用 JavaScript 来完成了。

17.1.2 学习JavaScript

网络上 JavaScript 的相关介绍资料、学习教程等非常多,这里列出了一些站点,供大家学习参考。也可以通过强大的搜索来解决问题,例如:如果要连接mysql、access,在 google 或百度输入"js mysql"、"js access"即可。

http://baike.baidu.com/view/16168.htm

http://www.w3school.com.cn/js/index.asp (强烈推荐)

http://www.dreamdu.com/javascript/

http://www.iselong.com/online/ebooks/javascript/

http://www.bccn.net/Article/web/javascript/Index.html

17.1.3 文档对象模型(DOM)

注: DOM 似乎是为 HTML 制定的概念, 但是可以套用到本系统中。

在使用 JavaScript 开发工程之前,必须解释一下文档对象模型(DOM: Document Object Model)。因为 JavaScript 做为一种通用的、面向对 象的脚本 语言,只有跟本系统的 DOM 相结合,才能发挥更强大的作用。对于不同系统而 言, JavaScript 语法是一致的,主要的不同就是 DOM。

那么究竟什么是 DOM 呢?简单的说,就是系统提供了哪些对象,这些对象 有哪些方法、属性、事件等。一般来说,DOM 是用树形结构 来描述,因为对象 包括子对象、子对象又包括子子对象、....,而这 些子对象又具有它们特有



的方法、属性、事件。

大家在参考网络上的 JavaScript 资料时,一般都会提到 HTML DOM, 有的 甚至不提而默认就是 HTML DOM, 这是因为 JavaScript 目前最广泛的应用就是做 为 Web 的脚本语言。而 HTML DOM 是不能在本系统使用的,因为系统不提供这 些对象模型,系统有自己专门的对象模型。

即使在同个系统中,但是不同环境下,DOM 也是不同的。例如:

Web 系统中,HTTP 服务器端的对象与客户端浏览器的对象不同。同样道理, 本系统画面窗口客户端的对象模型与数据库服务器端的对象模型也不同。

总之,在学习完 JavaScript 语法之后,最需要了解的就是系统所提供的 DOM。

当在开发环境编辑 JavaScript 时,当前环境的 DOM 就列在了工具箱的"对象"树中。请参考下节的"脚本编辑器"。

当前 JavaScript 主要包括画面端(包括窗口、三维窗口、定制图形等) 脚本和数据库端脚本。

17.1.4 一些说明

经常会有用户问诸如"软件的 js 支持 setInterval 函数吗?"、"软件 的 js 支持 alert 函数吗?"、"软件 js 与 web 浏览器的 js 一样吗?"此类 的问题。实际上就是没有弄清楚 js 语法与 js 对象函数库的概念。任何系统的 js 语法都一样,以及 js 标准内置的一些对象函数库(如数学、字符串、日期 等)都基本一样,区别就是不同系统的对象函数库,也可以说是上文介绍的 DOM 不一样。setInterval、alert 都是 web 浏览器的对象函数,所以本系统 不支持,但是有相关的对象函数代替,如窗口和数据库都有定时器组件, alert 可以用"画面.消息"代替。

为什么采用 js 脚本做为内置语言因为涉及到核心技术,原先并无计划解释 使用 js 脚本的原因,但是咨询的用户太多了,所以这里统一做个说明。

注:此处所说 js 是系统的内置语言,而系统本身是采用 C/C++开发。

1. js 是网络 Web 上最流行的脚本语言,是浏览器的标准语言, js 是国际标



准。

2. 因为第一点的原因,所以 js 是所有 IT 巨头(微软、苹果、Google 等)都 支持的脚本。

3. 因为第一点的原因,所以 js 是所有浏览器(包括 IE、Chrome、firefox、Safari、Opera、...)都支持的脚本。

4. 因为第一点的原因,所以 js 是所有主流平台(Windows、Wince、Linux、 Android、iPhone/iPad、...)都支持的脚本。

5. 随着智能移动的爆炸式发展,多平台、跨平台已经大势所趋, js 毫无疑问 是最适合跨平台的脚本语言。所以,不出意外的话,大部分脚本语言都将边缘化。

6. 正是借助了 js 的强大,公司研发出全球首套 HTML5 纯 Web 的组态软件基本架构,支持 Android、iPhone/iPad、 WP7 等平台的 Web 浏览,支持 IE、 Chrome、firefox、Safari、Opera 等浏览器。仅凭这一点,就可让其它脚本成为 浮云。

7. 简单、易学、强大,否则也不可能成为国际标准、WEB 标准。网络上相关 资料不计其数。只要输入 js 和你需要的解决的问题,即可搜索到需要的资料。

8. 因为 js 的强大,应用广泛,所以学习了 js,不仅仅本系统可以使 用,许 多领域(例如 Web 等)都可以使用。

9. 方便与其它系统集成, js 可以调用 COM、ActiveX 组件。而且几乎所有的 Web 组件库,如 Google、百度的地图库都是采用 js 开发的,因此具有天然的集成 性。

17.2 全局脚本

不少使用过其它传统组态软件的用户询问怎么没有全局脚本,这里对全局脚 本做个解释。

随着智能移动、物联网的发展,WEB 应用也将越来越广泛。因此,本系统与 传统组态软件的一个重大区别是:将WEB 支持做为产品的一个核心功能,天生就 是 B/S、C/S 架构。所有组件、对象要么位于画面端(客户端),要么位于数据库 端(服务器),脚本也不例外。所以,严格的说,全局脚本包括画面端全局脚本



和数据库端全局脚本。考虑到画面端全局脚本的实用价值不大,目前画面端只有窗口(三维窗口)脚本。数据库的JavaScript实际上就是数据库端全局脚本,因为它可以在系统启动、画面动作、触发器、定时器、...等许多地方随时被调用。 在单机版本中,画面端和数据库端在同个程序、同台机器中,因此可以把数据库的JavaScript 看成是全局脚本。

17.3 JavaScript 编辑器

JavaScript 程序在 JavaScript 编辑器中编辑的。JavaScript 编辑器具有 语法/词法高亮度显示、颜色块显示、行号、块折叠、智能感知、复 制/粘贴、 UNDO/REDO 等功能特点。

17.3.1 智能感知

智能感知即自动提示,具体体现在以下两方面:

1. 当在对象名后输入".",会列出该对象的所有成员,如图:

🚰 粗细	~
🍠 弾起	
🚰 顶	
😭 高	
🚰 过渡	
😭 过渡色	
🚰 可见	_
😭 宽	
🚰 名称	
🔷 绕点旋转	Y

2. 在某驱动的"取变量值"、"设变量值"、"连接变量"、"断开变量"等 方法的"("后输入""",则自动列出该驱动所有变量,在"_切换开关量"之 后则列出所有开关量,如图:





3. 在某对象的函数后输入"(",会提示当前函数的参数说明,如图:

圆角矩形1.绕点旋转	F (
	绕点旋转(X,Y,角)
	完点加42(4,1,用

17.3.2 工具箱

为方便用户开发 JavaScript,当打开 JavaScript 编辑器之后,工具箱 会显示"对象"、"运算符"、"语句"三个面板,如图:

对象: 当前环境的所有对象及对象成员(包括属性、 方法)。当鼠标停留在对象方法上时,也会提示该方法及 方法参数的描述,如上图 当鼠标停留在画面的"消息" 方法时,提示了该方法的详细描述。双 击对象树节点, 也会将节点内容复制到 JavaScript 编辑器中,例如,双 击画面的"消息"方法,则在编辑器的光标处增加内容 "画面.消息"几个字。

运算符:列出了 JavaScript 常用的运算符供参考。

语句:列出了 JavaScript 常用的语句供参考。

17.3.3JavaScript 编辑器选项设置

可以通过显示选项对话框来修改 JavaScript 编辑器 显示选项。通过系 统"工具"菜单或者系统工具栏,弹出 "显示选项"对话框,在 "JavaScript 编辑器"一栏进 行配置。如下图所示:





→ 显示选项 (U) 示选项 □ 系统 系统主题 05	
示选项 □ 系统 ■ 系统	(
□ 系统 系统字题 06	
	1
/示范工版 01	fice 2007 🛛 💌
9 80	
背景网格直	线
网格间距 50	
网格颜色 📃	c0c0c0
鼠标摄取间距 5	
三 三维窗口(未完成)	
网格间距 30	
网格颜色	000000
- JavaScript编辑器	
字体 宋	体
字号 12	
关键字颜色 🗾	0000ff
字符串颜色 📰	a31515
	009000

17.4 窗口javascript

17.4.1 概述

每个窗口(包括三维窗口、定制图形)都可以有一个对应的 JavaScript 脚本 程序。运行环境下当打开窗口时,会自动调用对应的 JavaScript 执行。

17.4.1.1 编辑窗口JavaScript

有多种打开窗口 JavaScript 编辑的方法,前面有提到的点击工具栏的 图标、 右键当前窗口弹出菜单选择"编写 JavaScript"、点击属性事件 栏的脚本图标,以 及在工程浏览画面端的窗口项右键弹出菜单选择 "编写 JavaScript"等。这些命令 都有图标。

17.4.1.2 事件驱动

画面窗口 JavaScript 与数据库 JavaScript 最大区别是: 窗口 JavaScript



是事件驱动(即事件触发、事件响应)的,比如按钮点击触发事件,然后执行事件里的响应函数。请参考窗口(或三维窗口)中关于窗口图形事件脚本的描述。

17.4.2 对象模型

在 JavaScript 编辑器一节已经介绍过了,当前脚本的对象都列在了工具箱的对象树中。如下图,对象模型分成3 个根对象:数据库、画 面、窗口,所有 其它对象都是这3 个根对象的子孙对象。

工具箱	×
□ 对象	
 	

- 数据库:数据库节点下包括数据库端的驱动、模块子对象,以及数据库本身的方法。
- 画面: 画面节点下包括画面端的模块子对象, 以及画面本身的方法。
- 窗口:窗口节点下包括当前窗口的所有图形子对象,以及窗口本身的方法、属性、事件等。窗口对象是默认对象,即调用窗口的属性、方法、子对象时,不必带"窗口."的前缀。例如修改窗口的背景色属性,可以写成"窗口.背景色=0xFF0000;",也可以去掉"窗口."前缀直接写成"背景色=0xFF0000;";再比如,修改窗口图形对象"编辑框1"的文本,可以写成"窗口.编辑框1.文本="新文本"",也可以去掉"窗口."前缀直接写成"编辑框1.文本="新文本""。

17.4.3 连接变量

本节内容比较实用。(自 WiSCADA 2.0 起,本节所介绍的功能可以用窗口的 触发器代替,但是在智能移动 Web 上和字符串变量情况下仍然无法 代替。)

上文已经提过了窗口 JavaScript 是事件驱动的。而为了系统高效的运行效 率,系统内核也是采用值改变了通知画面端的(这个也是本系统区别传统低效组态 软件的特点之一,传统组态软件是无论变量值是否改变,都不断获取变量值,然 后刷新画面,导致运行效率低下。请参考系统的特点说明)。系统的这种值变通 知模式刚好与事件触发机制完美结合起来了,因为可以把变量值改变当成一个事 件,然后设置一个事件触发函数即可。系统的所有变量都是隶属某个驱动、模块



{

的,这些驱动、模块都有一个共同方法(函数): "_连接变量("变量名",触发函数)",变量名——你想要监控的变量名称;触发函数——该变量值改变后的响应处理函数,该函数有一个参数,由脚本系统自动传入,表示变量当前改变的新值。

下面举个简单例子:假设想让窗口的"矩形1"图形对象的高度随数据库端的"油罐"驱动的"液位"变量值按1/10比例变化,那么就可以如下实现:

function 油罐液位改变(值) //这里的参数"值"就是变量改变后的新值, 脚本系统 自动传入,这里就是油罐液位的新值

矩形 1. 高 = 值 / 10; //这里的"值"就是函数参数(即变量新值),可以直接使用 }

数据库.油罐._连接变量("液位",油罐液位改变);//该语句将数据库端的 "油罐"驱动的"液位"变量改变与"油罐液位改变"函数对应起来,一旦数据 库的"油罐"的"液位"变量值改变,窗口会自动调用"油罐液位改变"函数执 行。

由此可见,与图形事件编写类似,只不过图形事件是将触发函数做为源函数 赋值给图形对象的事件的,例如"按钮1.弹起=按钮1 弹起;"

(请参考窗口中关于窗口图形事件脚本的描述),而这里触发函数是做为一 个参数传递给"连接变量"函数的。如果矩形的高度由多个变量决定,例如由 "液位 1"、"液位 2"、"液位 3"决定,则此时值无效,因为系统无法判断 究竟由哪个液位传入的, 得用"_取变量值"获取。

function 油罐液位改变(值) //这里的参数"值"不能用了

矩形 1. 高 = (数据库. 油罐._取变量值("液位 1″) + 数据库. 油罐._取变量值("液位 2″) + 数据库. 油罐._取变量值("液位 3″)) / 10;

}

{

数据库.油罐._连接变量("液位1",油罐液位改变);

数据库.油罐._连接变量("液位2",油罐液位改变);

数据库.油罐._连接变量("液位3″,油罐液位改变);

实际上,多个变量组合输出某个值可以用数据库的定制变量的"表达式"来



组态,该方法更高效、快捷,详情请参考"系统常见技巧、优化、注意事项"章 节。

当不需要监控变量时,可以调用"断开变量("变量名",触发函数)"。除非 你想动态地监控变量(这种情况比较少见),否则不需要调用"断开变量",因 为系统在关闭窗口时,会自动断开当前所有连接的变量。

其他说明:

一些用户似乎使用连接变量和触发器上瘾了,碰到任何情况都使用这方法, 但是一些情况下没必要使用。如:通过一个编辑框设置某个参数。连接变量的 方法是,增加一个画面端变量,接着编辑框连接这个变量,然后 js 中通过连接 变量触发某事件,最后在该事件中设置参数。

实际上不需要变量即可完成,做法如下:在编辑框的"文本改变"事件函数中处理即可,如:设置 XXX (编辑框 1.文本);

17.5 数据库jvavascript

17.5.1 概述

与窗口 JavaScript 不同,数据库端的 JavaScript 不是伴随某个窗口的,它是 独立的,因此也没有对象事件驱动(触发、响应)它执行,它一般是被动等待系统 的触发器、定时器、或触发动作调用执行。它们都是通过调用数据库对象的"运行 JavaScript"方法命令来执行的。

17.5.1.1 对象模型

在 JavaScript 编辑器一节已经介绍过了,当前脚本的对象都列在了工具箱的对象树中。

与窗口 JavaScript 不同,数据库 JavaScript 只能调用数据库的对象,因此对象树的根对象只有"数据库",所有其它对象都是这个根对象的子孙对象。

数据库节点包括数据库端的驱动、模块子对象,以及数据库本身的方法。 由于数据库也是默认对象,所以调用数据库的方法、子对象时,不必带"数据



库."的前缀。例如"数据库.打印报表("日报表");"可以直接写成"打印报表("日报表");","数据库.油罐.设变量值("液位",10);"可以直接写成"油罐.设变量值("液位",10);"。



17.5.2 新建

17.5.2.1 从工程数据库浏览窗口创建

在工程数据库浏览窗口中右键点击"JavaScript"节点弹出菜单后,点击"新增 JavaScript",然后在弹出的对话框中输入JavaScript 即可。

数据/	库 · · · · · · · · · · · · · · · · · · ·			新增JavaScript		
	 新增JavaScript Ctrl+N 添加数据库文件 		Ctrl+N	新增JavaScript名称:		
	×	打开 重命名 清除	Ctrl+0	UavaScript] 确定 取消		

17.5.2.2 通过命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。



新建文件	
类别: ■ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●	文件类型: 报表 报表
JavaScript脚本程序。 「将新建的文件添加到当前工程」 新建文件名: 确定	取消

一般这种方法用来创建临时 JavaScript 文件。

17.5.3 分组

当数据库 JavaScript 较多时,就可以采用分组来管理了。

17.5.3.1 新增JavaScript 组

在工程数据库浏览 JavaScript 中右键点 击"JavaScript"节点弹出菜单后, 点击"增 加 JavaScript 组",将新增一个 JavaScript 组,输入 JavaScript 组名称。

17.5.3.2 JavaScript 组的JavaScript 文件



右键点击 JavaScript 组, 弹出菜单后, 点击"新增 JavaScript", 将在该 JavaScript 组下新增 JavaScript 文件。

也可以通过鼠标拖拽,在JavaScript 组之间拖动JavaScript 文件,使得 JavaScript 放置到其它JavaScript 组中。



17.5.4 全局变量与内存JS

17.5.4.1 概述

为了方便 JS 的运行,并提高 js 的运行效率,系统提供了全局变量和 内存 脚本的配置来配合 js 的运行。

17.5.4.2 配置

在工程数据库浏览窗口中双击"系统配置"节点下的"JS 全局变量"子 节 点,打开 JS 全局配置对话框后即可进行配置。对话框中可以增加/删除/编辑 JS 全局变量以及设置内存 JS。

数据库 ×	JavaScript配置
■ 系统配置 ● 系统启动 ● 新统启动 ● 事件配置 ● 安全区 ● 安全区 ● 安全区 ● 安全区 ● 安全区 ● 安全区 ● 田户管理 ● 日户登录 ● Yeb配置 ● JavaScript 根東 JavaScript ● 系统模块 ● ●	JS全局变量 确定 工会员变量有两个作用: 取消 1. 变量驻留内存,因此变量值保存内存中; 取消 2. 变量可以在所有数据库端脚本共享。 取消 js全局变量 注:为防止全局变量与局部 变量与局部 变量分类更好 需要增加前缀 "g",如: 增加的全局变量好 常要为 "abc",则使用时为 "g_abc"。 取消 ** ** *

17.5.4.3 全局变量

由于数据库端的 JavaScript 不是驻留内存的,它的生存有效期是临时的,即 在需要的时候运行一下,然后就退出内存。为了能使 JS 的某 些变量值能保存在 内存中,系统提供了数据库端 JavaScript 全局变量。此外,JS 全局变量还有另 外一个作用,就是它能被所有数据库端的 JS 共享使用。

注意:为防止全局变量与局部变量冲突,使用全局变量时需要增加前缀 "g",如:增加的全局变量为 "abc",则使用时为 "gabc"。

JS 编辑器的智能感知:当增加了全局变量后,在开发环境中编辑数据库 JS



时,输入字符后,会自动列出已经增加的 JS 全局变量,如下图。从图中也可以 看到上图中增加的"ado数据库"、"excel 程序"、"二维数组"前面自动 增加了"g_"。



初始化:对于一些变量,尤其是复杂的变量,例如 ado 数据库对象、 excel 程序对象、数组等,需要对它们进行初始化后,才能被使用。 可以新建一 个数据库端 JS,在该脚本中对它们进行初始化,然后在 系统启动时,调用该脚 本。

例如新建一个"系统初始化" js 脚本,在脚本中执行如下语句:

```
g_excel 程序=new ActiveXObject("Excel. Application"); ...
g_ado 数据库= new ActiveXObject("ADODB. Connection"); ...
g_二维数组= new Array();
for(i = 0; i < 10; i++)
g_二维数组[i] = new Array();
...
```

然后在系统启动时,调用这个"系统初始化"脚本。



TET/14/61/F, BULT9/4 1 BUA/11 E	
1设置通道(串口=COM3)	1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
	一册除
	编辑
	上移
	下移
杭初始化后执行的动作	4001
行JavaScript("系统初始化")	増加
	删除
	编辑
	上移

17.5.4.4 内存JS

一般情况下,当系统需要执行数据库端 js 时,得从磁盘的组态工程中读出 js 文件,然后调入内存中,接着进行脚本的编译解释执行工作,执行该脚本后, 又被清理出内存。如果该 js 是频繁被调用的,则上述过程又被重复的执行了一 遍,显然影响运行效率。所以对频繁被调用的 js 可以驻留在内存中,这样就可 以提高运行效率了。

设置内存 js: 弹出脚本配置框后, 在内存 js 栏目中, 在需要驻留内存的 js 的复选框前打勾即可。

17.5.5 脚本参数

17.5.5.1 简介

数据库端的 JavaScript 是由"数据库.运行 JavaScript()"函数调用的,该函数的声明为"数据库.运行 JavaScript(JavaScript 名,参数1,参数2,参数3,...)",可以输入无限个参数,而在此 JavaScript 中,可以通过"参数[0]、参数[1],参数[2],..."来引用这些参数。

那么这些参数有什么用途呢?比如在某个画面窗口中,JavaScript 必须用户根据当前用户的交互输入情况(如编辑框内容、下拉框选择 等)进行相应的处理,此



时就可以将这些内容作为输入参数供此 JavaScript 使用。有些用户可能认为把这些编辑框、下拉框的输入连 接到某个数据库变量,也可以解决上述问题,这个方法 在单机版中是可以的,但是在网络 Web 版本中,有可能多个用户同时操作某个窗口,这些变量就存在同步的问题。

17.5.5.2 例子

例如一个数据库端的"脚本 A"的内容为:

"数据库. 驱动A._设变量值("变量A",参数[0] + 参数[1]);" 即将"参数1"与"参数2"的和保存到"数据 库. 驱动A. 变量A"中。

而画面某个窗口做如下调用

"数据库. 运行 JavaScript ("脚本A",编辑框 1. 文本,编辑框 2. 文本);"

这里的"编辑框 1. 文本"就成了"脚本 A"的"参数[0]", "编辑框 2. 文本"就成了"脚本 A"的"参数[1]"。

执行 "数据库. 运行 JavaScript ("脚本 A",编辑框 1. 文本,编辑框 2. 文本);" 后,就将"编辑框 1. 文本"与"编辑框 2. 文本"的和就保存到"数据库. 驱动 A. 变量 A"中了。

17.5.5.3 其它说明

内存脚本不支持参数。

17.6 调用com 组件

17.6.1 概述

17.6.1.1 概述

COM component (COM 组件) 是微软公司为了计算机工业的软件生产更加符 合人类的行为方式开发的一种软件开发技术。在 COM 构架下,人们可以开发出 各种各样的功能专一的组件,然后将它们按照需要组合起来,构成复杂的应用 系统。



Microsoft 的许多技术,如ActiveX、DirectX、OLE 等都是基于 COM 而 建立起来的。并且 Microsoft 的开发人员也大量使用 COM 组件来定制他们的 应用程序(如: Office、数据库、IE 等)及操作系统。

在 Windows 和 Wince 平台的系统中,用户可以通过 JavaScript 调用 COM,从而充分利用 COM 组件的强大功能,大大扩展了系统的应用 以及与外界 系统的交互。

有关COM 更多内容,请参考相关资料。

17.6.1.2 OPC

工控中使用很广泛的 OPC 实际上就是基于 COM 技术的,所以也可以通过 JavaScript 调用 OPC 服务器读写变量值。但是这样会导致 IO 授权变量变少 (导致授权价格降低,呵呵),所以本系统限制了 JavaScript 调用 OPC。

17.6.1.3 创建COM

创建 COM 很简单, 只要 "obj = new ActiveXObject("comName");"即可,

obj 是返回的对象, comName 是对象组件名。例如:

"Excel = newActiveXObject("Excel.Application");"

将创建一个 Excel 应用程序对象。 创建 COM 组件时要注意一点,就是该组件 已经安装在 JavaScript 要运行的机器上了。例如 Excel 程序组件得在机器上安装 了 Excel 程序后,创建组件才能成功。在网络 WEB 版本中,数据库端的 JavaScript 是在数据库运行的机器上运行,而画面窗口的 JavaScript 是在用户浏 览的机器运行,所以必须在相应的机器安装组件。下面两节将介绍组态系统中最常 用的两个 COM 组件: Excel 组件和数据库组件。

17.6.2 调用Excel 组件

微软的 Office 所有软件都提供了 COM 组件,但是在组态系统 Excel 应用最为 常见,所以本节简单介绍一下如何通过 JavaScript 调用 Excel,其它 Office 组件 的使用类似。演示工程也提供了一个例子,演示了如何将一个表格的数据写入 EXCEL。 上文已经提到,机器必须安装了 Excel 后,才能调用 Excel 组件。假设机 器已经安装了 Excel,脚本中写入



"ExcelObj = newActiveXObject("Excel.Application");"

即可创建一个 Excel 应用程序对象。 创建对象后,就可以通过 Excel0bj 来调用对象的方法、函数,从而实现系统与 Excel 的交互。

要熟悉 Excel 的使用,最关键是熟悉 Excel 的对象模型。下图是模型对象 图:






尽管在 Excel 的对象模型中包括 100 多个对象, 但程序设计主要集中

在如下五个对象上: Application、Workbook、Worksheet、Range、Chart。



这并不是说用不着使用其他对象,只是说明这五个对象最为常用

- Application 对象代表 Excel。使用 Application 对象可以控制应用程序 级的设置、内置的 Excel 函数以及高级方法,例如 InputBox 方法。
- Workbook 对象是指 Excel 中的工作簿,即是说 Excel 文件。在 VBA 环境中, 不说打开一个文件,而称为打开一个工作簿;也不说保存一个文件,而称为保 存工作簿。学习 Excel 时最先了解到的其中一 点就是,Workbook 中包括 Worksheet。Worksheet 是 Workbook 中独立的页,数据就保存在 Worksheet 中。
- Worksheet 中包括单元格 (Cell)。你也许会认为不得不编写大量的代 码以对 单元格对象进行控制,但是,实际上没有单元格这样的对象,只有单元格属 性。你将对范围 (Range) 对象进行处理。范围对象是指一个或者多个单元 格。

有关 Excel 更详细的资料请参考相关文档。下面是微软 MSDN 文档有 关 Excel 部分描述。

http://msdn.microsoft.com/zh-cn/library/bb386107.aspx

17.6.3 调用SQL 数据库组件

在 JavaScript 中一般是通过 ADO (ActiveX Data Objects) 组件来连接数据库的。

JavaScript 调用 ADO 更详细的资料可以参考: <u>http://www.w3school.com.cn/ado/index.asp</u>

<u>注:</u>下述资料都是从网络上搜索到的,就是为了充实用户手册之用,实际上用 户可以直接到网络上搜索相关资料,如"js ms sql""js access""js mysql""js oracle""js tinysql"、...。

17.6.3.1JavaScript 调用ADO 的基本步骤

在 JavaScript 程序中应用 ADO 有几个基本步骤,下面我们来详细地介绍一下:

1. 创建 ADO 对象



var oConn = new ActiveXObject("ADODB. Connection");

2. 指定连接的数据源

所连接的数据源使用 Connection. ConnectionString(连接字符串)属性指定,对于不同的数据提供者和数据源,连接字符串中所指定的参数也会有所不同。下面会专门对连接字符串做详细介绍。

3. 打开连接

ADO 打开连接的主要途径是使用 Connection. Open 方法。也可以在同一个操作中调用 Recordset. Open 快捷方法,既打开连接又通过该连接执行命令。它们的语法格式如下:

connection. Open (ConnectionString, UserID, Password, OpenOptions) recordset. Open (Source, ActiveConnection, CursorType, LockType, Options)

例如下面的Connection 对象声明和Open 方法:

var oConn = newActiveXObject("ADODB. Connection"); oConn. Open("DSN=sampleDB;UID=;PWD=;");

这两种打开连接的方法包含多个参数,可以通过几种方式指定方法参数。例如, Recordset. Open 接受 ActiveConnection 参数,该参数可 以是文字字符串、表示该字 符串的变量或者是表示打开的连接的 Connection 对象。

如果省略方法参数,许多对象都将具有可提供参数的属性。例如,通过设置 Connection 对象的 ConnectionString 属性,可以为 Connection. Open 方法提供连 接字符串信息,然后省略 Connection. Open 方法的 ConnectionString 参数。例如 下面的语句:

```
var oConn = new ActiveXObject("ADODB. Connection");
oConn. ConnectionString = "DSN=sampleDB;UID=;PWD=;";
oConn. Open();
```

下面重点介绍一下连接字符串。

17.6.3.2 连接字符串

应用程序必须调用数据库驱动程序才能建立与数据库的双向通信。在建立数据



库连接时,既可以使用 OLE DB 驱动程序,也可以使用 ODBC 驱动程序,都可以通 过定义连接字符串(ConnectionString)来实现调用驱动程序的目的。应用程序 通过使用连接字符串来调用驱动程序,连接字符串由建立 到数据库的连接所需的 全部信息(或参数)组成,它可用类似于以下示例的最简单形式指定驱动程序和数 据库:

Driver={Microsoft Access Driver(*.mdb)}; DBQ=C:InetpubwwwrootmyDatabase.mdb

根据驱动程序的不同,连接字符串中的参数可能不同。例如,如果使用 Microsoft Access 作为数据库,那么可以使用 Microsoft.Jet.OLEDB.4.0 驱动程 序作为连接字符串:

Provider=Microsoft. Jet. OLEDB. 4. 0; Data Source= C:InetpubwwwrootmyDatabase.mdb"

连接字符串可以包含指定 OLE DB 驱动程序的 Provider 参数。如果省略此参数,默认情况下,连接字符串使用 ODBC 驱动程序。假定我们创建的 DSN 名为 sampleDB,那么可以使用下面的语句创建连接:

oConn.ConnectionString = "Provider=MSDASQL;DSN=sampleDB;UID=;PWD=;"

其中, MSDASQL 即是 Microsoft OLE DB Provider for ODBC 数据库连接驱动。然后分别是 DSN 名、用户名和密码。

由于 Microsoft OLE DB Provider for ODBC 数据库连接驱动是默认的, 因此 也可以省略声明数据连接提供者 MSDASQL,直接使用下面的代码:

oConn. ConnectionString = "DSN=sampleDB;UID=;PWD=;"

17.6.3.3 连接字符串参数

对于不同的数据库,连接字符串包含由分号分隔的以下参数:

Provider

指定数据库的 OLE DB 驱动程序。例如,下面分别是 Access、SQL Server 和 Oracle 数据库的常用 OLE DB 驱动程序的参数:

Provider=Microsoft. Jet. OLEDB. 4. 0;...



Provider=SQLOLEDB;... Provider=OraOLEDB;...

其它 OLE DB 驱动程序的参数值,必须参考驱动程序供应商提供的文档。如果没有 包含"Provider"参数,则将使用 ODBC 的默认 OLE DB 提供程序 (MSDASQL)。

- Driver: 指定在没有为数据库指定 OLE DB 提供程序时所使用的 ODBC 驱动程 序。
- Server 或者 Data Source: 指定承载 SQL Server 数据库的服务器名,如果您的 Web 应用程序和承载 SQL Server 数据库的服务器运行在不同的服务器上,就必须定义这个值。
- Database 或者 Initial Catalog: 为 SQL Server 数据库的名称。
- DBQ: 是指向基于文件的数据库(如在 Microsoft Access 中创建的数据库) 的路径。该路径是在承载数据库文件的服务器上的路径。
- UID: 指定用户名。
- PWD: 指定用户密码。
- DSN: 为数据源名称。根据您在服务器上定义 DSN 的方式,可以省略连接字符串的其它参数。例如,如果在创建 DSN 时定义了其它参数,则 DSN=Results可以是有效的连接字符串。

技巧与提示:对于其它类型的数据库,连接字符串可能不使用上面列出的参数, 或者可能对于这些参数有不同的名称或者用途。有关更多信息,请参见数据库供应商 提供的文档。



十八、表达式

18.1 概述

系统中许多地方都用到了表达式,表达式虽然简单,但是它在很多模块提供 了不可缺少的作用,大大方便了用户开发组态工程。

与 JavaScript 比较而言,表达式简单、短小、高效,一般与变量值一起完成 一个简单的数学运算,而 JavaScript 能完成逻辑非常复杂的工作。

18.1.1 语法

表达式的语法非常简单,与C语言类似。一般由数值、变量值、操作符、函数、挂号"()"构成。如:

"(5 + 数据库.水厂.水流量)*sin(数据 库. 定制变量.角度)"。

18.1.2 表达式对话框

为了方便用户开发,表达式所有的函数和操作符都列在表达式对话框中,开 发环境中所有表达式相关功能,都有一个"表达式..."按钮, 点击该按钮就会 弹出这个对话框。





18.2 函数

表达式目前提供了一些常用的数学函数,未来还会陆续增加其它函数。

与大部分语言类似,函数参数用"()"挂号起来,用","分隔开各个参数.

当前已经支持的函数如下:

sin(angle): 正弦函数, 参数 angle 为弧度, 返回 angle 正弦值。 cos(angle):余弦函数,参数 angle 为弧度,返回 angle 余弦值。 tan (angle): 正切函数,参数 angle 为弧度,返回 angle 正切值。 asin(value):反正弦函数,返回value的反正弦弧度值。 acos(value): 反余弦函数, 返回 value 的反余弦弧度值。 atan(value):反正切函数,返回 value 的反正切弧度值。 sinh(angle):双曲正弦函数,参数 angle 为弧度,返回 angle 双曲正弦值。 cosh (angle):双曲余弦函数,参数 angle 为弧度,返回 angle 双曲余弦值。 tanh(angle):双曲正切函数,参数 angle 为弧度,返回 angle 双曲正切值。 asinh(value):双曲反正弦函数,返回 value 的双曲反正弦弧度值。 acosh(value):双曲反余弦函数,返回 value 的双曲反余弦弧度值。 atanh(value):双曲反正切函数,返回 value 的双曲反正切弧度值。 log2(n): 基(底)数为2 的 n 的对数。 log10(n): 基(底)数为10 的 n 的对数。 log(n): 基(底)数为10 的n 的对数。 1n(n): 基(底)数为e 的n 的对数。 exp(x): e 的x 次方。 sqrt(x): 平方根函数,返回 x 的平方根值。 sign(x): 符号函数, 如果 x<0 返回-1, 如果 x>0 返回 1, 否则返回 0。 rint(x): 取整函数,按照四舍五入取 x 的整数值。 abs(x):绝对值函数,返回x的绝对值。 min(x1, x2, x3,): 最小值函数,可以多个参数,返回所有参数的最小值。 max(x1, x2, x3,):最大值函数,可以多个参数,返回所有参数的最大值。 sum(x1, x2, x3,): 总和函数,可以多个参数,返回所有参数的总和。 avg(x1, x2, x3,): 平均值函数,可以多个参数,返回所有参数的平均值。 if(x1, x2, x3):条件函数,如果第一个参数 x1 值为真,那么函数返回 第二个参数 x2 值, 否则返回第三个参数 x3 值。

18.3 操作符

表达式目前提供了一些常见的操作符,这些操作符都具有一定的优先级,优 先级越大表示越优先执行,可以用"()"提高优先级。表达式提供了如下操作



符: (注:所有逻辑操作,如果结果为真,则返回1,否则为0。)

and: 逻辑与, a and b, 只有当 a 和 b 都为真时,结果才为真。优先级为 1。 or: 逻辑或, a or b, 只要 a 和 b 有一个为真时,结果就为真。优先级为 1。

xor: 逻辑异或, a xor b, 只有 a 与 b 不同时, 即 a 与 b 其中一个为 真, 另外一个为 假时, 结果才为真。优先级为 1。

<=: 小于或相等比较。优先级为2。

>=: 大于或相等比较。优先级为 2。

!=: 不相等比较。优先级为 2。

- ==:相等比较。优先级为2。
- <: 小于比较。优先级为2。
- >: 大于比较。优先级为2。
- +: 相加。优先级为3。
- -:相减。优先级为3。
- *: 相乘。优先级为4。
- /:相除。优先级为4。
- %: 求模。优先级为4。
- î: xîy, 表示 x 的 y 次方。优先级为5。



十九、报表

19.1 概述

报表是反应生产过程中的数据、状态等,并对数据进行记录的一种重要形 式。报表既能反映系统实时的生产情况,也能对长期的生产 过程进行统计分 析。监控系统通过报表来完成重要信息的报告。

系统提供了类似 Excel 表格状的报表组态。

19.2 创建报表

19.2.1 从工程数据库浏览窗口中创建

在工程数据库浏览窗口中右键点击"报表"节点弹出菜单后,点击"新增报 表",在弹出新增报表对话框中输入报表名称。



新增报表	X
新增报表名称:	
演示报表	
确定 取消	

19.2.2 通过命令创建

点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。



新建文件 类别:	文件类型:	X
 ■ 画面 ● 系統模块 ● 数据库 ● 系統模块 ● 系統模块 ● 通 記设备通(● 通 用设 ● 通 PLC 	言 :番	
报表。 又将新建的文件添加3	副当前工程	
新建文件名: 演示报:	長	

图 19-1

新建后,系统默认生成一个6行8列的表格。下面就可以进行报表设计了。

	A	В	С	D	E	F	G	н
1				l				
2			a) (a)		× × ×			
3								
4								
5								
6								

图 19-2

19.3 报表分组

当画面报表较多时,就可以采用分组来管理了。 新增报表组 在工程画面浏 览报表中右键点击"报表"节点弹出菜单后,点击"增加 报表组",将新增一个 报表组,输入报表组名称。





报表组的报表文件 右键点击报表组,弹出菜单后,点击"新增报表",将在 该报表组下 新增报表文件。 也可以通过鼠标拖拽,在报表组之间拖动报表文 件,使得报表放置 到其它报表组中。

19.4 报表设计

19.4.1 报表工具栏

报表工具栏和报表右键菜单能完成报表单元格的合并/拆分、行/列的增加/删除等功能。如下图所示:



19.4.2 设置报表属性

当点击报表空白处时,在属性栏将显示报表属性。目前主要属性有条件数据源、上下左右的空白边距(单位是毫米)和报表的描述。如图:



	条件数据源	name=条件源
	描述	
Ξ	页边距	
	左边距	20.000000
	右边距	20.000000
	上边距	30.000000
	下边距	30.000000
Ξ	文件属性	
	文件类型	报表
	本工程文件	是
	隶属	数据库
3	件教权通	

条件数据源主要供数据库报表使用,详情请参考"数据库报表"一节。

19.4.3 设置单元格属性

点中单元格,在属性栏就可以配置其属性了。如下图所示:

属	性栏	φx
	基本属性	
	对象	C4
	变量	
	文本	
	文本色	000000
	背景色	ffffff
	左	32.000000
	顶	24.000000
	宽	16.000000
	高	8.000000
	橫向对齐	居中
	纵向对齐	居中
	字体	
	字体	宋体
	字号	4.000000
	粗体	否
	下划线	否
	斜体	否
	删除线	否

可以设置文本、颜色、字体等属性。



19.5 单元格内容

单元格的"内容"属性是单元格最重要的属性,报表根据单元格的内 容进 行相应的数据处理及显示,实时报表、历史报表、日报表、周报表、月报表、数 据库报表等主要都是通过它来实现的。 在属性栏点击单元格的"内容"属性, 弹出如下对话框:

报表格子内容配置	报表格子内容配置 ※
内容类型: 统计计算 文本 文本 实时变量 自动历史时间 类型 (最大(历史时间表 系件源字段 表达式 序号	内容类型: 历史变量 ▼ 文本 文本 文本 文本 1 . hist2 1 . hist2 50 万史时间 50 50 50 50 50 50 50 50 50 50
确定 取消	<u>确定</u> 取消

上面的右图显示了单元格9种不同类型,除了文本类型外,点击上图红圈部分的按钮,会显示出对应的类型配置对话框。下面分别介绍这几种类型的单元格。

19.5.1 文本

文本类型的单元格直接显示所输入的单元格文本。

19.5.2 实时变量

实时变量类型的单元格直接显示所连接的实时变量的实时值。点击 上图红 圈部分的按钮,会显示出变量选择对话框供选择。

19.5.3 历史时间、历史变量

历史时间与历史变量的单元格结合一起使用可生成历史报表。报表根据历史 时间查询对应历史变量所有值,并自动逐行列出,因此历史变量单元格具有多行 (多值)的属性。



历史时间包括手动历史时间和自动历史时间。手动历史时间需要用户手工输 入查询时间。自动历史时间会根据用户预先定义好的时间差,然后与当前时间 进行计算,自动计算出要查询的时间。

有关历史时间、历史变量的详细介绍请参考历史报表一节。

19.5.4 变量历史值

可根据时间变量(或时间表达式)指定历史变量某个时间点的历史值。先选择要查询的历史变量,然后设置时间变量/时间表达式。时间表达式为某个画面时间变量与整数(或者变量、表达式)之和/差,如:画面.定制变量.早班时间+ 画面.定制变量.间隔+5。在网络WEB版本中,表达式里面的变量最好是画面端 变量,否则可能会出问题。

19.5.5 SQL 字段/条件源字段

SQL 字段/条件源字段的单元格主要供数据库报表使用。报表会根据 数据 源的查询值,自动逐行列出该字段所有值,因此 SQL 字段/条件 源字段单元格 具有多行(多值)的属性。

有关 SQL 字段/条件源字段的详细介绍请参考数据库报表一节。

19.5.6 统计计算

统计计算主要用来对"历史变量"、"SQL 字段"这两种具有多值(多行) 显示的单元格的所有值(即所有行)的自动计算,然后以计算后的 值填充该单元 格。下图是统计计算配置对话框。



统计计算	统计计算 🛛 🔀
计算类型: 个数 ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓ ✓	计算类型: 极差 ✓ 目标格子: ○
目标格子表示要进行统计计算的格子,格式为:第一个字符为格子的列标号(A~Z),后面紧跟行标号(数字),如:C8。	目标格于表示3策小值 子,格式为:3最大值 标号(A [·] Z), <u>极差</u> 字),如:C8。
确定 取消	确定 取消 图 19-5

目标格子:要进行统计计算的目标单元格。

从图中可知, 计算类型共6种, 分别是:

- **总和**:对要统计的单元格的所有值(所有行)求总和。
- **个数**:统计目标格子的所有值(所有行)的个数。
- **平均值**:统计目标格子的所有值(所有行)的平均值。
- **最大值**:取目标格子的所有值(所有行)的最大值。
- **最小值**:取目标格子的所有值(所有行)的最小值。
- **极差**:取目标格子的所有值(所有行)的最大值与最小值之差。

19.5.7 表达式

报表表达式除了具备普通表达式(有关表达式的使用请参考表达式相关章 节)功能外,还可以引用报表其它单元格的值。引用其它单元格的格式为: "格 *F.XX*"。其中 XX 为列行号,与统计计算中的单元格号相同,即第一个字符为单 元格的列标号(A[~]Z),后面紧跟着 行标号(数字),如"C8"。

例如: 格子. A6 + 格子. D3 + 数据库. IO. Tag1,表示将单元格 A6、单元 格 D3 与数据库 变量 IO. Tag1 相加。

表达式和统计计算单元格都涉及到对其它单元格计算的功能,他们主要区别 是,统计计算是针对"历史变量"、"SQL 字段"这两种具有多值(多行)显示 的单元格的所有值(即所有行)的计算,而表达式是针对单值(单行)类型的单 元格值的计算。



19.5.8 序号

序号主要配合"历史变量"、"SQL 字段"这两种具有多值(多行)显示 的单元格来使用,表示当前历史值(或记录)的序号。

19.5.9 定制多行值

配合"历史变量"、"SQL字段"这两种具有多值(多行)显示的单元格来使用,同时结合"取单元格值"、"设单元格值"等函数与"数据读取完毕"事件,通过 js 来实现自定义复杂的数值。

详情请参考"JS 定制数值"一节。

19.6 历史报表

历史报表主要通过内容属性为"历史时间(包括自动历史时间、手动 历史时间)"、"历史变量"的单元格完成。有关单元格内容的介绍请参考"单元格内容"一节。

创建历史报表得先在指定的单元格设置历史时间,然后设置对应的历史变量。

19.6.1 手动历史时间

手动历史时间需要用户手工输入查询时间,报表才会查询并列出对应历史变 量的数据。下图为手动历史时间单元格的时间配置对话框。

动历史日期时间面	35 (
时间标识	
标识名称:	SINTO .
如果不为空,则全	在报表浏览器的历史时
间中供于动输入,	同 尤 常 注 接 时 间 受 重 。
时间范围	
起始时间变量:	
终止时间变量:	
时间间隔	
0 小时 0	分钟 10 秒
确定	

手动设置时间可以通过两种方式实现,时间标识和指定时间变量。



 时间标识:如果设置了时间标识,则在运行时,点击报表浏览器工具栏的 "历史时间"按钮,会显示出所有的时间标识,双击这些标识,可以 设置历史时间范围。如图:

			-
上一页下一页	打印	合适 🗸	历史时间

历史时间		时间范围	×
时间标识 早班时间 中班时间 晚班时间	时间范围 未设置 未设置 未设置	起始日期时间 2011-11-16 ↓ 11:26:58 ↓ 终止日期时间 2011-11-17 ↓ 11:26:58 ↓	
	· · · · · · · · · · · · · · · · · · ·	-6	<u></u>

时间变量:如果设置了时间变量,则报表会按照时间变量指定的时间范围

- 查询。 ● **时间间隔,**加里指完的时间小千历中峦量孚集时间 则按昭孚集时间抽取
- 时间间隔:如果指定的时间小于历史变量采集时间,则按照采集时间抽取
 历史数据,否则按照指定的时间抽取数据。

19.6.2 自动历史时间

自动历史时间会根据用户预先定义好的时间差,然后与当前时间进行计算, 自动计算出要查询的时间。下图是自动历史时间单元格的 配置。包括根据日、 周、月计算三种类型。

自动历史日期时间配置	自动历史日期时间配置	自动历史日期时间配置
时间标识 标识名称: 日报表 如果不为空,则会在报表浏览器工具栏的 "历史时间"中提示手动输入时间。	时间标识 标识名称: 周报表 如果不为空,则会在报表浏览器工具栏的 "历史时间"中提示手动输入时间。	时间标识 标识名称: 月报表 如果不为空,则会在报表浏览器工具栏的 "历史时间"中提示手动输入时间。
交型 ●根据日计算 ○根据周计算 ○根据月计算	类型 ○根据日计算 ⊙根据周计算 ○根据月计算	类型 ○根据日计算 ○根据周计算 ④ <u>親据月计算</u>
 注:下面的起始/终止日期以报表打印或浏览的日期 为基准,表示该基准日期的前几天/周/月,0表示当 天/本周/本月。 起始日期时间 经止日期时间 日期:前 日期:前 	注:下面的起始/終止日期以报表打印或浏览的日期 力基准,表示该基准日期的前几天/周/月,0表示当 天/本周/本月。 起始日期时间 日期:前 1 周 星期→ ▼ 星期→ ▼ 星期→ ▼	注:下面的起始/终止日期以报表打印或浏览的日期 力基准,表示该基准日期的前几天/周/月,0表示当 天/本周/本月。 长山日期时间 终止日期时间 日期:前 1 月 日期:前 0 月 1 日 1 日
时间: 8:00:00) 时间: 7:59:题) 时间间隔 0 小时 0 分钟 10 秒	时间: 8:00:00 📚 时间: 7:59:59 📚 时间间隔 0 小时 0 分钟 10 秒	时间: 8:00:00) 时间: 7:59:59) 时间间隔 0 小时 0 分钟 10 秒
确定取消	确定取消	确定取消

图 19-7



从上图可以看出,这三种类型的起始、终止时间配置相同,而起始、终止日 期配置有点小区别,但是它们都是以当前日期为基准,向前追溯日期。下面分别介 绍一下。

- **根据日计算**: 起始和终止日期得输入前几日的日期。
- **根据周计算**: 起始和终止日期得输入前几周和具体周几的日期。
- 根据月计算: 起始和终止日期得输入前几月和具体哪日的日期。
- 时间间隔:如果指定的时间小于历史变量采集时间,则按照采集时间抽取
 历史数据,否则按照指定的时间抽取数据。
- **时间标识**:与手动历史时间相同,请参考手动历史时间介绍。

19.6.3 历史变量

报表根据历史时间查询对应历史变量所有值,并自动逐行列出,因此历史变量 单元格具有多行(多值)的属性。

历史变量设置在同行历史时间单元格右边。

19.6.4 历史时间与历史变量的对应

当报表逐行分析单元格时,如果发现了历史变量单元格,则会在该行往前 搜索同行的单元格是否有历史时间单元格,如果发现了则以该时间为查询时 间,如果没有发现,则认为错误。所以历史时间必须位于历史变量的同行的前 列。

由于一个历史时间可能对应多个历史变量,所以要求这个历史时间对应的 所有历史变量的采集周期必须相同,否则会出现各个历史变量值数(行数)不 一致的情况,这样报表就无法对应时间进行显示了。

19.7 数据库报表

数据库报表得先配置数据源,然后通过内容属性为"SQL 字段"的单元格 完成。有关单元格内容属性的介绍请参考"单元格内容"一节。



233 / 273

19.7.1 数据源

数据源包括"SQL 访问"模块和"条件数据源"。

"SQL 访问"请参考"数据库访问(SQL)"章节。

条件数据源是在"SQL 访问"基础上增加"WHERE"查询条件。如下图, 点击 报表属性栏的"条件数据源"进行配置。"系统数据源"即"SQL 访 问"。SQL 访问只能是"SELECT"模式,不是"表"模式。"条件"即"WHERE"查询条件, 系统会将"条件"增加到"SELECT"语句后面进行查 询,也可以用报表浏览器的 "修改 SQL 查询条件"来运行时动态修改。

居性栏 回 基本属性	×	报表条件数据调	1		
金件数据源 描述	name=条件源	增加	删除	确定 取消	ŧ.
日 東边距 左边距	20.000000	名称	系统数据源	条件(不必输入WHERE)	
右边距	20.000000 30.000000	条件源1	条件访问	是省支持 = True	
下辺距 日 文件属性	30.000000				
文件类型本工程文件	形表 是				
隶属	数据库				
条件教报题 在系统数据源基	础上增加条件。				

图 19-8

19.7.2 SQL 字段

将单元格设置内容属性为"SQL 字段",则弹出如下配置对话框,选择对应的 "SQL 访问"模块和字段即可。

数据库SQL字段	设选择		
SQL访问模块:	SQL访问	~	确定
SQL字段列表:	图形名 是否支持 创建时间 属性个数 备注		取消



19.7.3 条件源字段

将单元格设置内容属性为"条件源字段",则弹出如下配置对话框,选择对应 的"条件数据源"模块和字段即可。

条件数据源字	段选择	
条件数据源:	条件源1 🔽	确定
字段列表:	图形名 是否支持 创建时间 属性个数 备注	取消

19.7.4 显示说明

类型为"SQL 字段"/"条件源字段"的单元格是多值(多行)类型,报表 会对该类型进行多行显示。

19.8 报表显示

报表可以用画面窗口的"报表浏览器"控件显示查看。

19.8.1 报表浏览器

报表浏览器在画面窗口中显示。通过如下步骤创建:

打开或者新建一个画面窗口,点击"工具箱"的"高级图形"中的"报表浏览器",在窗口中创建。接着在属性栏中的报表属性选择一个报表,如下图所示:



00 • •	< 报表演示. t = d * 窗口控件. t wd 事件. t wd 报警. t wd 🗾 🔻	× 工具箱	ů,
 □ 弾出窗口 □ 高級窗口組 □ IO通信演示 ■ SQL数据库 	上一页 下一页 打印 合适 🗸	▲ U Windows控件 田 基本图形 □ 高级图形	
 根表演示 报警 板签替换 一次签替换 二次括替 (1) 一次 括表1 		141 报表浏览器 公 实时趋势图 公 历史趋势图 公 实时XY曲线 公 次时XY曲线 公 历史xY曲线	>
⑦ 属性 目 动画 ダ 事件 重画 否 ○ 位置 ご 165.000000 页 25.000000 宽 435.000000		 【聖 振客表格 ○ 事件表格 ① 配方表格 ④ 回图 ② 定制图形 	
高 545.000000 报表 报表 报表 诸选择一个数据库报表。		④ 管道	
54.74			

图 19-9

19.9 报表打印

19.9.1 从报表浏览器中打印

运行时,点击报表浏览器工具栏的"打印"按钮。

19.9.2 通过命令打印

调用画面的"打印报表"方法,将在画面端的打印机上打印指定的报表;调用数据库的"打印报表"方法,将在数据库端的打印机上打印指定的报表。

<u>该功能目前还在开发中。</u>

19.10 JS 定制数值

对于复杂的报表内容,也可以通过 JS 来实现。

当报表浏览器读入报表数据准备显示之前,会触发"数据读取完毕"事件。可 以在该事件中编写相关代码实现复杂的单元格数值。



报表浏览器的"取单元格值"和"设单元格值"函数:第一个参数表示单元格,第二个参数为第几个值(该参数仅对多行/值类型单元格有效)。

对于多行的单元格,得配置为"定制多行值"类型,才能通过"设单元格值" 函数进行设置。而对于单行的单元格,则不需要这个配置。



二十、数据库访问

20.1 概述

系统提供了一个方便、无需编程就可以访问关系数据库的模块——SQL 访问。为了保证访问关系数据库的高效性,系统都采取了特定数据库最高效的数据库引擎,例如: ACCESS 采用专门的 ACCESS DAO 引擎,MS SQL Server 采用 OLEDB,mySq1 采用 mySq1 提供的 C 接口,以及未来 Oracle 也将采用其本身提供的最底层的 C 接口 (采用 C 接口也有满足跨平台特性)。

此外,在Windows 平台上,也可以采用 JavaScript 调用 ADO 数据库 组件来 访问数据库,详情请参考 JavaScript 相关章节。

20.2 新建

在工程数据库浏览窗口中右键点击"系统模块"节点下的"SQL 访问"子节 点弹出菜单后,点击"新增 SQL 访问"即可新建 SQL 数据库访问。





20.3 配置

新建"Sq1 访问"后,出现如下界面:

ジ 数据源 ≰ ジ → 『	」 「刷新字」 記置Sq1数	⊛ 据源 居 类型	1
图形名		文本	
是否支	持	布尔	
创建时间		日期	
属性个数		短整数	
备注		备注	

点击"数据源"工具,弹出如下数据源配置对话框。

数据源类型:	Microsoft SQL Server 数据	库		
	Microsoft Access 数据库文	t件		
IsSQL服务器:	Microsoft SQL Server 数据	「库		
登录数据库				
用户名:	82			
密码:	*****			
数据库:	testdb 🔽	刷新库		
●数据表:	testtb 🔽 🤇	刷新表		
)SQL语句:	select 图形名, 创建时间 fi	rom 图库		
	注:此外不要输入"WHERE"	条件查询		

根据你的数据源情况选择对应数据源。目前已支持Access、MS Sq1 Server、 MySQL 等数据库。

"表"和"SQL 语句(即 SELECT)"只能选择其中之一。如果选择"SQL 语句",不能输入"WHERE",主要考虑系统有可能运行时动态改变查询条件。



1. 在数据源配置对话框中的"数据源类型"选择"Microsoft Access 数据 库文件"。

2. 在 "Access 文件"选择要访问的 Access 数据库文件。

3. 必要的情况下输入 Access 的用户名和密码。

4. 最后选择需要访问的数据库表。可以通过"刷新表"按钮显示当前数据库的所有数据库表。如下图所示。

以佑覙尖型;	Microsoft Access	数据库文件
access文件:	test	选择
登录数据库		
用户名		
密码:		
●数据表:	testtb	▶ 刷新表
●数据表:)SQL语句:	testtb select 图形名,创建	✓ 刷新表 时间 from 图库



20.5 访问SQL Server

1. 在数据源配置对话框中的"数据源类型"选择"Microsoft SQL Server 数据 库"。

2. 在 "SQL Server" 输入要访问的 SQL Server 数据库服务器。可以通过"搜索..." 按钮搜索 SQL Server 数据库服务器。

3. 输入登录的用户名和密码。

4. 选择需要访问的数据库。可以通过"刷新库"按钮显示当前数据库服务器的所有数据库。

5. 最后选择需要访问的数据库表。可以通过"刷新表"按钮显示当前数据库 的所有数据库表。

如下图所示。

数据源配置					
数据源类型:	Microsoft SQL Server 🖇	数据库 🖌 🖌 🖌			
MsSQL服务器:	WJHXP\SQLEXPRESS	✔ 搜索			
登录数据库					
用户名:	Sa				
密码:	*****				
数据库:	testdb 💌	刷新库			
 数据表: 	testtb 🗸	刷新表			
○ SQL语句:	select 图形名, 创建时间	from 图库			
	注:此处不要输入"WHER	E"条件查询			
C	确定 取消				



20.6 访问MySQL

1. 在数据源配置对话框中的"数据源类型"选择"MySQL数据库"。

2. 在"MySQL"输入要访问的 MySql 数据库服务器 I 或名称,如果端口不是 默认端口(3306),可以在后面加上":端口"。如果为空,则表示本机服务器。 如: MySQL 数据库服务器 IP 为 192. 168. 0. 66,端口为 3330,则输入 "192. 168. 0. 66:3330"。

3. 输入登录的用户名和密码。

4. 选择需要访问的数据库。可以通过"刷新库"按钮显示当前数据库服务器的所有数据库。

5. 最后选择需要访问的数据库表。可以通过"刷新表"按钮显示当前数据库的所有数据库表。

如下图所示。

数据源类型:	MySQL 数据库	~
MySQL服务器:	192. 168. 1. 33	说明
登录数据库		
用户名:	admin	
密码:	****	
数据库:	testdb 💌 🚺	刷新库
⊙数据表:	testtb 💌	刷新表
○ SQL语句:	select 图形名, 创建时间 fr	on 图库
	注:此处不要输入"WHERE"多	条件查询



20.7 操作使用

20.7.1 数据库报表

SQL访问可供数据库报表浏览使用,详情请参考"报表"相关章节。

20.7.2 数据库表格

SQL 访问可供画面窗口控件"数据库表格"浏览使用。

在某画面窗口上创建"数据库表格"后,在属性栏的"数据源"属性中 选择 要显示的"SQL 访问"模块,然后新建表格列,点中表格列,在表 格列"字段" 属性选择对应的"SQL 访问"模块中的字段变量。

如下图,分别为组态开发和运行时的"数据库表格"界面。

	4 (48.70				
·····································		创建时间	属性个数	备注	

ţt]	的 新研) (新聞	余) (修改	2 刷新			
	图形名	支持	创建时间	属性个数	备注	^
1	按钮	1	2007-05-08 00:00:00	12	windows 窗口控件-按钮	
2	标签	1	2007-03-06 00:00:00	5	windows窗口标签控件	
3	列表框	0	2010-08-06 00:00:00	15	windows 窗口列表框…	
4	编辑框	1	2007-05-06 00:00:00	10	windows 窗口编辑控…	
5	下拉框	0	2010-05-08 00:00:00	13	windows 窗口控件-下…	1
6	矩形	0	2010-05-08 10:00:00	15	基本图形-椭圆形	
7	椭圆形	0	2010-05-08 10:00:00	15	基本图形-椭圆形	
8	三角形	0	2010-05-08 10:00:00	15	基本图形-▲三角形	_
9	古绯	1	2007-05-08 00:00:00	6	其木图形 - 古线	Y



20.7.3 属性说明

刷新间隔: 该模块驱动还具备"刷新间隔"属性。当该属性值不为0时,系统会以该间隔不断刷新获取 SQL 字段值,实现数据库与变量之间的实时数据传递。由于此法效率低,所以若非特殊情况,否则尽量不要使用。

启动连接:表示系统启动时,是否连接数据库。默认不自动连接,可以用 "连接"函数命令手动连接。一般情况下,如果 SQL 访问模块 如果供报表或表 格使用,则不必连接;而其它方面的应用,则可以 "启动连接"。数据库一旦 打开连接,将保持着这个连接,如果不再 使用该模块,建议用"断开"函数命 令断开连接。

20.7.4 其它使用

用户也可以通过连接"SQL 访问"模块的变量和调用该模块的函数命令来实现对数据库的操作。

该模块的变量与 SQL 字段一一对应,变量值为该字段当前 SQL 游标(指针)所 指的字段值,游标可以通过模块的"第一条"、"最后一 条"、"上条"、"下 条"、"移动"等函数命令来移动。

此外,还可以通过调用"增加记录"、"删除记录"、"写记录"等函数命令 实现对 SQL 数据库的增删改。



二十一、其他辅助模块

21.1 概述

为了帮助开发人员更方便、快速地完成一个组态工程,系统提供了一些辅助功能模块。 这些模块包括: 画面端的定制变量、动作集,数据库端的定制 变量、 定时器、触发器、系统启动等。

21.2 定制变量

定制变量实际上跟一些组态软件的内部变量类似。画面端和数据库端都有定制 变量,分别供画面和数据库使用。画面变量一般作为画面模块临时使用,所以只 有数据库的变量才能具有历史记录、报警、 事件操作等功能。

21.2.1 新建

在工程数据库(或画面)浏览窗口中右键点击"系统模块"节点下的"定制 变量"子节点弹出菜单后,点击"新增定制变量"。





21.2.2 编辑

新建定制变量后,就可以进行编辑操作了,如下图所示。

<mark>き</mark> 増加変量	批量増加	∎ 删除变量	増加組	省 更名	↓ 🙀 🗍	♪ ↓					
变量		类型	初始值	表达式	读写方式	报警	历史记录	操作记录	保存值	数据转换	描述
演示变量		INT	0	-	读写	-	-				

编辑操作包括:增加变量、批量增加、删除变量、增加组、组更名、 导入、导出等。这些操作与 IO 设备通信中的驱动操作类似,具体请参考 IO 设备通信相关章节。

变量大部分相关属性也与 I0 设备通信中的变量类似,具体请参考 I0 设备 通信相关章节。这里主要讲述一下定制变量特有的"表达式"属性。如果设置 了某定制变量表达式属性,那么该变量的值就是该表达式 的值,将随表达式值 的变化而变化。点击表达式变量表达式格子"..." 按钮,将弹出表达式编辑对 话框,可以在该对话框中编辑表达式(关于如何使用表达式,请参考表达式相 关章节),如下图所示。

表达式编辑	
表达式	选择变量 表达式
v2.bi1*6	
	<u>></u>
	确定 取消

21.3 动作集

动作,系统将赋值操作和执行某个对象的方法(函数或者命令)称为动作。动作集,顾名思义,就是一系列动作的集合。如果某系列动作经常执行,就可以将它们设置为动作集,方便被调用。



21.3.1 新建

在画面浏览窗口中右键点击"系统模块"节点下的"动作集"子节点弹 出 菜单后,点击"新增动作集"。



21.3.2 编辑

新建动作集后,就可以进行编辑操作了,如下图所示。



操作包括:增加、删除、编辑、上移、下移等。



21.4 定时器

定时器就是按照指定时间定时执行一系列动作。

21.4.1 新建

在工程数据库浏览窗口中右键点击"系统模块"节点下的"定时器"子节点弹 出菜单后,点击"新增定时器"。



21.4.2 编辑

新建定时器后,就可以进行编辑操作了,如下图所示。

v3. a18 ≥ 10	△ 选择变量
	表达式
く	×
CHIMINHOWNER AND A	
定制变量1. 演示变量=v1. 演示变量	1 增加
定制变量1. 演示变量=v1. 演示变量 打印报表 ("报表1")	增加 删除
定制变量1. 演示变量=v1. 演示变量 打印报表 ("报表1")	增加 删除 编辑
定制变量1. 演示变量=v1. 演示变量 打印报表 ("报表1")	增加 删除 编辑 上移



定时时长:要定时执行的时间间隔,单位毫秒。

启用条件表达式: 当条件表达式为"真(即1或true)"时,才会执行动作列表。如果为空,则默认为真,启用定时器。

动作列表:每次定时要执行的动作列表。

21.5 触发器

触发器,是根据用户设置的表达式,当表达式值改变时,根据表达式值产 生触发事件,从而调用触发动作。 画面端和数据库端都有触发器,分别触发画 面端组件或数据库端组件的动作。

21.5.1 新建

在工程画面/数据库浏览窗口中右键点击"系统模块"节点下的"触发器" 子 节点弹出菜单后,点击"新增触发器"。



21.5.2 编辑

新建触发器后,就可以进行编辑操作了,如下图所示。



4. 演示变量 > 10000	选择变 表达式
	<u> </u>
	3
■ ■发执行的动作列表:	
★发执行的动作列表: 定制变量1、演示变量=v1、演示变量	増加
■ ■发执行的动作列表: 定制变量1. 演示变量=v1. 演示变量	
■ 定制变量1. 演示变量=v1. 演示变量	· · · · · · · · · · · · · · · · · · ·
触发执行的动作列表: 定制变量1. 演示变量=ᢦ1. 演示变量	

21.5.3 触发方式

包括以下几种触发方式:

值变触发:表达式值改变时触发。

值变与启动触发:表达式值改变和系统启动时触发。

0 触发: 表达式条件为假(等于0)时触发。

非0 触发: 表达式条件为真(不等于0)时触发。

字符串触发:所指定的字符串变量值改变时触发。由于触发器是根据表达式 触发的,但是目前表达式不支持字符串类型,所以专门做了字符串变量触发。字 符串触发只能连接字符串变量,不能连接表达式。

上图的例子中为"0 触发",而表达式为"v1.演示变量>10000",那么在"v1.演示变量"的值小于等于10000 时,表达式的值为0,这时就 触发了下面的动作列表。

21.6 动态库

"动态库"模块实现了对指定动态链接库(DLL)中的输出函数的调用。



21.6.1 新建

在工程数据库浏览窗口中右键点击"系统模块"节点下的"动态库"子 节点弹出菜单后,点击"新增动态库"。



21.6.2 编辑

新建动态库后,就可以进行编辑操作了,如下图所示。

DWORD CreateDCW (const WCHAR* lpszDr roid DeleteDC (DWORD hdc)	viver, const	而PFEAS	Device, cons	x	pszO
JNURD CreateDCM (const MCHAR* 1957D) void DeleteDC (DWORD hdc)	river, const	ULUNU TD23	tuevice, cons	ST NUMAR* 1	.pszu
1 T:T. (DWODD LJ :_1 :_1)					
nt Linelo (DWORD hdc, int x, int y) nt MoveToEx (DWORD hdc, int x, int y,	void*)				

先选择要引用的动态链接库,然后点击"增加函数"按钮,弹出配置 函数对话框,然后输入要引用的函数名、参数、返回类型等。


· 数相关配置	
参数列表: 增加参数	
参数类型	参数名(可为空)
DWORD (unsigned 1	hde
int (BOOL)	x
int (BOOL)	у

21.6.3 指针类型说明

指针类型分成两种。一种是输入类型,即DLL 的函数不会修改指针 所指内容的值,也可以认为指针内容是只读的。与之相反,另一种 便是输出类型,即DLL 的函数会修改指针所指内容的值,也可以认 为指针内容是可写的。

第一种类型主要有两个用途。

一是输入字符串,包括

const char(LPCSTR, 即笮字符串) const WCHAR(LPCWSTR, 即宽字符串, UNICODE 字符串)。

二是输入大量数据供 DLL 函数处理,可认为是数组数据,包括 const BYTE、 const WORD、const long、const DWORD等。因此这种类型在 JavaScript 中使用 时,需要传入一个 JavaScript 数组类型。

例如,一个函数的声明是:

void Foo(const char, const WCHAR, const int)*,则JavaScript 中可以如下调用:



```
datas = new Array(3);
datas[0] = 23;
datas[1] = 876;
datas[2] = 92;
xxx.Foo("abc", "组态软件", datas);
```

由于这种类型仅仅输入数据,所以在本系统中,配置该参数时都必须在前面增加 "const",即使 DLL 本身的函数没有 "const",但是实际是输入类型参数,在 配置该参数时,也必须在前面增加 "const"!其主要作用就是告诉本系统,当前 指针是输入类型,方便本系统进行处理。这点是下面 "严重警告"小节的一个例 外。例如,一个函数声明原来是 "void Foo(char)",即参数是一个字符串,但是 是输入类型,函数不修改字符串的值,这时在配置参数时就得声明为

"void Foo(const char)" .

第二种类型,即输出类型。js 通过一个初始化好的数组,即可将输出的值存入数组中。

例如: 一个函数的声明是: *void Foo(int* pOutDatas, int outCount)*,则 JavaScript 中可以如下调用:

datas = new Array(3); //假设返回的数组值个数为3 个
for(i=0; i<3; i++)
datas[i] = 0;
xxx.Foo(datas, 3); //则将返回的值存入 datas 中</pre>

21.6.4 严重警告

必须严格遵循下列2点,否则可能造成系统崩溃!

 1. 所有参数类型、顺序、个数必须准确无误!但是这里有个例外,即输入指针 类型,请参考"指针类型说明"小节。

2. 必须是 stdcal1(WINAPI)——Windows 默认调用方式。

21.7 系统启动

在工程数据库浏览窗口中双击"系统配置"节点下的"系统启动"子节点,





动短花指系统后动驱动、历史库约配置可以指定启动这些模块之前 统初始化前执行的动作	、 实时库、报告寺。 或者之后的动作列表。
mb1设置通道(串口=COM3)	增加
	册除
	编辑
	上移
	下移
统初始化后执行的动作	
统初始化后执行的动作 运行JavaScript("系统初始化")	增加
统初始化后执行的动作 运行JavaScript("系统初始化")	増加 一 删除
统初始化后执行的动作 运行JavaScript("系统初始化")	増加 一 一 一 開除 編辑
统初始化后执行的动作 运行JavaScript("系统初始化")	増加 一 一 一 一 一 開除 編辑 上移

根据实际情况可以在系统初始化各个模块之前或者之后执行相关动作。

21.8 模块开发包

如果用户需要特殊的辅助模块,用户可以通过 SDK 开发包来自行增加,详情 请参考《SDK 开发包》章节。



二十二、很重要!常用的优化、技巧、问题

22.1 模块平板拖动

在 WINCE (HMI) 版本中,可以通过窗口的拖动事件来实现类似平板电脑的 手指拖动功能,从而实现窗口切换。在窗口的脚本中,增加如下代码即可。

22.2 HMI 注意事项

}

22.2.1 旋转、倾斜

由于控件的任意角度旋转、倾斜会涉及到大量矩阵运算,在嵌入式 等 CPU 主频很低的设备上,大量复杂的图形的旋转、倾斜会严重影响画面刷新效率,因此 在这些设备上尽量少用这些功能。对于部分图形可以用"方向"、"方位"等属 性来代替。

<u>附:目前支持控件/图形的任意角度旋转、倾斜的功能的组态软件比较少,尤</u> 其是嵌入式 HMI 的组态软件更是寥寥无几。



22.2.2 图像

为了提高图像的显示速度,系统会将图像驻留内存,这样会占用一 定的内存空间,所以在嵌入式等内存较少的设备上,尺寸大的图片尽量不要太多。尤其是 GIF 动画图像可能内含多幅图片,会占用更大的内存。

曾经有个HMI项目,屏的内存为64兆,用户为了显示十几个80×20的按钮上的图片,每个按钮都采用了2000×500尺寸的图片,这样的一个图片得占用4兆内存,光十几个图片就占用了50兆内存,结果在HMI上项目无法启动。

22.3 窗口刷新、定时器

22.3.1 窗口定时器

也许是受到演示工程的脚本例子的影响,不少用户的画面刷新喜欢用窗口的 定时器事件。

窗口定时器包括两种:默认窗口定时器事件、定时器辅助组件。默认窗口定时器事件的时间间隔是 10 毫秒(嵌入式上是 100 毫秒),而定时器辅助组件可以自定义间隔。

这里要强调的是:尽量少用定时器事件,因为它对系统运行性能有 较大的影响。尤其是网络 web 版本在互联网运行时,在一个定时器 中不断从数据库取 值、设置,这几乎是致命的。

由于变量值改变而产生的画面刷新,应采用如下方法代替:

1.图形控件的变量相关属性,如编辑框的输入输出变量、标签的"数 据显示"属性等。

2. 图形控件的动画。

3. 窗口"触发器"组件(由于触发器基于表达式的,所以对于字符串变量, 效果不好,建议还是采用4点中的连接变量)。

4. 驱动的"_连接变量"方法(详情请参考"JavaScript 程序"->"窗口 JavaScript"->"连接变量"一节)! 新版本可以用第3 点中更加强大、灵 活、 易用的窗口"触发器"组件代替。



不是变量值产生的画面刷新,没必要使用触发器,一些用户为了使用触发器, 特意增加变量来实现触发器,这根本就是画蛇添足。

22.3.2 严重警告

目前有很多用户在网络 web 版本中利用窗口定时器,不断从数据库端取设变 量值来对图形控件设置属性,而且不加批量命令(参见网络 web 开发要点),从而 造成网络故障。

为什么会出现网络故障呢?窗口定时器 10 毫秒触发一次,假如定时器中有 10 条"*数据库.XXX._取变量值*"语句,而且不加批量命令,那么一秒钟内将产生 1000 次的发包请求并等待应答,这在互联网上就是致命的!!! 有关网络请求包说 明请参考《网络 web 开发要点》。

解决这个问题请采用上述的4个办法。

22.3.3 数据库端定时器

与窗口定时器类似,尽量少用定时器,而应该尽可能采用触发器。

22.3.4 其他说明

一些用户似乎使用连接变量和触发器上瘾了,碰到任何情况都使用这方法, 但是一些情况下没必要使用。

如:通过一个编辑框设置某个参数。连接变量的方法是,增加一个画面端变量,接着编辑框连接这个变量,然后 js 中通过连接变量触发某事件,最后在该事件中设置参数。

实际上不需要变量即可完成,做法如下:在编辑框的"文本改变"事件函数 中处理即可,如:设置 XXX (编辑框 1. 文本);

22.3.5 补充说明

一些用户用"系统变量. 秒"、甚至"系统变量. 运行时间",这还不如用定时器呢。



究竟用什么,其实是遵循这个原则: 在都能解决问题的情况下,哪个方法执行 周期更长,执行频率少,那么 CPU 就用的少, CPU 就可以腾出来做其他事情,就使 用哪种方法。这不仅仅是本系统的原则, 也是全世界任何系统遵循的原则!

上述的建议就是基于这个原则的,因为在大部分情况下定时器明显执行频率 多,所以效率低。但是某个触发器变量 300 毫秒变化一次,而定时器 2 秒执行一 次也能解决问题,这种情况下,就是触发器执行频率多、效率低,而定时器效率 高,所以应该选择定时器。

22.4 自定义图库

自定义图库是基于 SVG 格式的。所以得先准备一些 SVG 的图片。

22.4.1 SVG 图库

最好的方法是从网络上搜索,同时下面推荐几个。

1. 一个工业图库 http://www.reichard.com/。下载解压后,直接运行

SYMFAC_CN. EXE 程序。选择图形后导出为 SVG 格式。

2. Visio 是微软 Office 强大的图表制作工具,并且内置了大量的图库,这些图库可以保存为 SVG 格式。

3. 下面是网络上一些 SVG 图库地址。

https://www.flaticon.com/

http://www.iconsvg.com/

http://www.iconfont.cn/

22.4.2 复制到图库中

到软件的安装目录的 gallery 目录下,通过文件夹分组,将图形复制到对应文件夹中。



22.5 网络Web 组态开发说明及要点

22.5.1 概述

区别与传统组态软件,本组态是一款真正意义、完全的 B/S、C/S 架构的组态 软件。 有关本组态的特点与配置使用,请参考手册的《网络与 Web》相关章节, 本节主要介绍网络 Web 组态开发(尤其是 JavaScript 开发)的主要概念、原理、 要点。掌握了这些要点,网络 Web 功能就能发挥更强大的作用。

22.5.2 画面端与数据库端

系统的网络 Web 分成画面端和数据库端,即画面监控客户端(包括 Web 浏览器 监 控)和数据库运行服务器端。画面端的组件包括窗口、三维窗口、画面模块、 画面变量等,数据库端的组件包括驱动、数据库模块、数据库变量等。

有关画面和数据库具体模块可以通过工程浏览查看,如下图所示。详情可参考 "工程浏览"章节。

色面	×
 ■ 画面配置 ■ 圖 窗口 ■ 圖 三維窗口 ■ 圖 定制图形 ■ 圖 系统模块 	
國面 圖数据库	

一台机器只能运行一个数据库程序,其上只能运行一个组态工程,而可以有多 个机器 启动画面监控端连接到这个数据库端进行监控。

画面端可以调用数据库端组件,而数据库端不能调用画面端组件。试想一下, 当多个 画面端连接到数据库时,数据库应该调用哪个画面的组件呢?显然是做不 到的。所以,在 进行画面的组态开发时,会看到"画面.XXX"和"数据 库.XXX",就是表示当前是调用画面 或数据库的组件/变量。而进行数据库的组态 开发时,只能看到"XXX",这是因为,数据 库端只能调用数据库组件/变量,所 以"XXX"默认就是数据库端的组件/变量。

虽然数据库端不能调用画面端组件,但是画面端当前的监控窗口连接了数据库



端的某 些变量时,当这些变量的值改变了,就会通知画面端,从而触发画面执行 某些操作。例 如:窗口的编辑框的"输入变量"为数据库变量、某图形的动画的 表达式用到了数据库变 量、某窗口的 JS 脚本用了"数据库.xxx._连接变量("xxx 变量")"等,都表明画面端连 接了数据库变量。

画面端调用数据库端组件原理: 当画面端调用数据库端的方法时,实际上是通 过网络 发送了一个数据包,数据库收到包后,执行对应的方法,然后将结果通过 网络包返回给画 面端。因此,实际上执行的机器还是数据库端的机器。所以,类 似"画面.打印报表"将会 在画面所在的机器打印,而调用"数据库.打印报表" 将在数据库所在的机器打印,等等。

22.5.3 智能移动WEB 画面端

本系统是完全 B/S、C/S 架构, PC 上的浏览器客户端(包括 IE 浏览器)完全 具备单机版的画面浏览监控功能。虽然智能移动 WEB 画面端目前仅仅实现了画面端 部分监控浏览功能,但是本系统是按照真正 B/S 架构设计的,所以未来会逐步实现 其它画面监控功能。

已经实现的智能移动 Web 画面监控模块、图形会在用户手册相关章节专门说明,请参考手册相关模块、图形的说明。

22.5.4 画面端变量和数据库端变量

画面端变量供当前的画面监控端使用,而数据库端的变量供所有当前监控的画 面端共 享使用。只有正确地理解这个,才能正确地使用画面端变量和数据库端变 量!千万不能混淆画面端变量和数据库端变量的应用,要在适当的场合选择合适的 变量使用!下面举个例子说明它们的应用。

某个工程有三个监控站,某个窗口需要通过一个变量来选择其中一个站进行监控,根据这个变量值来监控对应的站,当值为1、2、3时分别监控站1、站2、站3,窗口上有个编辑框连接该变量。当前有3个用户通过不同机器连接该数据库,并同时打开这个窗口监控这些站。如果该变量为数据库端变量,其中一个用户想监控站2,于是将连接该变量的编辑框输入2,由于数据库端变量是所有画面端共享使用,数据库会将该变量值改变通知到所有当前连接的画面端,那么此时将出现如下情况:另外两个监控的用户将被迫也变成监控站2!也就是说:只要其中一个用户监控某个站,其他用户也随之改变!如果该变量为画面端变量,由于画面端变量



仅供当前画面端使用,它的改变不会影响其他画面端的 用户,那么每个用户可以 同时监控不同的站。

另外,画面端和数据库端都具有"_年"、"_月"、"_日"、"_时"、"_ 分"、"_秒"、"_运行时间"、"_日期时间"等系统变量。它们分别表示画面 端机器的时间和数据库端机器的时间。

22.5.5 尽量减少网络通信包

画面端每调用一次"数据库.XXX",就会发送一个网络请求包,如果网络通讯包太多了,会导致系统效率下降,增加数据库服务器的负担,还可能引起数据同步的问题。因此,应该尽量降低网络通信包。为了降低网络通信次数,组态开发时应该注意以下几点。

1) 中间变量的使用

在进行画面端的组态时中会用到一些临时、中间变量,应该采用画面端中间变量,因为如果采用数据库端中间变量,除了增加网络通信外,还可能引起数据库变量在各画面端 共享、同步的问题。

2) 画面端批量调用数据库端方法、组件

如果画面端某个功能涉及到批量调用数据库端方法、组件,而这些调用中没有 画面端的方法、组件,则可以在数据库端增加一个 JavaScript,在这个 JavaScript 中调用这些数据库端方法、组件,然后画面端调用这个 JavaScript 即 可。这样的话,本来需要多次通讯完成的功能,一次通讯即可实现。例如,本来是 如下的一个画面端调用:

> 数据库.组件1.方法1(); 数据库.组件1.方法2(); 数据库.组件2.方法1(); 数据库.组件2.方法2(); 数据库.方法1(); 数据库.方法2();

在数据库端增加一个"功能 1" JavaScript, 在这个 JavaScript 实现上述功能。

组件1.方法1();



组件1.方法2(); 组件2.方法1(); 组件2.方法2(); 方法1(); 方法2();

然后在画面端原来的位置用"*数据库.运行 JavaScript("功能1"*);"代替原来的语句。

3) 画面端批量读或写数据库端变量

有时画面必须读写大量不同数据库变量,那么可以采用如下的方法进行批 量读写。

取批量值: "画面.取批量变量值(变量列表)"返回一个数组,对应每个变量 值,参数"变量列表"是要获取值的变量集合,当中用空格分隔开。例如:

"arr = 画面. 取批量变量值("v1. a06 v1. a07 v1. a08"). toArray();",

则返回三个变量"v1.a06 v1.a07 v1.a08"的值分别存在 arr[0]、arr[1]、 arr[2]中。

设批量值: 与"取批量值"有较大不同。得在写变量值前增加"*画面. 开始批量请 求 (2);*",结束时增加*"画面. 结束批量请求();"*。(注:此处的*"画面. 开始批量请求"* 与 4)中的一样,只不过参数是 2,表示设变量值)。例如:批量往数据库的 v1 驱 动的"a0~ a99"100个变量写入值,这样的话只要一个往返包即可,否则得 100 个往返包。

> *画面.开始批量请求(2);* for(i=0; i<100; i++) { 数据库.v1._设变量值("a" + i, i); } *画面.结束批量请求();*

4) 画面端动态批量连接数据库端变量

在窗口打开/关闭时系统能自动批量连接/断开变量,但是有些工程要求动态地 连接 变量。这种情况下,由于有画面端的参数、对象参与执行,所以无法用 2)中 所提到的方法 来实现。得采用如下方法:在这些语句的前面增加"画面.开始批量



请求(1);",在这些语句结束时增加"画面.结束批量请求();"。(注:此处的 "画面.开始批量请求"与3)中的一样,只不过参数是1,表示连接/断开变量)。 例如在点击一个按钮时,执行如下语句:

> *画面. 开始批量请求(1);* 编辑框 1. 输入变量 = *"数据库. XXX1. 变量 1";* 编辑框 2. 输入变量 = *"数据库. XXX1. 变量 2";* 开关按钮 1. 变量 = *"数据库. XXX2. 变量 1";* 复选框 1. 变量 = *"数据库. XXX2. 变量 2";* 数据库. XXX3. _连接变量("变量 a", 触发函数 a); 替换所有变量("XXX1", "XXX2");

...... *画面. 结束批量请求()* ;

<u>注:上述批量调用功能在窗口打开/关闭时不能使用,因为窗口打开时,系统</u> 默认启用了批量调用功能。

22.5.6 数据库事务的处理

关于数据库事务的处理,切记住一条原则: "会改变数据库端组件状态的、且 与画面端无关"的事务一定要放在数据库端。下面的数据库端触发画面端事件和严 禁画面端定时执行数据库端事务就是该原则的应用演示。

22.5.7 数据库端触发画面端事件

画面端利用触发器或"数据库.XXX._连接变量("变量 X",触发函数);"连接 变量, 当该变量值改变时,会画面端会调用"触发动作"、"触发函数"执行。 在这些"触发动 作"、"触发函数"函数中,一般而言,不能调用"会改变数据 库端组件状态的、且与画面端无关函数"(例如写值、打印、保存变量值、运行外 部程序、...等)。但是可以调用读值函数,因为读值函数需要返回给客户端。也可 以调用与画面端相关的函数,例如某函数有个参数是画面端变量或图形当前状态 值。

试想一下以下这种情况:如果当前有3个用户连接到数据库,并且浏览同个窗口,当这个变量值改变时,同时触发了3个用户画面窗口的"触发函数",那么数据库端将执行3遍的"写值、打印、保存变量值、运行外部程序、...",这显然不符合常理。



那么如何解决这个问题呢?可以在数据库端增加一个"触发器",触发器的 触发条件 可以是这个值是否改变,然后将上述的"会改变数据库端组件状态 的、且与画面端无关函 数"移到这个触发器的触发执行动作中执行。如果触发 动作有些函数无法执行(因为有些 函数可能需要特殊参数),也可以增加一个 "JavaScript",然后将函数移到这个"JavaScript"中,然后触发动作调用 这个"JavaScript"。

22.5.8 严禁画面端定时执行数据库端事务

举个简单例子:需要对某个变量定时加1,如果利用数据库的定时器对该变量定时加1,则运行正常。

但是如果利用画面端的定时器对该变量定时加1,那会出现什么情况呢?假如5 个用户同时打开5个客户端,那么5个画面端都同时对该变量定时加1!而且为了从 数据库读写值,还生成了大量的网络通讯包。

22.5.9 严重警告

目前有很多用户在网络 web 版本中利用窗口定时器,不断从数据库端取设变 量值来对图形控件设置属性,而且不加批量命令,从而造成网络故障。

为什么会出现网络故障呢?窗口定时器 10 毫秒触发一次,假如定时器中有 10 条 "数据库.XXX._取变量值"语句,而且不加批量命令,那么一秒钟内将产生 1000 次的发包请求并等待应答,这在互联网上就是致命的!!!

解决这个问题请参考《窗口刷新、定时器》中描述的4个办法。

22.6 互联网云监控要点

越来越多的用户需通过互联网监控,本节就着重介绍一下互联网监控的一些 注意点。

22.6.1 互联网监控的方式

公有云 IP: 通过租借云上的公网 IP (如: 阿里云) 等, 一般上传运行工 程 后, 在客户端直接输入云 IP 即可开始监控。



Web:http://www.witlinc.cn E-mail:support@witlinc.cn 263 / 273

私有云 IP:本地拥有固定公网 IP 时,一般采用端口映射方法来实现

(包括路由器映射、利用 porttunnel 等端口映射软件等)。有关如何进行端 口映射相关资料较多,这里就不再详述了。

无公网 IP: 目前几乎所有组态软件都没有提供无公网 IP 情况下监控内网的 解决方案,但是可以借助"花生壳"等工具实现。

P2P 云:本系统采用独特的 P2P 穿透技术,可以不需要公网 IP 与其他辅助 工具情况下,实现外网穿透内网进行监控。详情请参考 P2P 云一节。

22.6.2 互联网监控要点——提高互联网的通讯效率

无论采用上述何种方式,它们都有共同点,即互联网的网速、稳定性远远低于局域网(局域网一般都是百兆,而互联网共享一兆都没有,考虑丢包等不稳定因素,局域网内通讯效率是互联网几百倍),

因此需要尽可能地提高互联网通讯的效率。 尽管本系统对工程文件、监控数据的传输做了大量的优化,但是了 解下面的设计要点将有助于进一步提高通 讯效率。 互联网监控体验:对于用户来说,影响互联网体验主要有 3 个响应 等 待时间,而响应等待时间主要是由请求应答包的个数决定。

A) 连接系统的响应时间。

B)窗口切换(窗口显示)的响应时间。 C)监控控制(即与数据库相关操作,主要是读写数据库变量的响应时间。

其中,A)连接系统时间由路由寻找时间、获取系统初始信息响应时间、初始 窗口显示时间组成,而路由与获取信息是必备的,已经无法优化,只剩窗口显示 时间,因此A)与B)实际上是同个问题。

本节的主要内容就是介绍如何提高本系统在互联网通讯(包括 P2P 云)的效率,即如何提高上述3个响应时间,也就是说如何降低请求应答包的个数。

可以从两大部分来提高通讯效率,数据传输与文件传输。

1. 数据传输:数据传输要解决的就是上述的C)监控控制响应时间。 主要在 组态开发时要注意如下2 个问题。

A)尽量避免在画面定时器中不断执行与数据库相关操作(比如读写数据库变量值)。



B)对于需要大量读写数据库变量、连接数据库变量的操作,务必加上"批量请求"相关 函数。

有关这 2 个问题的详细解决方法请参考《网络 WEB 组态开发说明及要点》 一节。

实践证明,只要注意这2个问题,数据传输对互联网监控体验影响几乎可 以忽略不计。因为经过优化后,几乎只要1个请求应答包。而且本系统是采用 变量值改变后才通知客户端的机制,因此数据传输的效率已经到达极致。

2. 文件传输:从目前本系统大量实际运行情况看,文件传输是影响用户 互联网监控体验最主要的因素,也就是上述的A)连接系统时间 与B)窗口切换 时间。在介绍要点之前,需要先介绍一下本系统的文件传输机制。

当客户端显示某个窗口(包括内部的媒体图像等文件)时,会先检测该 窗口是否需要下载。当窗口文件不一致,或者从来没有下载过,则会发送请 求下载。另外,系统是通过文件时间和大小来判断是否与服务器的文件一致。 下面的要点就是根据这个机制制定的。

A)局域网内先监控:有条件的情况下,可以在局域网内先监控,因为第一次监控时,会将工程文件自动下载到客户端,再次监控就不会下载了,目前影响监控效率的97%以上是工程文件引起的。尤其是某窗口如果使用了大量的图片,则此窗口第一次在互联网打开时间比较长,而如果在局域网先下载过,则不会出现此问题。

B)上传工程文件:当上传新的工程到服务器上时,尤其要注意,有些上 传工具会把工程文件(包括媒体图片等文件)时间修改为上传的时间,而实际 上这些文件没有修改过,但是客户端判断到这些文件时间与服务器的不一致, 则又会重新下载!建议采用 rar、zip 等 压缩工具把整个工程压缩打包上传, 然后解压,这样会保留实际的文件时间。附带说一句,WINCE 设备下载工程不 会有这个问题。

22.7 IO 设备通信采集说明

22.7.1 IO 采集包

当采集 I0 设备的数据时,组态软件会根据用户所配置的变量,将它们组合成



I0 采集包。为了提高 I0 通信效率,得尽量减少 I0 采集包。 因为包越少,那么每 个包被调度请求的频率就越高,这样采集效率就高了。一般 I0 通信协议都是规定采 集某个寄存器类型的地址段,所以组态软件会将连续的地址组合成地址段。所以在 相同数量的变 量个数情况下,如果寄存器类型越少、连续的地址段越多,则组合成 的包就越少了,这样采集效率就高了。

另外,组合包个数与设备通信能力相关,即设备每次通信允许的最大字节数。 当组合的包超过这个限制后,系统会自动新建一个包,然后再对剩余变量进行组 合。

注:系统会自动对变量的地址排序,然后组合包。所以即使组态时变量地址是 无序的,系统也会排序组合。排序与分组无关,即使不同组的变量,如果地址是连续的,也会组合在一起。

22.7.2 优先级

变量具有优先级(0~100),优先级越大,采集频率越高,0表示不采集 (手动调用驱动方法"_采集变量"时触发采集)。采集频率按优先级比例分配, 举例如下:变量1优先级为1,变量2优先级为10,变量3优先级为100,则当 总共采集次数为111次时,变量1采集了1次,变量2采集了10次,变量3 采集了100次。而实际上,变量组合成10采集包后,变量的采集频率就由这个 采集包的优先级决定了。而采集包优先级取自所有组合变量的最高优先级。上述 的例子中,如果变量地址是连续的,并且是同种寄存器,则三个变量有可能组合 成一个包,这种情况下,包的优先级为三个变量中的最高优先级(即100),三 个变量的优先级都变成100了。

22.7.3 诊断

未来计划用户点击 IO 驱动"诊断"按钮后,列出 IO 采集包的打包情况,包括:共打成的包个数、每个包由哪些变量组合成、包的优先级等。

22.7.4 优化采集

为了提高采集速度, I0 驱动提供了"优化采集"选项, 只有窗口的图形 有连接或者历史记录需要保存的变量, 系统才会采集。注:优化采集时, 脚本 中调用"_取变量值"的变量不会被采集, 而脚本中调用"_连接变量"的变量



会被采集。

22.8 变量位读写操作

可以有2种常见方法对变量位进行操作。

22.8.1 通过"位变量"辅助功能模块

1. 新建数据库"位变量"模块。

2. 点击"增加引用变量"按钮,增加引用变量。

3. 增加变量,并设置"位"属性列指定连接引用变量的某位,如果需要引用多 位,可以用"起始位[~]终止位"来指定,如: "3[~]6"表示引用 3[~]6 位。

<u>注:此法可以对引用变量的某位进行读写操作,但是需要计算授权点数,按</u> 实际变量的 2/3 数授权。

22.8.2 通过数据库定制变量的表达式

此法只能读取变量某位,但是不用计算授权点数。

1. 新建数据库"定制变量",并将它们的类型设置为"BOOL"(开关量)。

2. 在变量的"表达式"列中,分别输入"xxx 驱动.xxx 变量&1"、"xxx 驱动.xxx 变量&2"、"xxx 驱动.xxx 变量&4"、"xxx 驱动.xxx 变量&8"、"xxx 驱动.xxx 变量&16"、...等,分别表示读取"xxx 驱动.xxx 变量"的 第1、2、3、4、5、...位。注意上述表达式的"&"的前后得是""(空格),否则会被认为是连在一起的。

22.9 变量替换(窗口模板)

22.9.1 简介

替换变量包括组态设计时替换和运行时动态替换。组态时替换比较简单, 直接在窗口右键弹出替换对话框即可,这种方法还是需要建立多个窗口。下面



重点讲述运行时动态替换。 当多个窗口显示相同的图形控件,而仅仅连接的变量不同时,就可以使用运行时变量替换或相关功能,这样仅仅一个窗口就可以完成多个窗口需要完成的功能,大幅减少工程开发量。

22.9.2 用法

调用窗口的"替换所有变量"函数。如果需要网络WEB 监控,则需要调用"**一 面**. 开始批量请求(1)、**一**. 结束批量请求()",详情请参考《网络WEB 开发要点》一 节。

原型: 替换所有变量(旧文本, 新文本)。

函数说明:将窗口中所有图形控件的属性、事件、动画(脚本除外)所有 变量包含的"旧文本"替换成"新文本"。

例如:当前窗口中含有 "数据库. 驱动1. 变量A、数据库. 驱动1. 变量B、数据库. 驱动1. 变量C",当调用 "替换所有变量(". 驱动1. ", ". 驱动2.")"后,则当前连接变量变成 "数据库. 驱动2. 变量A、数据库. 驱动2. 变量B、数据库. 驱动2. 变量C",继续调用 "替换所有变量(". 驱动2.", ". 驱动3.")"后,则当前连接变量变成 "数据库. 驱动3. 变量A、数据库. 驱动3. 变量C"。

再如:当前窗口中含有 "数据库. 驱动. 站1 变量A、数据库. 驱动. 站1 变量B、数据 库. 驱动. 站1 变量C",当调用 "替换所有变量(". 站1", ". 站2")"后,则当前连接变量 变成 "数据库. 驱动. 站2 变量A、数据库. 驱动. 站2 变量B、数据库. 驱动. 站2 变量C",继 续调用 "替换所有变 量(". 站2", ". 站3")"后,则当前连接变量变成 "数据库. 驱动. 站 3 变量 A、数据库. 驱动. 站3 变量B、数据库. 驱动. 站 3 变量C"。

22.9.3 使用要点

1. 变量命名有一定的规律性,除一部分名称不一样外,其它部分相同。

2. 需要知道当前所连接的变量关键字,比如当前连接变量变成 "数据库. 驱动2. 变量A、数据库. 驱动 2. 变量B、数据库. 驱动 2. 变量C"后,调用"替换所有变量(". 驱动 1. ",". 驱动 3. ")",则没有效果,因为系统在所有图形控件中找不到含". 驱动 1." 的变量。所以一般需要用一个变量记住当前的关键字。如:

打开窗口时:



当前文本= ^{*}. 驱动 1. ^{*}; //先记住初始的关键字,与组态时所指定的变量对应,供下次替 换使用

.

替换为"驱动2": 替换所有变量(当前文本,".驱动2."); 当前文本=".驱动2."; //记住替换后的关键字,供下次替换使用 替换为"驱动3": 替换所有变量(当前文本,".驱动3."); 当前文本=".驱动3."; //记住替换后的关键字,供下次替换使用

3. 防止名称冲突。例如:当前窗口中含有 "数据库. A. 变量 A、数据 库. A. 变量
B、数据库. A. 变量 C",替换所有变量("A", "B"),则 "数据库. A. 变量 A" 被替换成
"数据库. B. 变量 B"。因此替换的关键字最好要带上特殊字符,如: "." "_"
等,替换所有变量(". A.", ". B."),就不会出 现上述问题。

22.9.4 其它说明

当窗口的变量名称不具有规律性时,则不能使用"替换所有变量"。此时可 以通过脚本直接赋值。例如:

编辑框1. 输入变量= "数据库. 驱动 2. 变量 A";

标签 1. 数据显示 = "模拟量文本=表达式(数据库. 驱动 A. 变量 1) /文本(5-<=5;10-<=10;30-<=30);前缀=aaa;后缀=bbb;*但是,这种方法目前无法对动画、事件动态设置,未来会考虑增加。

22.9.5 严重警告

不能在窗口打开事件函数中执行替换相关函数,如果需要在窗口打开时替换变量,直接放在函数之外即可,系统在读入 js 时自动会调 用函数之外的所有 js 语句。如下面的例子:

```
    替换所有变量("aaa.", "bbb."); //放在这里 0K!
    function 窗口_打开()
    {
        //替换所有变量("aaa.", "bbb."); //不能放在这里!请放在函数之外。
    }
```



22.10 定制图形(图形模板)

22.10.1 说明

当窗口中有大量相同的复杂图形时,即可采用定制图形开发,实现一次 开发多处引用,大幅提高组态效率。

22.10.2 与"组合"图形的区别

与"组合"图形相似,定制图形也是由很多图形组合而成。但是它们之间一个最重要、最直接的区别就是图形维护:当需要修改组合图形时,需要将所有窗口的所有组合图形修改一遍;而定制图形只要修改一次,所有引用到的定制图形会自动更新!

22.10.3 创建

1. 在工程画面浏览窗口中右键点击"定制图形"节点弹出菜单后,点击"新增 定制图形"。如图:



2. 也可以通过点击"文件"菜单的"新建文件"或者系统工具栏"新建文件"命令来创建。与创建窗口类似,请参考"新建窗口"一节。

22.10.4 增加定制属性、动画、事件

通过定制图形工具栏可以增加定制属性、动画事件。如图:





引用定制图形 在窗口工具箱的"高级控件"栏中,点击"定制图形",如图:



然后在定制图形的属性栏的"图形"属性中选择定制的图形。

22.11 SQL 关系数据库

系统有2 种方法与 SQL 关系数据库连接。

1. 系统内置辅助模块,包括"SQL 访问"(请参考"数据库访问(SQL)"章 节)、"实时 SQL"。

2. 通过 js 调用 COM 组件,请参考 "JavaScript 程序->调用 COM 组件-> 调用 SQL 数据库组件",此法实际上可以直接到网络上搜索,如: "js ms sql"、"js access"、"js mysql"、"js oracle"、"js tinysql"、…。 建议如果工程需要用到系统内置的窗口 SQL 数据库控件(如:数据库表格、数据库报表等) 情况下,则选择第一种方法,否则采用第二种方法。

也可以调用 "数据库. Sq1 查询(数据源,条件)"函数:该函数返回一个二维数组。下面是一个例子。

按钮1.弹起= function()



}

支持集 = 数组[1]. toArray(); 时间集 = 数组[2]. toArray(); 个数集 = 数组[3]. toArray(); 备注集 = 数组[4]. toArray(); for(i=0; i<图形集.length; i++) { if(表格 1. 行数() <= i) 表格 1. 增加行(); 表格 1. 设置格子文本(i, 0, 图形集[i]); 表格 1. 设置格子文本(i, 1, 支持集[i]); 表格 1. 设置格子文本(i, 2, 时间集[i]); 表格 1. 设置格子文本(i, 3, 个数集[i]); 表格 1. 设置格子文本(i, 4, 备注集[i]); }







北京伟联科技有限公司

E-mail: info@witlinc.cn Tel: 010-58449900 Fax: 010-58449900 地址: 北京海淀区双清路1号院文华楼310

WitLinc Technology Inc.

E-mail: info@witlinc.com Tel: +1778-300-9900 Fax: +1778-300-9080 Add: #712-63 W 2ND AVE VANCOUVER BC V5Y 0G8