General Industry Automation Monitor Configuration Software

世纪星通用工业自动化监控组态软件

用户手册

Windows98/2000/NT

北京世纪长秋科技有限公司

版权声明

本《用户手册》版权为北京世纪长秋科技有限公司全权所有。

未经北京世纪长秋科技有限公司书面许可，任何单位和个人不得以任何方式复制、储存、传输、影印、记录或抄袭本用户手册的任何内容。

版权所有，违者必究。

地址：北京市海淀区上地信息路 1 号国际创业园 A 栋 7 层
电话：（010）82896868（十线）
传真：（010）82893362　　邮政编码：100085
E-mail：Info@kingword.com.cn
Http: //www.kingword.com.cn

欢迎使用世纪星通用工业自动化监控组态软件系统

欢迎使用世纪星通用工业自动化监控组态软件（简称世纪星）—运行于Windows 98/2000/NT 平台上的人机接口（HMI）应用组态软件系统。世纪星作为一个实时的人机界面实用程序生成器，可以产生在管理级别上的监控和数据采集（SCADA）系统。

世纪星系统集强大功能和使用方便于一体。可视化 IE 风格界面、丰富的工具栏，使用它，操作人员可以直接进入开发状态，节省宝贵的时间；上百种图形控件，既提供所需的组件，又是画面制作向导；强大的全屏幕编辑功能，提供更大的制作空间；渐进颜色的使用，将用户带入三维动画世界；终身技术支持及软件升级服务，使用户系统永远保持最新的技术，有效地保护用户的投资。

世纪星的网络功能使企业的基层和其它部门建立起联系，现场操作人员和工厂管理人员都可以看到各种数据。管理人员不需要深入生产现场，就可以获得实时数据和历史数据，优化控制现场作业，提高生产效率和产品质量。

世纪星可用于电力、通讯、制冷、化工、石油、钢铁、煤炭、水泥、纺织、机械制造、造纸、食品、水处理、建材、环保、智能楼宇、实验室、交通等多种工程领域。无论您的应用场合如何，您都可以使用世纪星构造有效的监控和数据采集系统（SCADA）。

系统安装要求

如果需要安装运行世纪星，至少需要下列软硬件配置：

- Microsoft Windows 98 / 2000 或 Windows NT 操作系统

- IBM PC 及其兼容机，奔腾 266 MHz 以上的 CPU
- 32M 以上内存，建议使用 64 M 以上的内存
- 500 M 可用硬盘空间
- VGA/SVGA 显示卡

注：目前市场上出售的计算机都可满足要求。

安装世纪星监控组态软件系统

世纪星存于一张经过压缩的 CD-ROM 高密度光盘上。光盘上的安装程序 SETUP.EXE 将世纪星自动安装到用户的硬盘目录（缺省路径为 C:\CenturyStar），并在桌面上建立相应的应用程序图标。

世纪星的安装步骤：

1、启动 Windows 98 / 2000 或 Windows NT
2、在光盘驱动器中插入世纪星的光盘
3、世纪星光盘具有自引导安装功能，插入世纪星光盘后，系统会弹出安装向导，用户可根据需要进行选择安装
4、若不能自引导安装，请启动光盘驱动器窗口
5、双击安装程序 SETUP.EXE
6、根据安装提示，自动完成系统安装

安装硬件加密系统

世纪星硬件加密系统安装于计算机打印端口(LPT1)上。如果用户在打印端口上有其它外设，可以将外设接到本硬件上，本硬件不会影响其它外设的使用。

关于用户手册

本用户手册全面介绍了世纪星的系统特性、系统组成、各部分的系统命令及其使用、操作步骤、用户编程方法及其演示示例等，是用户快速学习和使用世纪星的基本手册。本手册的更新和升级事宜，由世纪长秋公司授权并组织实施。未经世纪长秋公司授权或书面许可，任何单位或个人无权对本手册内容进行修改或更正，世纪长秋公司概不负责由此而造成的客户损失。

技术支持

世纪长秋公司提供世纪星的全部技术支持，包括用户培训、软件升级及技术咨询。

世纪长秋公司提供了各种方式，尽最大可能帮助您解决关于世纪星产品的问题。如果您在使用世纪星的过程中遇到了问题，您可以在世纪星帮助系统中使用用户在线帮助功能。如果您希望得到比在线帮助更多的技术资料，请查阅世纪星的用户手册和参考手册。

如果您仍没找到您需要的资料，您可以通过以下方式得到更多的技术支持。

终身服务

在您购买世纪星的同时，您将填写一张世纪星用户登记卡。从此时起，您就成为众多世纪星用户大家庭中的一员。世纪长秋公司凭借此卡，向您提供终身技术支持及软件升级服务。

电话热线服务

在世纪星的使用过程中，不论您何时何地遇到问题，您都可以拨打技术支持电话，世纪长秋公司将以最快的速度给您满意的答复。

现场技术支持服务

如果您遇到通过电话不能解决的问题，您可以获得现场支持的服务。世纪长秋公司将授权技术支持工程师或技术人员帮助您到现场解决问题。

目　录

封面.....1
版权声明.....2
欢迎使用世纪星通用工业自动化监控组态软件系统 3
系统安装要求 3
安装世纪星监控组态软件系统 4
安装硬件加密系统 4
关于用户手册 5
技术支持 5
目录.....7
第一章　世纪星监控组态软件系统使用基础 26
1.1　世纪星监控组态软件系统的组成 26
1.1.1世纪星监控组态软件开发系统 26
1.1.2世纪星监控组态软件运行系统 26
1.2　开发系统主窗口 27
1.2.1世纪星监控组态软件开发系统主窗口描述 27
1.2.2工程项目管理器 28
1.2.2.1　新建 28
1.2.2.2　打开 30
1.2.2.3　连接 31
1.2.2.4　修改 31
1.2.2.5　删除 31
1.2.2.6　保护.....32
1.2.2.7　退出 33
1.3　系统工具栏 33
1.3.1常用工具栏 34
1.3.2绘图工具栏 35

1.3.3排列工具栏 36
1.3.4文字工具栏 38
1.3.5属性工具栏 39
1.3.5.1 线型 40
1.3.5.2 线宽 41
1.3.5.3 阴影图案 41
1.3.5.4 颜色 42
1.3.6系统状态行 43
1.3.7全屏显示 43
1.3.8栅格设定 44
1.3.9右键弹出菜单 44
1.3.10 用光标键移动图素 44
1.3.11 用鼠标和键盘选择图素..... 45
1.3.12 系统配置 45
1.3.12.1 开发系统配置 45
1.3.12.2 运行系统配置 47
1.3.12.3 报警系统配置 48
1.3.12.4 历史记录配置 51
1.3.12.5 网络系统配置 52
1.3.12.6 事件记录配置 52
1.3.13 系统安全设置 52
1.3.14 变量使用查询..... 56

第二章 画面图形绘制 60
2.1 画面文件管理 60
2.1.1创建新画面 60
2.1.2打开画面 61
2.1.3 导入画面..... 61
2.1.4保存画面 62
2.1.5关闭画面 63
2.1.6删除画面 64

2.1.7打印画面 65
2.1.8退出 65
2.2 图形绘制 66
2.2.1直线 66
2.2.2垂直/水平直线 67
2.2.3折线 68
2.2.4椭圆 68
2.2.5矩形 69
2.2.6圆角矩形 70
2.2.7多边形 70
2.2.8文本 71
2.2.9位图 71
2.2.10 历史趋势曲线 72
2.2.11 实时趋势曲线 73
2.2.12 报警窗口 73
2.2.13 按钮 75
2.2.14 改变文本 75
2.2.15 按钮文本 76
2.2.16 自定义图库 76
2.2.16.1 自定义图库管理 76
2.2.16.2 绘制自定义图库 78
2.2.16.3 创建自定义图库 80
2.3 图形编辑 81
2.3.1撤消 81
2.3.2重复 81
2.3.3剪切 81
2.3.4拷贝 81
2.3.5粘贴 81
2.3.6图形删除 82
2.3.7装载点位图 82
2.3.8粘贴点位图 82

2.3.9拷贝点位图 83
2.3.10 显示原始大小点位图 83
2.3.11 从屏幕抓取点位图 83
2.3.12 透明点位图 83
2.3.13 渐进色填充 83
2.3.14 全部选中..... 84
2.3.15 改变图素形状 85
2.4 图形排列 86
2.4.1图素后移 86
2.4.2图素前移 86
2.4.3图素对齐 87
2.4.4垂直等距 89
2.4.5水平等距 89
2.4.6垂直镜像 90
2.4.7水平镜像 90
2.4.8顺时针旋转 91
2.4.9逆时针旋转 91
2.4.10 合成单元 92
2.4.11 分裂单元 93
2.5 图形文本 93
2.5.1字体 93
2.5.2加粗 94
2.5.3斜体 95
2.5.4下划线 95
2.5.5字体缩小 95
2.5.6字体放大 96
2.5.7居左 96
2.5.8居中 96
2.5.9居右 96
2.6 窗口 97
2.7 帮助 98

第三章　变量数据库 100
3.1　变量类型 100
3.1.1系统变量 100
3.1.2内存变量 100
3.1.3I0变量 101
3.1.4特殊变量 102
3.2　变量的域 103
3.2.1离散变量的域 103
3.2.2整数、实数变量的域 104
3.2.3信息变量的域 105
3.2.4报警窗口的域 105
3.2.5历史曲线的域 105
3.3　变量数据库 106
3.4　数据库维护 114
3.4.1更新变量计数 114
3.4.2删除未用变量 115

第四章　动画连接 117
4.1　动画连接对话框 117
4.2　连接表达式 120
4.3　输入连接 121
4.3.1用户输入连接 121
4.3.1.1　离散值输入连接 122
4.3.1.2　模拟值输入连接 123
4.3.1.3　信息值输入连接 125
4.3.2滑动杆输入连接 127
4.3.2.1　水平滑动杆输入连接 128
4.3.2.2　垂直滑动杆输入连接 129
4.3.3按键输入连接 130
4.3.3.1　按键离散输入 130

4.3.3.2 命令语言输入 130
4.4 输出连接 132
4.4.1直线属性连接 132
4.4.2填充属性连接 134
4.4.3文本属性连接 135
4.4.4大小连接..... 135
4.4.4.1 高度变化连接 135
4.4.4.2 宽度变化连接 135
4.4.5移动连接 136
4.4.5.1 水平移动连接 137
4.4.5.2 垂直移动连接 138
4.4.6百分比填充连接 139
4.4.6.1 百分比水平填充连接 139
4.4.6.2 百分比垂直填充连接 140
4.4.7数值显示连接 141
4.4.7.1 离散值数值显示连接 141
4.4.7.2 模拟值数值显示连接 142
4.4.7.3 信息值数值显示连接 143
4.4.8特殊功能连接 144
4.4.8.1 隐含连接 144
4.4.8.2 闪烁连接 144
4.4.8.3 旋转连接 145
4.4.9访问权限设置 147

第五章 命令语言 148
5.1 命令语言编辑器 149
5.1.1 功能键 149
5.1.2 操作键 151
5.1.3 变量域输入 151
5.1.4 函数输入 151
5.2 命令语言描述 152

5.2.1 运算符 152
5.2.2 赋值语句 153
5.2.3 IF-THEN-ELSE语句 153
5.2.4 FOR-NEXT循环语句 156
5.2.5 命令语言注释 156
5.3 命令语言的函数 156
5.4 用户自定义函数 157
5.4.1 用户自定义函数格式 157
5.4.2 用户自定义函数调用格式 158
5.5 应用程序命令语言 158
5.6 热键命令语言 159
5.7 事件命令语言 161
5.8 数据变化命令语言 162

第六章 趋势曲线和报警窗口 165
6.1 实时趋势曲线 165
6.1.1 实时趋势曲线的含义 165
6.1.2 实时趋势曲线的定义 166
6.1.3 实时曲线定义对话框 167
6.2 历史趋势曲线 169
6.2.1 历史趋势曲线的定义 170
6.2.2 历史趋势曲线的曲线定义对话框 171
6.3 报警窗口 173
6.3.1 报警窗口的概念 173
6.3.2 报警窗口的定义 174
6.3.3 报警窗口定义对话框 175
6.4 与报警窗口有关的操作 177
6.4.1 报警组定义 177
6.4.2 报警配置 178

第七章 I/O 驱动设备 179

7.1 I/O 驱动设备管理概述 179
7.2 I/O 驱动设备分类 180
7.3 I/O 驱动设备管理 183
7.3.1 串口通信设备安装向导 183
7.3.2 板卡设备安装向导 186
7.3.3 DDE动态数据交换设备安装向导 188
7.3.4 网络设备安装向导 190
7.4 I/O设备驱动修改和删除 193
7.4.1 PLC 设备修改和删除 193
7.4.2 智能仪表设备修改和删除 194
7.4.3 智能模块设备修改和删除 195
7.4.4 变频器设备修改和删除 195
7.4.5 板卡设备修改和删除 197
7.4.6 DDE 设备修改和删除 197
7.4.7 网络设备修改和删除 198
7.5 I/O 仿真驱动系统 199
7.5.1 仿真驱动程序的菜单命令 199
7.6 “串行通讯方式”概述 201
7.6.1 连接一个串口 201
7.6.2 连接多个串口 202
7.7 RS-232 简介 203
7.7.1 79针到25针转换示意图 204
7.7.2 RS232/RS485 205

第八章 工程配方管理 206
8.1 工程配方及工程配方管理简介 206
8.2 工程配方模板的定义 207
8.3 使用工程配方管理 207
8.3.1 创建工程配方模板文件 207
8.3.2 调用工程配方管理函数 210
8.4 工程配方管理函数 211

第九章　世纪星监控组态软件运行系统 212
9.1　文件菜单 212
9.1.1打开画面 212
9.1.2关闭画面 213
9.1.3画面打印 213
9.1.4报表 213
9.1.5退出 214
9.2　系统菜单 214
9.2.1系统安全 215
9.2.1.1　用户登录 215
9.2.1.2　改变密码 216
9.2.1.3　配置用户 216
9.2.1.4　退出登录 216
9.2.2重建DDE连接 217
9.2.3重建未成功的DDE连接 217
9.2.4重新启动报警历史记录 217
9.2.5重新启动历史数据记录 218
9.2.6停止历史数据记录 218
9.3　关于菜单 218

第十章　组态报表及打印功能 220
10.1 打印功能总述 220
10.1.1　画面打印 221
10.1.2　历史数据打印 221
10.1.3　实时数据打印 221
10.1.4　历史报警打印 222
10.2 事件查询报表 223
10.3 使用命令语言实现定时打印 224
10.4 组态报表模板 226
10.5 组态报表 233

第十一章 SQL 功能 236
11.1 功能简介 236
11.2 SQL 管理 236
11.2.1 表格模板管理 236
11.2.2 记录列表管理 237
11.3 SQL 函数 239
11.4 ODBC 使用方法示例 240

第十二章 系统变量 245
12.1 $访问级别 245
12.2 $配置用户 245
12.3 $用户名 246
12.4 $用户登录 246
12.5 $密码输入 247
12.6 $改变密码 247
12.7 $日期 248
12.8 $日期信息 248
12.9 $年 248
12.10 $月 249
12.11 $日 249
12.12 $时间 249
12.13 $时间信息 249
12.14 $时 250
12.15 $分 250
12.16 $秒 250
12.17 $毫秒 251
12.18 $历史记录 251
12.19 $报警记录 251
12.20 $新报警 252
12.21 $垂直坐标 252
12.22 $水平坐标 252

12.23 $开始 DDE 会话.....253
12.24 $打印机错误 253
12.25 $系统 253

第十三章 .域 254
13.1 .Ack 254
13.2 .Alarm 254
13.3 .AlarmEnabled 254
13.4 .ChartLength 255
13.5 .ChartStart 255
13.6 .Comment 255
13.7 .DevTarget 256
13.8 .DisplayMode 256
13.9 .EngUnits 256
13.10 .Group 257
13.11 .HiHiLimit 257
13.12 .HiHiStatus 257
13.13 .HiLimit 258
13.14 .HiStatus 258
13.15 .LoLimit 258
13.16 .LoLoLimit 259
13.17 .LoLoStatus 259
13.18 .LoStatus 259
13.19 .MajorDevPct 260
13.20 .MajorDevStatus 260
13.21 .MaxRaw 260
13.22 .MaxRange 261
13.23 .MaxValue 261
13.24 .MinorDevPct 261
13.25 .MinorDevStatus 262
13.26 .MinRaw 262
13.27 .MinRange 262

13.28 .MinValue 263
13.29 .Name 263
13.30 .NextPage 263
13.31 .OffMsg 264
13.32 .OnMsg 264
13.33 .Pen1 ... Pen8 264
13.34 .PrevPage 264
13.35 .PriFrom 265
13.36 .Priority 265
13.37 .PriTo 265
13.38 .RocPct 266
13.39 .RocStatus 266
13.40 .ScooterPosLeft 266
13.41 .ScooterPosRight 266
13.42 .TotalPage 267

第十四章 函数 268
14.1 Abs( ) 268
14.2 Ack( ) 268
14.3 AckAll( ) 269
14.4 ActivateApp( ) 269
14.5 AlmMoveWindow( ) 270
14.6 ArcCos( ) 270
14.7 ArcSin( ) 271
14.8 ArcTan( ) 271
14.9 Beep( ) 271
14.10 Bit( ) 272
14.11 BitSet( ) 272
14.12 ChangePassword( ) 273
14.13 CloseWindow( ) 273
14.14 ConvertDateTime( ) 274
14.15 CopyScreenToPrint( ) 274

14.16 Cos( ) 275
14.17 Counter( ) 275
14.18 Delay( ) 276
14.19 DialogStringEntry( ) 276
14.20 DialogValueEntry( ) 277
14.21 DirectoryCreate( ) 278
14.22 DirectoryRemove( ) 278
14.23 DText( ) 278
14.24 EventsLogQuery( ).....279
14.25 Exit( ) 279
14.26 Exp( ) 280
14.27 FileCopy( ) 280
14.28 FileDelete( ) 281
14.29 FilelineDelete( ) 281
14.30 FileMove( ) 282
14.31 FileReadFields( ) 282
14.32 FileReadMessage( ) 283
14.33 FileReadWriteFields( ).....284
14.34 FileReadWriteTag( ).....285
14.35 FileSelectDialog () 285
14.36 FileWriteFields( ) 286
14.37 FileWriteMessage( ) 287
14.38 FileWriteTags( ) 287
14.39 FindFileOrDirectory( ).....288
14.40 FloatFromFileTable( ) 288
14.41 FormAutoSave( ) 289
14.42 GetHistDataualue( ).....290
14.43 GetWindowMilliseconds( ).....290
14.44 HideWindow( ) 291
14.45 HTGetPenName( ) 291
14.46 HTGetTimeAtScooter( ) 292
14.47 HTGetTimeStringAtScooter( ) 292

14.48 HTGetValue( ).....293
14.49 HTGetValueAtScooter( ).....293
14.50 HTGetValueAtZone( ) 294
14.51 HTScrollLeft( ) 295
14.52 HTScrollRight( ) 295
14.53 HTSelectTag( ) 296
14.54 HTSetPenName( ).....296
14.55 HTUpdateToCurrentTime( ) 297
14.56 HTZoomIn( ) 297
14.57 HTZoomOut( ) 298
14.58 In( ) 299
14.59 InfoAppActive( ) 299
14.60 InfoAppTitle( ) 299
14.61 InfoDisk( ) 300
14.62 InfoFile( ) 301
14.63 InfoAppDir( ) 301
14.64 InfoResource( ) 302
14.65 Int( ) 302
14.66 IOSetAppTopic( ) 303
14.67 IOSetItem( ) 303
14.68 Log( ) 304
14.69 Log10( ) 304
14.70 LogOff( ) 305
14.71 LogOn( ) 305
14.72 Out( ) 305
14.73 PI( ) 306
14.74 PlaySound( ) 306
14.75 PrintEventsQueryForm( ) 307
14.76 PrintFormTable( ) 307
14.77 PrintHistAlarm( ) 308
14.78 PrintHistData( ) 308
14.79 PrintRealAlarm( ) 309

14.80 PrintRealMessage( ) 309
14.81 PrintWindow( ) 310
14.82 pvAddRealPoint( ) 311
14.83 pvAddSetPoint( ) 311
14.84 pvClear( ) 312
14.85 pvGetValue( ) 313
14.86 pvInitSetCurve( ) 314
14.87 pvLoadData( ) 315
14.88 vModifySetCurve( ) 316
14.89 pvMoveSlide( ) 316
14.90 pvSaveData( ) 317
14.91 Rand( ) 318
14.92 RecipeDelete( ) 318
14.93 RecipeLoad( ) 319
14.94 RecipeSave( ) 320
14.95 RecipeSelectNextRecipe( ) 321
14.96 RecipeSelectRecipe( ) 322
14.97 RecipeSelectPreviousRecipe( ) 322
14.98 ReDDELink( ) 323
14.99 ReFailDDELink( ) 323
14.100 ReStartAlarmLog( ) 324
14.101 ReStartHistoryDataLog( ) 324
14.102 Round( ) 324
14.103 SendKeys( ) 325
14.104 Sgn( ) 326
14.105 ShowAt( ) 326
14.106 ShowTopLeftAt( ) 327
14.107 ShowWindow( ) 328
14.108 Sin( ) 328
14.109 SQLAppendStatement( ) 328
14.110 SQLClearStatement( ) 329
14.111 SQLClearTable( ) 329

14.112 SQLCommit( ) 330
14.113 SQLConnect( ) 330
14.114 SQLCreateTable( ) 331
14.115 SQlDelete( ) 331
14.116 SQLDisconnect( ) 332
14.117 SQLDropTable( ) 332
14.118 SQLEndSelect( ) 333
14.119 SQLExecute( ) 333
14.120 SQLFirst( ) 334
14.121 SQLGetRecord( ) 334
14.122 SQLInsert( ) 334
14.123 SQLInsertEnd( ) 335
14.124 SQLInsertExecute( ) 335
14.125 SQLInsertPrepare( ) 336
14.126 SQLLast( ) 336
14.127 SQLLoadStatement( ) 337
14.128 SQLNext( ) 337
14.129 SQLNemRows( ) 337
14.130 SQLPrepareStatement( ) 338
14.131 SQLPrev( ) 338
14.132 SQLRollback( ) 339
14.133 SQLSelect( ) 339
14.134 SQLSetParamChar( ) 340
14.135 SQLSeeParamDate( ) 341
14.136 SQLSetParamFloat( ) 341
14.137 SQLSetParamInt( ) 342
14.138 SQLStatement( ) 342
14.139 SQLTransact( ) 343
14.140 SQLUpdate( ) 343
14.141 SQLUpdateCurrent( ) 344
14.142 Sqrt( ) 344
14.143 StartApp( ) 345

14.144 StopHistoryDataLog( ) 345
14.145 StringASCII( ) 345
14.146 StringChar( ) 346
14.147 StringCompare( ).....346
14.148 StringFromChartLength( ).....347
14.149 StringFromFileTable( ).....347
14.150 StringFromIntg( ) 348
14.151 StringFromReal( ) 348
14.152 StringFromTime( ) 349
14.153 StringInString( ) 350
14.154 StringLeft( ) 350
14.155 StringLen( ) 351
14.156 StringLower( ) 351
14.157 StringMid( ) 352
14.158 StringReplace( ) 352
14.159 StringRight( ) 353
14.160 StringSpace( ) 354
14.161 StringTest( ) 354
14.162 StringToIntg( ) 355
14.163 StringToReal( ) 356
14.164 StringTrim( ) 356
14.165 StringUpper( ) 357
14.166 tagReadDisc( ) 357
14.167 tagReadInt( ) 357
14.168 tagReadFloat( ) 358
14.169 tagReadMess( ) 358
14.170 tagWriteDisc( ) 359
14.171 tagWriteFloat( ) 359
14.172 tagWriteInt( ) 360
14.173 tagWriteMess( ) 360
14.174 Tan( ) 361
14.175 Text( ) 361

14.176 Timer( ) 362
14.177 Trunc( ) 362
14.178 txAddMessage( ) 363
14.179 txClear( ) 363
14.180 txDisplayMessage( ) 364
14.181 txLoadFile( ) 364
14.182 txSaveFile( ) 465
14.183 UpdateRealChart( ) 366
14.184 UseDll( ) 366
14.185 WWControl( ) 366
14.186 xbClear( ) 367
14.187 xbSetHisPara( ) 368
14.188 xbSetParameter( ) 368
14.189 xbSetTagname( ) 369
14.190 xbSetTimeLength( ) 369
14.191 xxClear( ) 369
14.192 xxModify( ) 370
14.193 xyAddPoint( ) 371
14.194 xyClear( ) 371
14.195 xyGetValue( ) 372
14.196 xyLoadFile( ) 372
14.197 xySaveFile( ) 373
14.198 xySelectPen( ) 374
14.199 xySetMinMaxValue( ) 374

第十五章 图形控件 376
15.1 图形控件概述 376
15.2 图形控件使用方法 376
15.3 按钮类型 378
15.4 面板类型 384
15.5 指示灯类型 387
15.6 时钟类型 389

15.7 开关类型 391
15.8 垂直刻度类型 392
15.9 水平刻度类型 393
15.10 标注类型 394
15.11 仪表类型 396
15.12 滑动杆类型 401
15.13 历史曲线类型 403
15.14 阀门类型 408
15.15 泵类型 412
15.16 管道类型 414
15.17 温控曲线 419
15.18 棒图控件 422
15.19 XY曲线 424
15.20 带图控件 426
15.21 窗口控制控件 428

附录A 数据库数据转换及实用程序 DBExcel 430
附录B 历史数据记录及实用程序 HDExcel 433
附录C DDEDriver ToolKit 驱动系统 438
附录D 世纪星文件格式说明 448
附录E 常见问题解答..... 451

第一章 世纪星组态软件系统使用基础

1.1 世纪星组态软件系统的组成

世纪星是在 PC 机上开发的智能型人机接口软件系统，它以 Microsoft Windows 98/2000/NT 中文平台作为其操作系统，充分利用了 Windows 图形功能完备、界面一致性好、易学易用的特点。它比以往使用专用机开发的工业控制系统更有通用性，并且可以利用 PC 机丰富的软件资源进行二次开发。

世纪星由开发系统 CSMAKER 和运行系统 CSVIEWER 两部分组成。CSMAKER 和 CSVIEWER 是各自独立的 Windows 32 位应用程序，均可单独使用；两者又相互依存，在开发系统中设计开发的画面应用程序必须在 CSVIEWER 运行环境中才能运行。

1.1.1 世纪星监控组态软件开发系统

世纪星开发系统是其应用程序的集成开发环境。软件开发者在这个环境中完成界面的设计、数据库定义、动画连接等。开发系统具有先进完善的图形生成功能；数据库中有多种数据类型，能合理地抽象控制对象的特性，对数据的报警、趋势曲线、历史数据记录、安全防范等重要功能有简单的操作办法。利用图形控件功能可以大大减少设计界面的时间，从整体上提高工控软件的质量。

1.1.2 世纪星监控组态软件运行系统

CSVIEWER 是世纪星的实时运行环境，用于显示开发系统中建立的图形动画画面，并负责数据库与 I/O 服务程序的数据交换。它通过实时数据库管理从工业控

制对象采集到的各种数据，并用动画的方式形象地表示数据的变化，同时完成报警、历史数据记录、趋势曲线等监视功能，并可生成历史数据文件。

1.2 开发系统主窗口

1.2.1 世纪星监控组态软件开发系统主窗口描述

第一次运行世纪星软件时，先启动中文 Windows 98/2000/NT 操作系统之后，在桌面上用鼠标双击“世纪星开发系统”或从开始菜单启动世纪星开发系统 CSMAKER，系统弹出工程项目管理器，双击该工程项目器中的演示工程 1，系统弹出打开画面选择框，选择“生产线”画面，按确定按钮，系统会弹出如下图 1-1 所示。

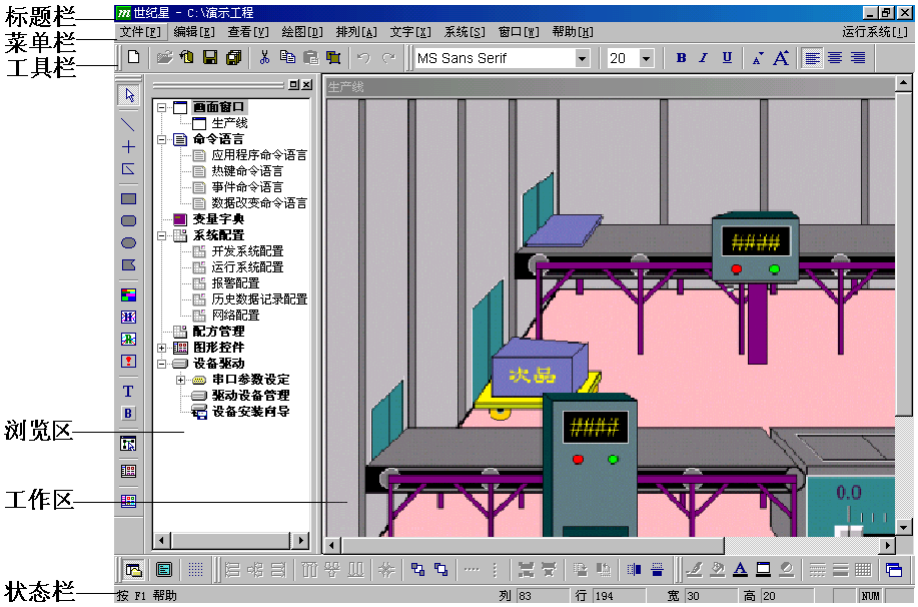


图 1-1

世纪星开发系统由标题栏、菜单栏、工具栏、浏览区、工作区窗口和状态栏组成。

屏幕最上端是标题栏，标题栏用于显示用户在开发系统配置中定义的标题及应用程序目录。标题栏下面是菜单栏和工具栏。中间部分是工作区窗口，即画面制作窗口。屏幕左端是浏览区窗口。屏幕最底端是状态栏。

世纪星支持屏幕显示的分辨率为：640 × 480；800 ×600；1024 × 768；1280 × 1024。建议使用屏幕分辨率为：800 × 600 或 1024 × 768。

1.2.2 工程项目管理器

启动世纪星开发系统后，用户第一步要做的事情就是为不同的工程项目设置相应的工程项目名。一旦工程项目名设置完成，用户应用程序的所有信息将被保存在该工程项目目录中。不同的工程项目应设置不同的工程项目名，以避免混淆和数据丢失。

启动世纪星开发系统后，在未打开画面之前，选择菜单“文件\工程项目管理”命令，则弹出下图 1-2 所示的设置窗口。

1.2.2.1 新建

此图标用于建立一个工程项目，单击“新建”图标，则弹出“新建工程项目”对话框如图 1-3 所示：

工程项目名称： 用户在此输入框中输入代表一个工程项目的名称，这个名称应象征性地代表了工程项目的主要内容。

工程项目路径： 编辑输入与工程名称相对应的工程目录。工程项目路径用于更换当前应用程序，因为应用程序由工程项目路径唯一标识，不同的应用程序应置于不同的路径下。用户也可以使用

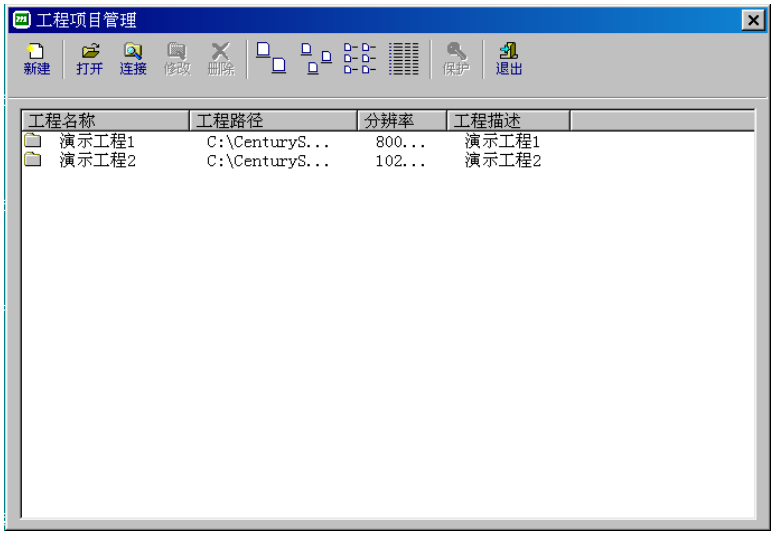

工程项目管理
新建
打开
连接
修改
删除
保护
退出
工程名称
工程路径
分辨率
工程描述
演示工程1
C:\CenturyS...
800...
演示工程1
演示工程2
C:\CenturyS...
102...
演示工程2

图 1-2

新建工程项目
工程项目名称
新工程
确定
工程项目路径
C:\新工程
浏览
取消
分辨率选择
640 x 480
800 x 600
1024 x 768
1280 x 1024
工程项目描述
新工程

图 1-3

“浏览”按钮来选择工程项目路径。单击“浏览”按钮，则弹出“选择路径”对话框如图 1-4 所示。

图 1-4

分　辨　率：用户定义项目工程的分辨率工作模式。打开时系统将按照系统显示分辨率自动匹配。

工程项目描述：当前工程项目的注释。输入文字，进一步说明和解释当前的工程项目。

1.2.2.2　打开

用户选中工程项目后，单击“打开”图标或者双击工程项目名，即可打开所选中的工程项目。

1.2.2.3 连接

此图标用于连接用户已经创建的工程项目，单击此图标，则弹出连接工程项目对话框如图 1-5 所示。

图 1-5

有关操作详见“新建/工程项目”的说明。

1.2.2.4 修改

此图标用于修改用户已经创建的工程项目，单击此图标，则弹出修改工程项目对话框如图 1-6 所示。

有关操作详见“新建/工程项目”的说明。

1.2.2.5 删除

此图标用于删除用户已经创建的工程项目，单击此图标，则弹出删除工程项目对话框如图 1-7 所示。

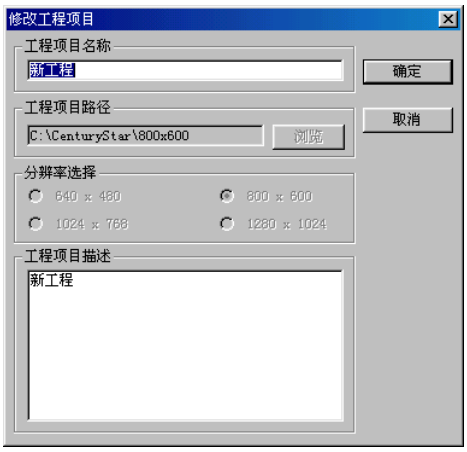


图 1-6

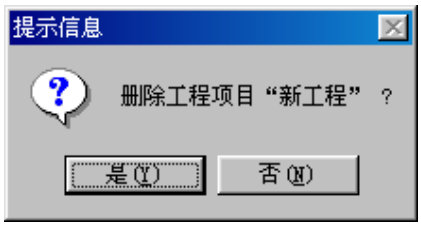


图 1-7

按“是”按钮，删除选中的工程项目，按“否”按钮,则取消删除操作。

注：正在使用的工程项目不能被删除。

1.2.2.6　保护

此图标用于对工程项目的加密保护，选中某工程后，单击此图标，则弹出设置工程项目密码对话框如图 1-8 所示：

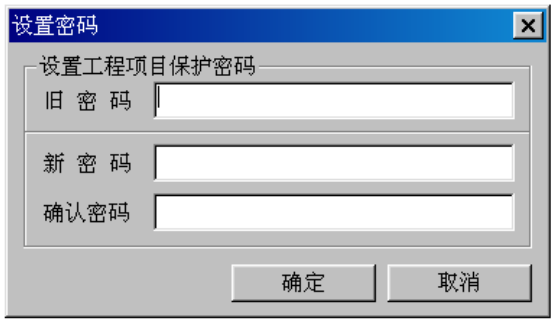


图 1-8

对已经加密的工程，需输入正确的旧密码，才能改变密码，对未加密的工程可不需输入旧密码，而直接输入要加的密码；要进入已经加密的工程，必须正确输入密码才可，而对没有输入加密的工程，可直接进入。

1.2.2.7 退出

单击此图标，则退出工程项目管理器。

1.3 系统工具栏

工具栏对应某些菜单的选项和命令，直接单击工具栏可执行相应的菜单选项命令，因此，使用工具栏比使用菜单命令更加直接、迅速。

世纪星包含六种工具栏：常用工具栏、绘图工具栏、属性工具栏、文字工具栏、排列工具栏和视图工具栏。

用户可以用鼠标按住工具栏左边的移动条(工具栏水平排列)或工具栏上边的移动条(工具栏垂直排列)移动工具栏到屏幕的任何地方。用鼠标左键双击工具栏，工具栏即可回到屏幕边框固定位置。

在主菜单的“查看”项中，选择工具栏对应的标题名，可以显示或隐含工具栏。

1.3.1　常用工具栏

常用工具栏如图 1-9 所示。

图 1-9

常用工具栏中各按钮的描述如下：

按钮图标	按钮名称	描　　述
	新画面	创建新画面
	打开画面	打开已有的画面
	关闭画面	关闭打开的画面
	保存画面	保存打开的画面
	保存所有画面	保存所有打开的画面
	剪切	剪切选定的内容到系统裁剪板中
	拷贝	拷贝选定的内容到系统裁剪板中

	复制	复制选定的内容到系统裁剪板中
	粘贴	粘贴系统裁剪板中的内容
	撤消	撤消最近一次操作
	重复	重复先前撤消的操作

1.3.2 绘图工具栏

绘图工具栏如图 1-10 所示。

图 1-10

绘图工具栏中各按钮的描述如下：

按钮图标	按钮名称	描　　述
	选择	选择画面窗口中的图素
	直线	绘制直线
	垂直/水平直线	绘制垂直/水平直线
	折线	绘制折线
	矩形	绘制矩形

	圆角矩形	绘制圆角矩形
	椭圆	绘制椭圆
	多边形	绘制多边形
	位图	绘制位图，点鼠标右键装载位图
	历史趋势曲线	绘制历史趋势曲线
	实时趋势曲线	绘制实时趋势曲线
	报警窗口	绘制报警窗口
	文本	输入文本
	按钮	绘制按钮
	全部选中	选中画面中的所有图素
	图形控件	绘制图形控件单元
	自定义图库	绘制自定义图库

1.3.3 排列工具栏

排列工具栏如图 1-11 所示。

图 1-11

排列工具栏中各按钮的描述如下：

按钮图标	按钮名称	描　　述
	向左对齐	选中的图素向左对齐
	水平向中对齐	选中的图素水平向中对齐
	向右对齐	选中的图素向右对齐
	向上对齐	选中的图素向上对齐
	垂直向中对齐	选中的图素垂直向中对齐
	向下对齐	选中的图素向下对齐
	向中心对齐	选中的图素向中心对齐
	图素后移	选中的图素移到其它图素之后
	图素前移	选中的图素移到其它图素之前
	水平等距	选中的图素在水平方向上间隔相等
	垂直等距	选中的图素在垂直方向上间隔相等
	合成单元	所有选中的图素合成为一个单元

	分裂单元	一个单元分裂成图素
	顺时针旋转	图素顺时针旋转 90 度
	逆时针旋转	图素逆时针旋转 90 度
	水平镜像	图素以垂直方向中心线翻转
	垂直镜像	图素以水平方向中心线翻转

1.3.4　文字工具栏

文字工具栏如图 1-12 所示。

图 1-12

文字工具栏中各按钮的描述如下：

Times New Roman		设定选定文本的字体
20		设定字体大小
按钮图标	按钮名称	描　　述
B	加粗	文本字体加粗

	斜体	文本字体变成斜体
	下划线	文本显示下划线
	缩小	文本字体缩小一号
	放大	文本字体放大一号
	居左	显示时文本向左对齐
	居中	显示时文本向中对齐
	居右	显示时文本向右对齐

1.3.5 属性工具栏

属性工具栏用于改变绘制图素的线型、线宽、阴影及颜色属性。如图 1-13 所示：

图 1-13

属性工具栏中各按钮的描述如下：

按钮图标	按钮名称	描　　述
	直线颜色	改变直线颜色

	填充颜色	改变图形填充颜色
	文本颜色	改变文本颜色
	背景颜色	改变背景颜色
	透明颜色	改变位图的透明颜色
	线型	改变线条的类型
	线宽	改变线宽
	阴影图案	改变填充阴影图案

1.3.5.1　线型

世纪星提供六种直线类型：无线、实线、虚线、点线、一点虚线和两点虚线。如图 1-14 所示。

图 1-14

当选中一个图素，并且其图形类型为直线、垂直/水平直线、折线、矩形、圆角矩形、椭圆、多边形和位图中的一种时，可以在绘图特性中改变直线的线型。

1.3.5.2 线宽

世纪星提供六种直线线宽：一象素线宽、二象素线宽、三象素线宽、四象素线宽、五象素线宽和六象素线宽。如图 1-15 所示。

图 1-15

当选中一个图素，并且其图形类型为直线、垂直/水平直线、折线、矩形、圆角矩形、椭圆、多边形和位图中的一种时，可以在绘图特性中改变直线的线宽。

1.3.5.3 阴影图案

世纪星提供六种阴影图案：45 度向上斜线组成的阴影图案、水平和垂直交叉组成的阴影图案、45 度斜线交叉组成的阴影图案、45 度向下斜线组成的阴影图案、水平线组成的阴影图案和垂直线组成的阴影图案。如图 1-16 所示。

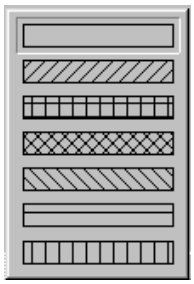


图 1-16

当选中一个图素，并且其图形类型为矩形、圆角矩形、椭圆、和多边形中的一种时，可以在绘图特性中改变填充的阴影图案。

1.3.5.4　颜色

绘图特性中颜色选择部分用于改变直线、填充、屏幕背景和文本的颜色。

世纪星调色板提供 16.7 百万种颜色。当用户用鼠标左键单击颜色区域时，弹出如图 1-17 的颜色调色板。

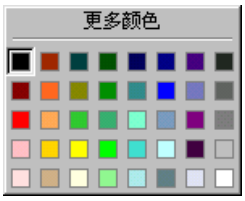


图 1-17

当用户在图 1-17 中用鼠标左键单击更多颜色按钮时，弹出如图 1-18 的颜色调色板。

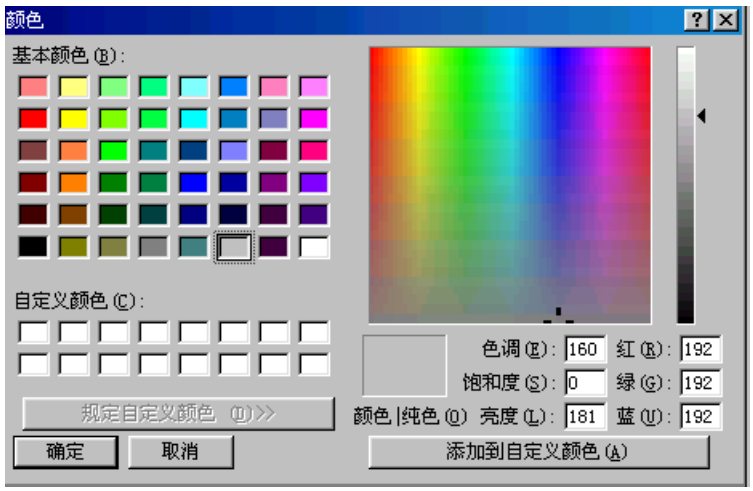


图 1-18

1.3.6 系统状态行

状态行既不接受用户输入信息也不产生命令消息，通过它可以显示一些文本信息，借以给用户一定的提示信息。

世纪星状态行如图 1-19 所示：

图 1-19

状态行显示信息如下：

（1）状态行左边为提示信息区。

（2）状态行中间部分显示象素坐标（列，行）即（X，Y）及象素高度和象素宽度。当用户选中一个图素时，状态行显示图素左上角的象素坐标（X，Y），以及图素的象素高度和象素宽度。当用户选中多个图素时，状态行显示所有图素最左边象素坐标（X）、最上边象素坐标（Y），以及整个选中图素的象素高度和象素宽度。当用户未选中图素时，状态行显示当前光标位置的象素坐标（X，Y），此时象素高度和象素宽度显示区域中没有显示。

（3）状态行右边部分显示键盘状态：NumLock，CapsLock，ScroolLock。

在主菜单的“查看”项中，选择状态行，可以显示或隐含状态行。

1.3.7 全屏显示

图 1-20

点击“全屏显示”按钮可切换至于全屏编辑和画面全屏显示。如图 1-20 所示。全屏编辑时，可利用右键弹出菜单加入工具栏和状态栏，进行全屏幕画面制作。

有两种方式退出全屏显示状态：1. 用全屏显示工具栏按钮；2. 用 ESC 键。

1.3.8　栅格设定

栅格设定用于在屏幕显示或隐含栅格，同时还可以设置栅格的宽度和高度象素值的大小。栅格设定对话框如图 1-21 所示：

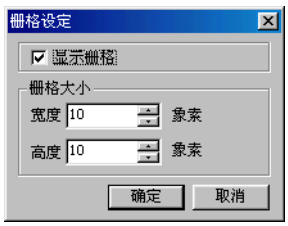


图 1-21

1.3.9　右键弹出菜单

在世纪星开发系统运行时，可随时按鼠标右键弹出如下图 1-22 所示的快捷菜单（对于不同的图形单击右键，则弹出不同的快捷菜单）。最大限度地满足用户开发应用程序的需要。

1.3.10　用光标键移动图素

世纪星规定，用光标键移动图素，必须遵守以下原则：

（1）用光标键 LEFT,TOP,RIGHT,BOTTOM 每次移动 1 个象素。

（2）用 Shift 键+光标键 LEFT,TOP,RIGHT,BOTTOM 每次移动 10 个象素。

（3）用 Ctrl 键+光标键 LEFT,TOP,RIGHT,BOTTOM 每次移动 50 个象素。

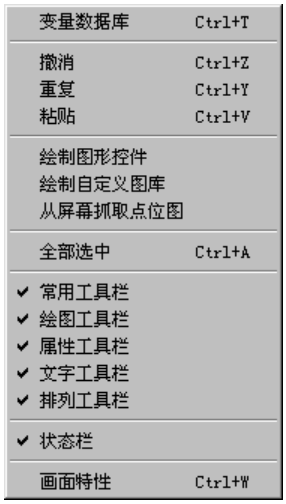


图 1-22

1.3.11 用鼠标和键盘选择图素

世纪星规定，用 CTRL+鼠标可选定图素，具体操作如下：

按住 CTRL 键后，用鼠标左键单击要选定的图素，可多次重复操作，以选定同一画面中的多个图素。此功能有利于图形编辑。

1.3.12 系统配置

选择菜单项：系统/系统配置菜单项，可对开发系统、运行系统、报警系统、历史数据记录和网络系统进行相应的设定，以最大程度地满足用户的各自需要。

1.3.12.1 开发系统配置

选择菜单项：系统/系统配置→开发系统，弹出开发系统配置对话框如图 1-23 所示：

图 1-23

对话框的参数设置如下：

标　　题：输入开发系统在启动时出现在标题栏中的标题。
栅格大小：选择开发系统启动时是否带有栅格，及其宽度和高度的象素值。
系统参数：线选中（偏离中心线）：用鼠标选中图素时，选中直线的范围。
撤消次数：系统能取消操作的最多次数。取值范围 2～25。
启动信息记录窗口：设置是否启动信息记录窗口。

注：若“撤消[UNDO]次数”小于 2 或大于 25，则弹出提示信息框，如下图 1-24 所示。

图 1-24

确定设置后弹出另一提示信息框，如图 1-25 所示。

图 1-25

1.3.12.2 运行系统配置

一、运行系统参数设置

选择菜单项：系统/系统配置→运行系统，弹出运行系统配置对话框如图 1-26 所示：

图 1-26

对话框的参数设置如下：

标　　题： 输入世纪星在运行时出现在标题栏中的标题。
显示应用程序目录： 选择此选项使当前应用程序目录显示在标题栏中。
系统主菜单： 选择此选项使系统在运行时具有菜单。
控制菜单： 选择此选项使系统在运行时标题栏中带有系统菜单框。
大小控制： 选择此选项使系统在运行时，可以改变窗口的大小。
最大化控制： 选择此选项使系统在运行时标题栏中带有最大化按扭。
最小化控制： 选择此选项使系统在运行时标题栏中带有最小化按扭。
基 准 时 间：系统运行时，系统时间中断的时间间隔。
数据采样频率：系统运行时从下位机数据采样的时间间隔。
闪 烁 频 率： 系统运行时，图素闪烁的频率。
运行时保持最大化：系统运行时保持全屏幕显示。
启动信息记录窗口：选择此选项在系统运行时启动信息记录窗口。

二、初始窗口设定

用于设定运行系统启动时打开的窗口。在图 1-26 对话框中，选择初始窗口，得到初始窗口设定对话框如图 1-27 所示：

1.3.12.3　报警系统配置

一、报警系统参数设置

选择菜单项“系统/系统配置→报警系统”，弹出运行系统配置对话框如图 1-28 所示。在此窗口中可以进行有关报警参数的设置。

二、报警系统记录格式

用于设定报警系统的记录格式。在图 1-28 对话框中，选择记录格式窗口，得到记录格式窗口设定对话框如图 1-29 所示：

运行系统配置
系统配置
初始窗口
定制菜单
主菜单
系统帮助
警告信息
退出画面
在线帮助
报警窗口
历史曲线
生产线
反应车间
确定
取消
应用(A)

图 1-27

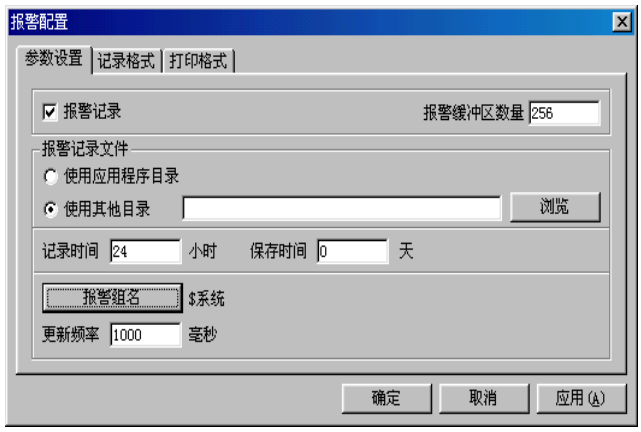

报警配置
参数设置
记录格式
打印格式
报警记录
报警缓冲区数量
256
报警记录文件
使用应用程序目录
使用其他目录
浏览
记录时间
24
小时
保存时间
0
天
报警组名
$系统
更新频率
1000
毫秒
确定
取消
应用(A)

图 1-28

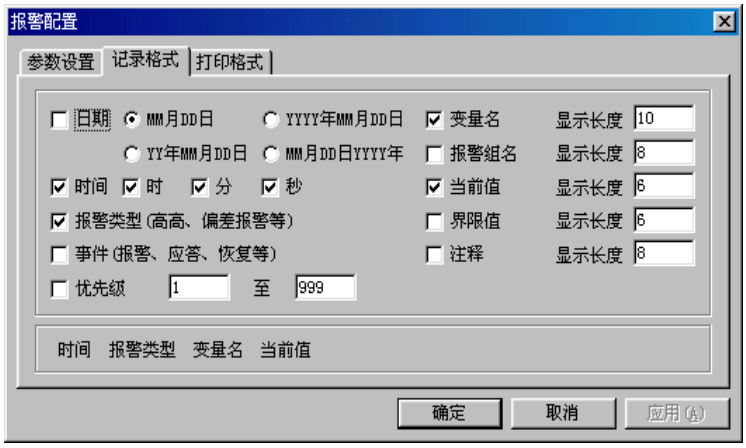


图 1-29

三、报警系统打印格式

用于设定报警系统的打印格式。在图 1-28 对话框中，选择打印格式窗口，得到打印格式窗口设定对话框如图 1-30 所示：

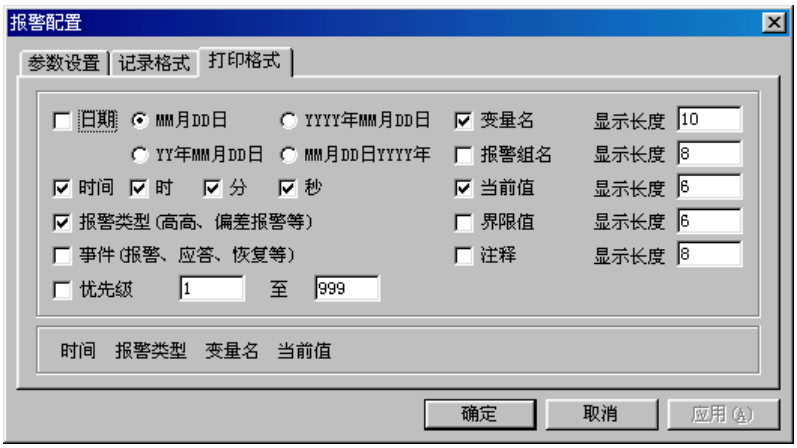


图 1-30

1.3.12.4 历史数据记录配置

选择菜单项“系统/系统配置→历史数据记录”，弹出历史数据记录配置对话框如图 1-31 所示：

图 1-31

在“历史数据记录配置”窗口内选择“浏览”按键，则弹出如图 1-32 窗口：

图 1-32

1.3.12.5　网络系统配置

选择菜单项“系统/系统配置→网络系统”，弹出网络系统对话框如上图 1-33 所示。

图 1-33

1.3.12.6　事件记录配置

事件记录配置用于定制一些特殊事件记录，方便事件查询双击开发系统的“系统配置→事件记录配置”，系统弹出如下图 1-34 对话框：

在“事件记录配置”窗口选择“增加”按键，则弹出如图 1-35 窗口。

1.3.13　系统安全设置

在主菜单的“系统”项中，选择系统安全，即可进行系统安全设置。

世纪星为用户设置了两个缺省用户：

（1）用户名：**超级用户**；密码：CenturyStar；访问级别：9999。
（2）用户名：NONE；密码：**无**；访问级别：0。

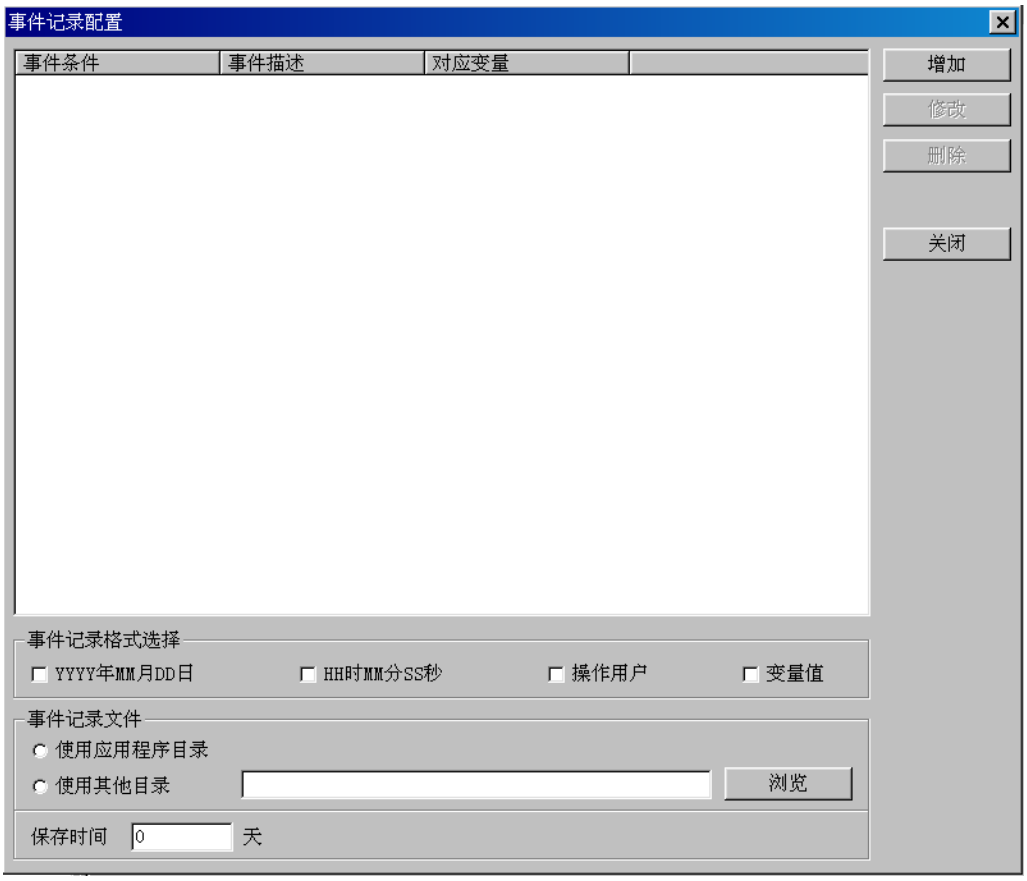


图 1-34

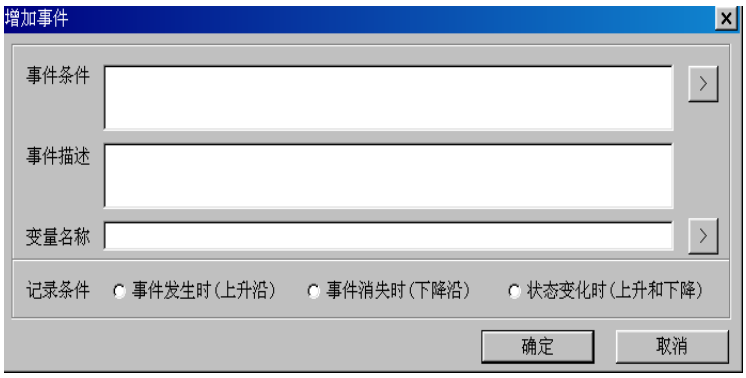


图 1-35

世纪星规定：

（1）只有访问级别大于 9000 的用户才能配置用户。
（2）只有访问级别大于 9000 的用户才能执行具有系统安全设置的功能。
（3）密码区分大小写。

一、用户登录

用户登录对话框如图 1-36 所示：

图 1-36

用户初次登录时，只有上述两个用户登录，并且只有超级用户才能配置用户。在配置其它用户以后，配置的用户就可以在此对话框上登录。

二、改变密码

用户成功登录后，可以改变其自己的密码。要改变其它用户的密码，登录用户的访问级别必须大于 9000，并且只能在配置用户对话框中进行。改变密码对话框如图 1-37 所示：

图 1-37

三、配置用户

用户成功登录后，如果访问级别大于 9000，就可以配置用户。否则，显示如图 1-38 所示错误信息。

图 1-38

配置用户对话框如图 1-39 所示，参数设置如下：

增加：增加新的用户，同时设置其密码和访问级别。

修改：修改用户的密码和访问级别。“**超级用户**”和“NONE”的用户名和访问级别不能被修改，但可以修改“**超级用户**”密码。

删除：删除选中的用户。不能删除“**超级用户**”和“NONE”。

保存：保存用户信息到数据库中。

图 1-39

四、退出登录

用户退出当前登录，同时自动切换至“NONE”用户。

1.3.14　变量使用查询

在用开发系统开发自己的应用程序时，用户经常会碰到这样一种情况：变量在某画面或某命令语言中使用过，用“更新变量计数”和“删除未用变量”时，无法删除该变量，而用户又想改变变量属性或删除该变量。此时，用“变量使用查询”功能就非常有必要。你可以找到某变量在哪个具体的画面和哪一种命令语

言中使用过。

在开发系统中的“系统”菜单下，选择“变量使用查询”功能，如下图 1-40 所示：

图 1-40

系统弹出如下图 1-41 对话框：

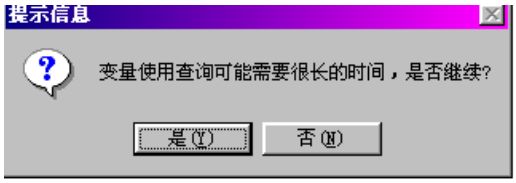


图 1-41

单击“取消”表示不进行变量使用查询，单击“确定”，进入如图 1-42 对话框：

在该对话框中，单击“>”图标，系统弹出该工程中用户所定义的所有变量列表，如图 1-43 所示：

图 1-42

图 1-43

在变量列表中用鼠标左键双击或选中某变量后选择“确定”按钮，如选择“反应罐温度”，系统弹出如图 1-44 对话框：

在上图所示对话框中，选择“取消”按钮，表示放弃操作，单击“>”图标，表示重新选择变量，选择“确定”按钮，系统开始查询变量使用情况，当查询结束后，弹出查询结果，如按以上操作，会弹出如图 1-45 对话：

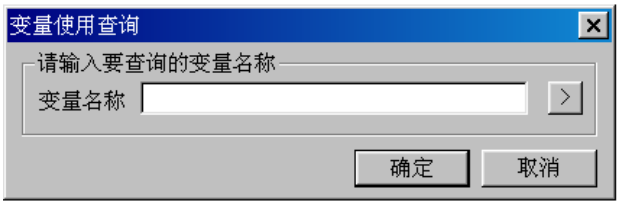


图 1-44

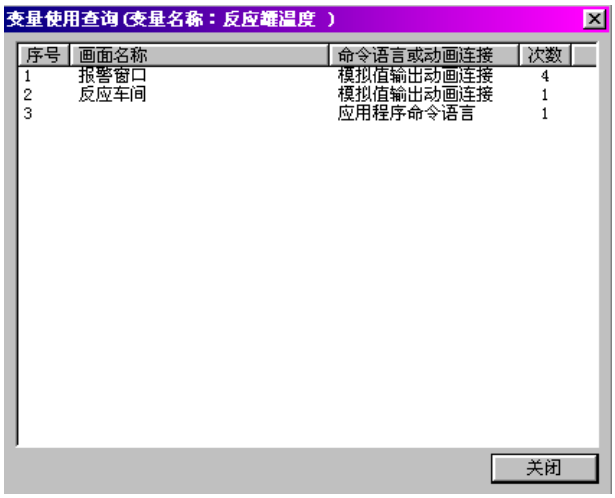


图 1-45

有了上表列出变量的使用情况，在相应画面或命令语言中查询变量就会方便很多。

第二章　画面图形绘制

2.1　画面文件管理

画面文件管理是对画面文件的图形数据进行管理，主要包括：创建新画面、打开画面、导入画面、画面关闭、画面保存、删除画面、打印画面和退出等一系列操作。

2.1.1　创建新画面

用鼠标左键单击“文件→新画面”菜单选项或使用热键 Ctrl+N 后，弹出“新画面”对话框，在对话框中可定义画面的名称、背景颜色、风格和画面创建时所处的位置。

创建新画面的对话框如图 2-1 所示：

图 2-1

新画面参数设置如下：

名　称：新画面的名称，画面保存时就以此名称作为文件名。画面名最长为32字符，画面名中不能包含下列字符：　\ / : * ? ” < > |

背景色：用于设置窗口的背景色，用鼠标按下背景色区域，弹出浮动的调色板窗口，可从中选择一种颜色。

注　释：与画面有关的注释信息。

边　框：画面创建时边框的类型。

类　型：有三种画面类型可供选择。“覆盖式”画面出现时，它重叠在当前画面之上。关闭后被覆盖的画面又可见。“替换式”画面出现时，所有与之相交的画面自动从屏幕上和内存中删除。建议尽量使用“替换式”画面以节约内存。“弹出式”画面类似于“覆盖式”画面，一般用于对话框或信息显示。

位　置：画面的左上角坐标、宽度和高度的值，以象素为单位计算。

选　项

标 题 杆：用于决定画面是否有标题杆，若有标题杆，标题杆上将显示画面名称。

大小可变：用于决定画面是否能由用户改变大小。改变画面大小的操作与改变WINDOWS窗口相同。

2.1.2　打开画面

选择此命令后，弹出列表框，列表框中列出了当前工程路径所有未打开的画面，可用鼠标选择一个或多个画面，单击“确定”打开所有选中的画面，或单击“取消”废弃当前操作。当然，也可进行“全部选中”和“全部清除”画面的操作。如图2-2所示。

2.1.3　导入画面

选择此命令后，弹出如图 2-3 所示的对话框，您可在此搜寻需要导入的窗口所在的目录，并选择您所有导入的窗口名，单击打开即完成导入画面操作。

要注意的是，导入的画面是不带动画连接的。

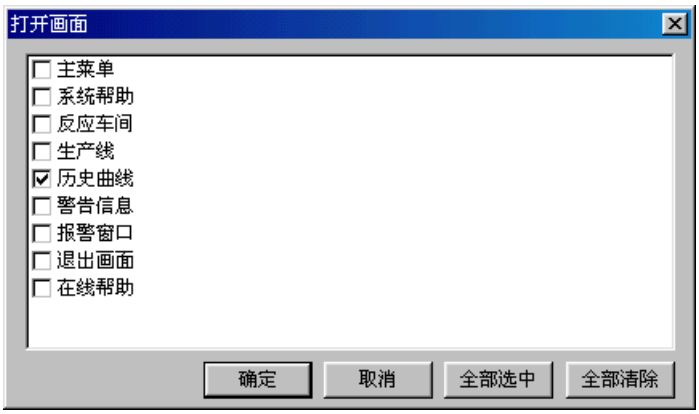


图 2-2

图 2-3

2.1.4　保存画面

选择此命令后，图 2-4 对话框中列出所有已经打开的画面清单。

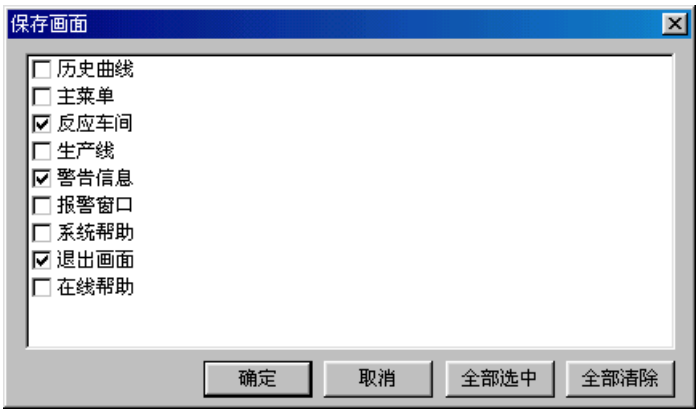


图 2-4

可用鼠标选取一个或多个画面，单击“确定”存入所有选中的画面；单击“取消”不做存储。当然，也可进行“全部选中”和“全部清除”画面的操作。

2.1.5 关闭画面

选择此命令后，弹出列表框如图 2-5 所示:

列表框中列出所有已经打开的画面。可以用鼠标选择其中一个或多个画面，然后单击“确定”关闭所有选中的画面，或单击“取消”放弃当前操作。当然，也可进行“全部选中”和“全部清除”画面的操作。如果用户改变了画面而没有保存，则弹出如图 2-6 对话框让用户选择是否保存画面。

选择“是(Y)”命令后，“世纪星监控组态软件”将所有已经打开并且内容发生改变的画面存入对应的文件。

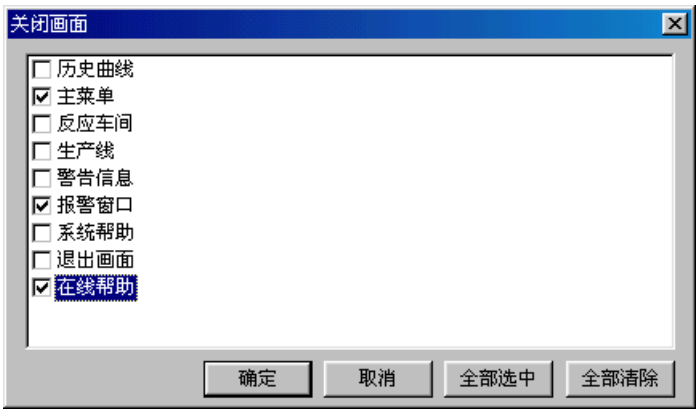


图 2-5

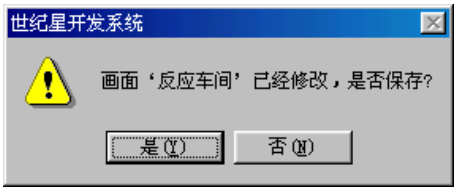


图 2-6

2.1.6 删除画面

先关闭所有已打开的画面，然后选择此命令，则弹出如图 2-7 所示的对话框，其中列出了当前工程路径下的所有画面的清单 ：

可以用鼠标选择一个或多个需要删除的画面。

单击“确定”后执行删除操作。单击“取消”放弃删除操作。CSMAKER 在实际删除每个画面之前会让用户确认删除操作。

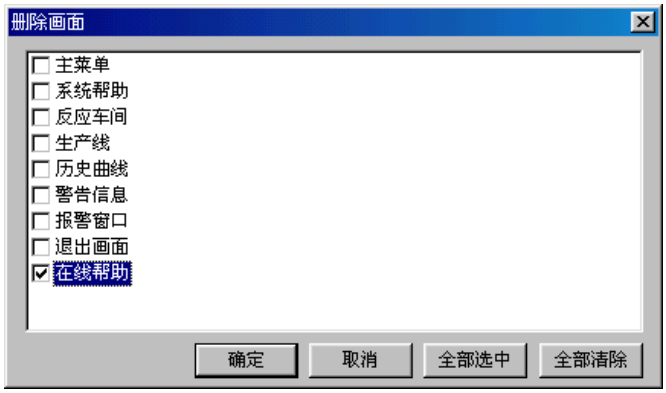


图 2-7

2.1.7 打印画面

打印当前有效画面。

2.1.8 退出

此命令用于退出开发系统，如果画面内容已改变但没有保存，则程序会提示用户选择是否保存。如图 2-8 所示。

图 2-8

如果要保存已修改的画面内容，单击“是”或击字母键 Y。若不保存，单击“否”

或击字母键 N。单击“取消”按钮取消退出操作，回到开发系统 CSMAKER。

2.2 图形绘制

设置应用程序目录后，就可以为每个应用程序建立数目不限的画面，在每个画面上生成互相关联的静态或动态图形对象。这些画面都是由世纪星提供的类型丰富的图素组成的。

世纪星为用户提供了直线、矩形（圆角矩形）、椭圆（圆）、点位图、多边形、文本等基本图形对象，及按钮、趋势曲线窗口、报警窗口等复杂的图形对象。提供了对图形对象在窗口内任意移动、缩放、改变形状、复制、删除、对齐等编辑操作，全面支持键盘、鼠标绘图，并可提供对图形对象的颜色、线型、阴影图案进行改变的操作工具。

世纪星采用面向对象的编程技术，使用户可以方便地建立画面的图形界面。用户构图时可以象搭积木一样利用系统提供的图形对象完成画面的生成。同时支持画面之间的图形对象拷贝，可重复使用以前的开发结果。

2.2.1 直线

选择“直线”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置（即直线的起点）。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的直线，移动鼠标到新的位置（即直线的终点），然后松开左键。直线在两点之间画出。

绘制时，按住 Shift 键，可画出正斜线（与水平方向成 45 或 135 度角的直线）。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的直线如图 2-9 所示：

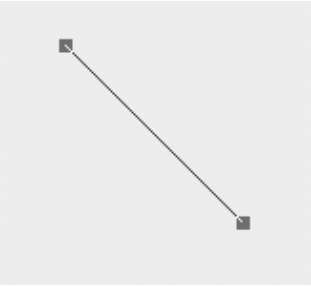


图 2-9

2.2.2 垂直/水平直线

选择“垂直/水平直线”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置（即垂直/水平直线的起点）。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的垂直/水平直线，移动鼠标到新的位置（即垂直或水平直线坐标的终点），当上述两点在水平方向的位移大于垂直方向上的位移时，就画出水平直线，反之则画出垂直直线。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的垂直/水平直线如图 2-10 所示：

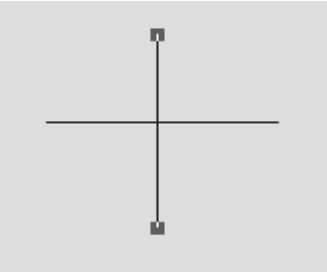


图 2-10

2.2.3　折线

选择“折线”菜单命令，此时鼠标光标变为十字形，折线绘制过程如下：

（1）将鼠标光标置于一个起始位置，单击鼠标左键，即绘制折线的起始点。
（2）移动鼠标，此时屏幕上出现一个随鼠标移动而变化的直线，单击鼠标左键，可以绘出折线的一条直线。
（3）依此方法绘制折线的各条直线。
（4）最后单击鼠标右键完成折线绘制。

绘制的折线如图 2-11 所示：

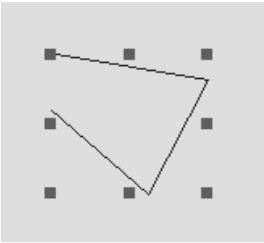


图 2-11

2.2.4　椭圆

选择“椭圆”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的椭圆框，此椭圆框表示所绘制椭圆的大小，移动鼠标到新的位置，然后松开左键。完成椭圆绘制。

绘制时，按住 Shift 键，可画出正圆。单击右键弹出快捷菜单，可对其进行渐进色填充。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的椭圆如图 2-12 所示：

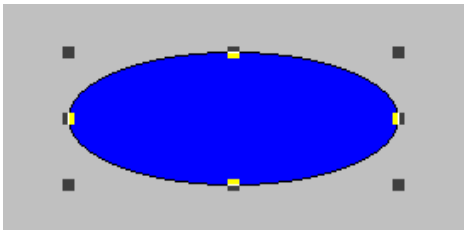


图 2-12

2.2.5 矩形

选择“矩形”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制矩形的大小，移动鼠标到新的位置，然后松开左键。完成矩形绘制。

绘制时，按住 Shift 键，可画出正方形。单击右键弹出快捷菜单，可对其进行渐进色填充。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的矩形如图 2-13 所示：

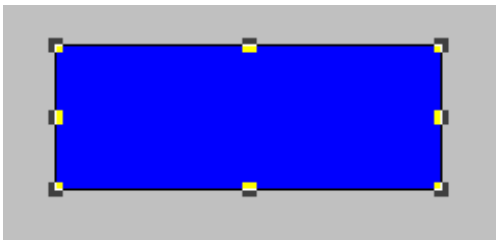


图 2-13

2.2.6　圆角矩形

选择“圆角矩形”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的圆角矩形框，此圆角矩形框表示所绘制圆角矩形的大小，移动鼠标到新的位置，然后松开左键。完成圆角矩形绘制。

绘制时，按住 Shift 键，可画出圆角正方形。单击右键弹出快捷菜单，可对其进行渐进色填充。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的圆角矩形如图 2-14 所示：

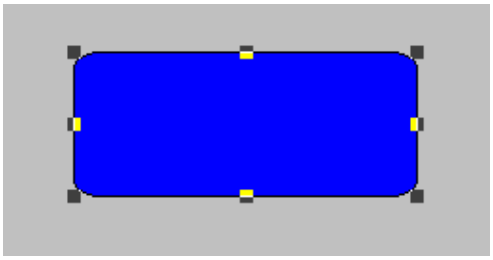


图 2-14

2.2.7　多边形

选择“多边形”菜单命令，此时鼠标光标变为十字形，多边形绘制过程如下：

（1）将鼠标光标置于一个起始位置，单击鼠标左键，即绘制多边形的起始点。
（2）移动鼠标，此时屏幕上出现一个随鼠标移动而变化的直线，单击鼠标左键，可以绘出多边形的一条边。
（3）依此方法绘制多边形的各边。
（4）最后单击鼠标右键，所绘制各边自动闭合成多边形。

绘制的多边形如图 2-15 所示：

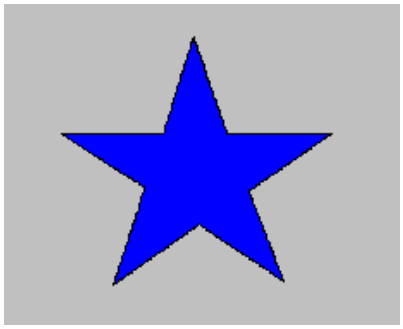


图 2-15

2.2.8 文本

选择“文本”菜单命令，此时鼠标光标变为I字形，将鼠标光标置于一个起始位置。按下鼠标左键，即可输入文本。输入文本结束后，按鼠标右键或用鼠标左键单击画面中其他地方，完成文本绘制。图2-16的立体文本效果是由两个相同内容，但不同颜色的文本错位叠加而成。

图 2-16

2.2.9 位图

选择“位图”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制位图的大小，移动鼠标到新的位置，然后松开左键。完成位图绘制。单击右键弹出快捷菜单，可对其进行位图的装载、粘贴、拷贝、

透明等操作。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的位图如图 2-17 所示：

图 2-17

2.2.10　历史趋势曲线

选择“历史趋势曲线”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制历史趋势曲线的大小，移动鼠标到新的位置，然后松开左键。完成历史趋势曲线绘制。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的历史趋势曲线如图 2-18 所示：

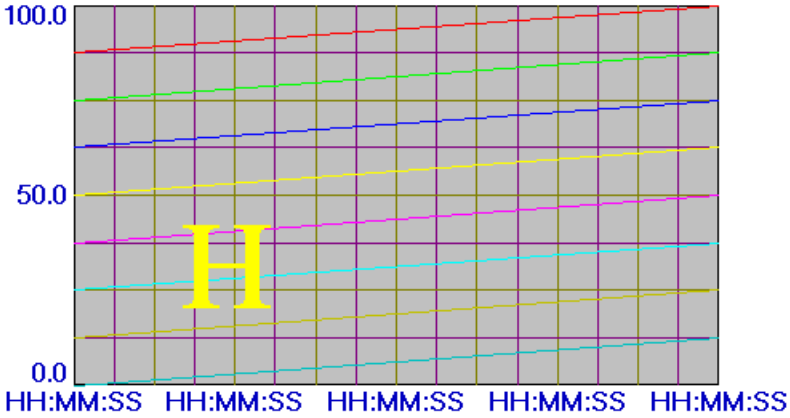


图 2-18

2.2.11 实时趋势曲线

选择“实时趋势曲线”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制实时趋势曲线的大小，移动鼠标到新的位置，然后松开左键。完成实时趋势曲线绘制。

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的实时趋势曲线如图 2-19 所示：

2.2.12 报警窗口

选择“报警窗口”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制报警窗口的大小，移动鼠标到新的位置，然后松开左键。完成报警窗口绘制。

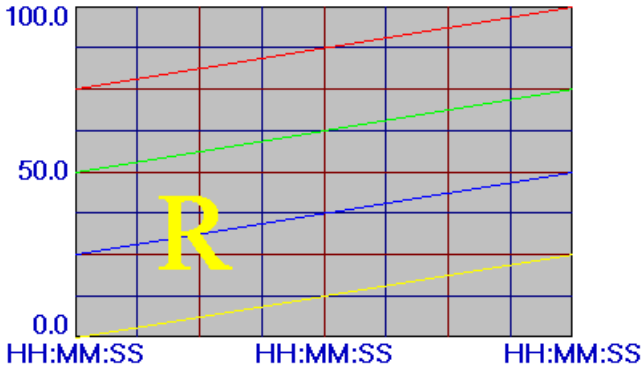


图 2-19

在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的报警窗口如图 2-20 所示：

图 2-20

2.2.13 按钮

选择“按钮”菜单命令，此时鼠标光标变为十字形，将鼠标光标置于一个起始位置。按下鼠标左键并拖动鼠标，此时屏幕上出现一个随鼠标移动而变化的矩形框，此矩形框表示所绘制按钮的大小，移动鼠标到新的位置，然后松开左键。完成按钮绘制。

绘制时，按住 Shift 键，可画出正方形按钮。在绘制过程中，如果需要取消绘制，同时按下鼠标右键即可。

绘制的按钮如图 2-21 所示：

图 2-21

2.2.14 改变文本

当选中一个文本图素时，“绘图/改变文本”菜单命令或鼠标右键快捷菜单中的“改变文本”命令由禁止使用变成允许使用。选取改变文本菜单项，弹出改变文本对话框如图 2-22 所示：

图 2-22

改变文本内容对话框参数：

原文本：改变之前的文本。
新文本：改变之后的文本

2.2.15　按钮文本

当选中一个文本图素时，“绘图→按钮文本”菜单命令或鼠标右键快捷菜单中的“按钮文本”命令由禁止使用变成允许使用。选取按钮文本菜单项，弹出按钮文本对话框如图 2-23 所示：

按钮文本对话框参数：

原文本：按钮文本改变之前的文本。
新文本：按钮文本改变之后的文本

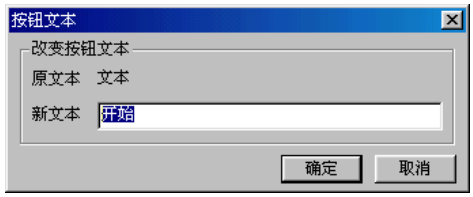


图 2-23

2.2.16　自定义图库

用户自定义图库功能可以不受限制扩充图库或新建图库，以满足不同行业者的需求。

2.2.16.1　自定义图库管理

选择“绘图/自定义图库管理”菜单弹出如图 2-24 所示对话框。在对话框中可以对自定义图库进行新建、连接、修改、删除等管理工作。

单击“新建”按钮，则弹出如图 2-25 所示对话框。

图 2-24

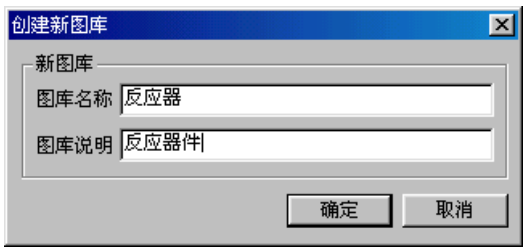


图 2-25

单击“连接”按钮，则弹出如图 2-26 所示对话框。

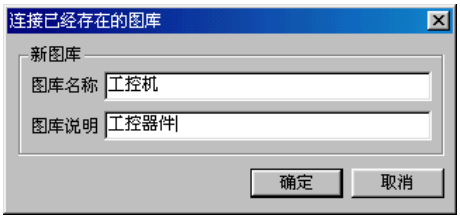


图 2-26

单击“修改”按钮，则弹出如图 2-27 所示对话框。

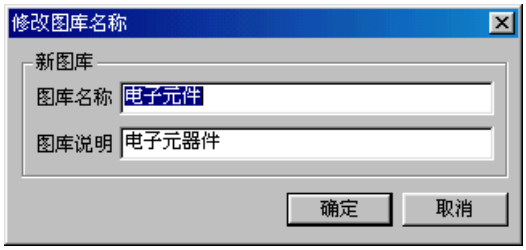


图 2-27

单击“删除”按钮，则弹出如图 2-28 所示对话框。

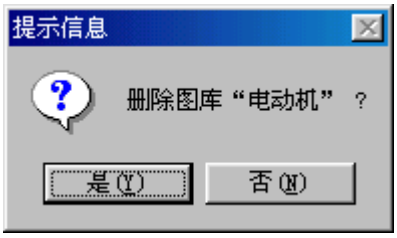


图 2-28

2.2.16.2　绘制自定义图库

选择“绘图→绘制自定义图库”菜单，或绘图工具栏中的自定义图库按钮，

弹出如图 2-29 所示对话框。

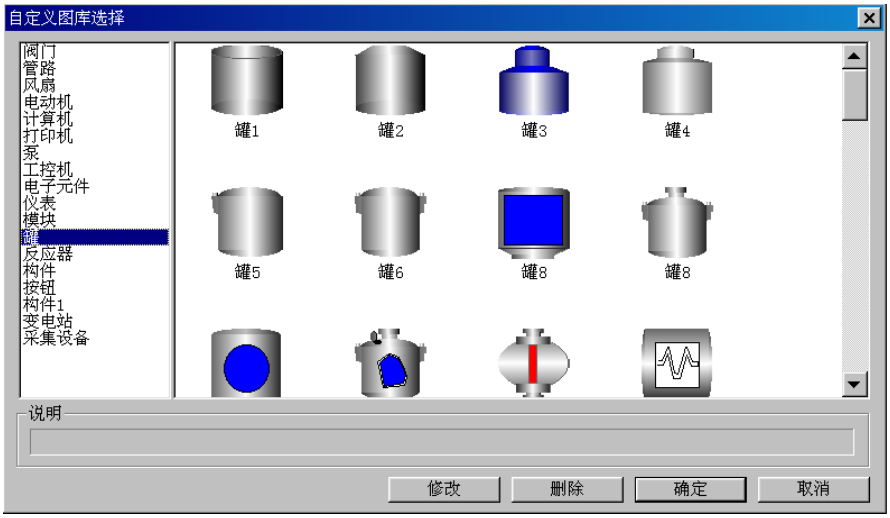


图 2-29

此命令用于对已建立的自定义图库单元名称的修改、删除。 单击“修改”按钮，则弹出如图 2-30 所示对话框。

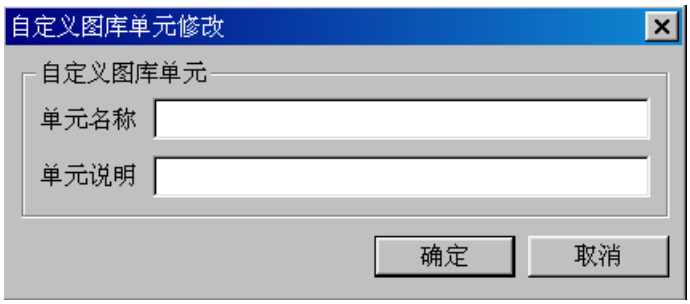


图 2-30

单击“删除”按钮，则弹出如图 2-31 所示对话框。

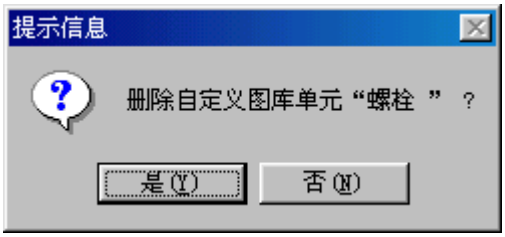


图 2-31

2.2.16.3　创建自定义图库

选中新创建的图形，此时“绘图→创建自定义图库”菜单由不可选变为可选，选择此菜单或单击右键，选择创建自定义图库菜单；弹出如图 2-32 所示对话框。

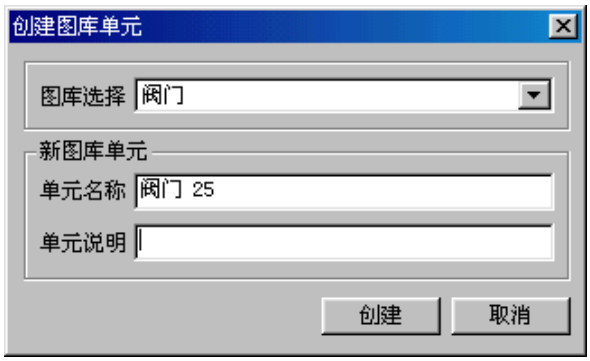


图 2-32

此时可选择新建图形要存放的图库名、图库单元名称及单元说明。单击创建，则新建图形被加入到指定的图库中，完成自定义图库的创建。

2.3 图形编辑

编辑菜单中的菜单命令主要对用户选中的图素进行编辑处理，主要功能有图形编辑撤消、图形编辑重复、图形编辑剪切、图形编辑拷贝、图形编辑粘贴、图形删除、粘贴点位图、显示原始大小点位图、全部选中、改变图素形状等等。

2.3.1 撤消

对上一次的图形编辑操作进行撤消处理。

2.3.2 重复

对上一次的撤消操作进行重复处理。

2.3.3 剪切

此命令是将选中的一个或多个图素从画面中删除，并复制到本系统内部的裁剪板。剪切命令与复制命令的相同之处是都把选中的一个或多个图素复制到系统内部的裁剪板之中，不同之处是剪切命令删除画面中选中的一个或多个图素，而复制命令保留画面中选中的一个或多个图素，其操作方式与复制命令完全相同。

2.3.4 拷贝

此命令将选中的一个或多个图素复制到本系统内部的裁剪板之中。当有图素选中时，灰色的“拷贝”命令将变为正常的显示颜色，表示此命令可对当前选中的一个或多个图形对象进行拷贝操作。

2.3.5 粘贴

此命令将本系统内部裁剪板之中的一个或多个图素复制到指定位置。只有执行了复制命令或剪切命令之后此命令才有效，这时“粘贴”项由灰色变成正常颜

色。

2.3.6　图形删除

删除一个或多个选中的图素。只有选中图素之后，删除命令才有效。

2.3.7　装载点位图

此命令将硬盘中的点位图图形(BITMAP)装载到当前选中的点位图对象中，并且复制的点位图将进行缩放以适应点位图对象的大小。选取当前点位图，执行“装载点位图”命令，则弹出选择点位图文件对话框如图 2-33 所示。

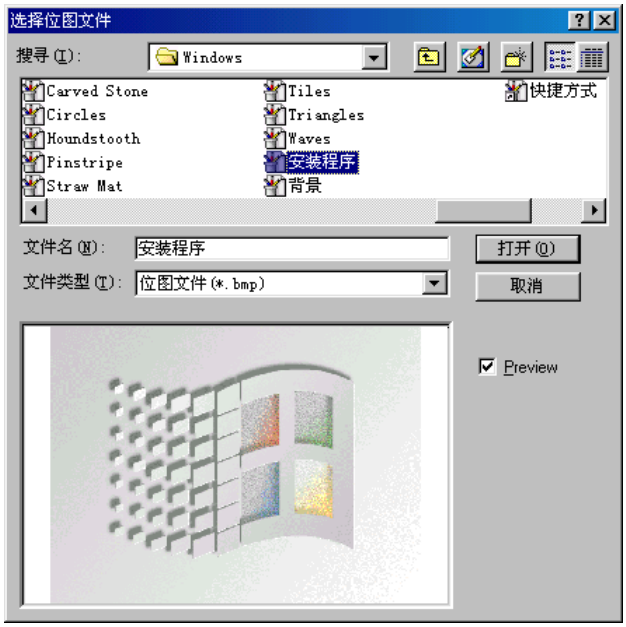


图 2-33

2.3.8　粘贴点位图

此命令将裁剪板之中的点位图(BITMAP)复制到当前选中的点位图对象中，并且复制的点位图将进行缩放以适应点位图对象的大小。如果裁剪板之中的内容不

是点位图格式，则此项菜单无效。

2.3.9 拷贝点位图

此命令将当前选中的点位图对象拷贝到系统裁剪板之中。

2.3.10 显示原始大小点位图

此命令使选中的点位图对象中的点位图恢复到原有尺寸，而不管点位图对象矩形框的大小。点位图恢复到原有尺寸是为了避免缩放引起图象失真。

2.3.11 从屏幕抓取点位图

选择“编辑→从屏幕抓取点位图”菜单时，光标变为小箭头显示，将光标置于所需抓取图形的起始位置，按下鼠标左键并拖动鼠标至图形结束位置，然后松开鼠标左键，完成从屏幕抓取点位图操作。所抓取的点位图可随意移动、缩放。

2.3.12 透明点位图

此命令将选中的点位图对象按指定色进行透明处理。点位图透明处理是为了在本系统加入不规则的点位图图形。

操作示例如图 2-34 所示：

2.3.13 渐进色填充

选择“编辑→渐进色填充”将弹渐进色填充对话框。此命令用于将图形对象进行渐进色填充。对于不同的图形则弹出不同的对话框。

对于矩形、圆角矩形、多边形，选择渐进色填充，弹出图 2-35 所示对话框。对于椭圆，选择渐进色填充，弹出图 2-36 所示对话框。

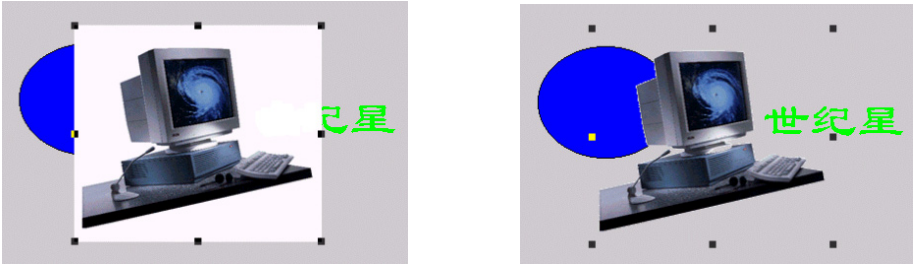


图 2-34

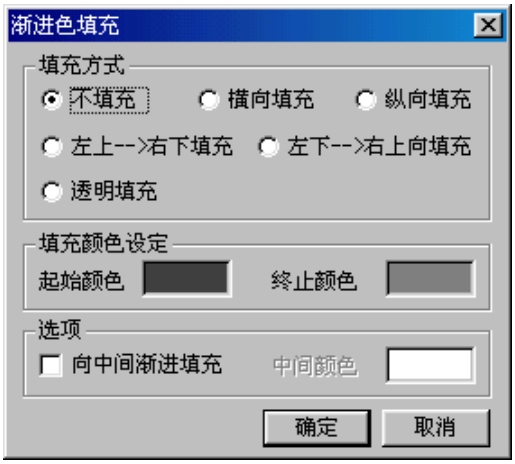


图 2-35

2.3.14　全部选中

选中画面上的所有图素。

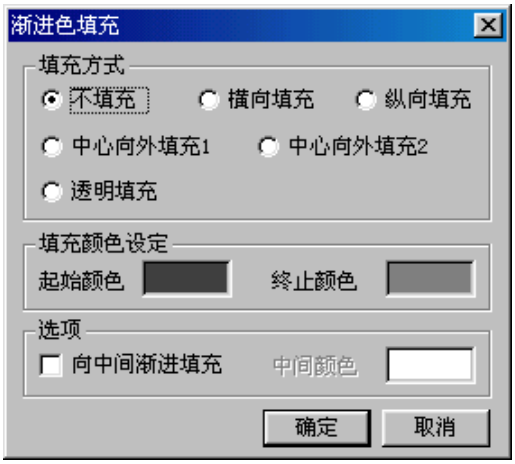


图 2-36

2.3.15 改变图素形状

用于改变多边形或多边线的各顶点的相对位置。用鼠标点中对象后，对象各顶点以小方框表示，其中较大的方框是拖动的焦点，用鼠标切换、拖动此焦点，可改变图素的形状。也可用键盘的 Tab 键或 Shift+Tab 键切换焦点，或用光标键拖动这些焦点。如图 2-37 所示。

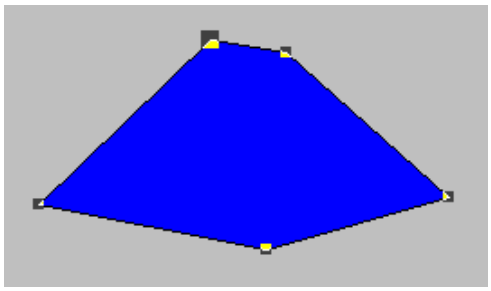


图 2-37

操作方法如下：

（1）用鼠标单击“改变图素形状”菜单命令，此时鼠标光标变为小箭头，同时在菜单“编辑→改变图素形状”命令前面出现“√”号。
（2）用鼠标单击图形对象，则对象顶点以小方框表示，其中较大的方框是拖动的焦点。
（3）用鼠标切换、拖动焦点，可改变图素的形状。
（4）用键盘的 Tab 键或 Shift+Tab 键切换焦点，用光标键拖动这些焦点，直到用户满意为止。

注：键盘移动焦点规则：

（1）用光标键 LEFT,TOP,RIGHT,BOTTOM 每次移动 1 个象素。
（2）用 Shift 键+光标键 LEFT,TOP,RIGHT,BOTTOM 每次移动 10 个象素。
（3）用 Ctrl 键+光标键 LEFT,TOP,RIGHT,BOTTOM 每次移动 50 个象素。

2.4　图形排列

图形排列由一系列调整画面中图形对象排列方式的命令组成。主要图形排列功能有图素后移、图素前移、图素对齐、垂直等距、水平等距、垂直镜像、水平镜像、顺时针旋转、逆时针旋转、合成单元、分裂单元等等。

2.4.1　图素后移

选择“图素后移”菜单，使所有选中的图素移至其它图素后面。“图素后移”操作命令与“图素前移”命令执行相反操作，两者的使用方法完全相同。

图素后移操作示例如图 2-38

2.4.2　图素前移

选择“图素前移”菜单，使所有选中的图素移至其它图素前面。“图素前移”操作命令与“图素后移”命令执行相反操作，两者的使用方法完全相同。

图素前移操作示例如图 2-39 所示：

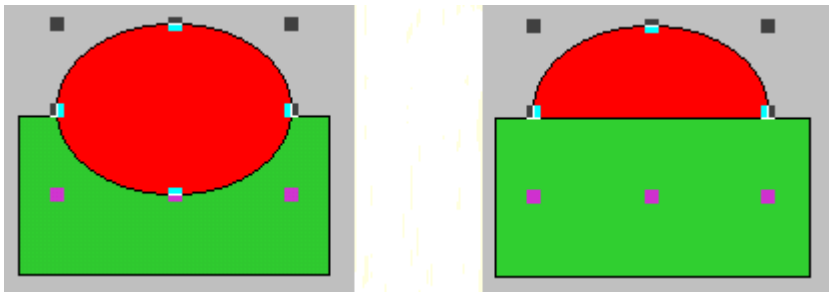


图 2-38　　图素后移

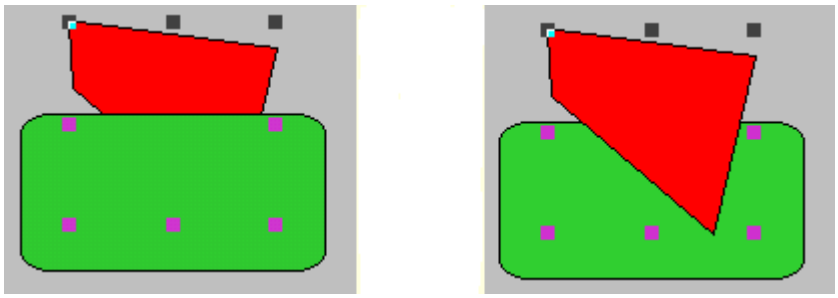


图 2-39　图素前移

2.4.3　图素对齐

选择“对齐”菜单，使所有选中的图素按照选定的方式进行对齐，对齐方式有：左对齐、水平向中对齐、右对齐、上对齐、垂直向中对齐、下对齐和向中心对齐。

各对齐方式示例如图 2-40、图 2-41、图 2-42、图 2-43、图 2-44 所示：

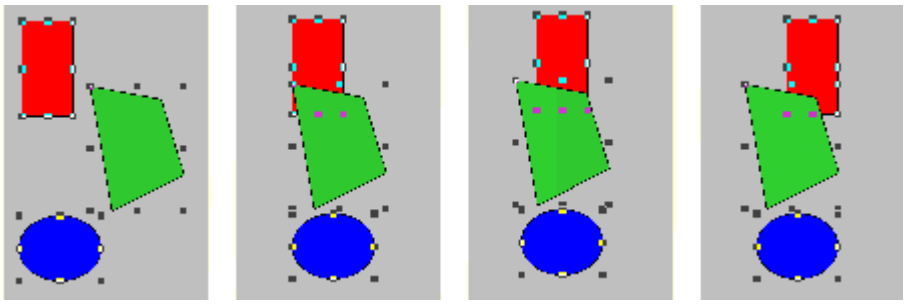


图 2-40　原始图、左对齐、水平向中对齐、右对齐

图 2-41　上对齐

图 2-42 垂直向中对齐

图 2-43　下对齐

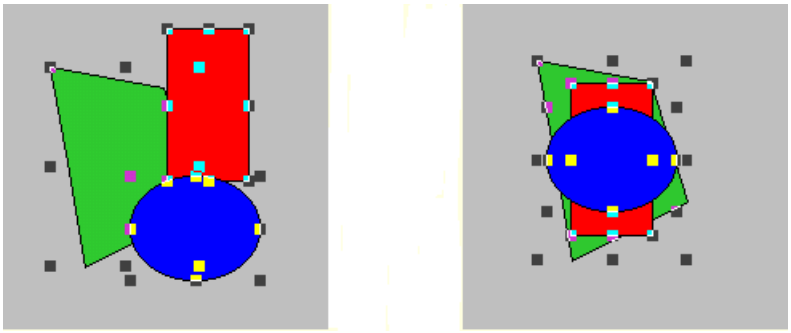


图 2-44　向中心对齐

2.4.4　垂直等距

选择“垂直等距”菜单，使所有选中的图素在垂直方向上图素之间的距离相等。计算图素之间的距离是指各图素中心之间的距离。

垂直等距示例如图 2-45 所示：

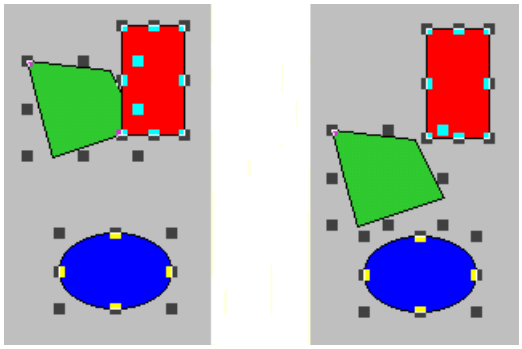


图 2-45 垂直等距

2.4.5　水平等距

选择“水平等距”菜单，使所有选中的图素在水平方向上图素之间的距离相等。计算图素之间的距离是指各图素中心之间的距离。

水平等距示例如图 2-46 所示：

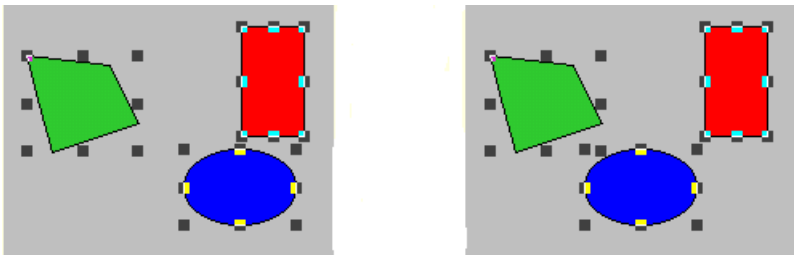


图 2-46　水平等距

2.4.6　垂直镜像

选择“垂直镜像”菜单，使所有选中的图素在垂直方向上，以所有选中图素在垂直方向上的中心线作为镜面，进行垂直方向镜像。

垂直镜像示例如图 2-47 所示：

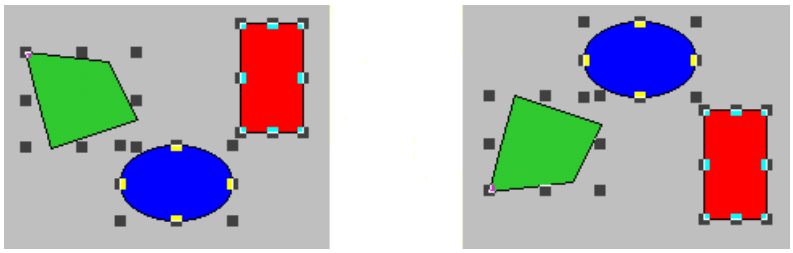


图 2-47　垂直镜像

2.4.7　水平镜像

选择“水平镜像”菜单，使所有选中的图素在水平方向上，以所有选中图素在水平方向上的中心线作为镜面，进行水平方向镜像。

水平镜像示例如图 2-48 所示：

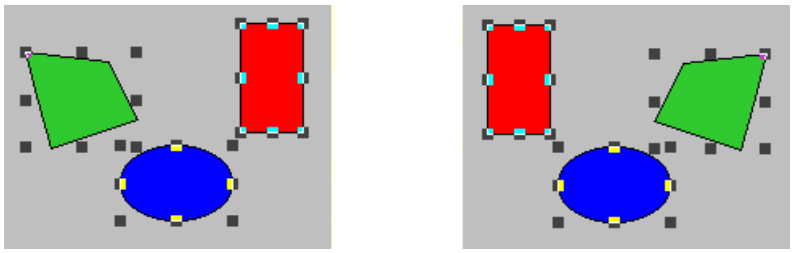


图 2-48　水平镜像

2.4.8　顺时针旋转

选择“顺时针旋转”菜单，使所有选中的图素绕其中心点顺时针方向旋转 90 度。

顺时针旋转示例如图 2-49 所示：

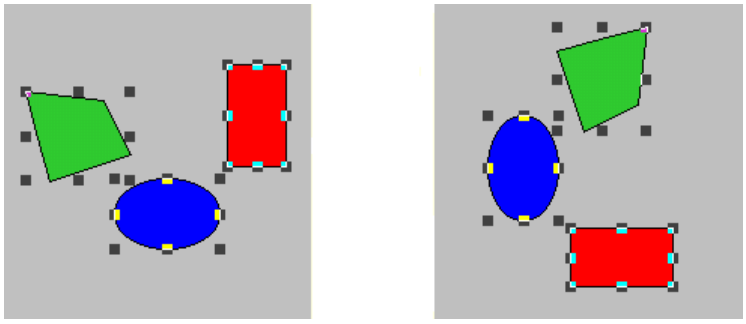


图 2-49　顺时针旋转

2.4.9　逆时针旋转

选择“逆时针旋转”菜单，使所有选中的图素绕其中心点逆时针方向旋转 90 度。

逆时针旋转示例如图 2-50 所示：

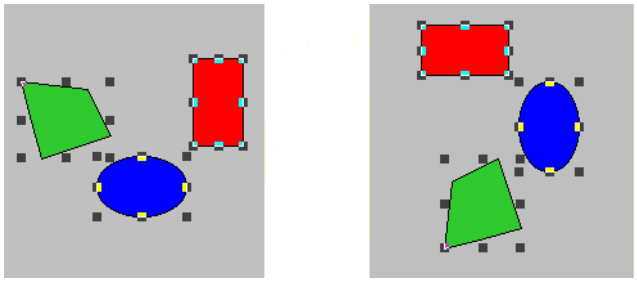


图 2-50　逆时针旋转

2.4.10　合成单元

选择“合成单元”菜单，使所有选中的图素合成一个单元。

注意：在为合成单元而选中的所有图素中不能有已经经过合成单元的图素，否则不能再成单元。

合成单元示例如图 2-51 所示：

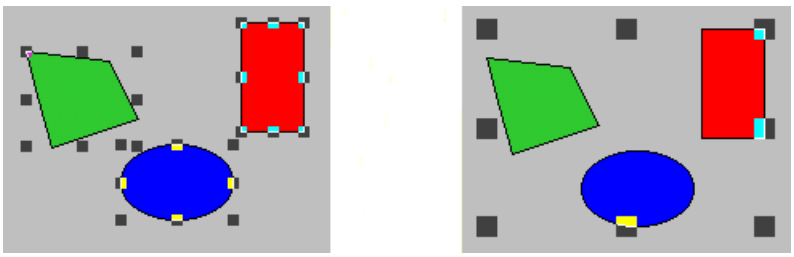


图 2-51 合成单元

2.4.11 分裂单元

选择“分裂单元”菜单，将一个选中的单元分裂成图素。

分裂单元示例如图 2-52 所示：

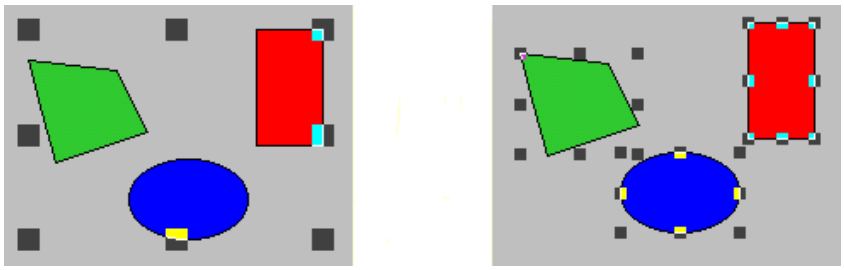


图 2-52 分裂单元

2.5 图形文本

图形文本包括字体、加粗、斜体、下划线、字体缩小、字体放大、居左、居中、居右等功能。

2.5.1 字体

选择“字体”菜单，设定选中文本或按钮文本的显示字体。世纪星中的字体包括操作系统（Windows98/2000/NT）中的所有字体，选择字体的对话框如图 2-53 所示。当然，用户亦可在文字工具栏中进行字体、字体样式及大小的设定，如图 2-54 所示。字体、字号的选择方式如同其他文字编辑软件（如 Word）的操作方法完全相同。

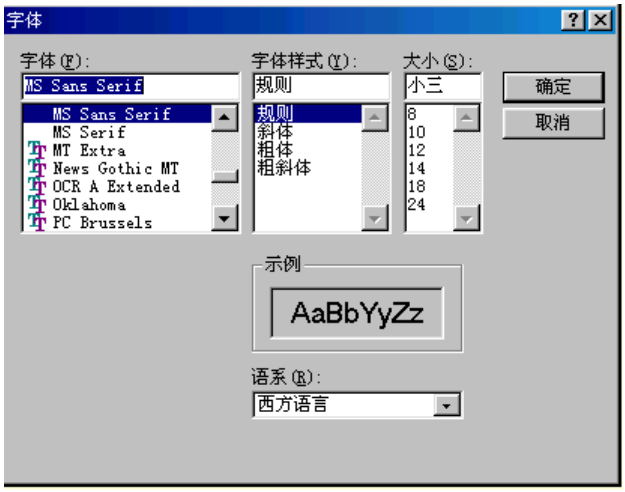


图 2-53 字体对话框

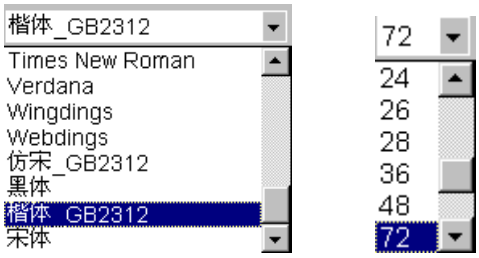


图 2-54

2.5.2　加粗

选择“加粗”菜单，使选中的文本或按钮文本加粗显示。

加粗示例如图 2-55 所示：

图 2-55 加粗

2.5.3 斜体

选择“斜体”菜单，使选中的文本或按钮文本斜体显示。

斜体示例如图 2-56 所示：

图 2-56 斜体

2.5.4 下划线

选择“下划线”菜单，使选中的文本或按钮文本下划线显示。

下划线示例如图 2-57 所示：

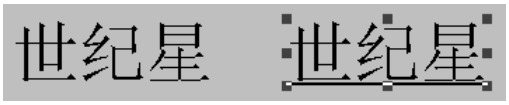


图 2-57 下划线

★世纪星监控组态软件系统规定：在字号大于 72 时，文本不再有下划线。

2.5.5 字体缩小

选择“字体缩小”菜单，使选中的文本或按钮文本缩小一号字体。

字体缩小示例如图 2-58 所示：

图 2-58　字体缩小

2.5.6　字体放大

选择“字体放大”菜单，使选中的文本或按钮文本放大一号字体。

字体放大示例如图 2-59 所示：

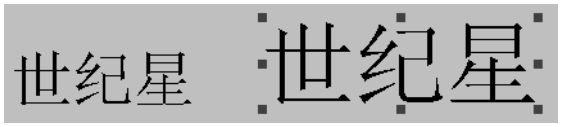


图 2-59　字体放大

2.5.7　居左

选择“居左”菜单，使选中的文本或按钮文本在运行时居左显示。

2.5.8　居中

选择“居中”菜单，使选中的文本或按钮文本在运行时居中显示。

2.5.9　居右

选择“居右”菜单，使选中的文本或按钮文本在运行时居右显示。

2.6 窗口

在本菜单下方列出“画面窗口特性”以及已经打开的画面名称，选取其中的一项可激活该画面，使之显示在屏幕上。当前画面的左边有“√”号。如图 2-60 所示。

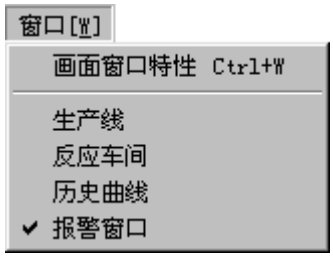


图 2-60

选择“画面窗口特性”菜单命令，则弹出如图 2-61 所示的画面特性窗口：

图 2-61

画面特性参数设置如下：

名　称：画面的名称，画面保存时就以此名称作为文件名。画面名最长为 32 字符，画面名中不能包含下列字符：　\ / : * ? ” < > |

背景色：用于设置窗口的背景色，用鼠标按下背景色区域，弹出浮动的调色板窗口，可从中选择一种颜色。

注　释：与画面有关的注释信息。

边　框：画面创建时边框的类型。

类　型：有三种画面类型可供选择。“覆盖式”画面出现时，它重叠在当前画面之上。关闭后被覆盖的画面又可见。“替换式”画面出现时，所有与之相交的画面自动从屏幕上和内存中删除。建议尽量使用“替换式”画面以节约内存。“弹出式”画面类似于“覆盖式”画面，一般用于对话框或信息显示。

位　置：画面的左上角坐标、宽度和高度的值，以象素为单位计算。

选项

标 题 杆：此选择用于决定画面是否有标题杆，若有标题杆，标题杆上将显示画面名称。

大小可变：此选择用于决定画面是否能由用户改变大小。改变画面大小的操作与改变 WINDOWS 窗口相同。

2.7　帮助

选中此选项可查看系统帮助文件。如图 2-62 所示。

世纪星拥有丰富的在线帮助系统，当您遇到问题时，您可以随时查阅帮助系统。

世纪星的帮助系统中，详细说明了世纪星的特点、系统的组成。对于所有的菜单命令、工具箱都有详细的说明。并有专门的章节详细论述图库、报警事件、历史趋势曲线、控件等专门问题。

帮助系统中对世纪星与硬件系统的连接问题，给予了特别的重视。详细说明了各硬件设备的特点、与世纪星连接时的设置，并且有详细的硬件连接图，方便您的使用。

帮助系统采用 Windows 98/2000/NT 的通用标准，使用方便灵活。

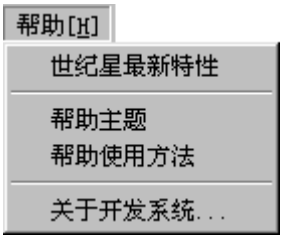


图 2-62

第三章 变量数据库

世纪星提供的变量数据库是一个实时数据库。在运行系统中，数据库中保存的是所有变量的实时数据。运行系统将数据库中的数据同用户输入的数据以及工业现场传送来的数据进行实时处理，再将数据送回工业现场，同时更新变量数据库中变量的实时数据。

3.1 变量类型

在变量数据库中定义变量时，用户必须为每个变量定义一种数据类型，这样用户才能使用该变量。本系统提供四大类型变量：系统变量、内存变量、IO 变量和特殊变量。

3.1.1 系统变量

系统变量是系统预先设置的变量，这些变量用户可以直接使用。系统变量又分为系统离散、系统整数、系统实数、系统信息和系统报警组变量。系统变量设有只读和读写属性，只读变量如系统时间等，由系统自动更新，用户不能改变这些变量的数值；对于具有读写属性的系统变量，用户可以改变变量的数值。

3.1.2 内存变量

内存变量是用户定义在系统内部的变量，这些变量不同外部进行数据交换，主要存放计算处理的中间值，以及在系统仿真时模拟 IO 变量。

内存变量又分为内存离散变量、内存整数变量、内存实数变量和内存信息变量四种。

(1)、内存离散变量

布尔（BOOL）型变量。
内存离散变量的值只能为：0（关，OFF，FALSE）和 1（开、ON，TRUE）两种，一般用于表示开关量。

(2)、内存整数变量

32 位整数变量。
内存整数变量的取值范围：-2,147,483,648 ～ 2,147,483,647。

(3)、内存实数变量

十进制浮点型变量。
内存实数的取值范围：-3.4e 38 ～ 3.4e 38。

(4)、内存信息变量

字符串变量。
内存信息变量的取值范围：最长能存储 131 个字符。

3.1.3 IO 变量

IO 变量是能其他应用程序进行数据交换的变量。本系统的 IO 变量能以多种数据交换协议同外部应用程序进行数据交换，如 Windows 动态数据交换（DDE）协议、OPC、网络、串口、总线、板卡等通讯协议。

具有读写属性的 IO 变量数据变化时，系统立即将 IO 变量的值写到外部应用程序。IO 变量的值也可以由外部应用程序更新。

IO 变量又分为 IO 离散变量、IO 整数变量、IO 实数变量、IO 信息变量四种。

(1)、IO 离散变量

布尔（BOOL）型变量。
IO 离散变量的值只能为：0（关，OFF，FALSE）和 1（开、ON，TRUE）两种，一般用于表示开关量。

(2)、IO 整数变量

32 位整数变量。
IO 整数变量的取值范围：-2,147,483,648 ～ 2,147,483,647。

(3)、IO 实数变量

十进制浮点型变量。
IO 实数变量的取值范围：-3.4e 38 ～ 3.4e 38。

(4)、IO 信息变量

字符串变量。
IO 信息变量的取值范围：最长能存储 131 个字符。

3.1.4　特殊变量

特殊变量类型有报警窗口变量、历史曲线变量两种。主要用于系统报警显示和历史趋势曲线显示。

(1)、报警窗口变量

报警窗口变量主要用来将用户绘制的报警窗口同数据库连接，达到显示报警的目的。用户可以用命令语言来设置或改变报警窗口的一些特性，如上、下翻页等。

报警组的定义是采用树型结构，系统已经预定义了名为“$系统”的报警组根

节点。

(2)、历史曲线变量

历史曲线变量将用户绘制的历史趋势曲线同数据库连接，达到显示历史趋势数据的目的。用户可以用命令语言编制程序来设置或改变历史趋势曲线的一些特性，如改变历史趋势曲线的起始时间或显示的时间长度等。

3.2 变量的域

变量的域：反映变量具有的属性。如实数变量的报警具有“高报警限”、“低报警限”等属性，历史曲线变量具有曲线起始时间、曲线时间长度等属性。

在定义变量时，同时需要设置变量的域值。

用户可以用命令语言编制程序来读取或设置变量的域，变量的域具有只读和读写两种类型。

变量的域的表示方法：变量．域。

3.2.1 离散变量的域

离散变量具有以下的域：

.Ack： 变量报警是否被应答，离散型，读写型。
.Alarm： 变量是否允许报警，离散型，只读型。
.AlarmEnabled： 变量是否有报警，离散型，只读型。
.Comment： 变量数据库中变量的注释，离散型变量，读写型。
.Group： 变量所属的报警组名，信息型变量，读写型。
.Name: 以信息类型描述变量名，字符串型，只读型。
.OffMsg: 离散值为关时信息，信息型，读写型。

.OnMsg:　离散值为开时信息，信息型，读写型。
.Priority：　变量的报警优先级，整数型，读写型。

3.2.2　整数、实数变量的域

整数、实数变量具有以下的域：

.Ack：　变量报警是否被应答，离散型，读写型。
.Alarm：　变量是否有报警，离散型，只读型。
.AlarmEnabled：　变量是否允许报警，离散型，只读型。
.Comment:　变量数据库中变量的注释，信息变量，读写型。
.DevTarget:　大小偏差报警的目标值，模拟型，读写型。
.EngUnits:　工程单位，信息变量，读写型。
.Group：　变量所属的报警组名，信息变量，读写型。
.HiHiLimit：　变量高高报警限，模拟型，读写型。
.HiHiStatus：　变量高高报警状态，离散型，只读型。
.HiLimit：　变量高报警限，模拟型，读写型。
.HiStatus：　变量高报警状态，离散型，只读型。
.LoLimit：　低报警限，模拟型，读写型。
.LoLoLimit：　低低报警限，模拟型，读写型。
.LoLoStatus：　低低报警状态，离散型，只读型。
.LoStatus：　低报警状态，离散型，只读型。
.MajorDevPct：　大偏差报警限，模拟型，读写型。
.MajorDevStatus：　大偏差报警状态，离散型，只读型。
.MaxRaw：　最大原始值，I/O 整数、实数型，只读型。
.MaxValue：　变量最大值，实数型，只读型。
.MinRaw：　最小原始值，I/O 整数、实数型，只读型。
.MinValue：　变量最小值，实数型，只读型。
.MinorDevPct：　小偏差报警限，模拟型，读写型。
.MinorDevStatus：　小偏差报警状态，离散型，只读型。
.Name:　以信息类型描述变量名,信息型，只读型。
.Priority：　变量的报警优先级，整数型，读写型。

.RocPct： 变化率报警限，模拟型，读写型。
.RocStatus： 变化率报警状态，离散型，只读型。

3.2.3 信息变量的域

.Comment: 变量数据库中变量的注释，信息变量，读写型。
.Name: 以信息类型描述变量名,信息型，只读型。

3.2.4 报警窗口的域

.Comment: 变量数据库中变量的注释，信息变量，读写型。
.Name: 以信息类型描述变量名,信息型，只读型。
.NextPage: 报警窗口后一页，报警组变量，读写离散型。
.PrevPage： 报警窗口前一页，报警组变量，读写离散型。
.PriFrom： 报警窗口显示的最小报警优先级，报警组变量，只读整数型。
.PriTo： 报警窗口显示的最大报警优先级，报警组变量，只读整数型。
.TotalPage： 报警窗口显示最大页数，报警组变量，只读整数型。

3.2.5 历史曲线的域

历史曲线变量具有以下的域：

.ChartLength： 历史趋势曲线的时间长度，整数型，读写型。
.ChartStart： 历史趋势曲线的起始时间，整数型，读写型。
.Comment: 变量数据库中变量的注释，信息变量，读写型。
.DisplayMode: 历史趋势曲线显示数值方式，模拟型，读写型。
.MaxRange： 历史趋势曲线显示的最大值百分数，实数型，读写型。
.MinRange： 历史趋势曲线显示的最小值百分数，实数型，读写型。
.Name: 以信息类型描述变量名,信息型，只读型。
.Pen1，.Pen2，
.Pen3，.Pen4，
.Pen5，.Pen6，

.Pen7，.Pen8：　　历史趋势曲线显示的变量，变量名，读写型。
.ScooterPosLeft：　　左指示器的位置，实数型，读写型。
.ScooterPosRight：　　右指示器的位置，实数型，读写型。

3.3　变量数据库

选择“系统→变量数据库”菜单，或选择浏览器中的“变量字典”项，弹出变量数据库管理对话框如图 3-1 所示：

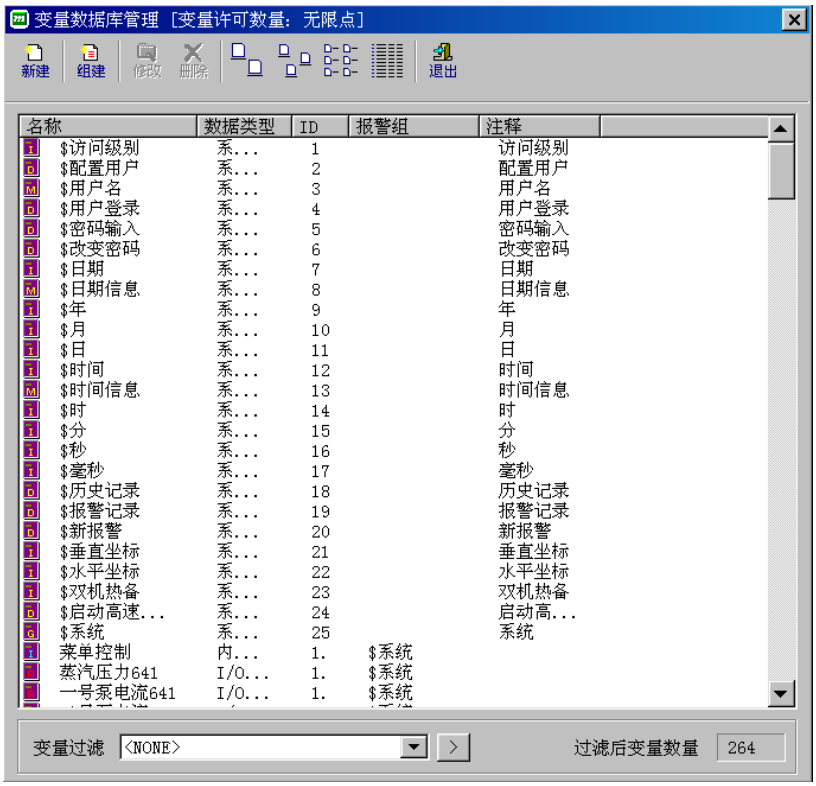


图 3-1

变量数据库管理的对话框说明如下：

新　　建：进入变量数据库对话框(图 3-2)，创建新变量。

组　　建：进入成组创建变量对话框(图 3-3)，成组创建变量

修　　改：进入变量数据库对话框，修改新变量。用鼠标左键双击变量名也可进入修改状态。

删　　除：删除指定的变量，若变量为系统变量或已经被使用（变量计数大于零），比如此变量和某一个图素建立了动画连接或出现在命令语言中，则提示不能删除。

退　　出：退出“变量数据库”对话框。

变量过滤：按用户指定条件过滤变量数据库，进入变量过滤对话框。

过滤后变量数量：按用户指定的条件过滤后变量数据库中变量的数量，默认条件下是指没有经过过滤的变量数据库中变量的数量。

变量数据库的对话框中包含变量的如下参数：

变量定义

变量名：变量在变量数据库中的名称，同一应用程序中数据变量不能重名，数据变量名区分大小写，最长不能超过 32 个字符。用鼠标单击编辑框的任何位置进入编辑状态，用户此时可以输入变量名字，变量名可以是汉字或英文名字或$、@、#、_、？，但第一个字符不能是数字且变量名不能是命令语言的关键字或函数名称。

类　型：在对话框中只能定义以下基本类型中的一种：内存离散、内存整数、内存实数、内存信息、IO 离散、IO 整数、IO 实数、IO 信息、历史曲线、报警窗口。

描　述：变量的注释信息。

报警组名：是指变量所属的报警组，定义报警组是通过菜单命令“系统→报警组定义”完成的。

只读、只写、读写属性：表明变量的读写属性。

数据设定

初始值：系统开始运行时变量的初始值。初始值与所定义的变量类型有关，定义整数或实数类型变量时可以输入一个整数或实数；定义离散类型变量时可以选择开或关；定义信息类型变量时可以输入字符串。

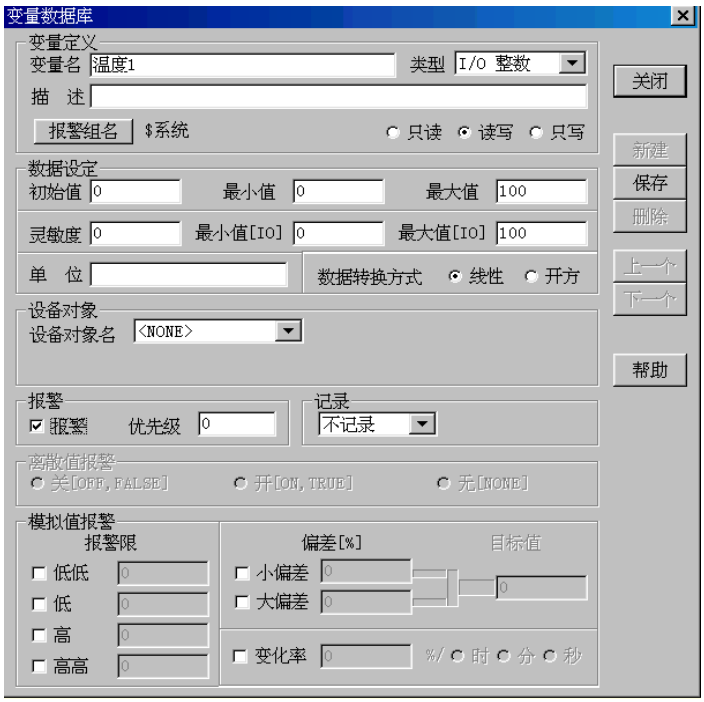


图 3-2

最小值：变量在数据库中的最小值。

最大值：变量在数据库中的最大值。

灵敏度：变量的数据类型为整数类型、实数类型时此项有效。只有当该数据变量的值变化幅度超过“数据变化灵敏度”时，系统才更新与之相连接的图素。

最小值[I/O]：I/O 变量的最小原始值。

最大值[I/O]：I/O 变量的最大原始值。

单　　位：变量的工程单位。

数据转换方式：规定 IO 整数和 IO 实数对应的原始数值到数据库使用数值值的转换方式。

线　　性：IO 整数和 IO 实数对应的原始数值和数据库使用值用线性插值进行转换。

开　　方：用 IO 整数和 IO 实数对应的原始数值的平方根进行转换。

设备对象

设备对象名：选择已安装的驱动设备，配置与所定义变量有关的参数。

数据 类型：选择从设备对象中采集得到的数据类型。

采集 频率：选择从设备对象中采集数据的频率。

寄　存 器：选择设备对象中数据采集的寄存器。

报警参数

报　警：变量是否具有报警功能。选择此项下面的离散报警（对离散类型变量）或报警参数（对整数类型、实数类型变量）才有效。

优先级：是报警事件重要程度的度量，数字 1 的级别最高，999 为最低级别，给每个要监视的变量规定一个报警优先级可以分层次管理报警事件。

记录参数

记　录：当变量值发生变化时，是否将此时的变量值存放到磁盘上（即：历史记录），具有离散类型、整数类型和实数类型的变量可以记录。包括不记录、按灵敏度记录、定时记录等三种选择。

灵敏度：只有变量值的变化幅度大于此值时才被记录到磁盘上。当“是否记录”选项有效时，“记录变化灵敏度”选项才有效，其缺省值为 1，用户可根据需要修改。

示例：

数据库中有一个模拟型变量，如果要将此变量的值记录到硬盘上，而且规定其记录变化灵敏度为 3，则其记录过程如下所示：如果第一次记录值是 10，当第二次的变量值为 12 时，由于 12－10=2<3，也就是第二次变量值相对第一次记录值的变化小于设定的“记录变化灵敏度”，所以第二次变量值不记录到硬盘上；当第三次变量值为 14 时，由于 14－10=4>3，即变化幅度大于设定的“记录变化灵敏度”，所以此次温度值记录到磁盘上。

间隔：定时存储的时间间隔，范围：1～3153600 秒.

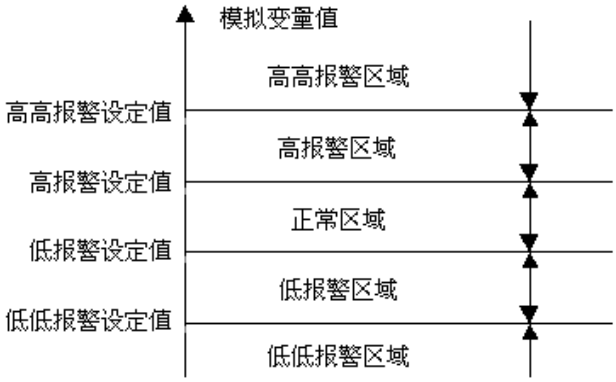
离散值报警

是对离散量的开或关进行报警。

模拟值报警

只对变量类型为整数类型、实数类型的变量有效。有 7 种报警：低低、低、高、高高、小偏差、大偏差、变化率。

报警限：变量报警的上下限，整数类型、实数类型的变量有四种报警可选项：低低、低、高、高高。在低低和低报警的情况下，每当变量值小于报警下限时就出现报警；在高高和高报警的情况下，每当变量值大于报警上限值时就出现报警。正常区域和报警区域的确定要根据低低，低，高，高高这四种报警设定值来进行。其确定示意图如下所示：



帮助：在定义变量时，可以不退出定义变量对话框而进入帮助系统。

示例：设定某锅炉控制温度在高高报警值=150，高报警值=130，低报警值=50，低低报警值=30，则当锅炉温度变化时会出现以下控制情况：

锅炉温度>=150　　高高报警

130=<锅炉温度<150高报警

50<锅炉温度<130　正常工作状态
30<锅炉温度<=50　低报警
锅炉温度<=30 低低报警

偏差：是经模拟量相对目标值（基准值）上下波动的百分比来定义，有小偏差和大偏差两种报警条件，目标值（基准值）由后面编辑框输入。当波动的百分比小于小偏差或大于大偏差时，分别出现报警。

偏差=((当前值－目标值)÷(最大值－最小值))×100

由于偏差有正负，在偏差范围内相对目标值（基准值）上下波动的模拟量的最小分界值称为最小当前值，相对目标值（基准值）上下波动的模拟量的最大分界值称为最大当前值，则有：

最小当前值=目标值-(偏差/100)×(最大值－最小值)；
最大当前值=目标值+(偏差/100)×(最大值－最小值)；

示例：

某变量的最小值=－1000，最大值=1000，设定其小偏差=10，大偏差=15，目标值=500，则可计算出小偏差报警和大偏差报警的条件如下：

小偏差报警：
最小当前值=500－(10/100)×(1000－(－1000))=500－200=300
最大当前值=500+(10/100)×(1000－(－1000))=500+200=700
则：模拟变量值>=700 或模拟变量值<=300 时，出现小偏差报警
300<模拟变量值<700 时，为正常工作范围

大偏差报警：
最小当前值=500－(15/100)×(1000－(－1000))=500－300=200
最大当前值=500+(15/100)×(1000－(－1000))=500+300=800
则：模拟变量值>=800 或模拟变量值<=200 时，出现大偏差报警

200<模拟变量值<800 时，为正常工作范围

变化率：选择模拟量在给定时间内的变化率报警。每当变量发生变化时，就计算变化率以决定是否报警。若计算结果大于报警限值，则出现报警。时，分，秒按钮用于选择时间单位。单位的选择不影响检查变化率的时间，每次变量变化时，就计算变化率。变化率的计算公式如下：

变化率=((当前值－上一次值)/(最大值－最小值)/(当前时间－上一次值变化的时间))×100%

变量操作键

新建： 单击此按钮后“变量名”编辑栏加亮，而且上一次在编辑栏中输入的有效变量名保留下来供用户编辑，这样当变量名相近时，用户只需要改少量的部分就可完成，以提高创建变量的速度。用户也可直接输入新的变量名。

保存： 单击此按钮后保存新建的变量名。若变量名不合法，会弹出提示对话框提醒用户修改变量名。

删除： 单击此按钮后删除指定的变量，若变量正被使用（变量计数大于零），比如此变量和某一个图素建立了动画连接或出现在命令语言中，则此按钮会变成灰色，表示不可删除此变量。

上一个：用于查看当前变量的前一个变量。

下一个：用于查看当前变量的后一个变量。

浏　览：单击此按钮后弹出列表框，列出了数据词典里的所有变量的简要信息，包括变量名，序号，变量类型。在列表框中单击任一变量，在“变量定义”对话框中显示此变量的详细信息。

关　闭：单击此按钮后关闭“变量定义”对话框，若有未保存的变量会提示保存。

成组创建变量对话框

单击变量数据库管理对话框中的“组建”按钮，进入成组创建变量对话框，如图 3-3 所示：

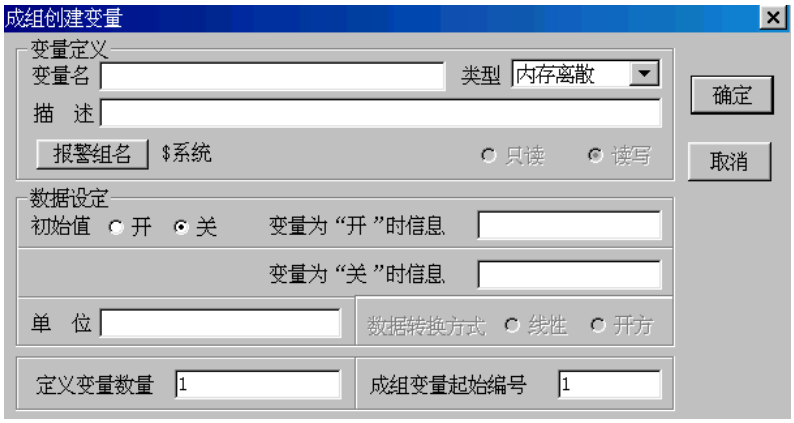


图 3-3

成组创建变量与新建一个变量基本类似，有以下几点需要注意：

1、成组创建变量中的“变量名”是指此次所创建的变量都是以该变量名开头，后面接数字作为变量名。
2、“定义变量数量”是指您此次想要定义的变量数量，范围为：1～1000。
3、“成组变量起始编号”是指要成组定义变量的起始编号。范围为：1～9999。

例如：在成组创建变量中“变量名”输入“IO”,“定义变量数量”中输入 10，“成组变量起始编号”中输入 1，其他不选择或输入，单击“确定”后会在变量数据库中定义好 IO1～IO10 十个变量。

变量过滤对话框

单击变量数据库管理对话框中的 > 按钮，进入变量过滤对话框，如图 3-4 所示：

图 3-4

过滤名称：在此栏中可输入一个新过滤名称，也可选择已有的过滤名称

过滤条件：这里有变量名称、变量类型、设备对象、报警组、注释等 5 个条件任意组合作为过滤的条件。

您还可删除您认为不需要的过滤名称。

3.4 数据库维护

数据库维护在大部分工作都是由系统自动完成的，设计者需要做的是在完成最后阶段“删除未用变量”。在删除未用变量之前需要更新变量计数，目的是确定变量是否有动画连接或是否在命令语言中使用过，只有没使用过（变量计数=0）的变量才可以删除。更新变量计数之前要求关闭所有画面。

3.4.1　更新变量计数

选取菜单“系统→更新变量计数”，系统提示如图 3-5 所示：

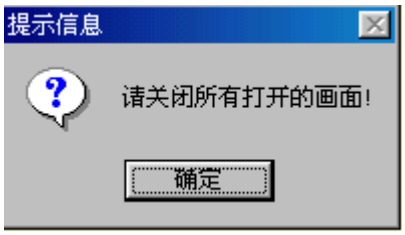


图 3-5

单击“确定”按钮，系统关闭所有画面，统计变量使用情况。如图 3-6 所示：

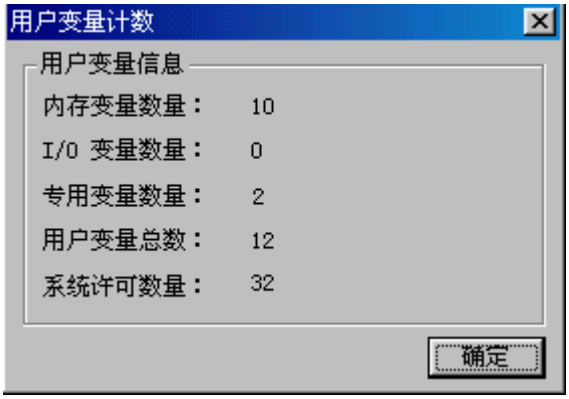


图 3-6

3.4.2 删除未用变量

检查完毕后，选取菜单“系统→删除未用变量”，如果没有未使用的变量，系统提示图 3-7 所示：

图 3-7

否则列出所有未使用的变量如图 3-8 所示：

图 3-8

在对话框中选择未用变量，然后单击“删除”按钮删除被选项，删除完毕，单击“关闭”按钮完成操作

第四章　动画连接

动画连接就是建立画面的图素与数据库变量的对应关系。这样，工业现场的数据，比如温度、液面高度等，当它们变化时，通过 I/O 接口，将引起实时数据库中变量的变化，如果设计者曾经定义了一个画面图素（比如指针）与这个变量相关，我们将会看到指针在同步偏转。动画连接的引入是设计者人机接口的一次突破，它把程序员从重复的图形编程中解放出来，为程序员提供了标准的工业控制图形界面，并且有可编程的命令语言连接来增强图形界面的功能。图形对象与变量之间有丰富的连接类型，给程序员设计图形界面提供了极大的方便。

图形对象可以按动画连接的要求改变颜色、尺寸、位置、填充百分数等，一个图形对象又可以同时定义多个连接。把这些动画连接组合起来，应用软件将呈现出令人难以想象的图形动画效果。

4.1　动画连接对话框

给图形对象定义动画连接是在连接定义对话框中进行的。在 CSMAKER 中双击图形对象（不能有多个图形对象同时被选中），弹出动画连接对话框如图 4-1 所示。不同类型的图形对象弹出的对话框大致相同。对于特定对象，有些按钮变成灰色表明此动画连接不适应于该对象，或者该对象定义了与此动画连接不相容的其他动画连接。

对话框的第一行标识出被连接对象的名称。

下面分组介绍所有的动画连接种类。

图 4-1

用户输入：所有的图形对象都可以定义为三种用户输入连接中的一种，输入连接使被连接对象在运行时成为触敏对象。当CSVIEWER运行时，触敏对象周围出现反显的矩形框，可由鼠标或键盘选中此触敏对象。按 SPACE 键、ENTER 键或鼠标左键，会弹出输入对话框，可以从键盘键入数据以改变数据库中变量的值。

直线属性：直线属性连接规定了图形对象的颜色、线型属性如何随变量或连接表达式的值变化而变化。线类型的图形对象和填充形状的图形对象可定义直线属性连接。

填充属性：填充属性连接规定了图形对象的填充类型属性如何随变量或连接表达式的值变化而变化。填充形状的图形对象可定义填充属性连接。

文本属性：文本属性连接规定了图形对象的文本属性如何随变量或连接表达

式的值变化而变化。文本对象可定义文本色连接。

滑 动 杆：所有的图形对象都可以定义两种滑动杆输入连接中的一种，滑动杆输入连接使被连接对象周围出现反显的矩形框。鼠标左键拖动有滑动杆输入连接的图形对象可以改变数据库中变量的值。

大小改变：这两种连接（高度、宽度）规定了图形对象如何随变量值的变化而改变大小。不是所有的图形对象都能定义这两种连接。单击任一按钮弹出相应的连接对话框。

移　　动：这两种连接（水平、垂直）规定了图形对象如何随变量值的移动而改变位置。不是所有的图形对象都能定义这两种连接。单击任一按钮弹出相应的连接对话框。

百分比填充：这两种连接（水平、垂直）规定了图形对象如何随变量值的变化而改变填充百分比。不是所有的图形对象都能定义这两种连接。单击任一按钮弹出相应的连接对话框。

按　　键：这两种连接（离散值、命令语言）规定了图形对象如何随按键的改变而发生变化。不是所有的图形对象都能定义这两种连接。单击任一按钮弹出相应的连接对话框。

数值显示：只有文本图形对象能定义三种值输出连接中的某一种。这种连接用来在画面上输出文本图形对象的连接表达式的值。运行时文本字符串将被连接表达式的值所替换，输出的字符串的大小、字体和文本对象相同。按动任一按钮弹出相应的输出连接对话框。

访问权限：此编辑框用于输入被连接图形对象的访问权限级别。当软件在 CSVIEWER 中运行时，只有权限级别不小于此值的操作员才能访问它，这是世纪星保障系统安全的一个重要功能。

特殊功能：具有可见/隐含、闪烁、旋转等三种连接，这是三种规定图形对象

可见性的连接。按动任一按钮弹出相应连接对话框。

4.2 连接表达式

连接表达式是定义动画连接的主要内容，因为连接表达式的值决定了画面上图素的动画效果。表达式由数据字典中定义的变量、变量的域或报警组名，数值常量以及各种运算符组成，与 C 语言中的表达式非常类似。

在连接表达式中不允许有赋值语句以及非计算函数，表达式的值在“世纪星监控组态软件”运行时计算。变量名和报警组名可以直接出现在表达式中，不必加引号，但区分大小写。

表达式中的算符

－	取反，将正数变为负数（单目）。
～	取补码，将整型变量变成 " 2 " 的补码。
!	逻辑非
*	乘法
**	乘方
/	除法
%	模运算
+	加法
－	减法（双目）
&	整型量按位与
|	整型量按位或
^	整型量异或
&&	逻辑与
||	逻辑或
＜	小于
＞	大于
＜＝	小于或等于
＞＝	大于或等于
＝＝	等于

◇	不等于
＝	赋值
（）	括号，保证运算按所需次序进行。
{}	大括号，用于注释；必须成对使用。

算符的优先级

下面列出算符的运算次序，首先计算最高优先级的算符，再依次计算较低优先级的算符。同一行的算符具有相同的优先级。

（）　　最高优先级
－（单目），！，～
＊，／，％
＋，－
＜，＞，＜＝，＞＝，＝＝，◇
＆，｜，＾
＆＆，｜｜
＝　　最低优先级

表达式举例：

单独的变量或变量的域：　开关、液面高度.alarm

复杂的表达式：　开关==1、（液面高度＞50）＆＆（液面高度＜8）、
（开关 1｜｜开关 2）＆＆（液面高度.alarm）

4.3　输入连接

输入连接主要包括用户输入连接、滑动杆输入连接、按键输入连接等功能。

4.3.1　用户输入连接

用户输入连接包括离散值输入连接、模拟值输入连接、信息值输入连接。所

有的图形对象都可以定义为三种用户输入连接中的一种，输入连接使被连接对象在运行时成为触敏对象。当 CSVIEWER 运行时，触敏对象周围出现反显的矩形框，可由鼠标或键盘选中此触敏对象。按 SPACE 键、ENTER 键或鼠标左键，会弹出输入对话框，可以从键盘键入数据以改变数据库中变量的值。

4.3.1.1　离散值输入连接

离散值输入连接是使被连接对象在运行时成为触敏对象，单击此对象后弹出对话框，可在对话框中输入离散值，以改变数据库中某个离散类型变量的值。

示例：

建立一个内存离散变量：“按钮”，当“按钮”=1 时，为“按下”状态，“按钮”=0 时，为“弹起”状态。

离散值输入连接的设置方法是：在“动画连接”对话框中单击“离散值输入”按钮，弹出对话框如图 4-2 所示。(在设计状态下。)

图 4-2

对话框中各项设置的意义如下：

离散变量

名　　称：可以输入合法的离散变量名称，单击右侧的“>”按钮可查看已定义的变量名。

提示信息：运行时出现在弹出对话框上用于提示输入内容的字符串。

离散值信息

变量为真时，输入信息：规定表达式为真时，被连接对象(文本)输入的内容。

变量为假时，输入信息：规定表达式为假时，被连接对象(文本)输入的内容。

激活键

定义激活键，这些激活键可以是键盘上的单键也可以是组合键（Ctrl，Shift　和键盘单键的组合)，在 CSVIEWER 运行画面时可以用激活键随时弹出输入对话框，以便输入修改新的离散值。

图 4-3 是在上述设置条件下在 CSVIEWER 中的运行状态。

图 4-3

4.3.1.2　模拟值输入连接

模拟值输入连接是使被连接对象在运行时成为触敏对象，单击此对象或按下指定的热键将弹出对话框，用户在对话框中可以输入连接变量的新值，以改变数据库中某个模拟型变量的值。

本例中建立一个矩形框，设置“模拟值输入”连接以改变变量“温度”的值。

模拟值输入连接的设置方法是：在“动画连接”对话框中单击“模拟值输入”按钮，弹出对话框如图 4-4 所示：

图 4-4

对话框中各项设置的意义如下：

模拟变量：

名　　称：要改变的模拟类型变量名称。单击右侧的“>”按钮可查看已定义的变量名。

提示信息：运行时出现在弹出对话框上用于提示输入内容的字符串。

模拟值范围：

规定键入值的范围。它应该是要改变的变量在数据库中的最大最小值。

激活键：定义激活键，这些激活键可以是键盘上的单键也可以是组合键(Ctrl，Shift 和键盘单键的组合)，在 CSVIEWER 运行画面时可以用激活键随时弹出输入对话框，以便输入修改新的模拟值。

当 Ctrl 和 Shift 字符左边的选择框中出现“√”符号时，分别表示 Ctrl 键和 Shift 键有效，单击“键”按钮，则弹出如 图 4-5 所示的对话框。在此对话框

中用户可以选择一个键，再单击“确定”按钮完成热键设置。

注：只有当 Ctrl 和（或）Shift 键选中时，激活键有效。

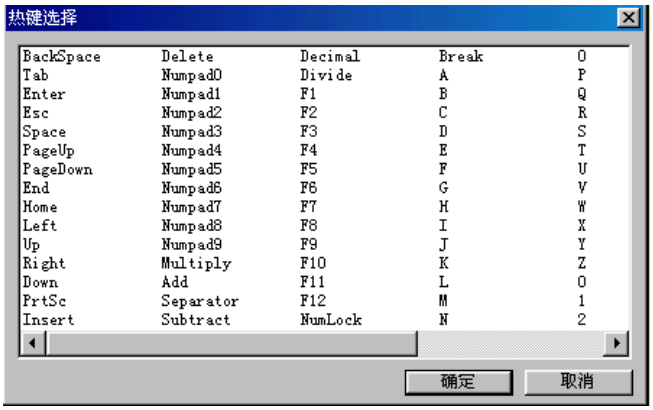


图 4-5

在运行时单击矩形框，弹出输入对话框。用户在此对话框中可以输入变量的新值。如果在 CSMAKER 中选中了菜单“数据库运行时使用模拟键盘”，程序运行中弹出输入对话框的同时还将显示模拟键盘窗口，在模拟键盘上单击按钮的效果与键盘输入相同。如图 4-6 所示：

4.3.1.3 信息值输入连接

信息值输入连接是使画面中文本对象的内容在程序运行时被数据库中的某个信息变量的值所取代。

示例：

建立一个内存信息变量，图 4-7 是设计状态，图 4-8 是在 CSVIEWER 中的运行状态。

字符串输入连接的设置方法是：在“动画连接”对话框中单击“字符串输入”按钮，弹出对话框如图 4-7 所示：

图 4-6

图 4-7

对话框中各项设置的意义如下：

信息变量

名　　称：输入要显示值内容的信息变量。单击右侧的“>”按钮可查看已定义的变量名。

提示信息：运行时出现在弹出对话框上用于提示输入内容的字符串。

密码输入方式

用户对信息变量的设置，可以通过密码输入方式完成。

激活键

定义激活键，这些激活键可以是键盘上的单键也可以是组合键（Ctrl，Shift 和键盘单键的组合），在 CSVIEWER 运行画面时可以用激活键随时弹出输入对话框，以便输入修改新的信息值。

注：只有当 Ctrl 和（或）Shift 键选中时，激活键有效。

图 4-8 为上述设置条件下在 CSVIEWER 中的运行状态。

图 4-8

4.3.2　滑动杆输入连接

所有的图形对象都可以定义垂直滑动杆输入或水平滑动杆输入两种滑动杆输入连接中的一种，滑动杆输入连接使被连接对象在运行时成为触敏对象。当CSVIEWER 运行时，此触敏对象周围出现反显的矩形框。鼠标左键拖动有滑动杆输入连接的图形对象可以改变数据库中变量的值。

4.3.2.1　水平滑动杆输入连接

当有水平滑动杆输入连接的图形对象被鼠标拖动时，与之连接的变量的值将会被改变。当变量的值改变时，图形对象的位置也会发生变化。经常使用滑动杆输入连接处理用户的输入。

水平滑动杆输入连接的设置方法是：在“动画连接”对话框中单击“水平滑动杆输入”按钮，

弹出如图 4-9 所示对话框：

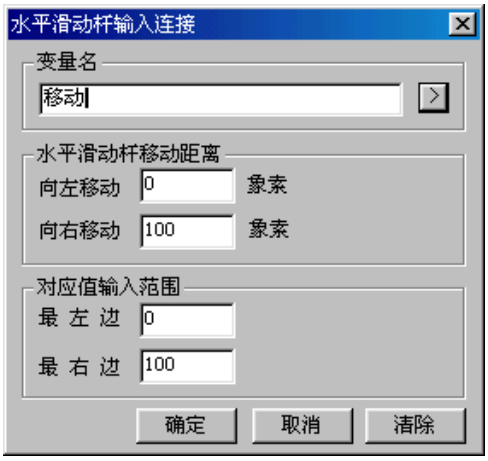


图 4-9

对话框中各项设置的意义如下：

变 量 名：与图形对象相联系的变量。

水平滑动杆移动距离

向左移动：图形对象从设计位置向左移动的最大距离。

向右移动：图形对象从设计位置向右移动的最大距离。

对应值输入范围

最 左 边：图形对象在最左端时对应变量的值。

最 右 边：图形对象在最右端时对应变量的值。

4.3.2.2 垂直滑动杆输入连接

垂直滑动杆输入连接与水平滑动杆输入连接类似，只是图形对象的移动方向不同。设置方法是：在“动画连接”对话框中单击“垂直滑动杆输入”按钮，弹出对话框如图 4-10 所示。

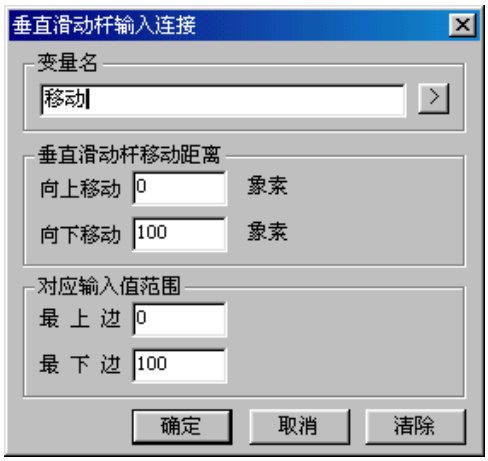


图 4-10

对话框中各项的意义解释如下：

变 量 名：与产生滑动输入的图形对象相联系的变量。单击“>”按钮查看所有已定义的变量名。

垂直滑动杆移动距离

向上移动：图形对象从设计位置向上移动的最大距离。

向下移动：图形对象从设计位置向下移动的最大距离。

对应值输入范围

最 上 边：图形对象在最上端时变量的值。

最 下 边：图形对象在最下端时变量的值。

4.3.3　按键输入连接

按键输入连接主要有按键离散输入和命令语言输入两种功能。

4.3.3.1　按键离散输入

选择“按键/ 离散值”有效，则弹出如图 4-11 所示的对话框，用户可进行按键离散值连接设置：

图 4-11

4.3.3.2　命令语言输入

选择“按键/ 命令语言”有效，则弹出如图 4-12 所示窗口，用户可进行按键命令语言连接设置：

命令语言连接会使被连接对象在运行时成为触敏对象。当 CSVIEWER 运行时，触敏对象周围出现反显的矩形框。命令语言有三种：“按下时”、“弹起时”和“按住时”，分别表示鼠标左键在该触敏对象上按下、弹起、按住时执行连接的命令语言程序。定义“按住时”的命令语言连接时，还可以指定按住鼠标后每隔多少毫秒执行一次命令语言，这个时间间隔在编辑框内输入。

可以指定一个等价键，操作者在键盘上用等价键代替鼠标，等价键的按下、弹起、按住三种状态分别等同于鼠标的按下、弹起、按住状态。单击任一种“命令语言连接”按钮，将弹出对话框用于输入命令语言连接程序。

在对话框右边有一些能产生提示信息的按钮，可让用户选择已定义的变量名及域，系统预定义函数名、画面窗口名、报警组名、算符、关键字等。还提供剪切、复制、粘贴、复原等编辑手段，使用户可以从其他命令语言连接中复制已编好的命令语言程序。详细内容参见第五章“命令语言”。

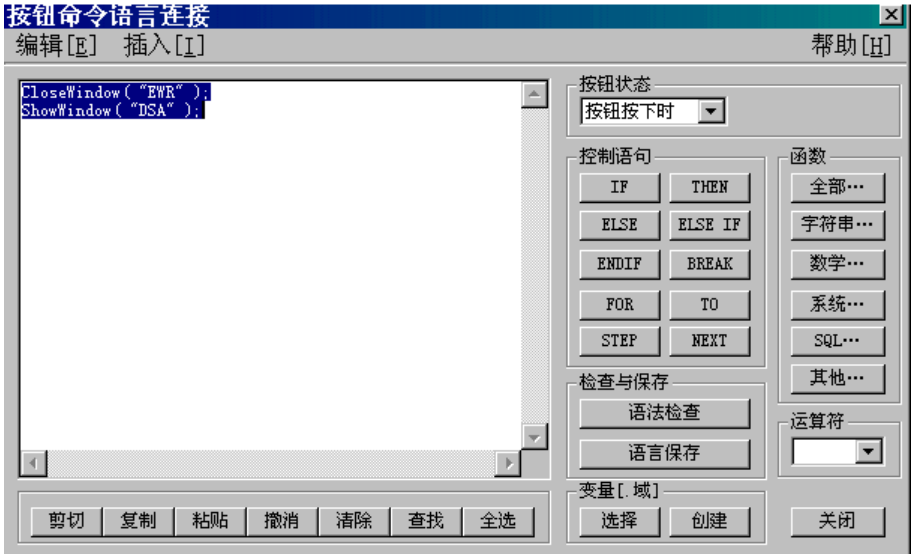


图 4-12

4.4　输出连接

输出连接主要包括直线属性连接、填充属性连接、文本属性连接、大小变化连接、移动连接、百分比填充连接、数值显示连接、特殊功能连接和访问权限设置等功能。

4.4.1　直线属性连接

在“动画连接”对话框中，单击“直线属性”按钮，弹出连接对话框。直线属性连接是使被连接对象的边框或线的颜色和线型随连接表达式的值而改变。定义这类连接需要同时定义分断点（断点值）和对应的直线属性。利用连接表达式的多样性，可以构造出许多很有用的连接。

例如：可以用线颜色表示离散变量 EXAM 的报警状态，只需在连接表达式中输入 EXAM.Alarm，然后把下面的两个笔属性颜色对应的值改为 0（蓝色），1（红色）即可。软件在运行时，当警报发生时（EXAM.Alarm==1），线就由蓝色变成了红色；当警报解除后，线的颜色又变为蓝色。

直线属性连接对话框中有四种设置方式：离散值，模拟值，离散报警和模拟报警。

离散值直线属性连接的各项设置的意义如下：（如图 4-13 所示）

图 4-13

表达式：用于输入连接表达式，单击“>”按钮可以查看已定义的变量名。
离散值颜色设定：根据表达式值的不同可以设定不同的颜色。
模拟值直线属性连接的各项设置的意义如下：（如图 4-14 所示）

图 4-14

表达式：用于输入连接表达式，单击“>”按钮可以查看已定义的变量名。
模拟值颜色设定：根据表达式值的不同可以在不同的断点上设定不同的颜色。

离散值报警直线属性连接的各项设置的意义如下：（如图 4-15 所示）

图 4-15

变量名：用于输入连接变量名，单击“>”按钮可以查看已定义的变量名。
离散值报警颜色设定：根据变量值的不同可以设定不同的颜色进行报警。
模拟值报警直线属性连接的各项设置的意义如下：（如图 4-16 所示）

图 4-16

变量名：用于输入连接变量名，单击“>”按钮可以查看已定义的变量名。
模拟值报警颜色设定：根据变量值的不同可以设定不同的颜色进行低低、低、正常、高、高高报警。

4.4.2　填充属性连接

填充属性连接使图形对象的填充颜色和填充类型随连接表达式的值而改变，通过定义一些分段点（包括断点值和对应填充属性），使图形对象的填充属性在一段数值内为指定值。

本例为封闭图形对象定义填充属性连接，断点值为 0 时填充属性为白色，断点值为 100 时为黄色，断点值为 200 时为红色。画面程序运行时，当变量“温度”的值在 0 至 100 之间时，图形对象为白色；在 100 至 200 之间时为黄色，变量值大于 200 时，图形对象为红色。

“填充属性”动画连接对话框中有四种设置方式：离散值，模拟值，离散报

警和模拟报警。

其具体的设置方法与直线属性连接完全一致，请参考相应的直线属性连接的内容。

4.4.3 文本属性连接

文本色连接是使文本对象的颜色随连接表达式的值而改变，通过定义一些分段点（包括颜色和对应数值），使文本颜色在特定数值段内为指定颜色。如定义某分段点，断点值是 0，文本色为红色，则当连接表达式的值在 0 到下一个断点值之间时，对象的文本色为红色。

文本色连接的设置对话框中有四种设置方式：离散值，模拟值，离散报警和模拟报警。

其具体的设置方法与直线属性连接完全一致，请参考相应的直线属性连接的内容。

4.4.4 大小变化的连接

大小改变连接包括：高度变化连接、宽度变化连接，这两种连接规定了图形对象如何随变量值的变化而改变大小。

4.4.4.1 高度变化连接

高度变化连接是使被连接对象在画面中随连接表达式值的改变而改变高度。其参数设置如图 4-17 所示：

4.4.4.2 宽度变化连接

宽度变化连接是使被连接对象在画面中随连接表达式值的改变而改变宽度。其参数设置如图 4-18 所示：

图 4-17

图 4-18

4.4.5　移动连接

移动连接包括：水平移动连接、垂直移动连接，这两种连接规定了图形对象如何随变量值的变化而改变位置。

4.4.5.1 水平移动连接

水平移动连接是使被连接对象在画面中随连接表达式值的改变而水平移动。移动距离以象素为单位，以被连接对象在画面制作系统中的原始位置为参考基准。水平移动连接常用来表示图形对象实际的水平运动。

水平移动连接的设置方法为：在“动画连接”对话框中单击“移动/水平”按钮，弹出“水平移动连接”对话框如图 4-19 所示：

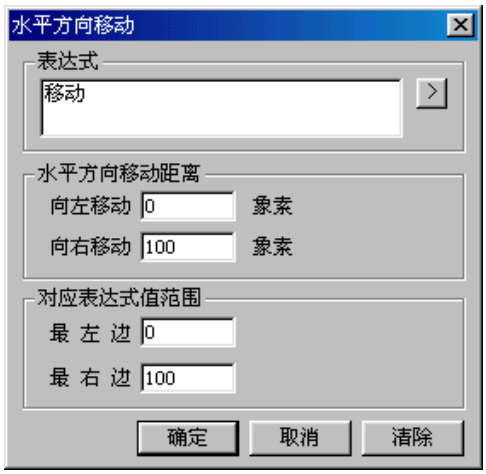


图 4-19

对话框中各项设置的意义如下：

表 达 式：在此编辑框内输入合法的连接表达式，单击“>”按钮可查看已定义的变量名。

水平方向移动距离

向左移动：输入图素在水平方向向左移动（以被连接对象在画面中的原始位置为参考基准）的距离。

向右移动：输入图素在水平方向向右移动（以被连接对象在画面中的原始位置为参考基准）的距离。

对应表达式值范围

最 左 边：输入与图素处于最左边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向左（以原始位置为参考基准）移到最左边规定的位置。

最 右 边：输入与图素处于最右边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向右（以原始位置为参考基准）移到最右边规定的位置。

4.4.5.2　垂直移动连接

垂直移动连接是使连接对象在画面中的位置随连接表达式的值而垂直移动。移动距离以象素为单位，以被连接对象在画面制作系统中的原始位置为参考基准。垂直移动连接常用来表示对象实际的垂直运动，单击“动画连接”对话框中的“移动/垂直”按钮，弹出如图 4-20 所示对话框：

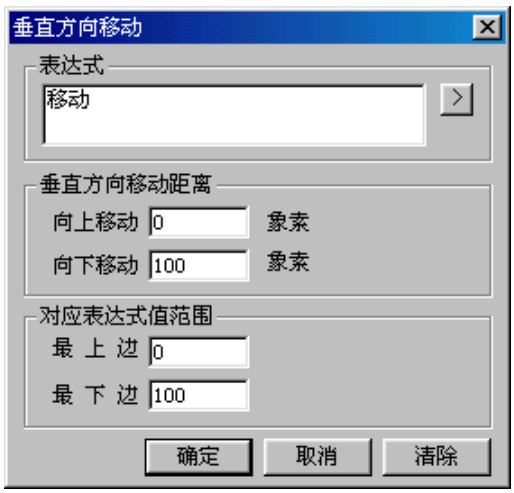


图 4-20

对话框中各项设置的意义如下：

表 达 式：在此编辑框内输入合法的连接表达式，单击“>”按钮可查看已定义的变量名。

垂直方向移动距离

向上移动：输入图素在垂直方向向上移动（以被连接对象在画面中的原始位置为参考基准）的距离。

向下移动：输入图素在垂直方向向下移动（以被连接对象在画面中的原始位置为参考基准）的距离。

对应表达式值范围

最 上 边：输入与图素处于最上边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向上（以原始位置为参考基准）移到最上边规定的位置。

最 下 边：输入与图素处于最下边时相对应的变量值，当连接表达式的值为对应值时，被连接对象的中心点向下（以原始位置为参考基准）移到最下边规定的位置。

4.4.6 百分比填充连接

百分比填充连接是使被连接对象的填充物（颜色和填充类型）占整体的百分比随连接表达式的值而变化。

百分比填充连接方式有两种：百分比水平填充和百分比垂直填充。

4.4.6.1 百分比水平填充连接

百分比水平填充连接的设置方法是：在“动画连接”对话框中单击“百分比填充/水平”按钮，弹出对话框如图 4-21 所示：

对话框中各项设置的意义如下：

表 达 式：在此编辑框内输入合法的连接表达式，单击“>”按钮可查看已定义的变量名。

最小填充：输入对象填充宽度（或高度）最小时占据的被连接对象的宽度（或高度）的百分比及对应的表达式的值（对应数值）。

最大填充：输入对象填充宽度（或高度）最大时占据的被连接对象的宽度（或高度）的百分比及对应的表达式的值（对应数值）。

填充方向：规定填充方向。

填 充 色：规定填充颜色。

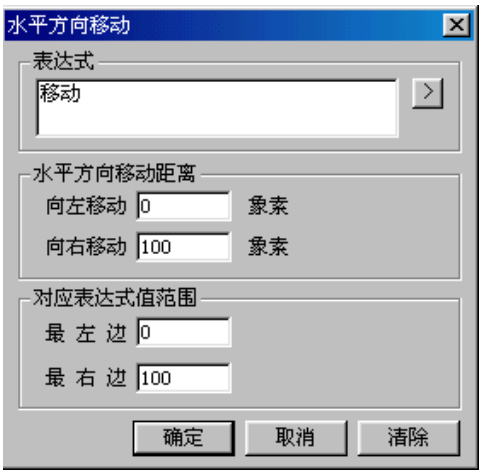


图 4-21

4.4.6.2　百分比垂直填充连接

百分比垂直填充连接的设置方法是：在“动画连接”对话框中单击“百分比填充/垂直”按钮，弹出对话框如图 4-22 所示：

图 4-22

百分比垂直填充连接的设置方法和百分比水平填充连接的设置方法相同，在此不一一详述了。

4.4.7 数值显示连接

数值显示连接是使文本对象的内容在程序运行时被连接表达式的值所取代。

数值显示连接对话框中有三种设置方式：离散值、模拟值和信息值。

4.4.7.1 离散值数值显示连接

离散值数值显示连接的设置方法是：在“动画连接”对话框中单击“数值显示/离散值”按钮，弹出对话框如图 4-23 所示：

图 4-23

对话框中各项设置的意义如下：

表达式：在此编辑框内输入合法的连接表达式，单击“>”按钮可查看已定义的变量名。

输出信息

变量为真时，输入信息：规定表达式为真时，被连接对象(文本)输入的内容。
变量为假时，输入信息：规定表达式为假时，被连接对象(文本)输入的内容。

4.4.7.2　模拟值数值显示连接

模拟值数值显示连接的设置方法是：在“动画连接”对话框中单击“数值显示/模拟值”按钮，弹出对话框如图 4-24 所示：

图 4-24

模拟值输出显示和信息值显示输出的设置过程同离散值输出显示设置方法。

对话框中各项设置的意义如下：

表 达 式：在此编辑框内输入合法的连接表达式，单击“>”按钮可查看已定义的变量名。

输出格式：可选择科学计数法或一般计数法，并规定小数点位数。

4.4.7.3 信息值数值显示连接

信息值数值显示连接的设置方法是：在“动画连接”对话框中单击“数值显示/信息值”按钮，弹出对话框如图 4-25 所示：

图 4-25

设置后将表达式的值显示于画面中的信息值位置。

4.4.8　特殊功能连接

特殊功能连接主要包括隐含连接、闪烁连接和旋转连接等功能。

4.4.8.1　隐含连接

隐含连接是使被连接对象根据条件表达式的值而显示或隐含。

隐含连接的设置方法是：在“动画连接”对话框中单击“特殊功能/可见/隐含”按钮，弹出对话框如图 4-26 所示：

图 4-26

对话框中各项设置的意义如下：

表　达　式：输入显示或隐含的条件表达式，单击“>”可以查看已定义变量名。

表达式为真时：规定当条件表达式值为 1（TRUE）时，被连接对象是显示还是隐含。当表达式的值为假时，定义了“显示”状态的对象自动隐含，定义了“隐含”状态的对象自动显示。

4.4.8.2　闪烁连接

闪烁连接是使被连接对象在条件表达式的值为真时闪烁。闪烁效果易于引起注意，故常用于出现非正常状态时的报警。

闪烁连接的设置方法是：在“动画连接”对话框中单击“特殊连接/闪烁”按钮，弹出对话框如图 4-27 所示：

图 4-27

对话框中各项设置的意义如下：

表达式：输入闪烁的条件表达式，当此条件表达式的值为真时，图形对象开始闪烁。表达式的值为假时，闪烁则自动停止。单击“>”按钮可查看已定义的变量名。

闪烁方式：规定闪烁时颜色是否改变。

闪烁速度：规定闪烁的频率。

4.4.8.3 旋转连接

旋转连接是使对象在画面中的位置随连接表达式的值而旋转。

旋转连接的设置方法为：在“动画连接”对话框中单击“旋转连接”按钮，弹出对话框如图 4-28 所示：

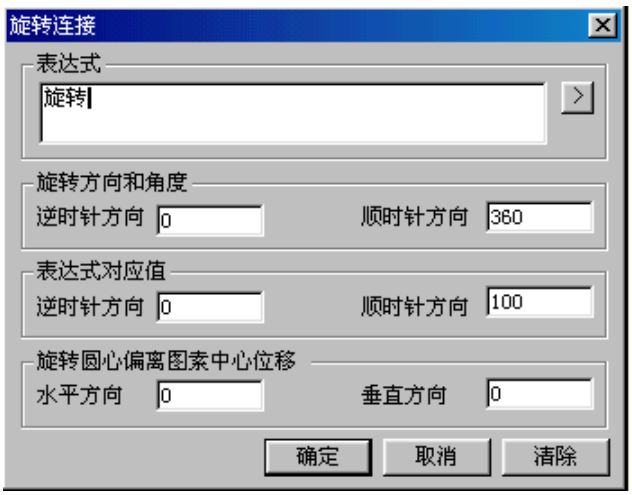


图 4-28

对话框中各项设置的意义如下：

表达式：在此编辑框内输入合法的连接表达式，单击“>”按钮可查看已定义的变量名。

旋转方向和角度

逆时针方向：被连接对象逆时针方向旋转所能达到的角度及对应的表达式的值（对应数值）。旋转中心点垂直向上为 0 度。

顺时针方向：被连接对象顺时针方向旋转所能达到的角度及对应的表达式的值（对应数值）。旋转中心点垂直向上为 0 度。

表达式对应值

旋转方向的旋转角度所对应的值。

旋转圆心偏离图素中心位移

被连接对象旋转时所围绕的圆心坐标距离被连接对象中心的值，水平方向为圆心坐标水平偏离的象素数（正值表示向右偏离），垂直方向为圆心坐标垂直偏离的象素数（正值表示向下偏离），该值可由状态行显示位置坐标计算得出。

4.4.9 访问权限设置

利用此功能用户可以进行输入连接的访问权限设置，如图 4-29 所示：

图 4-29

第五章　命令语言

世纪星除了在定义动画连接时支持连接表达式，还允许用户定义类似于C语言的命令语言来驱动应用程序，极大地增强了应用程序的灵活性。

命令语言有五种形式，其区别在于命令语言执行的时机或条件不同：

1.　应用程序命令语言

可以在程序启动时执行、关闭时执行或者在程序运行期间定时执行。如果希望定时执行，还需要指定时间间隔。

2.　热键命令语言

被链接到设计者指定的热键上，软件运行期间，操作者随时按下热键都可以启动这段命令语言程序。

3.　事件命令语言

规定在事件发生、存在、和消失时分别执行的程序。离散变量名或表达式都可以作为事件。

4.　数据改变命令语言

只链接到变量或变量的域。在变量或变量的域的值变化到超出变量字典中所定义的变化灵敏度时，它们就被执行一次。

5.　高速命令语言

有关高速采集时用，当 $启动高速命令语言为 1（真） 时，该语言才会执行。

5.1 命令语言编辑器

“命令语言编辑器”对话框如图 5-1 所示。

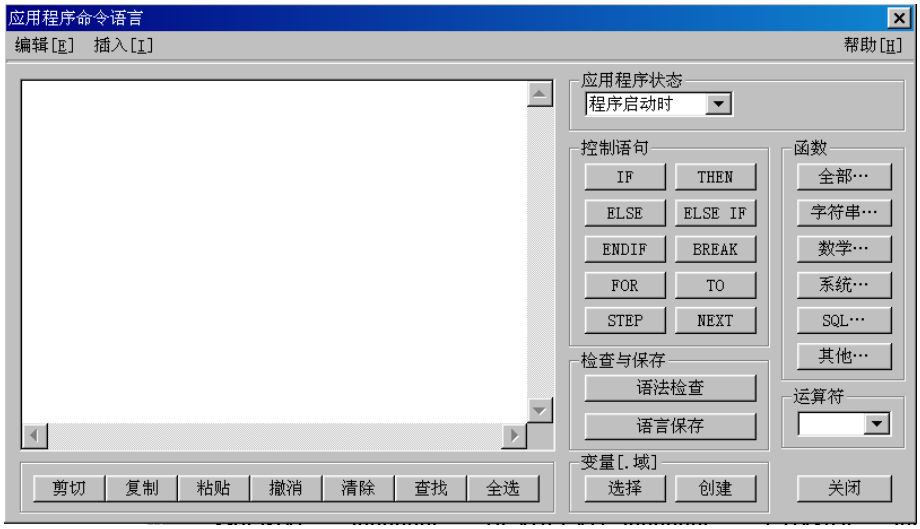


图 5-1

命令语言的句法和 C 语言非常类似，是 C 的一个子集，具有完备的语法查错功能和丰富的运算符、数学函数、字符串函数、控件函数和系统函数。各种命令语言通过“命令语言”对话框编辑输入，在世纪星中被编译执行。

在命令语言编辑器中，可以进行功能键、操作键、变量域输入、函数输入等操作。

5.1.1 功能键

命令语言编辑器中功能键主要有：剪切、复制、粘贴、撤消、清除、查找、全选、语法检查和语言保存。

剪　　切：选择此按钮从命令语言程序中剪切所选的文本块，并将其保存 WINDOWS 的裁剪板上，所选的文本从程序中删除，在一种命令语言中剪切完后可在其他命令语言或在 Windows 其他文件中（如写字板、WORD 文档等）粘贴。

复　　制：选择此按钮，则将所选的文本拷贝至 WINDOWS 的裁剪板上，但所选的文本不从程序中删除，其他功能与“剪切”功能相似。

粘　　贴：选择此按钮，则将 WINDOWS 的裁剪板上剪切或拷贝的内容，在当前光标位置上粘贴到命令连接程序中。

撤　　消：选择此按钮，则使最后编辑操作撤消。

清　　除：清除命令语言编辑区中的所有程序内容。

查　　找：单击此按钮，进行查找和替换操作，具体操作如下图 5-2 所示：

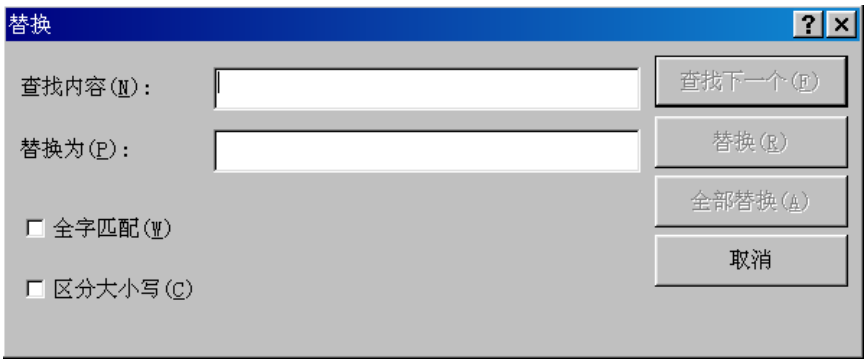


图 5-2

全　　选：全部选中命令语言编辑区中的所有程序内容。

语法检查：对当前正在编辑的命令语言进行语法检查，在编辑框的下方会弹出语法检查中所查出的错误信息，每行前有错误所在命令语言的行号，双击该错误信息行，光标会自动指到命令语言中的出错行。

语言保存：保存当前正在编辑的命令语言，如在按此功能键前没有对命令语言进行语法检查，在保存前还进行语法检查。

5.1.2 操作键

命令语言编辑器中操作键主要有：IF,THEN,ELSE, ELSE IF, END IF, FOR, TO, STEP, NEXT, BREAK 等。

5.1.3 变量域输入

“变量[.域]选择”对话框如图 5-3 所示：

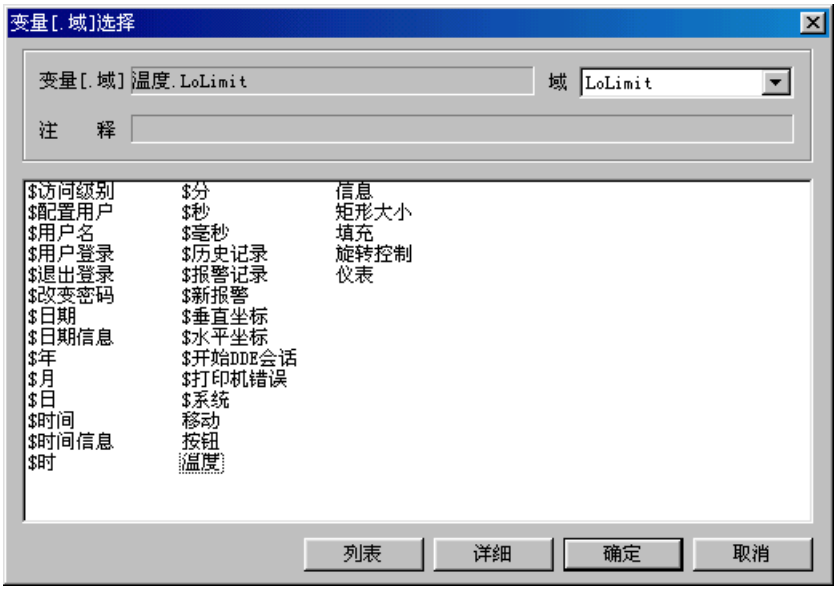


图 5-3

有关变量域输入的详细说明，请参见有关变量.域的相关内容。

5.1.4 函数输入

“函数选择”对话框如图 5-4 所示：

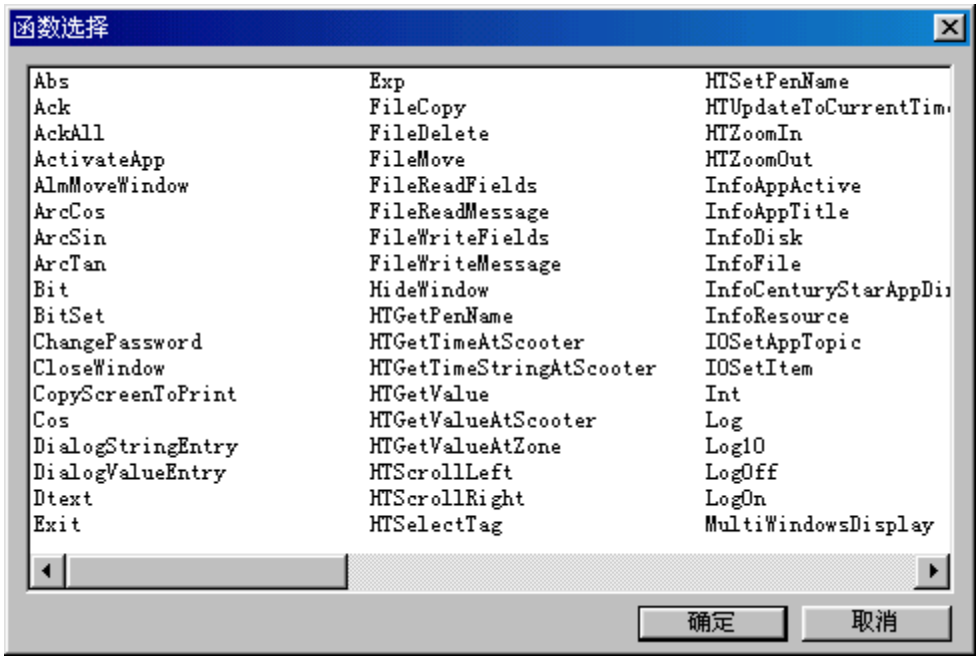


图 5-4

有关函数输入的详细说明，请参见具体函数一章的相关说明。

5.2　命令语言描述

命令语言可以进行赋值、比较、数学运算，还提供了可执行 IF-ELSE 条件语句及 FOR-NEXT 循环语句的逻辑操作能力。

5.2.1　运算符

用运算符连接变量或常量就可以组成较简单的命令语言语句，如赋值、比较、数学运算等。命令语言中可使用的运算符以及算符优先级与连接表达式相同，参

见“连接表达式”一节。

5.2.2 赋值语句

赋值语句用得最多，语法如下：

变量（变量的可读写域）=表达式；

可以给一个变量赋值，也可以给可读写变量的域赋值。

示例：

自动开关=1；表示将自动开关置为开(1 表示开，0 表示关)
颜色=2；　　将颜色置为黑色(如果数字 2 代表黑色)
反应罐温度.Priority=3；　表示将反应罐温度的报警优先级设为 3

5.2.3 IF-THEN-ELSE 语句

IF-THEN-ELSE 语句根据表达式的状态有条件的执行不同的指令程序，以下的比较运算符用于在 IF-THEN-ELSE 语句中建立条件。

< 小于
> 大于
<= 小于等于
>= 大于等于
== 恒等于
<> 不等于

以下是命令语言的示例：

1、不带 ELSE 语句的 IF-THEN 语句:

```
IF a <> 0 THEN
   a = a + 100;
ENDIF;
```

2、带一个 ELSE 语句的 IF-THEN-ELSE 语句:

```
IF temp > 500 THEN
Disc = 1;
Real = 43.7;
ELSE
Disc = 0;
Real = 93.4;
ENDIF;
```

3、带一个 ELSE IF 语句不带 ELSE 语句的 IF-THEN-ELSE 语句:

```
IF temp > 500 THEN
Disc = Disc * 10;
ELSE
IF temp > 250 THEN
x = y / z;
a = abc + def;
ENDIF;
ENDIF;
```

4、带一个 ELSE IF 语句和一个 ELSE 语句的 IF-THEN-ELSE 语句:

```
IF temp > 500 THEN
Disc = Disc - 10;
ELSE
IF temp < 250 THEN
Disc = Disc + 10;
```

```
ELSE
Disc = Disc + 50;
Real = 100;
ENDIF;
ENDIF;
```

5、带多个 ELSE IF 语句和一个 ELSE 语句的 IF-THEN-ELSE 语句:

```
IF temp > 100 THEN
temphihi = 1
Disc = 50;
ELSE
IF temp > 80 THEN
temphi = 1;
ELSE
IF temp < 10 THEN
templo = 1;
ELSE
IF temp < 30 THEN
templolo = 1;
ELSE
tempok = 1;
ENDIF;
ENDIF;
ENDIF;
ENDIF;
```

6、测试条件 1 和条件 2 的 IF-THEN-ELSE 语句：

```
IF (pump1 < 50.0) || (pump2 < 50.0) THEN
   Alarm_1 = 1;
ELSE
```

```
  Alarm_1 = 0;
ENDIF;
```

注释：每个 IF 语句必须和 ENDIF 语句匹配。

5.2.4 FOR-NEXT 循环语句

FOR-NEXT 循环用于完成在命令语言中多次执行一条函数命令或一组函数命令。以下是 FOR-NEXT 循环语句的通用格式：

```
FOR 模拟变量 = 起始值 TO 终止值 [STEP 步长；]
...statements...
IF (条件表达式) THEN
[BREAK;]
ENDIF;
...statements...
NEXT;
```

注释：步长可以为正数或负数。如果步长为正数表示递增，步长为负数表示递减。步长缺省值为 1。

5.2.5 命令语言注释

在大括号{…}内所包含的内容为注释。

5.3 命令语言的函数

世纪星支持使用内建的复杂函数，其中包括字符串函数、数学函数、系统函数、控件函数、SQL 函数及其他函数，共计 199 个，具体的使用方式请参见有关函数一章：

5.4 用户自定义函数

用户自定义函数是“世纪星”最突出的功能之一，它能方便用户在命令语言中随时增加命令语言函数，函数中又可编制各种算法和用户需要的一些其他的应用程序函数，来增强应用系统的整体功能，达到更精确控制应用系统的目的。

5.4.1 用户自定义函数格式：

```
FUNCTION TYPE① FuncName(TYPE② Param1, TYPE Param2, … TYPE ParamN)
TYPE③ LocalVarName1;
TYPE LocalVarName2;
⋮
TYPE LocalVarNameN;

… statements …

(RETURN ReturnValue④;)
ENDFUNC;
```

参数说明：

FUNCTION — 自定义函数标志。
ENDFUNC — 自定义函数结束标志。
TYPE — 自定义函数支持的数据类型。
DISCRETE — 离散类型
INTEGER — 整数类型
REAL — 实数类型
MESSAGE — 信息类型

① 函数返回值数据类型，包括 DISCRETE、INTEGER、REAL、MESSAGE 和 VOID（无返回值）。

② 函数参数，数据类型，包括 DISCRETE、INTEGER、REAL、MESSAGE。
③ 用户自定义变量（仅用于函数体内），数据类型 DISCRETE、INTEGER、REAL、MESSAGE。
④ 函数返回值的数据类型不为 VOID 时，必须加 RETURN 返回值；语句。

注释：1、关键字不区分大小写。
2、函数体内支持 IF—ELSE—ENDIF 语句、FOR—NEXT 语句，系统函数调用。
3、用户自定义函数必须在调用之前定义，且必须定义在命令语言的前面。

5.4.2　用户自定义函数调用格式：

```
FuncName(Param1，Param2，… ParamN);            或
ReturnValue=FuncName(Param1，Param2，… ParamN);
```

前一种为无返回值或者不需要返回值时的调用方式，后一种方式可取得函数的返回值。

示例：

```
FUNCTION INTEGER AddInteger(INTEGER P1,INTEGER P2,INTEGER P3)
INTEGER D1,D2;                  {用户定义变量}
D1=P1+P2;
D2=D1+P3;
RETURN D2;                      {函数返回值}
ENDFUNC;                        {用户自定义函数结束}
Result=AddInteger(5,10,15);     {调用用户自定义函数}
```

5.5　应用程序命令语言

根据程序设计者的要求，应用程序命令语言可以在程序启动时执行、关闭时执行或者在程序运行期间定时执行。如果希望定时执行、还需要指定时间间隔。

选择菜单“系统→命令语言→应用程序命令语言”，将出现如图 5-5 所示对话框：

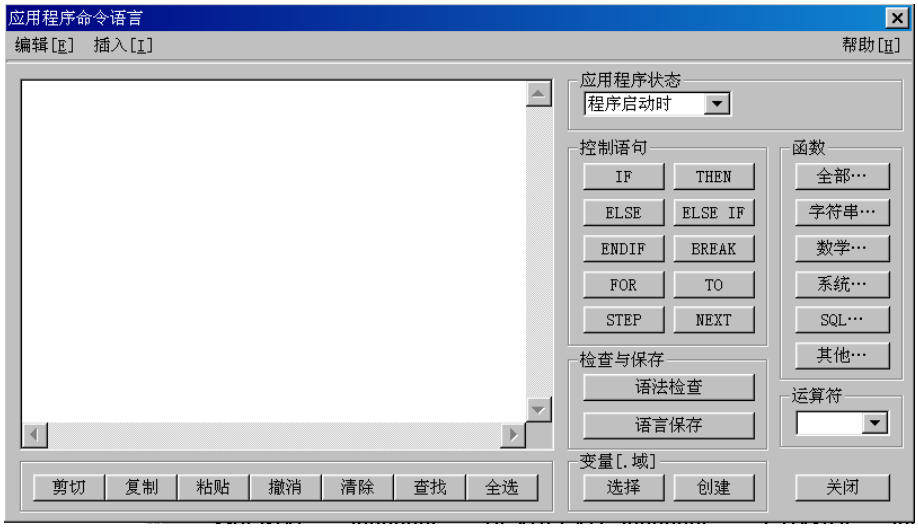


图 5-5

5.6 热键命令语言

“热键命令语言”链接到设计者指定的热键上，软件运行期间，操作者随时按下热键都可以启动这段命令语言程序。热键命令语言可以指定使用权限。

输入热键命令语言请选择菜单“系统→命令语言→热键命令语言”，弹出“热键命令语言”对话框如图 5-6 所示：

图 5-6

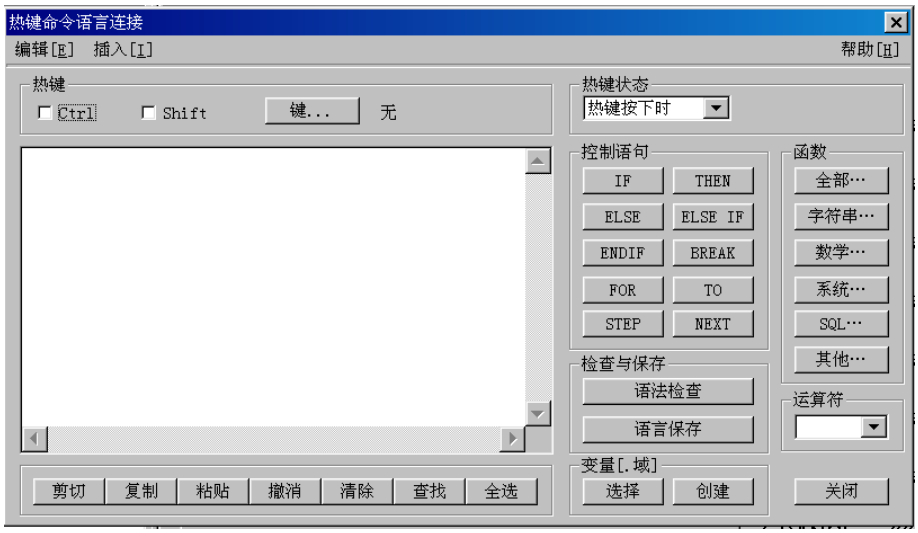


图 5-7

“热键命令语言”对话框用于管理画面程序中所有的热键命令语言。单击“增加”按钮，则弹出“热键命令语言”对话框如图 5-7 所示。用户可以定义热键、

设置热键操作权限并输入命令语言：

热键定义 参考“第四章 动画连接”中关于热键定义的说明。

5.7 事件命令语言

事件命令语言可以规定在事件发生、存在、和消失时分别执行的程序。离散变量名或表达式都可以作为事件。

编辑“事件命令语言”，请选择“系统→命令语言→事件命令语言”菜单。弹出“事件命令语言”对话框如图 5-8 所示：

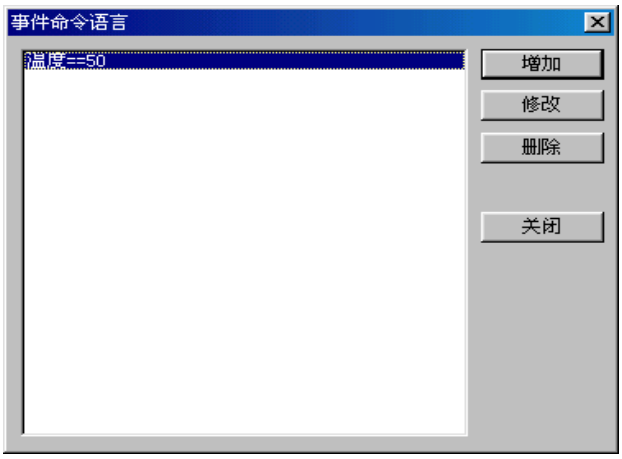


图 5-8

单击“增加”按钮，则弹出“事件命令语言连接”对话框如图 5-9 所示。在对话框中输入事件名（离散变量名或条件表达式）、注释，选择“事件状态”，然后输入命令语言程序。

事件描述：用户可通过该输入栏输入作为事件的离散变量名或条件表达式。
注　　释：通过该输入栏可以为输入的“事件”作其他备注。
事件状态：指定命令语言执行的条件。
发 生 时：　当事件发生时将执行命令语言程序。
存 在 时：　当事件存在时将按指定时间间隔执行命令语言程序。
消 失 时：　当事件消失时将执行命令语言程序。

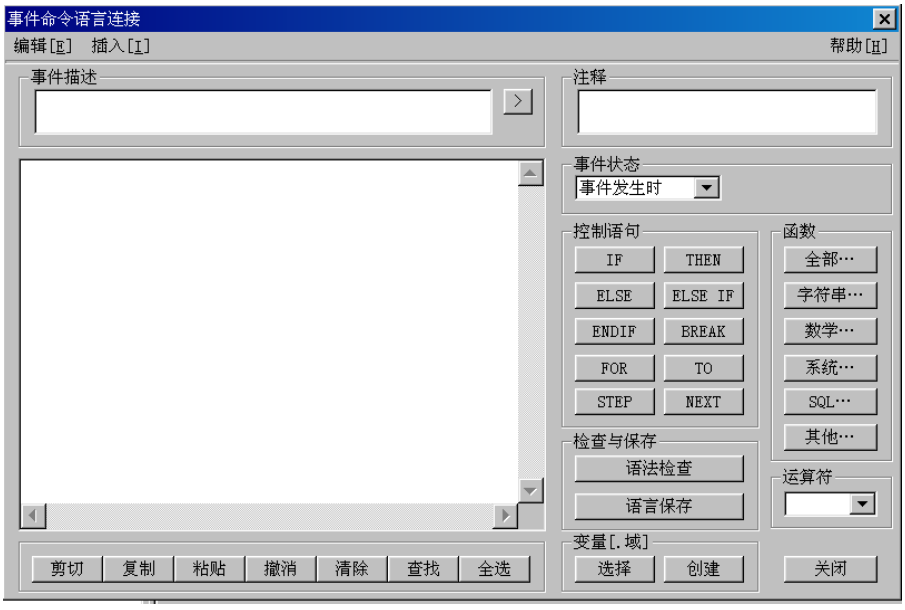


图 5-9

5.8　数据变化命令语言

“数据改变命令语言”只链接到变量或变量的域。在变量或变量的域的值变化到超出变量字典中所定义的变化灵敏度时，它们就被执行一次。

数据改变命令语言
温度
增加
修改
删除
关闭

图 5-10

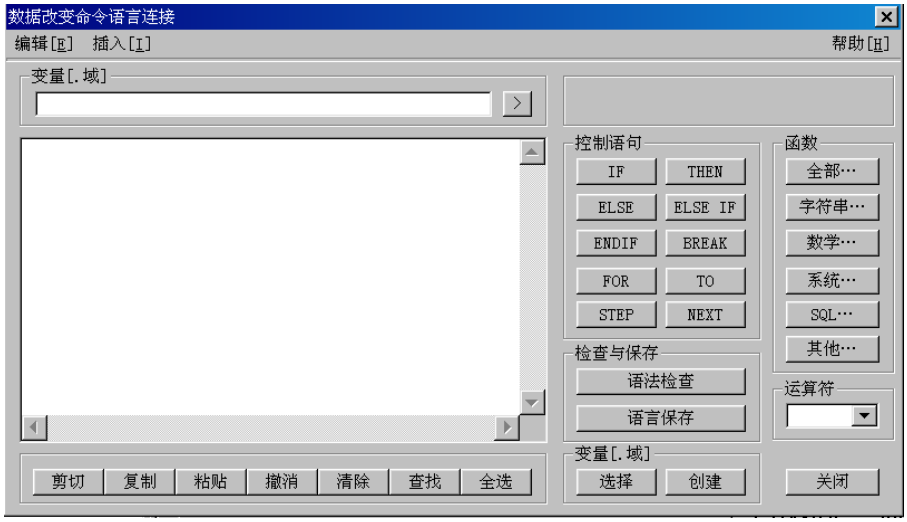

数据改变命令语言连接
编辑[E]
插入[I]
帮助[H]
变量[.域]
控制语句
IF
THEN
ELSE
ELSE IF
ENDIF
BREAK
FOR
TO
STEP
NEXT
函数
全部…
字符串…
数学…
系统…
SQL…
其他…
检查与保存
语法检查
语言保存
运算符
变量[.域]
选择
创建
关闭
剪切
复制
粘贴
撤消
清除
查找
全选

图 5-11

编辑数据改变命令语言，请选择“系统→命令语言→数据改变命令语言”菜单。将弹出“数据改变命令语言”对话框如图 5-10 所示：

单击“增加”按钮，弹出“数据改变命令语言”对话框如图 5-11 所示：

在“变量[. 域]”输入栏中输入变量或变量的域。当变量或变量的域的值变化到超出变量字典中所定义的变量灵敏度时，将执行这段命令语言程序。

注意：

在使用“事件命令语言”或“数据改变命令语言”过程中要注意防止死循环。

示例：

变量 A 变化引发数据改变命令语言程序中含有命令 B=B+1，若用 B 变化再引发事件命令语言或数据改变命令语言的程序中不能再有类似 A=A+1 的命令。

第六章　趋势曲线和报警窗口

趋势分析和运行报警是控制软件必不可少的重要功能，世纪星对以上两种功能提供了强有力的支持和简单的控制方法。

趋势曲线有实时趋势曲线和历史趋势曲线两种。实时趋势曲线用于实时显示数据的变化情况，历史趋势曲线用于查看历史数据。两种趋势曲线的外观、定义方法是相同或相似的。趋势曲线的外形类似于坐标纸，X 轴代表时间，Y 轴代表变量值。主要用于观察变量的变化趋势。同一个趋势曲线中最多可同时显示四个变量的变化情况，而一个画面中可定义数量不限的趋势曲线。

软件运行时，实时趋势曲线可以自动卷动，以快速反映变量随时间的变化。历史趋势曲线并不自动卷动，它一般与功能按钮一起工作，共同完成历史数据的查看工作。这些按钮可以完成翻页、设定时间参数、启动/停止记录、打印曲线图等复杂的功能。

报警处理是“工业现场监控系统”中的一个重要内容。当变量的数值或数值的变化异常时，将产生报警，以便操作者采取必要的措施。

6.1　实时趋势曲线

6.1.1　实时趋势曲线的含义

实时趋势曲线用于实时显示数据的变化情况。在画面运行时实时趋势曲线对象由系统自动更新。数据将从趋势的右边进入，同时趋势将从右向左移动。运行时，实时趋势曲线的外观如图 6-1 所示：

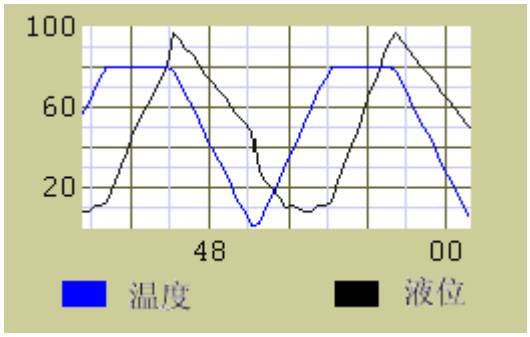


图 6-1

6.1.2　实时趋势曲线的定义

实时趋势曲线的定义方法如下：

1、绘制曲线

在 CSMAKER 中制作画面时，选择菜单“绘图→历史趋势曲线”项或者单击工具箱中的“画实时趋势曲线”按钮，此时鼠标在画面中变为十字形，在画面中用鼠标画出一个矩形，实时趋势曲线就在这个矩形中绘出，如 图 6-2 所示：

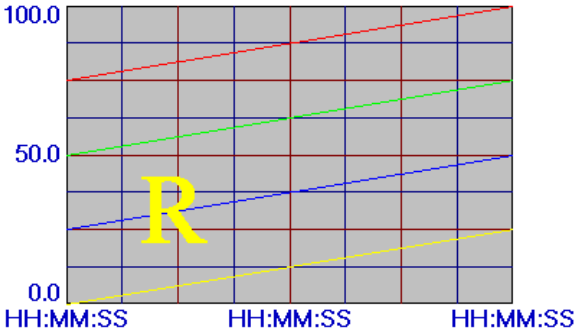


图 6-2

实时趋势曲线对象的中间有一个带有网格的绘图区域，表示曲线将在这个区域中绘出，网格下方和左方分别是 X 轴（时间轴）和 Y 轴（数值轴）的坐标标注。

2、调整曲线

可以通过选中实时趋势曲线对象（周围出现 8 个小矩形）来移动位置或改变大小。通过调色板工具或相应的菜单命令可以改变趋势曲线的笔属性和填充属性，笔属性是趋势曲线边框的颜色和线型，填充属性是边框和内部网格之间的背景颜色和填充模式。

注意：用户有时会发现趋势曲线中没有标注数字。这是因为背景颜色和字体颜色正好相同，更改背景颜色或字体颜色即可。

3、曲线定义

双击趋势曲线后，可弹出“实时趋势曲线配置”对话框如图 6-3 所示。可以对趋势曲线的外观、四条曲线所要显示的表达式、颜色属性等进行定义。同时亦可定义趋势曲线坐标轴的标识信息，主要用于定义趋势曲线的标识信息。

6.1.3 实时曲线定义对话框

实时曲线定义对话框选项如图 6-3：

实时趋势曲线注释
对所定义的实时趋势曲线进行必要的文字标识。

标注位置
标注位置共有四种选择：上边、下边、左边和右边。用户可根据自己的需要灵活选择不同的标注位置。

颜色
分别规定绘图区域的边框和背景(底色)的颜色。

时间轴

分别规定主分割线、次分割线、标注数量的具体数值和颜色以及时间格式选择参数。

图 6-3

主分割线：在趋势区内输入可见主划分线数目。这些线的颜色在邻近颜色方块中设置。

次分割线：在主分割线内输入可见次划分线数目。线的颜色在邻近颜色方块中设置。

标注数量：在每个主分割线内的标注数量值。数值颜色在邻近颜色方块中设置。

时间格式选择：时间格式可按时、分、秒来选择。

数值轴

分别规定主分割线、次分割线、标注数量的具体数值和颜色以及最大值、最小值参数。

主分割线：在趋势区内输入可见主划分线数目。这些线的颜色在邻近颜色方块中设置。

次分割线：在主分割线内输入可见次划分线数目。线的颜色在邻近颜色方块中设置。

标注数量：在每条主分割线内的标注数量值。数值颜色在邻近颜色方块中设置。

最大值、最小值参数：输入设置数值轴的范围。此范围为是工程单位，且对所有的趋势变量都是一样的。

时间长度设定

设定实时趋势曲线的显示长度数值以及单位。

数据采样频率

设定实时趋势曲线的采样频率数值以及所使用的单位。

曲线

定义所绘的 1～4 条曲线对应的表达式。同时，每条曲线可由右边的选择框分别选择颜色和线型。

说明：

一个实时趋势曲线可以同时显示 4 个变量或表达式的变化，因为各变量的数值范围可能相差很大，为方便每个变量都能表现清楚，实时趋势曲线的每根曲线可反应一个表达式的变化趋势，且变量或表达式在数值轴上以实际数表示，数值轴的范围是可由开发者自己输入。

6.2 历史趋势曲线

历史趋势曲线用于历史数据的查看工作。历史趋势曲线可使您得到过去的时间和数据。软件运行时，历史趋势曲线并不自动卷动，它一般与功能按钮一起工作，利用历史趋势曲线变量的域或者使用与历史趋势曲线有关的函数可以完成对历史趋势曲线的控制，这些按钮可以完成翻页、设定时间参数、启动/停止记录、打印曲线图等复杂的功能。

6.2.1　历史趋势曲线的定义

历史趋势曲线的定义方法如下：

1、绘制曲线

在 CSMAKER 中制作画面时，选择菜单“绘图→历史趋势曲线”项或者单击工具箱中的“画历史趋势曲线”按钮，此时鼠标在画面中变为十字形，在画面中用鼠标画出一个矩形，实时趋势曲线就在这个矩形中绘出，如图 6-4 所示：

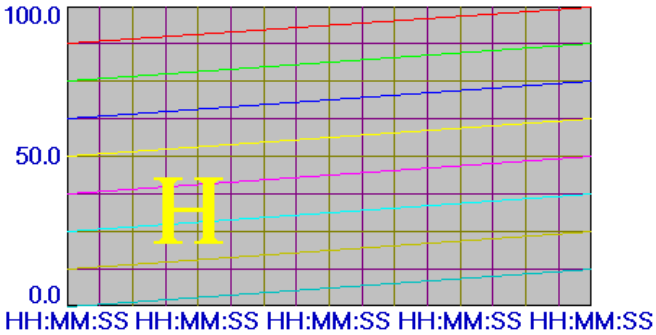


图 6-4

历史趋势曲线对象的中间有一个带有网格的绘图区域，表示曲线将在这个区域中绘出，网格下方和左方分别是时间轴和数值轴的坐标标注。

2、调整曲线

可以通过选中历史趋势曲线对象（周围出现 8 个小矩形）来移动位置或改变大小。通过调色板工具或相应的菜单命令可以改变趋势曲线的笔属性和填充属性，笔属性是趋势曲线边框的颜色和线型，填充属性是边框和内部网格之间的背景颜色和填充模式。

注意：用户有时会发现趋势曲线中没有标注数字。这是因为背景颜色和字体颜色

正好相同，更改背景颜色或字体颜色即可。

3、曲线定义

双击历史趋势曲线后，可弹出“历史趋势曲线配置”对话框。可以对趋势曲线的外观、四条曲线所要显示表达式、颜色属性等进行定义。

6.2.2 历史趋势曲线的曲线定义对话框

历史曲线定义对话框选项如图 6-5：

图 6-5

历史趋势曲线名

对历史趋势曲线窗口的标识，必须在“变量数据库”中先进行定义。

标注位置

标注位置共有四种选择：上边、下边、左边和右边。用户可根据自己的需要灵活选择不同的标注位置。

颜色

分别规定绘图区域的边框和背景(底色)的颜色。

时间轴

分别规定主分割线、次分割线、标注数量的具体数值和颜色以及标注格式参数选择。

主分割线：在趋势区内输入可见主划分线数目。这些线的颜色在邻近颜色方块中设置。

次分割线：在主分割线内输入可见次划分线数目。线的颜色在邻近颜色方块中设置。

标注数量：在每条主分割线内的标注数量值。数值颜色在邻近颜色方块中设置。

标注格式：标注格式可按年、月、日方式或者时、分、秒方式来选择。

数值轴

分别规定主分割线、次分割线、标注数量的具体数值和颜色以及显示方式参数选择。

主分割线：在趋势区内输入可见主划分线数目。这些线的颜色在邻近颜色方块中设置。

次分割线：在主分割线内输入可见次划分线数目。线的颜色在邻近颜色方块中设置。

标注数量：在每条主分割线内的标注数量值。数值颜色在邻近颜色方块中设置。

显示方式：显示方式可按最大值/最小值方式或者平均值方式来选择。

按实际值显示:系统状态为按百分比显示，选择此功能按实际值显示。

曲线

定义所绘的 1～8 条曲线对应的表达式。同时，每条曲线可由右边的选择框分别选择颜色和线型。

历史趋势曲线的补充说明

由于历史趋势曲线的数据来源于历史数据文件，所以应确保历史数据记录的正常工作。当使用历史趋势曲线时，应特别注意：

1、选择历史记录有效。历史趋势曲线中显示的变量在变量定义时，应选择历史记录有效。

2、确保历史数据文件的存储目录有效。在工程管理器“系统配置”节点下双击“历史数据记录”，则弹出历史记录配置对话框。在此对话框中输入记录历史数据文件在磁盘上的存储路径和其它属性(如数据文件记录时数，记录起始时刻，数据保存天数)。

3、重新启动历史数据记录。在 CSVIEWER 的菜单条上单击“系统”菜单项，再从弹出的菜单命令中选择“重启历史数据记录”命令项，此选项用于重新启动历史数据记录，在没有空闲磁盘空间时，系统就自动停止历史数据记录。当发生此情况时，还会显示一信息框，通知用户，用户将数据转移到其他地方后，空出磁盘空间。此操作可调用此命令重启历史数据记录。

6.3 报警窗口

报警处理是“工业现场监控系统”中的一个重要内容。当变量的数值或数值的变化异常时，将产生报警，以便操作者采取必要的措施。

6.3.1 报警窗口的概念

报警窗口用于显示报警事件。世纪星自动对“变量定义”对话框中“变量报警定义”有效的数据变量进行监视。当发生报警事件时，把这些事件存于内存中

的报警事件缓冲器中。报警事件在缓冲器中是以先进先出的原则存贮，所以只有最近的报警事件存于内存中。历史报警窗口的报警事件多是取自报警缓冲器。如果一个数据变量包含于报警窗口的报警组，而且变量定义时报警优先级不大于报警窗口的报警优先级时，报警事件就会在报警窗口中显示出来，其显示格式由报警窗口定义时报警信息格式选项规定。

报警窗口有两种类型：实时报警窗口和历史报警窗口。实时报警窗口只显示当前处于报警状态的报警事件，历史报警窗口则显示与报警有关的所有的报警事件（报警发生、报警应答、报警恢复）。

6.3.2　报警窗口的定义

在世纪星开发系统中制作画面时，选中菜单“绘图→报警窗口”项或者单击工具箱中的“报警窗口”按钮，这时鼠标在画面中变为十字形，在画面中用鼠标画出一个矩形，在画面中用鼠标画出一个矩形，报警窗口就在这个矩形中绘出，如图 6-6 所示。

图 6-6

6.3.3 报警窗口定义对话框

双击报警窗口对象可弹出报警窗口定义对话框，如图 6-7 所示：

图 6-7

报警窗口名：规定报警窗口在数据库中的变量登记名，此报警窗口变量名可在为操作报警窗口建立的命令语言连接程序中使用。

窗口类型：用户可选择报警窗口类型，实时报警窗口或历史报警窗口。

标　　题：选中此选择框将在报警窗口中显示标题，同时右面的“背景色”按钮和“文本色”按钮会加亮。其中“背景色”按钮是选择标题条的背景色，“文本色”按钮是选择标题文本色，用鼠标按下相应

按钮出现一个浮动的颜色选择对话框，使用方法与“趋势曲线”类似。

颜　色：报警事件显示时的颜色设置

边框色：是报警事件窗口的边框显示颜色。

背景色：是报警事件窗口的背景显示颜色。

报警色：是报警事件窗口的报警显示颜色。

应答色：是报警事件中应答报警事件的显示文本颜色。

恢复色：是报警事件中报警恢复事件的显示文本颜色。此项能否选择与实时报警窗口和历史报警窗口的选择有关，在实时报警窗口中此项被禁止，因为它不显示此类报警事件。

报警信息格式

报警事件的显示格式

日　　期：选择是否记录日期及如何记录日期，有四种格式可供选择。

时　　间：选择时、分、秒是否记录，可选择较短的格式。

报警类型：是否记录报警事件的种类（高高、偏差报警等）。

事　　件：是否记录是何种报警事件（报警、应答、恢复），这几类事件在报警窗口中以不同的颜色来显示。

优 先 级：是否记录报警事件的优先级。

变 量 名：是否记录发生报警事件的变量名，后面的数字是变量名所占的字符宽度，如果变量名超出此宽度，将被截断。

报警组名：是否记录报警组名，后面的数字是报警组名所占的字符宽度。

当 前 值：是否记录报警发生时的数据变量值，后面的数字是变量值所占的字符宽度。

界 限 值：是否记录报警发生时变量报警定义时的报警设定值，后面的数字是此值所占的字符宽度。

注　　释：是否记录报警发生时数据变量的描述，这使用户可以利用描述来给操作员解释报警事件的性质或提示报警事件的处理方法，后面的数字是变量描述所占的字符宽度。

新报警发生时，报警窗口自动滚动：窗口是否总是显示最后出现的报警事件，

滚动选择项有效时报警事件从下向上推移。否则，需要建立有命令语言连接的按钮来操作以查看报警窗口的内容。

显示报警优先级：指报警窗口显示报警事件必须达到的优先级，可以在编辑区直接输入。

6.4 与报警窗口有关的操作

与报警窗口有关的操作包括报警组定义和报警配置功能等。

6.4.1 报警组定义

在开发系统的“系统”菜单命令下，选中“报警组定义”功能，则弹出“报警组定义”窗口，它显示已定义报警组名的树状结构，可用报警组名指定一个报警窗口来监视一个报警组的变量，如图 6-8 所示：

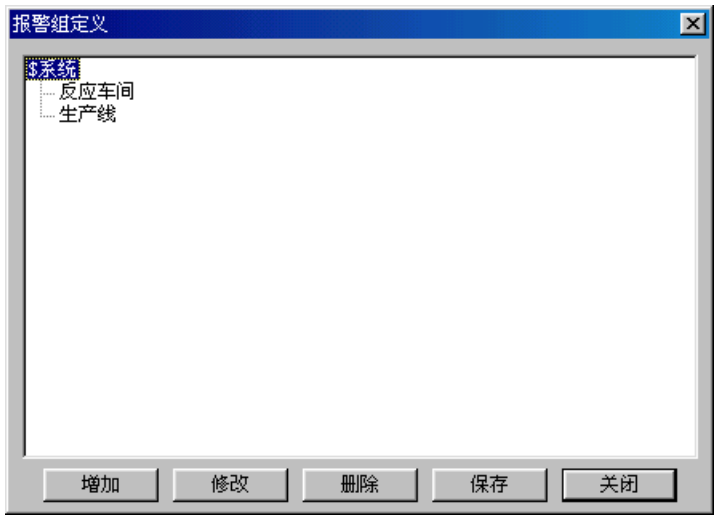


图 6-8

6.4.2 报警配置

在开发系统中，选择菜单命令“系统→系统配置→报警系统”可以弹出“报警配置”对话框，它用于定义报警事件的参数设置、报警事件的记录格式和报警事件的打印格式。有关内容请参阅第一章的 1.3.11.3 节。

第七章　I/O 驱动设备

世纪星的设备驱动采用 OLE 自动化（即 COM 组件）接口技术，使设备驱动程序和世纪星有机的整合在一起，来完成数据采集和实时控制。对于不同的硬件设备，利用世纪星的设备安装向导，配置相应的设备驱动程序即可。

世纪星支持的硬件设备包括：可编程控制器（PLC）、智能仪表、智能模块、变频器、板卡、电力载波、有线和无线 Modem、工业以态网、电台通讯、DDE 动态数据交换、网络设备、调用用户动态库等等。用户不必关心具体的通信协议，只需要在世纪星中，按照“设备安装向导”的提示安装设备驱动程序，设置必要的参数即可，在安装不同的设备时，可参考在线帮助中的“世纪星支持的硬件设备”一章中的相关硬件安装说明。

7.1　I/O 驱动设备管理概述

世纪星的设备驱动管理采用树形层次结构，在浏览器中，列出了世纪星已经配置的驱动设备对象名，每个设备对象名实际上是具体设备的逻辑名称，每一个设备对象名对应一个相应的驱动程序。另外，世纪星的设备管理采用驱动设备安装向导，用户只要按照设备安装向导的提示设置相应的参数，选择 I/O 设备的生产厂家、设备名称、通信方式，并指定设备对象名和通信地址，则世纪星自动完成驱动程序的安装，并在运行系统中启动设备驱动程序，完成数据采集和通信，不需要用户人工进行。

世纪星浏览器中设备驱动树形结构如图 7-1 所示：

图 7-1

7.2　I/O 驱动设备分类

世纪星中驱动设备分为串口通信设备、板卡设备（即总线型设备）、DDE 设备和网络通信设备等几种，用户根据自己的实际情况通过世纪星的设备安装向导配置这些逻辑设备。

1、 串口通信设备

串口通信设备是指通过串口（COM1、COM2 等）与世纪星进行通信的 I/O 设备。包括可编程控制器（PLC）、智能仪表、智能模块、变频器等。世纪星与串口类逻辑设备之间的关系如图 7-2 所示：

2、 板卡驱动设备

板卡驱动设备是指插入计算机总线扩展槽中的 I/O 设备，因此，一个板卡驱动设备也就代表了一个实际插入计算机总线扩展槽中的 I/O 板卡。世纪星与板卡类逻辑设备之间的关系如 图 7-3 所示：

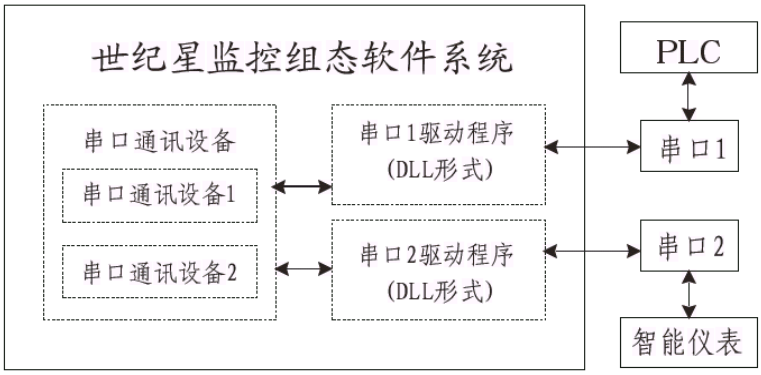


图 7-2

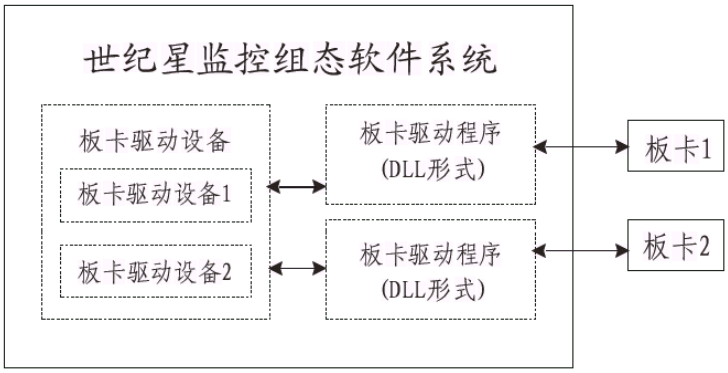


图 7-3

世纪星根据用户指定的板卡驱动设备自动调用相应板卡驱动程序。

3、 DDE 动态数据交换设备

DDE（动态数据交换）是 Microsoft 公司设计的一种基于 Windows 消息驱动机制的动态数据交换协议。应用程序采用三层标识系统，即服务程序名、话题名

和项目名，在两个或多个应用程序之间动态传送数据。DDE 数据交换时，提供数据的应用程序为服务应用程序，接收数据的应用程序为客户应用程序。

世纪星通过应用程序名、话题名和项目名标识 I/O Server 中的数据变量。I/O Server 通过上述三个会话参数从客户服务程序（世纪星）获取数据。系统运行时，世纪星与 I/O Server 自动建立数据交换通道实现数据传递。

当其它应用程序需要从世纪星中获取数据时，世纪星规定：

1. 应用程序名：CSVIEWER
2. 话题名：TAGNAME
3. 项目名：世纪星变量数据库中定义的项目名。

示例：

在 Excel 中访问世纪星的变量，可在 Excel 单元中敲入如下命令：

=CSVIEWER|TAGNAME!'变量项目名'

世纪星与 DDE 设备之间的关系如图 7-4 所示：

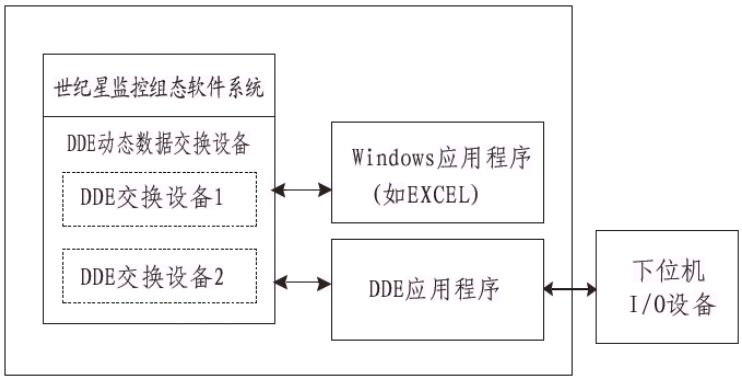


图 7-4

4、 网络驱动设备

世纪星利用以太网和 TCP/IP 协议与专用的网络通信模块进行连接。

7.3 I/O 驱动设备管理

世纪星采用设备驱动向导，由用户进行 I/O 设备配置，使用这些配置向导页可以方便快捷地增加、修改、删除设备驱动程序。

选择“文件/驱动设备管理”或在浏览器中双击“驱动设备管理”，则弹出驱动设备管理对话框如图 7-5 所示

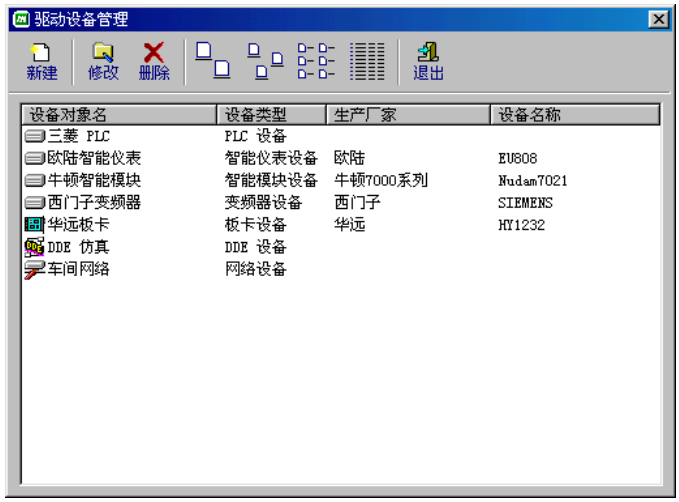


图 7-5

7.3.1 串口通信设备安装向导

根据设备安装向导就可以完成串口通信设备的安装，操作步骤如下：

1、在“驱动设备管理”中选择“新建”项，或在浏览器中双击“设备安装向导”

则弹出设备安装向导对话框如图 7-6 所示：

用户从树形设备列表区中可选择 PLC、智能仪表、智能模块、变频器等节点中的一个。然后选择要配置串口设备的生产厂家、设备对象名称、通信方式；PLC、智能仪表、智能模块、变频器等设备通常与计算机的串口相连进行数据通信。

图 7-6

2、单击“下一步”按钮，则弹出设备安装向导对话框如图 7-7 所示：

设备对象名称：要安装串口通信设备的名称。

通 信 端 口：串口设备与计算机相连的串口号，该下拉式串口列表框共有 32 个串口号供用户选择。

设 备 地 址：串口通信设备的设备地址。

尝试恢复间隔：在世纪星运行期间，如果有一台设备如 PLC1 发生故障，则世纪星能够自动诊断并停止采集与该设备相关的数据，但会每隔一段时间尝试恢复与该设备的通信，如图所示，尝试时间间隔为 3 秒。

最长恢复时间：若世纪星在一段时间之内一直不能恢复与PLC1的通信，则不再尝试恢复与PLC1通信，这一时间就是指**最长恢复时间**。

使用动态优化：世纪星对全部通信过程采取动态管理的办法，只有在数据被

上位机需要时才被采集。

单击“上一步”按钮，则可返回上一个对话框进行修改。

图 7-7

3、继续单击“下一步”按钮，则弹出设备配置向导对话框如图 7-8 所示：

图 7-8

此向导页显示已配置的串口通信设备的设备信息，供用户查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，完成设备安装。在浏览器的“当前驱动设备”中将增加新安装的设备。

7.3.2　板卡设备安装向导

根据设备安装向导就可以完成板卡设备的安装，操作步骤如下：

1、 在“驱动设备管理”中选择“新建”项，或在浏览器中双击“设备安装向导”则弹出设备安装向导对话框如图 7-9 所示：

图 7-9

用户从树形设备列表区中可选择一个板卡设备。然后选择要配置板卡设备的生产厂家、设备对象名称。

2、单击“下一步”按钮，则弹出设备安装向导对话框如图 7-10 所示：

设备对象名称：要安装板卡设备的名称。
板卡设备地址、初始化字：用户要为板卡设备指定板卡地址、初始化字。

端 口 选 择：转换器的输入方式（单端或双端）。

其中初始化字是针对具有 8255 芯片的板卡而言的，因为 8255 芯片同时具有 DI（数字量输入）和 DO（数字量输出）功能，而初始化字将决定它到底是 DI 还是 DO。下表为 8255 芯片的初始化表：

图 7-10

	D4	D3	D1	D0
CW	PA0-PA7	PC4-PC7	PB0-PB7	PC0-PC3
80H	output	Output	Output	Output
81H	Output	Output	Output	Input
82H	Output	Output	Input	Output
83H	Output	Output	Input	Input
88H	Output	Input	Output	Output
88H	Output	Input	Output	Input
8AH	Output	Input	Input	Output
8BH	Output	Input	Input	Input
90H	Input	Output	Output	Output
91H	Input	Output	Output	Input
92H	Input	Output	Input	Output
93H	Input	Output	Input	Input
98H	Input	Input	Output	Output
99H	Input	Input	Output	Input

9AH	Input	Input	Input	Output
9BH	Input	Input	Input	Input

而单端、双端是为了适应不同信号的要求，模拟信号放大器不同的使用方法，若要修改板卡设备的逻辑名称，单击“上一步”按钮，则可返回上一个对话框。（具体含义见联机帮助）。

单击“上一步”按钮，则可返回上一个对话框进行修改。

3、继续单击“下一步”按钮，则弹出设备配置向导对话框如图 7-11 所示：

图 7-11

此向导页显示已配置的板卡设备的设备信息，供用户查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，完成设备安装。在浏览器的“当前驱动设备”中将增加新安装的设备。

7.3.3　DDE 动态数据交换设备安装向导

根据设备安装向导就可以完成 DDE 动态数据交换的安装，操作步骤如下：

1、 在“驱动设备管理”中选择“新建”项，或在浏览器中双击“设备安装向导”则弹出设备安装向导对话框如图 7-12 所示：

图 7-12

用户从树形设备列表区中可选择一个 DDE 动态数据交换设备。然后选择要配置设备的设备对象名称、服务程序名、话题名等。

2、 单击“下一步”按钮，则弹出设备安装向导对话框如图 7-13 所示：

设备对象名称： 要安装 DDE 动态数据交换设备的名称。

服务程序名： 是与“世纪星”交换数据的程序名，一般是 I/O 服务程序，或者是 Windows 应用程序。

话　题　名： 是本程序和服务程序进行 DDE 连接的话题名（Topic）。

数据交换协议： 是指 DDE 会话的两种方式：I/O 驱动程序交换协议是本公司开发的 DDE 动态数据交换程序采用的协议；标准的 WINDOWS 交换协议是按照标准的 Windows DDE 交换协议开发的 DDE 服务程序。

单击“上一步”按钮，则可返回上一个对话框进行修改。

图 7-13

3、 继续单击“下一步”按钮，则弹出设备配置向导对话框如图 7-14 所示：

此向导页显示已配置的 DDE 动态数据交换设备的设备信息，供用户查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，完成设备安装。在浏览器的“当前驱动设备”中将增加新安装的设备。

7.3.4　网络设备安装向导

根据设备安装向导就可以完成网络设备的安装，操作步骤如下：

1、在“驱动设备管理”中选择“新建”项，或在浏览器中双击“设备安装向导”则弹出设备安装向导对话框如图 7-15 所示：

图 7-14

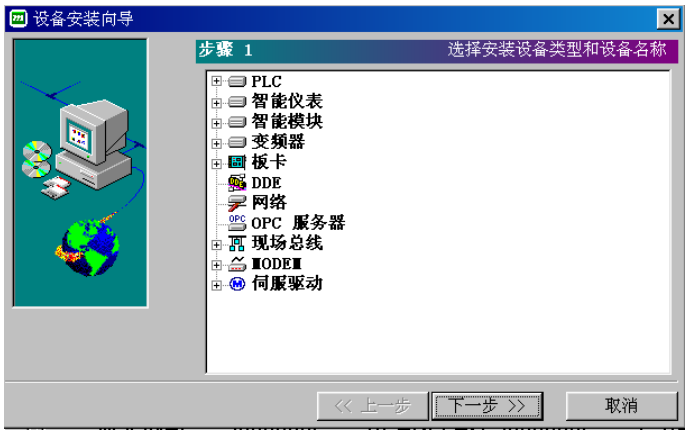


图 7-15

用户从树形设备列表区中可选择一个网络设备。然后选择要配置设备的设备对象名称、网络设备参数等。

2、 单击“下一步”按钮，则弹出设备安装向导对话框如图 7-16 所示：

设备对象名称：输入一个名称以标志该设备对象名。
网络设备参数：输入需要连接的对方接点计算机名或其 IP 地址。

当使用 ET-LAN 网络模块单元与上位机相连时 ，PLC 地址在世纪星中的设置方法是在设备地址中输入 IP 地址号:PLC 站号。例如，PLC 的 IP 地址是 223.103.10.90，PLC 站号是 1，那么输入 223.103.10.90:1。IP 地址由用户根据实际要求设置，PLC 的站号范围从 1 到 64。

图 7-16

ET-LAN 单元 IP 地址号与 PLC 站号的设置在 ET-LAN 单元及对应型号的 PLC 使用说明书中有详细解释。

3、继续单击“下一步”按钮，则弹出设备配置向导对话框如图 7-17 所示：

此向导页显示已配置的网络设备的设备信息，供用户查看，如果需要修改，单击“上一步”按钮，则可返回上一个对话框进行修改，如果不需要修改，单击“完成”按钮，完成设备安装。在浏览器的“当前驱动设备”中将增加新安装的设备。

图 7-17

7.4 I/O 设备驱动修改和删除

在驱动设备管理对话框中，用户可对串口通信设备（包括 PLC 设备、智能仪表设备、智能模块设备和变频器设备等）、板卡、DDE 设备和网络设备等进行必要的修改或删除操作。

7.4.1 PLC 设备修改和删除

若对 PLC 设备进行修改或删除，则弹出如图 7-18 所示对话框：

设备生产厂家：列出设备生产厂家名称。
设 备 名 称：列出设备系列名称。
通 讯 方 式：列出设备采用的通讯方式。
设备对象名称：要安装串口通信设备的名称。
通 信 端 口：串口设备与计算机相连的串口号，该下拉式串口列表框共有 32 个串口号供用户选择。
设 备 地 址：串口通信设备的设备地址。

尝试恢复间隔：在世纪星运行期间，如果有一台设备如 PLC1 发生故障，则世纪星能够自动诊断并停止采集与该设备相关的数据，但会每隔一段时间尝试恢复与该设备的通信，如图所示，尝试时间间隔为 3 秒。

最长恢复时间：若世纪星在一段时间之内一直不能恢复与PLC1的通信，则不再尝试恢复与PLC1通信，这一时间就是指最长恢复时间。

使用动态优化：世纪星对全部通信过程采取动态管理的办法，只有在数据被上位机需要时才被采集。

图 7-18

7.4.2　智能仪表设备修改和删除

若对智能仪表设备进行修改或删除，则弹出如 图 7-19 的对话框：

窗口中有关内容的说明同 PLC 设备，在此不在详述。

7.4.3 智能模块设备修改和删除

若对智能模块设备进行修改或删除，则弹出如图 7-20 所示对话框：

图 7-19

窗口中有关内容的说明同 PLC 设备，在此不在详述。

7.4.4 变频器设备修改和删除

若对变频器设备进行修改或删除，则弹出如图 7-21 所示对话框：

窗口中有关内容的说明同 PLC 设备，在此不在详述。

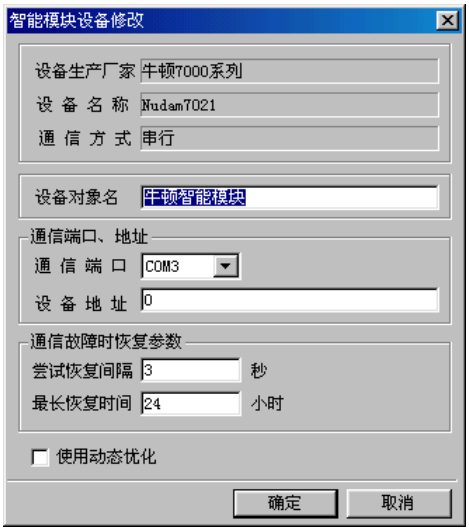

智能模块设备修改
设备生产厂家 牛顿7000系列
设 备 名 称 Nudam7021
通 信 方 式 串行
设备对象名 牛顿智能模块
通信端口、地址
通 信 端 口 COM3
设 备 地 址 0
通信故障时恢复参数
尝试恢复间隔 3 秒
最长恢复时间 24 小时
使用动态优化
确定
取消

图 7-20

变频器设备修改
设备生产厂家 西门子
设 备 名 称 SIEMENS
通 信 方 式 串行
设备对象名 西门子变频器
通信端口、地址
通 信 端 口 COM4
设 备 地 址 0
通信故障时恢复参数
尝试恢复间隔 3 秒
最长恢复时间 24 小时
使用动态优化
确定
取消

图 7-21

7.4.5 板卡设备修改和删除

若对板卡设备进行修改或删除，则弹出如图 7-22 所示对话框：

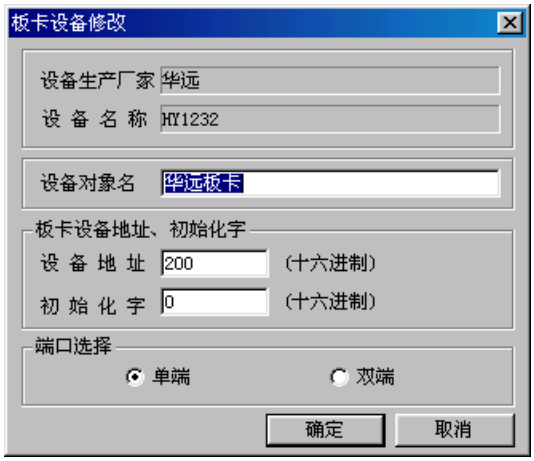


图 7-22

设备生产厂家：列出设备生产厂家名称。
设 备 名 称：列出设备系列名称。
设备对象名称：要安装板卡设备的名称。
板卡设备地址、初始化字：用户要为板卡设备指定板卡地址、初始化字。
端 口 选 择：转换器的输入方式（单端或双端）。

7.4.6 DDE 设备修改和删除

若对 DDE 设备进行修改或删除，则弹出如图 7-23 所示对话框：

设备对象名称：要安装 DDE 动态数据交换设备的名称。
服务程序名：是与“世纪星”交换数据的程序名，一般是 I/O 服务程序，或者是 Windows 应用程序。
话 题 名：是本程序和服务程序进行 DDE 连接的话题名（Topic）。

数据交换协议：是指 DDE 会话的两种方式：I/O 驱动程序交换协议是本公司开发的 DDE 动态数据交换程序采用的协议；标准的 WINDOWS 交换协议是按照标准的 Windows DDE 交换协议开发的 DDE 服务程序。

图 7-23

7.4.7　网络设备修改和删除

若对网络设备进行修改或删除，则弹出如图 7-24 所示对话框：

图 7-24

设备对象名称：要安装 DDE 动态数据交换设备的名称。

网络设备参数：输入需要连接的网络节点机器名。

7.5 I/O 仿真驱动系统

世纪星监控组态软件系统 I/O 仿真驱动软件 SIMULATE 是一个 Windows 应用程序，可模拟下位机向用户应用程序提供数据，为用户程序的调试提供方便。

7.5.1 仿真驱动程序的菜单命令

配置

图 7-25

图 7-25 菜单命令用来设置仿真驱动程序运行所必需的参数。

系统配置

选择此菜单，弹出“系统配置”对话框，如图 7-26 所示。

设置文件路径：在此编辑框内输入需要调试的应用程序的工程路径。仿真驱动程序在此路径下查找所有与系统运行相关的文件。若工程路径不正确则仿真驱动程序不能正确运行。当然，用户还可以将设置的文件路径定为缺省文件路径名。

通 信 频 率：仿真驱动程序与外设进行数据交换的工作频率。

会话定义

选择此菜单，弹出“会话数据定义”对话框，如图 7-27 所示。

图 7-26

图 7-27

项 目 名：仿真驱动程序需要仿真的项目变量名。

注　　释：关于项目名变量的有关说明。

仿真数据

数据类型：可定义仿真数据的数据类型，如：I/O离散，I/O整数，I/O实数，I/O信息等。当然用户可以通过查阅变量字典来进行数据类型的定义。

最 小 值：规定仿真程序产生数据的的最小值。

最 大 值：规定仿真程序产生数据的的最大值。

递　　增：仿真程序提供的数据从“最小值”递增到“最大值”，递增量为 1。

递　　减：仿真程序提供的数据从“最大值”递减到“最小值”，递减量为-1。

随　　机：仿真程序提供的数据在“最大值”和“最小值”之间随机生成。

设　　置：可设置仿真值为“关[0, FALSE]”或“开[1,TRUE]”。

更新频率：指定从外设采集数据的时间间隔，在仿真驱动程序中就是仿真程序向“世纪星监控组态软件”系统提供数据的时间间隔。

7.6　“串行通讯方式”概述

这是世纪星与 I/O 设备之间最常用的一种数据交换方式。串行通讯方式使用“世纪星计算机”的串口，I/O 设备通过 RS-232 串行通讯电缆连接到“世纪星计算机”的串口。如果您的计算机拥有多个串口，您可以同时与多个 I/O 设备。世纪星最多可与 32 个串口设备相连。如图 7-28 所示。

任何具有串行通讯接口的 I/O 设备都可以采用此方式。大多数的可编程控制器(PLC)、智能模块、智能仪表采用此方式。

7.6.1　连接一个串口

最简单的情况下，使用世纪星的计算机只与一个 I/O 设备相连。I/O 设备使用标准的 RS-232 电缆与计算机主机后面的串口连接。如图 7-29 所示。

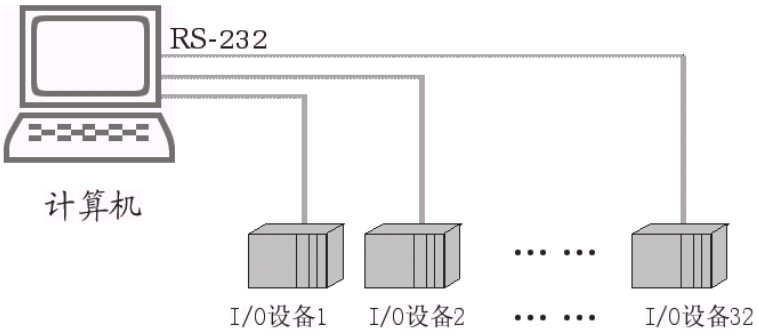


图 7-28

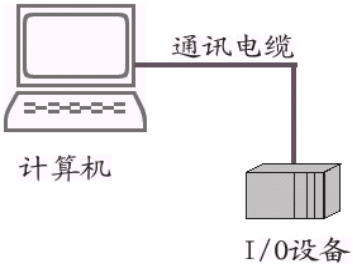


图 7-29

当世纪星计算机与多个 I/O 设备相连时，由于 RS-232 是一个点对点的标准，可以将 RS-232 转化为 RS-485，再经过一次 RS-485 到 RS-232 的转化，实现一个计算机串口连接多个 I/O 设备。如图 7-30 所示：

7.6.2　连接多个串口

当世纪星计算机有多个串口时，可以直接连接多个串行通讯方式的 I/O 设备。世纪星最多支持 32 个串口。如图 7-31 所示。

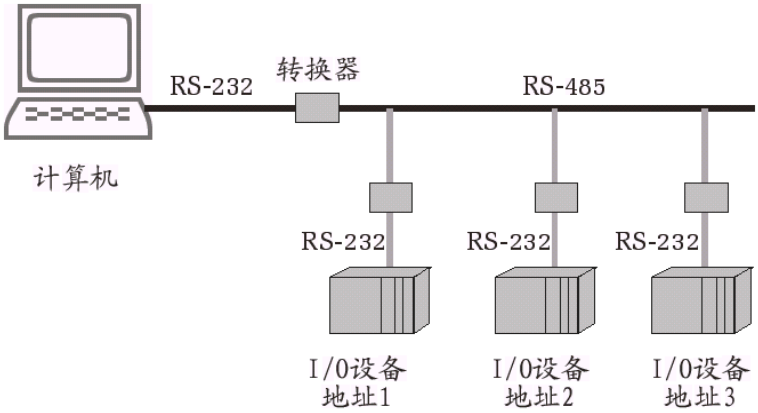


图 7-30

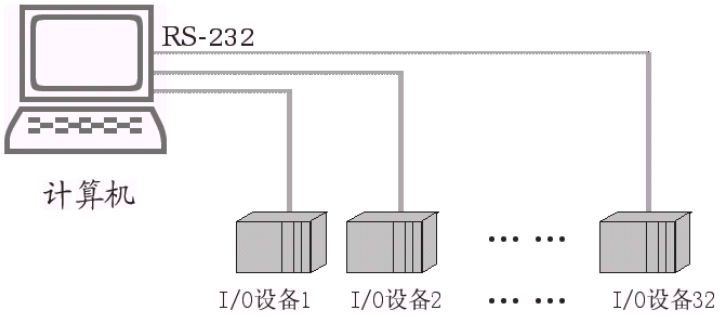


图 7-31

7.7 RS-232 简介

RS-232 又称为 EIA-232C 或 RS-232C，是最通用的一种串行通讯标准。它是一种点到点的通讯方式，只能连接两个通讯设备。通讯是全双工的，最简单的情况

下只需 3 根线。19200 波特率时，最大距离为 75 米；900 波特率时，最大距离为 900 米。计算机的串口即为标准的 RS-232 接口。如图 7-32、7-33 所示。

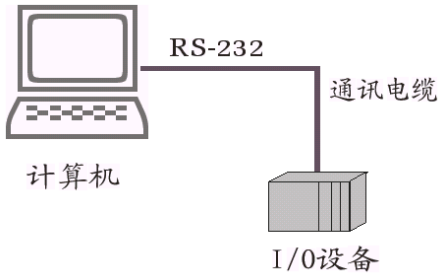


图 7-32

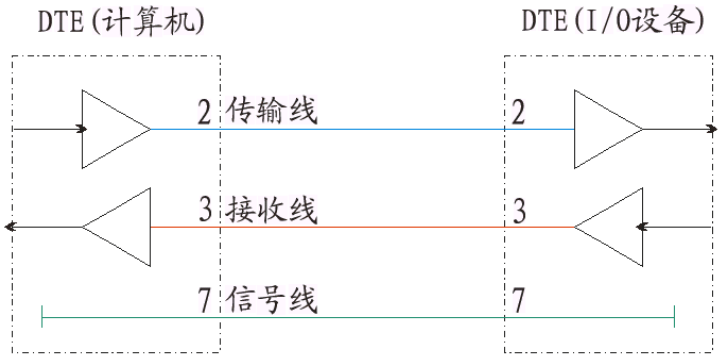


图 7-33

7.7.1　9 针到 25 针转换示意图

RS-232 一般为 25 针，如果需要将 9 针转化为 25 针，参见 图 7-34：

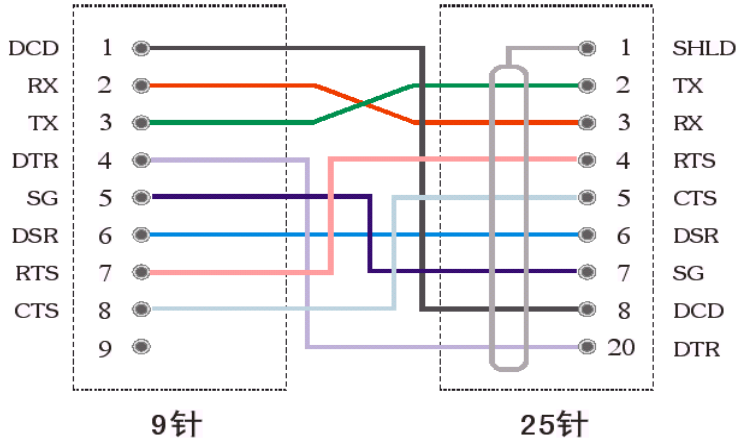


图 7-34

7.7.2 RS232/RS485

使用 RS232/485 转换器可以免掉一个 RS-422/485 串行接口板。与 RS-232 相比，RS-485 传输距离长、传输速率高，而且支持多点连接。参见 图 7-35：

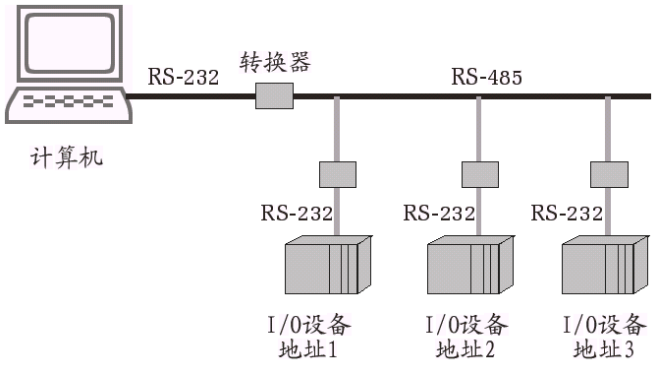


图 7-35

第八章 工程配方管理

8.1 工程配方及工程配方管理简介

在制造领域，工程配方是用来描述生产一件产品所用的不同配料之间的比例关系，是生产过程中一些变量对应的参数设定值的集合。例如，一个水泥制品厂生产水泥制品时有一个基本的配料配方，此配方列出所有要用来生产水泥制品的配料成分表（如水泥，沙石，水，添加剂等），而这些可选配料成分可以被添加到基本配方中用以产生各种各样的水泥制品。下表为某一水泥制品厂生产水泥制品时的配方：

	配方 1	配方 2	配方 3
配料名称	预制板	水管	电线杆
水泥	28	24.5	31.5
沙石	95	98	89
水	71	65	68
添加剂	2.9	1.6	3.6
钢筋	36	27	45

注：表中的数字是示意性的。

又如，在供电局里，一个工程配方可能就是机器设置参数的一个集合，而对于批处理，一个工程配方可能被用来描述批处理过程中的不同步骤。

世纪星支持对工程配方的管理，用户利用此功能可以在控制生产过程中得心应手，提高效率。比如，当生产过程需要大量的控制参数时，如果一个接一个地设置这些变量参数就会耽误时间，而使用工程配方，则可以一次设置大量的控制变量参数，满足生产过程的需要。

8.2 工程配方模板的定义

所有工程配方都在工程配方模板文件中定义和存储。每一个工程配方模板文件以扩展名为 csv 的文件格式存储，一个工程配方模板文件是通过工程配方定义模板产生的。

定义工程配方模板文件时需要定义所有的变量名（即项目名）、变量类型（即项目类型）、工程配方名及该工程配方所对应的所有变量的配方值。

工程配方模板的结构如下：

变量名 (项目名)	变量类型（项目类型）	配方名 1	配方名 N
变量 1	离散类型、整数类型、实数类型、信息类型	配方值	配方值
变量 2	离散类型、整数类型、实数类型、信息类型	配方值	配方值
变量 3	离散类型、整数类型、实数类型、信息类型	配方值	配方值
变量 N	离散类型、整数类型、实数类型、信息类型	配方值	配方值

注：“配方名 N”由用户设定，具体的“配方值”表示变量的数值大小。

8.3 使用工程配方管理

8.3.1 创建工程配方模板文件

世纪星工程管理器能够创建和管理工程配方模板文件。在工程管理器中选“工程配方管理”选项弹出如图 8-1 所示对话框：

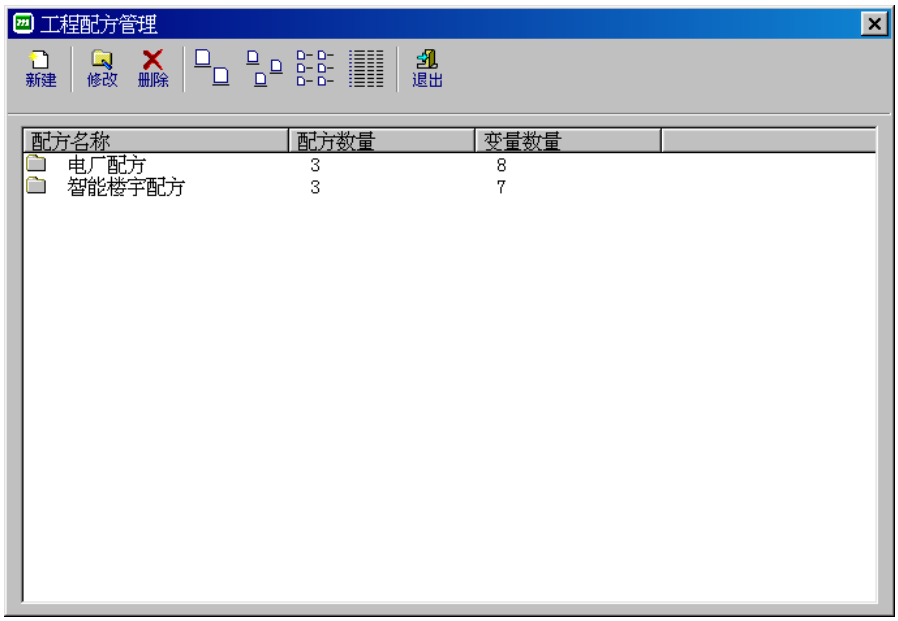


图 8-1

单击“新建”图标，弹出“新建工程配方”对话框，如图 8-2 所示。

图 8-2

输入配方名后单击“确定”按钮，弹出“工程配方定义”对话框，如图 8-3 所示：

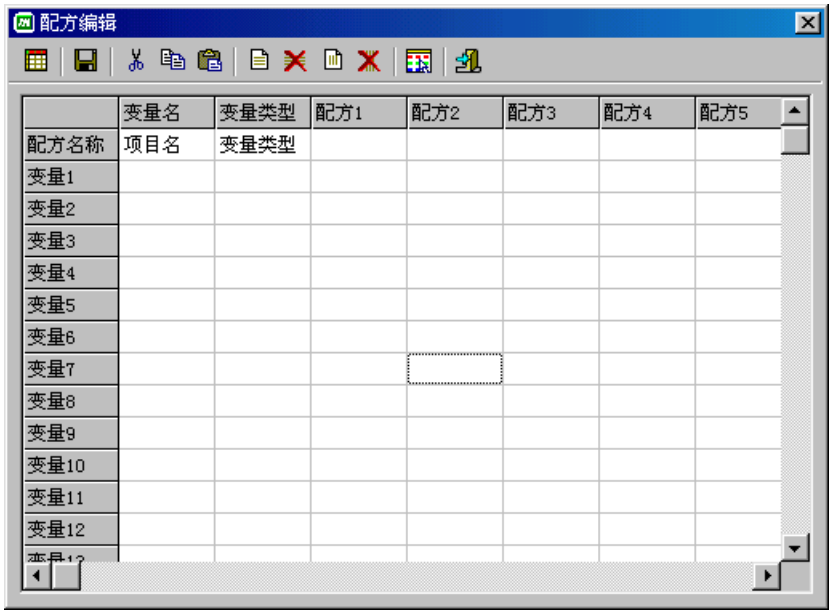


图 8-3

项 目 名：用户定义的变量名。

变量类型：为实数型、整数型、离散型、字符串中的一种。

配方名N：设定一组与变量相对应的配方值，其他每一个配方也有一组配方值与之对应。

使用工程管理器中的工程配方定义模板来生成工程配方模板文件的方法方便而简单，只需要在相应表格中填入项目名、变量类型、变量名、配方值即可自动创建 csv 格式的工程配方模板文件。

还可以使用 WINDOWS 操作系统中的写字板编写。使用这种方法要按照 csv 格式自己编写，因此比前一种方法要复杂一些。

工程配方模板文件的 csv 格式如下：

“项目名”，　“变量类型”，　“配方名 1”，“配方名 2”，　“配方名 N”，
“变量 1”，　“ 变量类型”，　“配方值”，　“配方值”，　“配方值”，
“变量 2”，　“ 变量类型”，　“配方值”，　“配方值”，　“配方值”，
“变量 N”，　“ 变量类型”，　“配方值”，　“配方值”，　“配方值”，

在这种格式中每一项都要有双引号，而且每一项之间要用逗号隔开，变量类型有实数类型、整数类型、离散类型、信息类型四种。“配方值”为特定配方的特定配料（变量）的类。

例如：用写字板写一个 csv 格式的工程配方模板文件“建材生产厂”，编辑过程如图 8-4 所示：

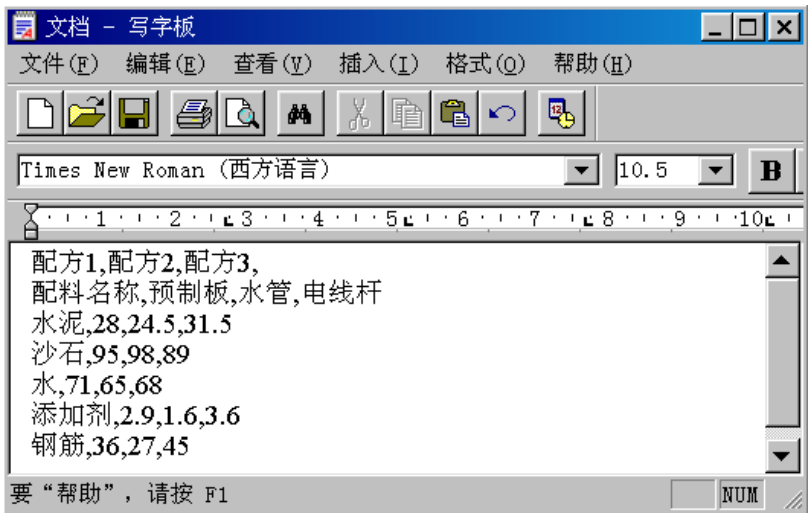


图 8-4

8.3.2　调用工程配方管理函数

工程配方模板文件中的工程配方定义模板完成以后，在世纪星系统中运行时

可以通过工程配方管理函数进行各种工程配方的调入、修改等操作。对工程配方模板文件的操作由工程配方管理函数来完成。

8.4 工程配方管理函数

世纪星的工程配方管理函数还是标准命令语言函数的扩展，可以被用于任何形式的命令语言中，通过这些函数用户可以对已有的工程配方模板文件进行选择、修改、调用、插入、删除等操作。

工程配方管理函数的一般使用格式为：

RecipeFunction(Argument1,Argument2,ArgumentN);

或

RecipeName= RecipeFunction(Argument1,Argument2,ArgumentN);

世纪星提供的工程配方管理函数如下，详细内容请参阅有关函数的相关章节：

(1) RecipeDelete
(2) RecipeLoad
(3) RecipeSave
(4) RecipeSelectRecipe
(5) RecipeSelectNextRecipe
(6) RecipeSelectPreviousRecipe

第九章　世纪星监控组态软件运行系统

世纪星软件包由开发系统 CSMAKER 和运行系统 CSVIEWER 两部分组成。在 CSMAKER 系统中设计开发的画面应用程序必须在 CSVIEWER 运行环境中才能运行。CSMAKER 和 CSVIEWER 各自独立，一个应用程序可以同时被编辑和运行，这对于程序的调试是非常方便的。

9.1　文件菜单

单击运行系统 CSVIEWER 中菜单条上的“文件”菜单项，打开下拉式菜单如图 9-1：

图 9-1

文件菜单主要菜单命令有：打开画面、关闭画面、画面打印、报表和退出运行系统等。

9.1.1　打开画面

选择此命令后，将在对话框中列出当前路径下所有未打开画面的清单。用鼠标选择一个或多个窗口后，单击“确定”按钮打开所有选中的画面，或单击“取

消”放弃当前操作。如图 9-2 所示:

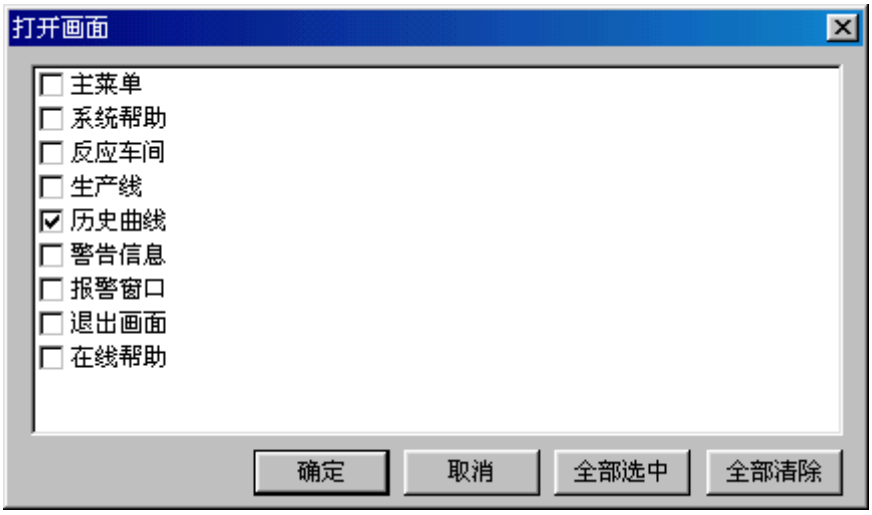


图 9-2

9.1.2 关闭画面

选择此命令后，将在对话框中列出所有已打开画面的清单。用鼠标选择一个或多个窗口后，单击“确定”按钮关闭所有选中的画面，或单击“取消”废弃当前操作。如图 9-3 所示。

9.1.3 画面打印

画面打印功能是打印运行环境中的当前画面，点击“画面打印”菜单，系统弹出打印设置对话框，操作员可根据需要选择打印或退出打印。

9.1.4 报表

选择报表，系统弹出如图 9-4 所示：

世纪星报表包括：历史数据报表、实时数据报表、历史报警报表、组态报表、事件查询报表等，有关这种报表的具体使用参看相关章节。

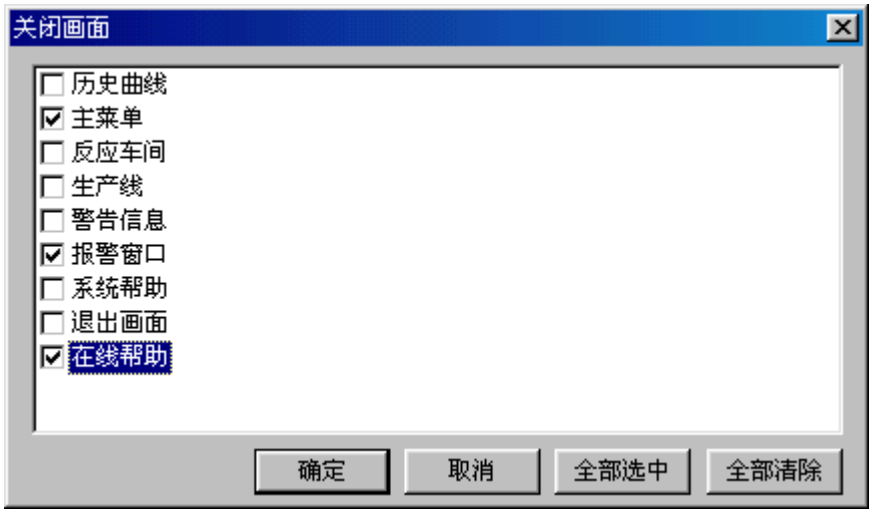


图 9-3

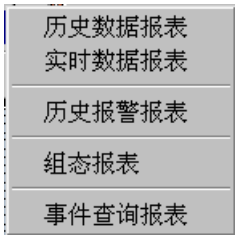


图 9-4

9.1.5　退出

选择此命令后退出世纪星运行程序 CSVIEWER。

9.2　系统菜单

单击运行系统 CSVIEWER 中菜单条上的“系统”菜单项，打开下拉式菜单如图 9-6：

图 9-6

系统菜单主要菜单命令有：系统安全、重建 DDE 连接、重建未成功的 DDE 连接、重新启动报警历史记录、重新启动历史数据记录、停止历史数据记录等。

9.2.1 系统安全

系统安全命令项是专为用户提供有关系统安全操作的相关事项。单击菜单条上的“系统安全”菜单项，弹出如图 9-7 所示的菜单：

图 9-7

系统安全菜单主要有用户登录、改变密码、配置用户和退出登录等功能。

9.2.1.1 用户登录

此选项用于打开有密码设置的图形元素或对象。在 CSVIEWER 运行环境下，当运行画面打开后，单击此选项，则弹出如图 9-8 所示对话框：

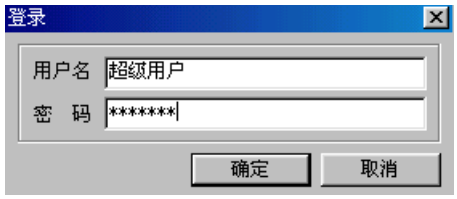


图 9-8

9.2.1.2　改变密码

此选项用于修改图形元素或对象的密码设置，在 CSVIEWER 运行环境下，当运行画面打开后，单击此选项，则弹出如图 9-9 所示对话框：

图 9-9

9.2.1.3　配置用户

此选项用于重新设置用户的访问权限和密码，当操作员的访问权限大于 9000 时，此选项有效。如图 9-10 所示：

9.2.1.4　退出登录

此选项用于退出当前用户登录，当前用户变成“NONE”，使当前用户的访问级别为 0。

图 9-10

9.2.2 重建 DDE 连接

CSVIEWER 先中断已经建立的 DDE 连接，然后重新建立 DDE 连接。

9.2.3 重建未成功的 DDE 连接

重新建立启动时未建立成功的 DDE 连接。已经建立的 DDE 连接不受影响。

9.2.4 重新启动报警历史记录

此选项用于重新启动报警历史记录。在没有空闲磁盘空间时，系统自动停止报警历史记录。当发生此种情况时，将显示一信息框，通知用户。为了重启报警历史记录，用户须清理出一定的磁盘空间，并选择此命令。

9.2.5　重新启动历史数据记录

此选项用于重新启动历史数据记录。在没有空闲磁盘空间时，系统自动停止历史数据记录。当发生此种情况时，将显示一信息框，通知用户。为了重启历史数据记录，用户须清理出一定的磁盘空间，并选择此命令。

9.2.6　停止历史数据记录

此选项用于用户干涉停止历史数据记录。

9.3　关于菜单

单击画面运行系统 CSVIEWER 中菜单条上的“关于”菜单项，打开下拉式菜单：

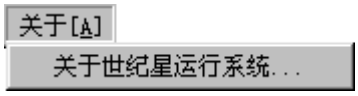


图 9-11

关于世纪星监控组态软件

菜单命令项用于显示世纪星监控组态软件的版权信息，弹出如图 9-12 所示窗口：

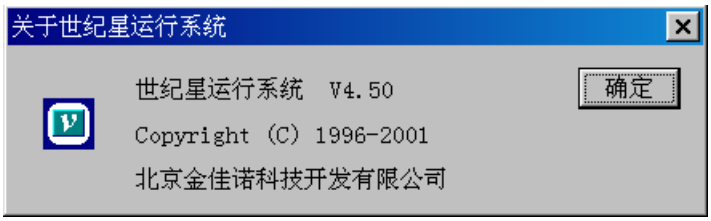

关于世纪星运行系统
世纪星运行系统 V4.50
Copyright (C) 1996-2001
北京金佳诺科技开发有限公司
确定

图 9-12

第十章　　组态报表与打印功能

10.1 世纪星打印功能综述:

在世纪星中，提供丰富的画面打印及报表功能，有画面图形打印、历史数据打印、历史曲线打印、报警事件打印、实时数据打印、组态报表等共六种打印功能。分别对应于“CSVIEWER”中“FILE”菜单下的五个打印子功能及 PrintHistData 函数。其中的历史数据打印功能可以按数据输出、也可以按图形输出。其对应关系如图 10-1 所示：

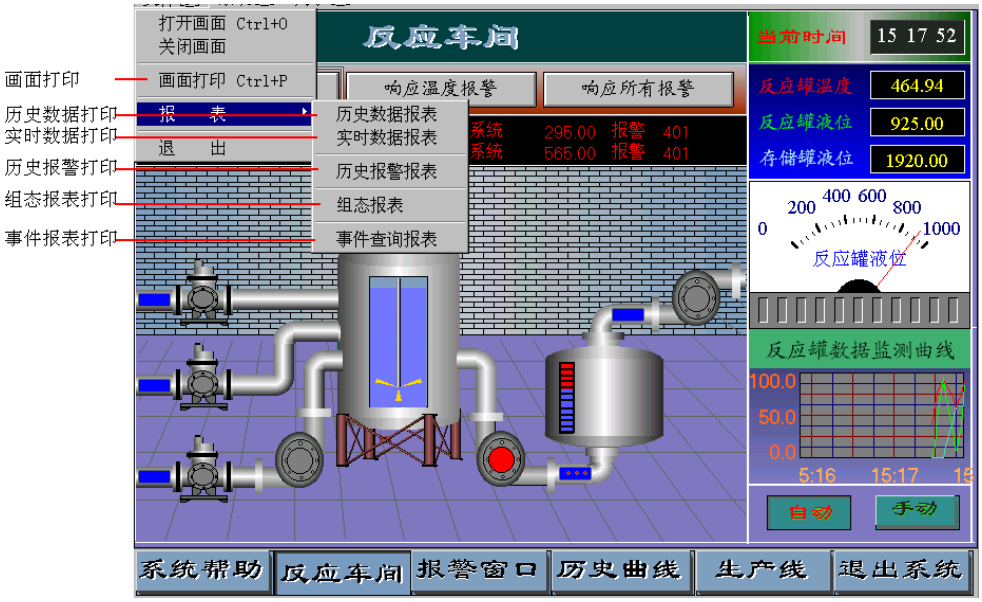


图 10-1

因为打印功能适用于大多数应用场合。因此，世纪星中的打印功能是运行系

统框架内置的功能。也就是说在开发一个工程时，无需设计打印功能也可以实现打印。当然，通过命令语言也可以实现“打印按钮”或软件“定时自动打印”。打印功能自动调取 Windows 的打印设备并支持网络打印，无需单独为某种打印机编写打印驱动程序。

10.1.1 画面打印

世纪星的画面打印功能既可以适用于开发系统，也适用于运行系统。其打印范围为“客户区”的全部,而不包括窗口和框架。它们分别位于开发系统和运行系统中的“文件”菜单下。参见打印功能综述。

10.1.2 历史数据打印

参见打印功能综述，使用历史数据打印功能会出现下面的对话框。只需要指定变量、起止日期、时间间隔，然后按“报表生成”就完成了报表提取工作。而打印时，可以按文本输出也可以按曲线输出。曲线打印最多可打印六条，且每根曲线都可选择线型和线宽，以区别不同的曲线所对应的变量。

历史数据的提取有赖于历史数据的保存，可以在开发系统中的“系统设置→历史数据记录配置”菜单下找到相应的历史数据保留天数设定，具体参见图 10-2 历史记录配置。

10.1.3 实时数据打印

实时数据打印是打印当前时刻，指定变量的数值及报警状态。这里只需指定变量然后按“报表生成”后，如图 10-3 所示：

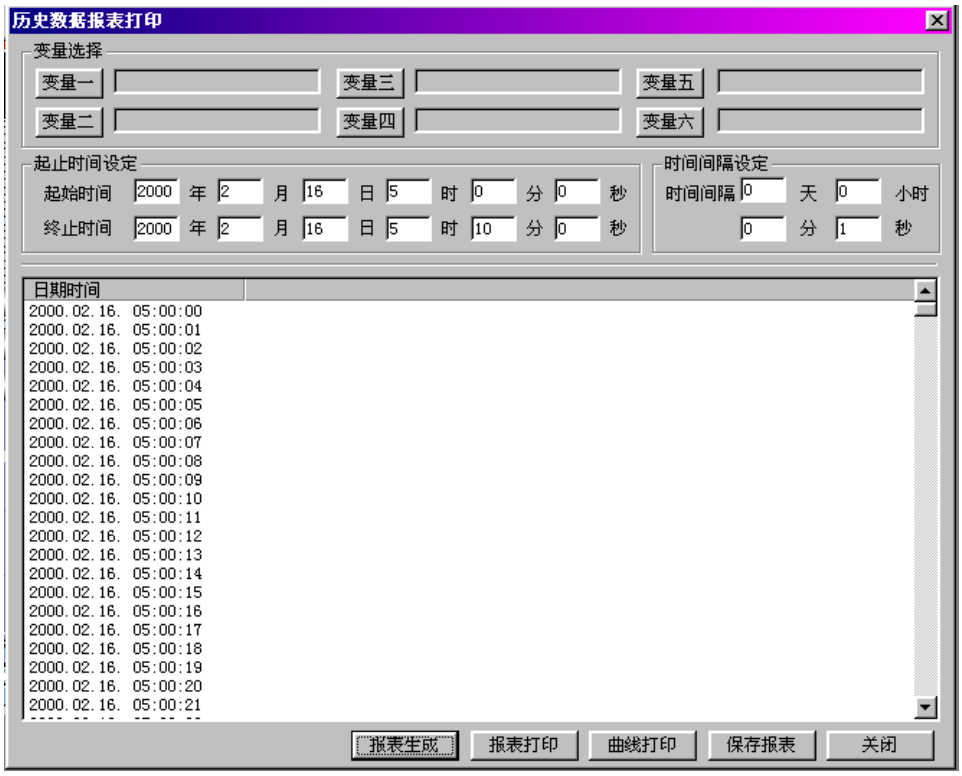


图 10-2

10.1.4　历史报警打印：

历史报警报表打印需指定需打印相关变量报警的变量名或指定需打印报警的优先级范围，在指定时间范围后按“报表生成”按钮后生成历史报警报表。其报警格式在开发系统中的“系统配置→历史报警配置”菜单下可以找到相应的配置（报警记录格式和打印格式），具体参见报警记录配置。按“报表打印”就可以实现报表输出。如图 10-4 所示：

10.2 事件查询报表

在运行系统中选中“报表→事件查询报表”或调用函数 EventsLogQuery，系统弹出图 10-5 菜单：

可在“按事件查询”中选择你要查询的事件，并可打印生成的报表。

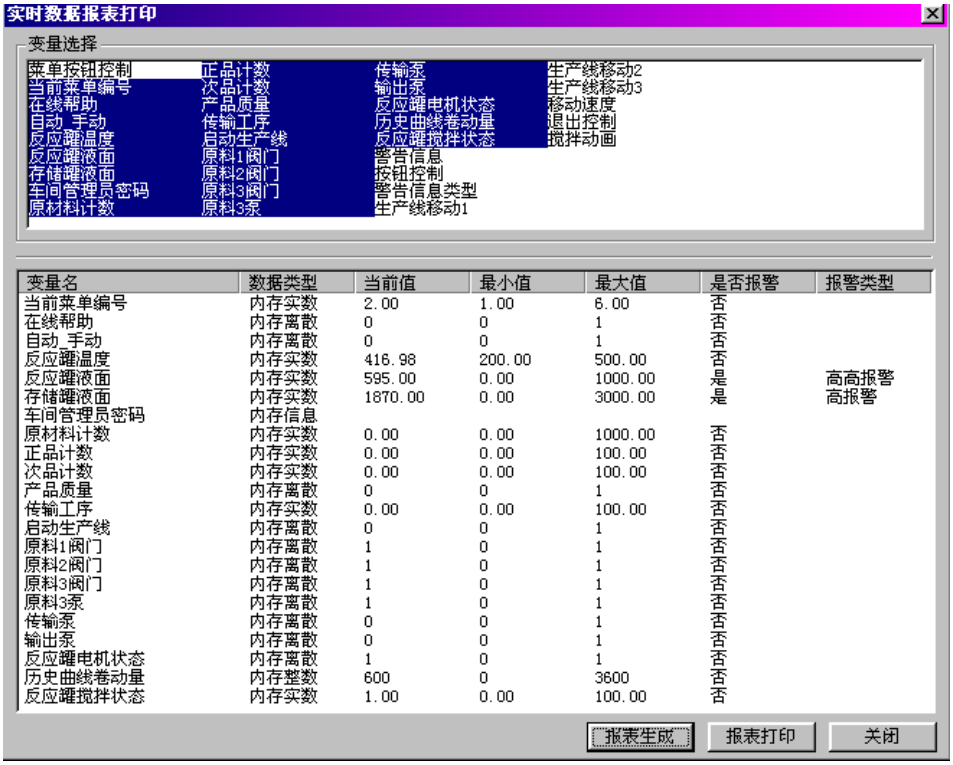


图 10-3

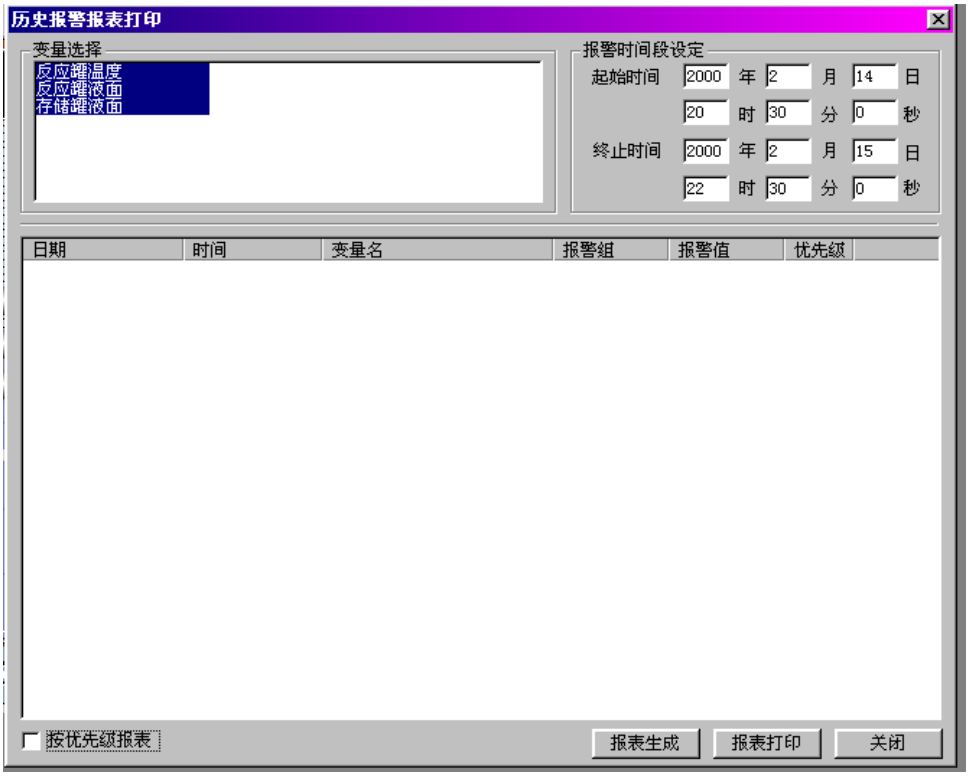


图 10-4

10.3 使用命令语言实现定时打印：

命令语言中的“PrintHistData”函数支持对历史数据的打印。其格式如下：

PrintHistData（”变量 1”，”变量 2”，”变量 3”，”变量 4”，”变量 5”，”变量 6”，起始时间，终止时间，间隔时间，打印方式）；

变量 1～6：指定的打印变量名；
起始时间：70 年 1 月 1 日到起始指定的时间。单位：秒。

终止时间：70 年 1 月 1 日到结束指定的时间。单位：秒。

间隔时间：显示数据的时间间隔。单位：秒。

打印方式：0…不设置打印参数，直接输出。

1…打印报表。

2…打印曲线。

3…打印报表及曲线。

4…调用调用对话框。由用户定制输出打印。

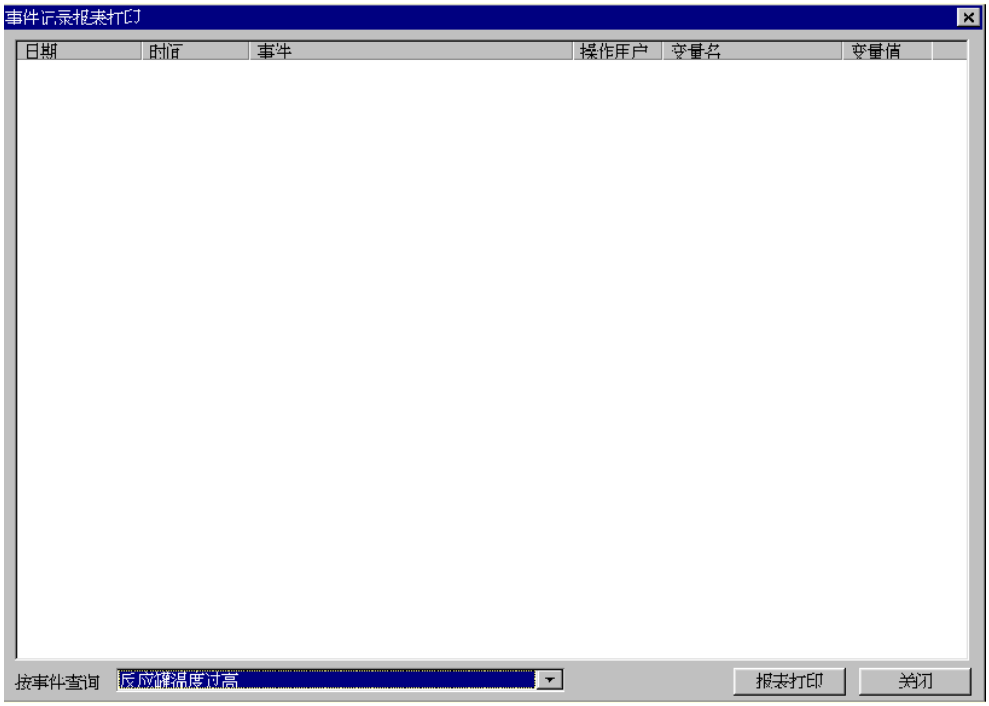


图 10-5

编程实例：

```
IF ($时==8) && ($分==0) && ($秒==0) THEN
    PrintHistData ("变量 1","变量 2","变量 3","变量 4","","",($日
```

```
                                    期-1)*24*3600,$日期*24*3600,60,2);
        ENDIF
```

说明：

在“应用程序命令语言”中的“程序运行时”里加上上面的几句命令语言后，每日的上午 8 点程序执行自动打印“变量 1”到“变量 4”的上一天运行曲线。

10.4. 组态报表模板

世纪星提供特有的组态报表，开发人员可在世纪星的开发系统中用组态报表模板中建立模板，在运行系统中，调用组态报表模板以生成组态报表，用于打印和浏览组态报表。

在世纪星的开发系统中，选择“文件→组态报表模板”，弹出编辑组态报表模板对话框，如图 10-6 所示：

为方便使用，世纪星组态报表模板的使用方法有些类似电子表格，每个单元用所在的行列数唯一标识，行用 1、2、3...100 等标识，缺省最大行数为 100，可增加行、列用 A、B、C、...Z 等标识，缺省最大列到 Z，可增加列，增加后的列用 AA、AB、AC...等标识。要在某单元中输入数据，应用鼠标左键双击该单元。图 10-7 是组态报表模板的按钮功能示意图：

在世纪星组态报表模板中，各功能按钮介绍如下：

“新模板”、“打开模板”、“保存模板”、“浏览”、“剪切”、“拷贝”、“粘贴”、“撤消”、“重复”、“字体”、“退出”等十一个功能按钮的用法与其他 Windows 应用程序的用法一样，这里就不多作介绍。

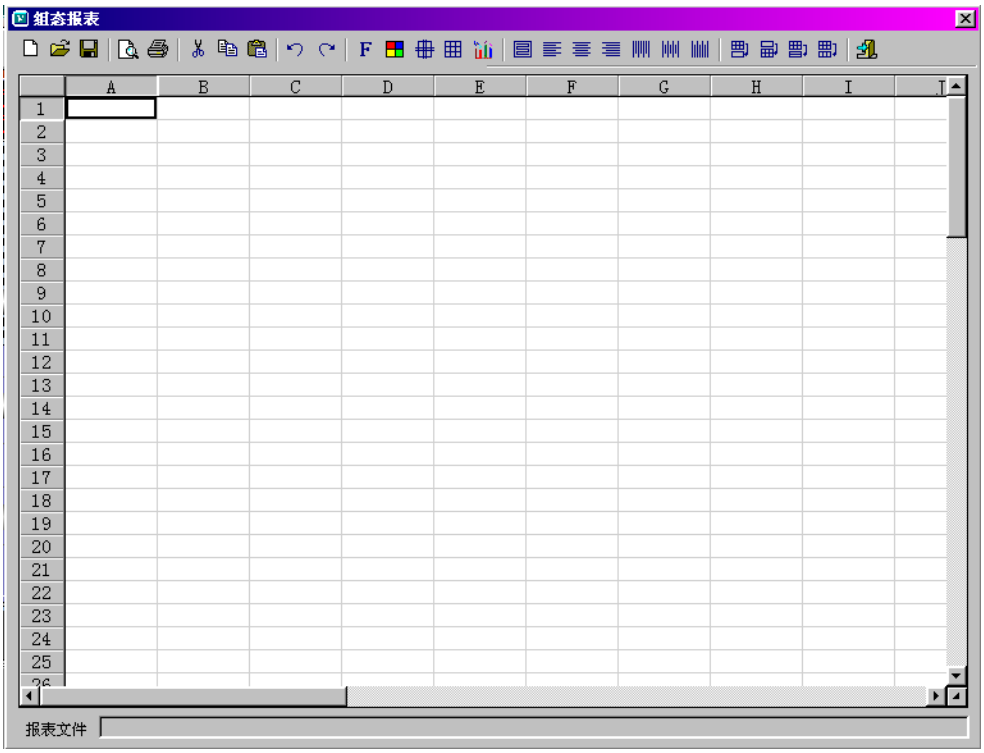

组态报表
报表文件

图 10-6

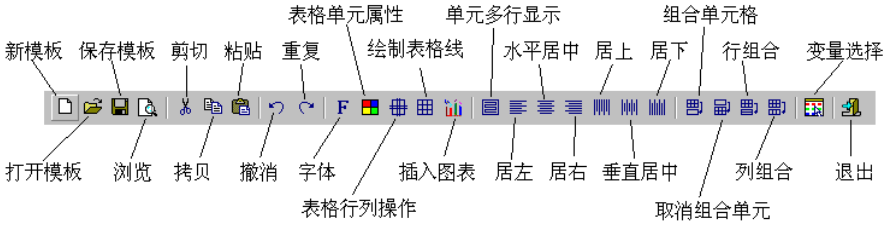

表格单元属性
单元多行显示
组合单元格
新模板
保存模板
剪切
粘贴
重复
绘制表格线
水平居中
居上
居下
行组合
变量选择
打开模板
浏览
拷贝
撤消
字体
插入图表
居左
居右
垂直居中
列组合
退出
表格行列操作
取消组合单元

图 10-7

表格单元属性

在选择一个或多个表格单元后，单击“表格单元属性”按钮后，弹出如下图所示对话框，在该对话框中，您可以定义已选定的单元的颜色设置（文本颜色、背景颜色）、数据格式（数值用科学计数法、数值用逗号分隔、数值转换为百分数、只读属性）、小数点位数等属性，如下图 10-8 所示。

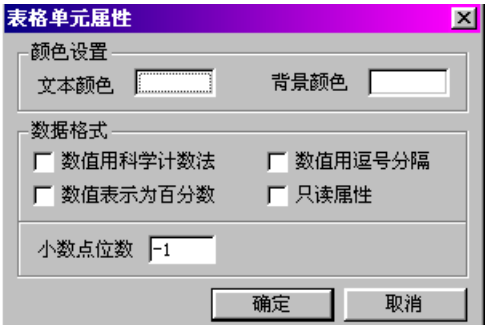


图 10-8

表格行列操作：

在选择该功能按钮后，弹出如下图所示的对话框，在此可进行插入行、追加行，删除行、插入列、追加列，删除列等操作。对于删除行列操作，应选定要删除的范围，如下图 10-9 所示。

图 10-9

绘制表格线

在选择该功能按钮后，弹出如下图所示的对话框，画线有“网线、框线、上边线、下边线、左边线、右边线、对角线（左、右两个方向）”等几种，同时可选择线宽和颜色属性，在此对话框中可分别做画线和抹线操作，如图 10-10 所示。

图 10-10

插入图表

要想在指定位置插入图表，应先选定一个范围，再选择该功能按钮后，弹出如图 10-11 所示的对话框。

此对话框的使用方法与 Excel 中的图表使用方法类似，也有柱状图、折线图、离散图、饼图等几种，这里简单简介如下：

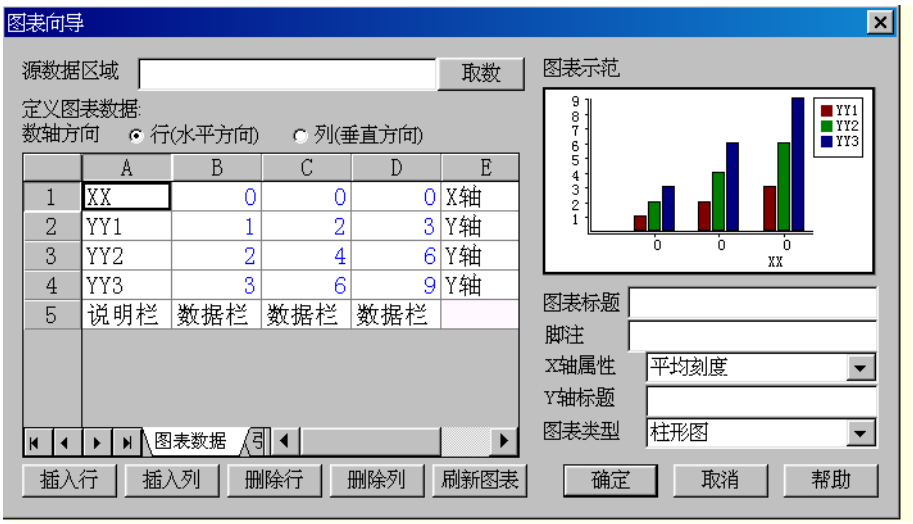


图 10-11

源数据区域：是指该图表所要反映的数据区域，在此框中输入如“A1：D3”（表示从 A1 区到 D3 区）。

取　　数：从指定的源数据区域取出数据，在图表数据显示。

定义图表数据：可定义数轴方向（行、列），同时可进行插入行、删除行、插入列、删除列、刷新图表等操作。

图表标题和脚注：可根据需要在图表标题和脚注中输入内容。

X 轴属性：平均刻度和按 X 数轴的值可选。

Y 轴属性：可任意输入需要的内容。

图表类型：柱状图、折线图、离散图、饼图等可选。

单元多行显示：指把组合单元格后的单元中的内容超过列宽的按多行显示。

居左、居右、居上、居下、水平居中、垂直居中等六种方式，可方便编辑。

组合单元格、取消组合单元、行组合、列组合等四种有关单元格的操作，能方便单元格操作。

选择变量：单击选择变量按钮，弹出如下图所示的对话框，在变量列表中选中某变量后按“确定”或有鼠标左键双击要选择的变量，格式模板

中的该单元格中会有“@变量名”字样，在世纪星运行系统中的组态报表中可得到该变量的实时数据。

注意

1、需要在组态报表中取得变量的历史数据从而生成报表，需在组态报表模板的某单元格中输入如下格式：

@[变量名][年,月,日,时,分,秒][间隔][区域][排列]

其中：

逗号等符号必须为英文半角。

[变量名]是指该变量有记录属性且在系统配置中选择了允许记录的变量；

[间隔]的单位为秒，整数或整数类型变量；

[年,月,日,时,分,秒]是指查询的时间，整数或整数类型变量；

[排列]：[0] -- 纵向排列，[1] -- 横向排列，缺省为纵向排列；

[区域]是指如“A1:G24”字样（[排列]为[0]表示从 A1 到 A24 的 24 个单元纵向排列，[排列]为[1]表示从 A1 到 G1 的 7 个单元横向排列）。

示例：在组态报表模板的 A1 单元中输入

@[反应车间温度][2000,7,10,0,0,0][3600][A1:A24][0]

在世纪星运行系统的组态报表中，能得到从 2000 年 7 月 10 日 0 时 0 分 0 秒整开始， 每隔 1 小时（3600 秒）的反应车间温度的数据，并写到表中的 A1 到 A24 中，即得到这一整天的历史数据（每隔 1 小时）。

2、得到历史数据的表达式应放在要填数据的单元中，以便组态报表中不会出现“@[变量名][年,月,日,时,分,秒][间隔][区域]”字样,如要放在 B2：B25，则此格式放在 B2：B25 之间的任何一单元。

3、组态报表中同时提供按一定方式添加时间，具体格式如下：

@[{年,月,日,时,分,秒}][间隔][区域][方式][排列]

其中：逗号等符号必须为英文半角，
“[{”和“}]”是固定格式。
[方式]中的[1]表示显示“年月日”，[2]表示显示“时分秒”；
[间隔]的单位为秒，整数或整数类型变量；
[年,月,日,时,分,秒]是指查询的时间，整数或整数类型变量；
[排列]：[0] -- 纵向排列，[1] -- 横向排列，缺省为纵向排列；
[区域]是指如“A1:G24”字样（[排列]为[0]表示从 A1 到 A24 的 24 个单元纵向排列，[排列]为[1]表示从 A1 到 G1 的 7 个单元横向排列）。

示例：@[{2000,7,10,0,0,0}][3600][A1:G24][1][1]

4、世纪星组态报表模板支持许多函数，如常用函数、统计函数、字符串函数、表函数、其它函数等，在组态报表模板中，第一次输入函数时，按“=”号，弹出表达式计算对话框，单击：“函数”按钮，得到如下图所示的对话框；如要修改已经输入函数的单元，用鼠标左键双击该单元，就会弹出图 10-12 对话框。

每个函数都有详细说明，这里就不再介绍。

5、从 CSV 格式文件中调入数据，用下面方法：

格式：@["文件路径名"][起始位置]
示例：@["C:\TEST\报表.CSV"][C6]
表示把"C:\TEST\报表.CSV"文件中的数据调入报表中，起始位置是 C6。
注释：CSV 文件中数据由“,”分隔，本报表能自动区分整数、实数和字符串，用 ",," 或 ", ,"表示本格不用显示。

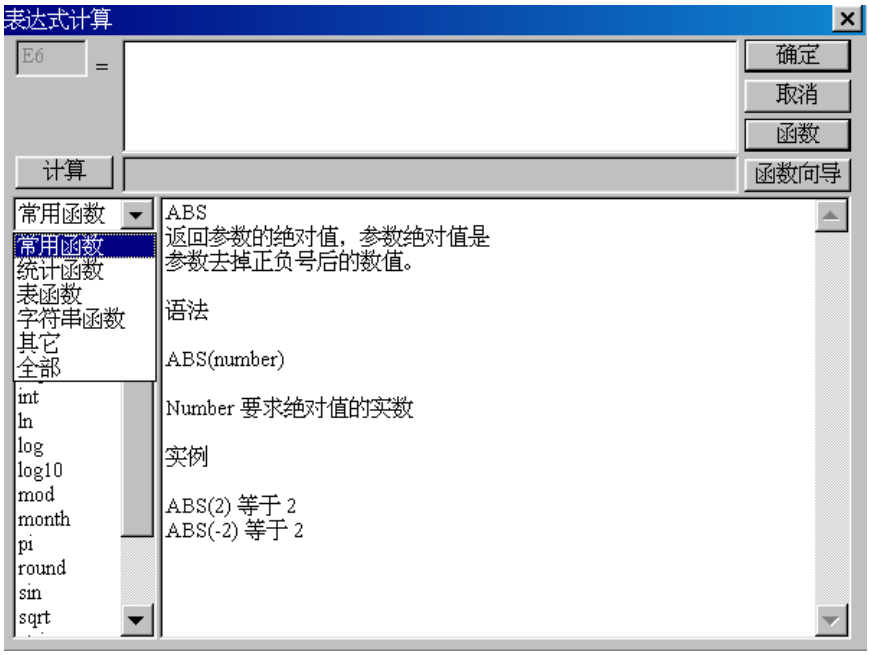


图 10-12

10.5 组态报表

在世纪星的运行系统中，选择“文件→报表→组态报表”或用函数 PrintFormTab，弹出组态报表对话框，如图 10-13 所示：

下图标出各按钮的功能，组态报表的各功能按钮介绍如下图 10-14：

在组态报表中除多一个打印功能和少一个插入变量功能，其它功能基本相似，包括函数等都一样。要想得到实时数据和历史数据，必须在世纪星开发系统的组态报表模板中做好模板，在世纪星运行系统的组态报表中才能得到相应的报表。图 10-15 是一个已经做好的报表例子。

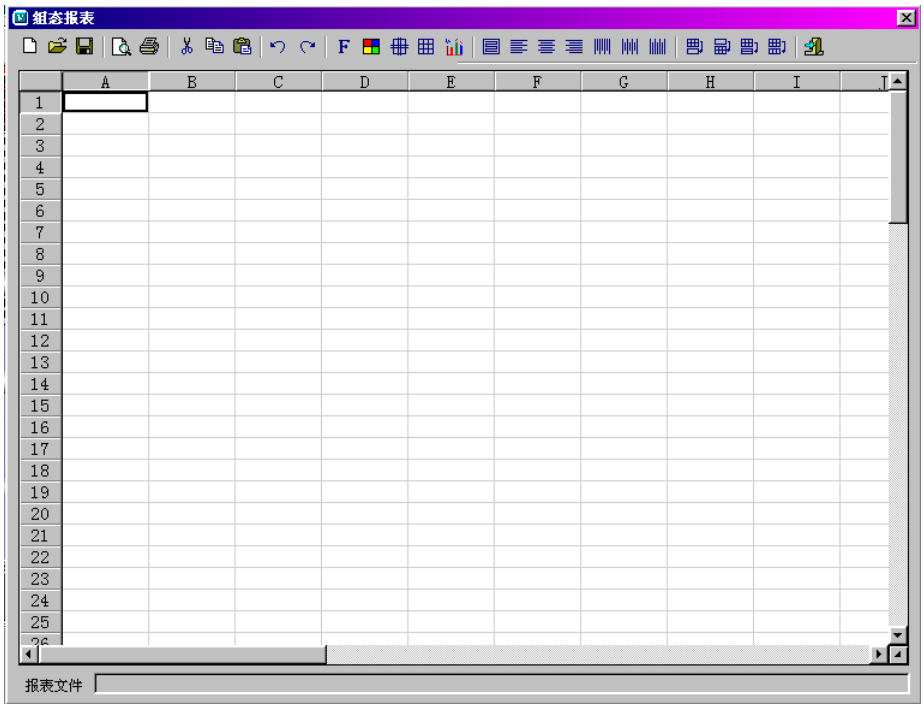

组态报表
报表文件

图 10-13

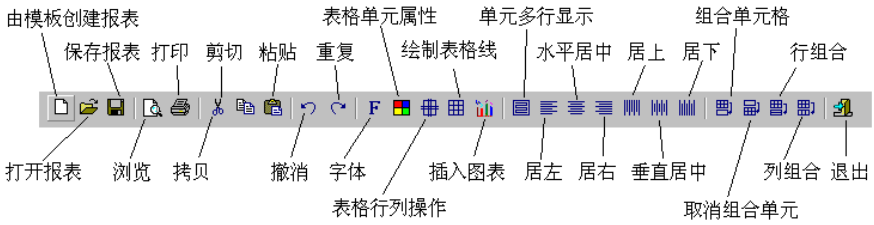

由模板创建报表
保存报表
打印
剪切
粘贴
重复
表格单元属性
绘制表格线
单元多行显示
水平居中
居上
居下
组合单元格
行组合
打开报表
浏览
拷贝
撤消
字体
表格行列操作
插入图表
居左
居右
垂直居中
取消组合单元
列组合
退出

图 10-14

火力发电站发电量日报表

单位：万千瓦　　日班

时间 \ 参数 数值	一号机	二号机	三号机	四号机	总发电量
08:00--09:00	35.68	23.58	65.34	45.68	170.28
09:00--10:00	40.65	23.37	68.95	44.38	177.35
10:00--11:00	38.45	22.68	66.28	48.63	176.04
11:00--12:00	39.66	21.98	64.38	47.69	173.71
12:00--13:00	34.87	28.65	70.98	44.95	179.45
13:00--14:00	41.35	29.64	71.25	49.67	191.91
14:00--15:00	33.25	26.35	68.51	47.56	175.67
15:00--16:00	34.86	24.87	66.89	46.58	173.20
班小计	298.77	201.12	542.58	375.14	1417.61

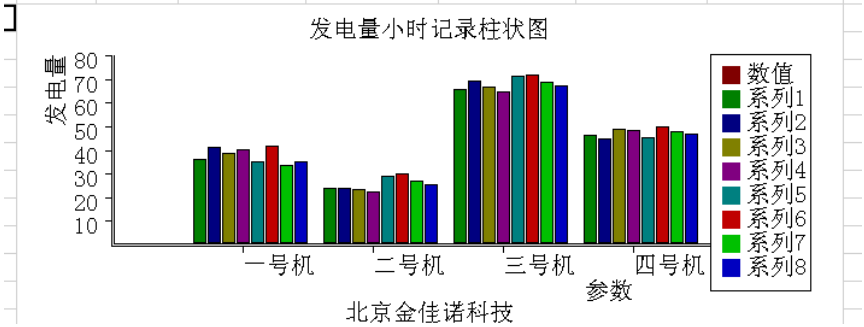


图 10-15

第十一章 SQL功能

11.1 SQL功能简介

世纪星SQL功能是为了实现世纪星和其他数据库之间的数据传输，通过ODBC接口实现世纪星与其他数据库之间的数据交换。它包括世纪星SQL管理和SQL函数两部分。

SQL管理用来建立数据库列和世纪星变量之间的联系。通过表格模板在数据库中创建表格，表格模板信息存储在SQL.DEF文件中；通过记录体建立数据库表格列和世纪星之间的联系，允许世纪星通过记录体直接操纵数据库中的数据。这种联系存储在BNID.DEF文件中。

SQL函数可以在世纪星的任意一种命令语言中调用。这些函数用来在数据库中创建表格，编辑已有的表格，插入、删除、更新记录，清空、删除表格，查询记录等。

11.2 SQL管理

世纪星提供集成的SQL管理。在世纪星工程浏览器的左栏中，可以看到SQL管理。

SQL管理包括表格模板管理和记录列表管理两部分，当世纪星执行SQLCreateTable()指令时，使用的表格模板将定义创建的表格的结构。当执行SQLInsert(), SQLSelect()或SQLUpdate()时，记录表中定义的连接将使世纪星中的变量和数据库表格中的变量相关联。

11.2.1 表格模板管理

用户可以设置特定的表格模板，然后以这种模板在数据库中添加表格。单击表格模板，在右栏中可以新建和管理表格模板。单击新建，弹出图11-1创建表对话框：

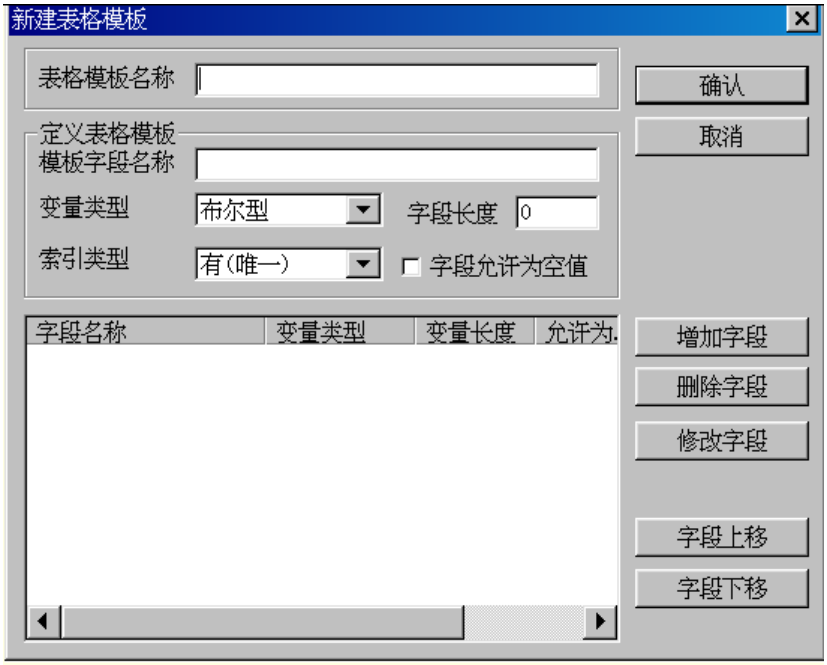


图11-1

其中：

表格模板名称：所建的表格模板的名称，在SQL函数中要用到，可以是英文、中文、数字等字符组成的字符串；

模板字段名称：将要创建的表格的列项名称；

字段类型：将要创建的表格的列项的类型，有布尔型、整型、浮点型、定长字符型、变字符型、日期型等可选；

索引类型：将要创建的表格的列项的类型，有“有（唯一）”、“有（不唯一）”“无”等三种可选；字段是否允许为空：可选择；

增加字段：在输入完“模板字段名称”、“ 字段类型”、“ 索引类型”后，点击此按钮，在对话框中会增加一个列表。

删除字段：在对话框中选择一个字段，按此按钮可删除字段。

修改字段：在对话框中选择一个字段，按此按钮可修改字段。

字段上移：在对话框中选择一个字段，按此按钮可把该字段上移，用于排序。

字段下移：在对话框中选择一个字段，按此按钮可把该字段下移，用于排序。

11.2.2 记录列表管理

记录列表用来连接表格的列和世纪星数据词典中的变量。单击记录体，工程浏览器右栏中显示你已经创建的记录列表，你可以新建和修改记录体。单击新建，弹出图11-2创建记录列表对话框：

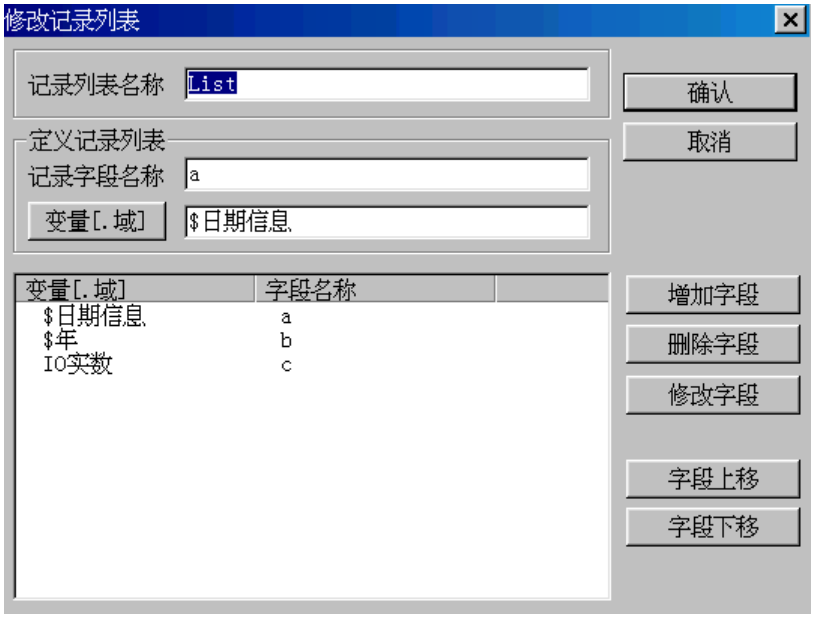


图11-2

其中：

记录列表名称：在SQL函数中要用到，可以是英文、中文、数字等字符组成的字符串；

模板名称：可选择已经建好的表格模板名称；

字段名称：选择已经建好的表格模板中的字段名称；

变量[.域]：选择已经建好的变量或变量的域；

注意：

1、每建好一个字段，按增加字段即可，根据需要建好每个字段，同时还能对建好的字段进行删除、修改及字段上移、下移等操作。

2、保持记录列表中字段的顺序和数据库中表格的顺序一致。

11.3 SQL函数

以下列出所有的SQL函数。SQL功能是同步的，在SQL功能结束之前，控制权不能返回世纪星。(包括趋势曲线等将被挂起)

所有的SQL函数（除SQLNumRows()之外）都会返回一个结果代码。如果这个代码不为零，表明调用函数失败。结果代码可以通过SQLErrorMsg()函数得到。

SQL函数的通常格式如下，具体可参见在线帮助的函数部分：

SQLFunction(Parameter1,Parameter2,…)

以下列出世纪星中的所有SQL函数清单，具体使用可参见函数一章。

SQLAppendStatement	使用SQLStatement的内容继续一个SQL 语句
SQLClearStatement	释放和SQLHandle指定的语句相关的资源
SQLClearTable	删除表格中记录，但保留表格
SQLCommit	定义一组transaction命令的结尾
SQLConnect	连接世纪星到connectstring指定的数据库中
SQLDelete	删除一个或多个记录
SQLCreatTable	使用表格模板中的参数在数据库中
SQLDisconnect	断开和数据库的连接
SQLDropTable	破坏一个表格
SQLErrorMsg	返回一个文本错误信息，此错误信息和特定的结果 代码相关
SQLExecute	执行一个SQL语句。如果这个语句是一个选择语句，捆绑表中的参数所指定的名字用来捆绑数据词典中变量和数据库的列。
SQLFirst	选择由SQLSelect()选择的表格中的首项记录
SQLGetRecord	从当前选择缓存区返回由RecordNumber指定的记录
SQLInsert	用捆绑表中指定的变量中的值在表格中插入一个新记录。捆绑表中的参数定义了世纪星中变量和数据库表格列的对应关系。
SQLInsertEnd	释放插入语句
SQLInsertExecute	执行已经准备的语句
SQLInsertPrepare	准备一个插入语句

SQLLast	选择SQLSelect()指定表格中的末项
SQLLoadStatement	读取包含在FileName中的语句，它类似于SQLSetStatement()创建的语句，能被SQLAppendStatement()挂起，或由SQLExecute()执行，每个文件中只能包含一个语句。
SQLNext	选择表中的下一条记录。
SQLNumRows	指出有多少条记录符合上一次SQLSeclect()的指定
SQLPrepareStatement	本语句为SQLSetParam()准备一个SQL语句。一个语句可以由SQLSetStatement(),或SQLLoadStatement()创建
SQLPrev	选择表中的上一条记录SQLRollback
SQLSelect	访问一个数据库并返回一个表的信息，此信息可以被SQLFirst(),SQLLast(),SQLNext()和SQLPrev().
SQLSetParamChar	将指定的参数设置成特定的字符串。本函数可以调用多次，以设置参数值
SQLSetParamDate	将指定的日期参数设置成特定的字符串
SQLSetParamFloat	将指定的参数设置成特定的值
SQLSetParamInt	将指定的参数设置成特定的值
SQLSetStatement	启动一个语句缓存区，此语句缓存区由SQLStatement()使用
SQLTransact	定义了一组访问指令的开始。在SQLTransact()指令和SQLCommit()指令之间的一组指令称为一个访问组。一个访问组可以象单个访问一样操作。在SQLTransact()指令执行后，所有接下的操作都不委托给数据库，直到SQLCommit()指令执行
SQLUpdate	用当前的世纪星变量更新数据库中的记录SQLUpdateCurrent

11.4 ODBC使用方法示例

本节以世纪星与Access数据库通过ODBC接口进行数据交换作为例子，详细介绍世纪星中有关SQL功能的使用方法及一般顺序。

建库

1.在刚打开Access的时候选择空数据库。

2. 保存数据库。
3. 选择新建，然后选择设计视图，如图11-3所示：
4. 在表填写子段名和类型即可。

注意：建数据库是必要的，而建表除可在Access中建外，也可通过《世纪星》的SQL语句实现。

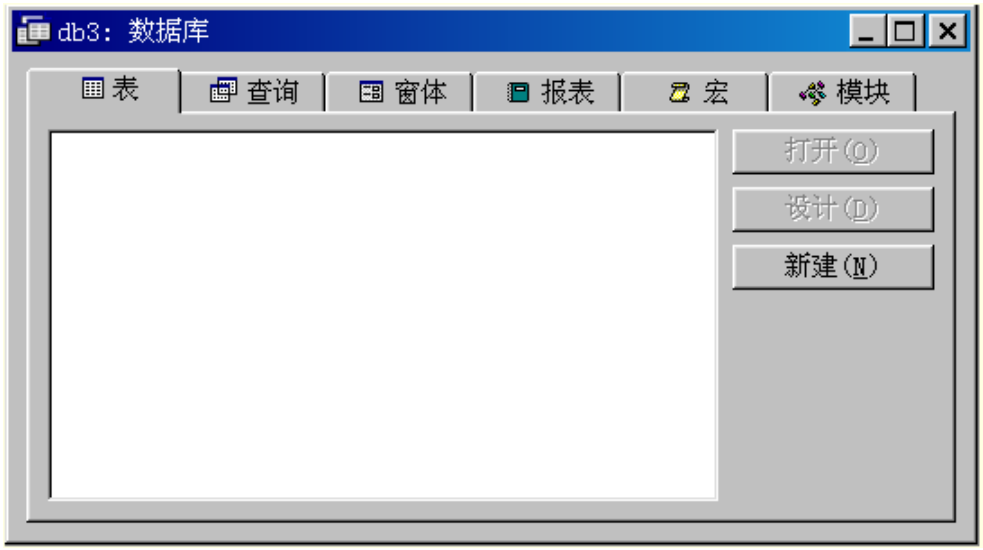


图11-3

设置数据源

1. 在控制面板中选择ODBC数据源。
2. 在系统DSN中选择“添加...”，如图11-4所示：
3. 选择Microsoft Access Driver(*.mdb)，单击完成，如图11-5所示：
4. 在Data Source Name中填写该数据源名称。Select...用于选择具体的Access数据库。然后单击OK就可以了，如图11-6所示：

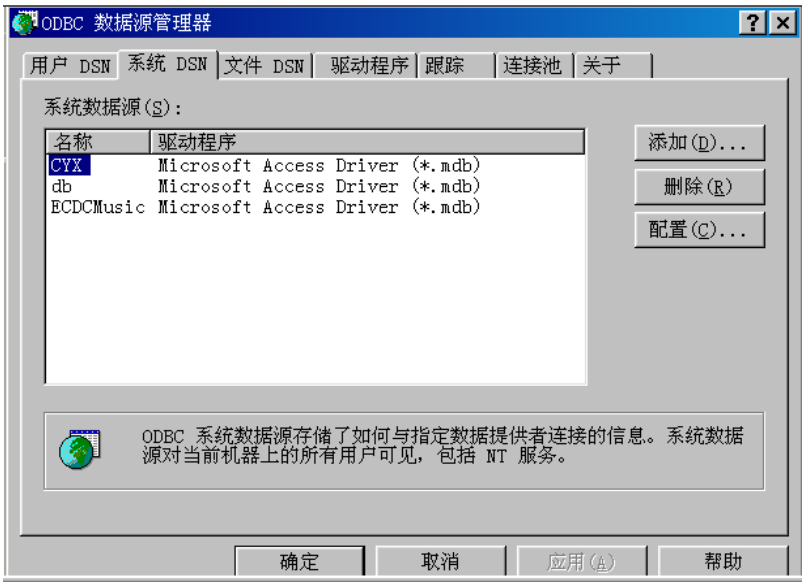

ODBC 数据源管理器
用户 DSN
系统 DSN
文件 DSN
驱动程序
跟踪
连接池
关于
系统数据源(S):
名称
驱动程序
CYX
Microsoft Access Driver (*.mdb)
db
Microsoft Access Driver (*.mdb)
ECDCMusic
Microsoft Access Driver (*.mdb)
添加(D)...
删除(R)
配置(C)...
ODBC 系统数据源存储了如何与指定数据提供者连接的信息。系统数据源对当前机器上的所有用户可见，包括 NT 服务。
确定
取消
应用(A)
帮助

图11-4

ODBC 数据源管理器
创建新数据源
选定您需要的驱动程序来安装数据源(S):
名称
版本
Microsoft Access Driver (*.mdb)
4.00.
Microsoft dBase Driver (*.dbf)
4.00.
Microsoft Excel Driver (*.xls)
4.00.
Microsoft FoxPro Driver (*.dbf)
4.00.
Microsoft ODBC for Oracle
Microsoft Paradox Driver (*.db)
4.00.
Microsoft Text Driver (*.txt; *.csv)
4.00.
Microsoft Visual FoxPro Driver
6.00.
SQL Server
3.70.
< 上一步(B)
完成(F)
取消

图11-5

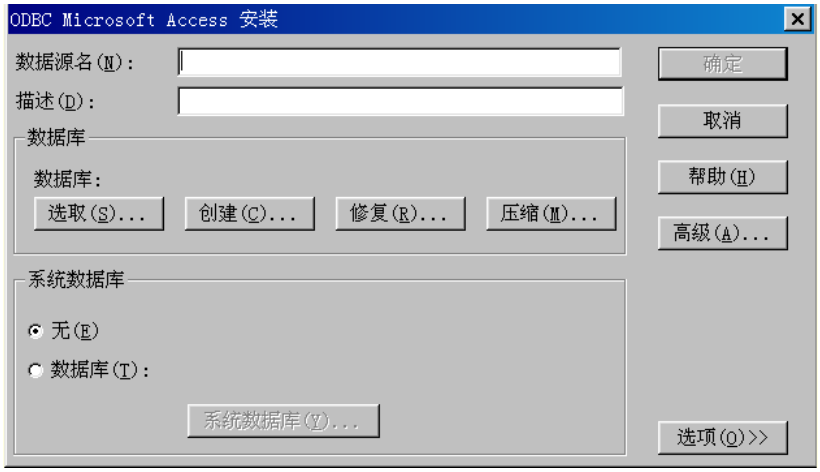


图11-6

SQL管理

1、《世纪星》SQL管理中的“表格模板管理”是定义表的格式，包括每字段的长度、数据类型等属性，如您在数据库中已经建好了表，则在此您要把数据库中已建表的字段进行描述。

2、《世纪星》SQL管理中的“记录列表管理”是定义表的字段与《世纪星》中变量的对应关系，“表格模板管理”及“记录列表管理”是必须按要求完成。

SQL函数使用

顺序及主要函数：

1、连接数据库

格式：ConnectionID= SQLConnect(szConnectString)

该函数用于通过ODBC连接某个具体的数据库。返回的是该连接的唯一标识。

例：

```
ConnectString ="DSN=dsn;UID=uid;PWD=pwd"
ConnectionID = SQLConnect(ConnectString);
```

说明：

DSN是ODBC中的数据源名，UID是具体数据库的用户，PWD为口令。其中DSN、UID和PWD必须大写。ConnectionID是一个很大的整数，建议在《世纪星》变量中建一个9位的内存整数，如连接句柄。SQLConnect函数，是在引用所有SQL函数之前必须引用的一个函数。

2、创建表

----不是必要的，如已经建好表，则不用此步骤

SQLCreateTable(ConnectionID, TableName, TemplateName)

该函数用于在数据库中创建一个表。

ConnectionID：数据库连接的标识

TableName：目标表名

TemplateName：表模板的名称。在该模板中定义了表的所有属性，系统会分析该模板，并按照定义进行建表。

3、添加记录

SQLInsert(ConnectionID, TableName, BindList)

ConnectionID：数据库连接的标识

TableName：目标表名

BindList：捆绑列表名函数首先从BindList中取出要插入的值，然后进行插入操作。

4、修改表记录

SQLUpdate(ConnectionID, TableName, BindList, WhereExpr)

具体使用和标准SQL语句中的SQLUpdate类似。

5、删除记录

SQLDelete(ConnectionID, TableName, WhereExpr)

在表中删除记录

ConnectionID：数据库连接的标识

TableName：目标表名

WhereExpr：查询条件

6、释放连接

SQLDisconnect(ConnectionID)

释放连接函数是用于要使《世纪星》与数据库断开连接时。

第十二章　系统变量

12.1　$访问级别

当前注册用户的访问级别。

使用方法：$访问级别
注　　释：在运行系统中通过选择配置用户菜单来访问本变量。
数据类型：整数(只读)
值 范 围：0 ～ 9999
示　　例：在按钮动画的可见性连接中，如定义：$访问级别**=2000;**
则可由命令语言控制窗口显示和隐含。

```
IF $访问级别 <=500 THEN
   ShowWindow ("禁止访问窗口");
ELSE
   ShowWindow("访问窗口");
ENDIF;
```

参　　阅：$密码输入; $配置用户

12.2　$配置用户

显示配置用户对话框。

使用方法：$配置用户
注　　释：设置为 1 才能显示配置用户对话框。当对话框关闭时，其值自动复位为 0。要显示对话框，用户的$访问级别必须>=9000.
数据类型：离散(只写)
值 范 围：0，1

示　　例：定义一离散变量“更改用户名”，按钮命令语言：

```
IF 更改用户名==1 THEN
   $配置用户=1;
ENDIF;
```

当按下按钮时，系统变量 $配置用户=1，如$访问级别>=9000，将显示配置用户对话框。

参　　阅：$密码输入；$访问级别

12.3　$用户名

当前注册的用户名。

使用方法：$用户名
注　　释：当前注册的用户名。
数据类型：信息(只读)
参　　阅：$密码输入；$注册用户

12.4　$用户登录

显示用户登录对话框。

使用方法：$用户
数据类型：离散(只写)
值 范 围：0，1
注　　释：设置为 1 才能显示用户登录对话框。当对话框关闭时，其值自动复位为 0。
示　　例：定义一离散变量“更改用户登录”，按钮命令语言：

```
IF 更改用户登录==1 THEN
   $用户登录=1;
```

```
ENDIF;
```

当按下按钮时，系统变量 $用户登录=1，将显示用户登录对话框。

参　　阅：$密码输入；$用户名

12.5　$密码输入

用户有效密码输入。

使用方法：$密码输入
数据类型：信息(只写)
注　　释：读本变量时，总返回空串。
参　　阅：$改变密码

12.6　$改变密码

改变用户密码。

使用方法：$改变密码
数据类型：离散(只写)
值 范 围：0，1
注　　释：设置为 1 才能显示改变密码对话框。当对话框关闭时，其值自动复位为 0。
示　　例：定义一离散变量“更改注册用户密码”，按钮命令语言：

```
IF 更改注册用户密码==1 THEN
    $改变密码=1;
ENDIF;
```

当按下按钮时，系统变量 $改变密码=1，将显示改变密码对话框。

参　　阅：$密码输入；$用户名

12.7　$日期

包含从 1970 年 1 月 1 日到现在的天数。

使用方法：$日期
数据类型：整数(只读)
示　　例：StringFromTime($日期×24×3600，1);
返回当前日期信息。

12.8　$日期信息

当前日期，格式：YY 年 MM 月 DD 日。

使用方法：$日期信息
数据类型：信息(只读)
示　　例：StringRight($日期信息，4);
返回"DD 日"。

12.9　$年

当前年。
使用方法：$年
数据类型：整数(只读)
值 范 围：1970 ～ 2035
示　　例：自 1980 年至今年的年数：$年-1980。

12.10 $月

当前月

使用方法：$月
数据类型：整数(只读)
值 范 围：1 ～ 12

12.11 $日

当前天。

使用方法：$日
数据类型：整数(只读)
值 范 围：1 ～ 31

12.12 $时间

包含从 0 点到现在的秒数。

使用方法：$时间
数据类型：整数(只读)

12.13 $时间信息

当前时间，格式：HH 时 MM 分 SS 秒。

使用方法：$时间信息
数据类型：信息(只读)

12.14 $时

当前小时。

使用方法：$时
数据类型：整数(只读)
值 范 围：0 ～ 23

12.15 $分

当前分钟。

使用方法：$分
数据类型：整数(只读)
值 范 围：0 ～ 59

12.16 $秒

当前秒。

使用方法：$秒
数据类型：整数(只读)
值 范 围：0 ～ 59

12.17 $毫秒

当前毫秒。

使用方法：$毫秒
数据类型：整数(只读)
值 范 围：0 ～ 999

12.18 $历史记录

启动或停止历史记录。

使用方法：$历史记录
数据类型：离散(读写)
值 范 围：0，1
注　　释：1 -- 启动历史记录
0 -- 停止历史记录

12.19 $报警记录

启动或停止报警记录。

使用方法：$报警记录
数据类型：离散(读写)
值 范 围：0，1
注　　释：1 -- 启动报警记录
0 -- 停止报警记录

12.20　$新报警

新报警发生时为 1。

使用方法：$新报警
数据类型：离散(读写)
值 范 围：0，1
注　　释：当新报警发生时为 1

12.21　$垂直坐标

选中目标的中心的垂直坐标(即 Y 坐标)。

使用方法：$垂直坐标
数据类型：整数(只读)
参　　阅：$水平坐标

12.22　$水平坐标

选中目标的中心的水平坐标(即 X 坐标)

使用方法：$水平坐标
数据类型：整数(只读)
参　　阅：$垂直坐标

12.23 $双机热备状态

配置双机热备状态。

使用方法：$双机热备
数据类型：整数(读写)
注　　释：1–主机工作正常
2–主机工作异常

12.24 $启动高速命令语言

开始连续执行高速命令语言。

使用方法： $启动高速命令语言
数据类型： 离散(读写)
值 范 围： 0，1
注　　释： 0 -- 停止执行高速命令语言
1 -- 开始连续执行高速命令语言

12.25 $系统

缺省报警组。

使用方法：$系统
数据类型：系统报警组(只读)
注　　释：如果一个变量未定义报警组名，则自动将$系统作为缺省报警组
所有定义的报警组为$系统的下级

第十三章 .域

说明：变量的域用于展示变量、历史趋势曲线和窗口控制的特性。“.域”可以用来显示或修改这些特性。变量的域可以用两种方式访问：通过“变量.域”直接访问，或者通过命令语言函数间接访问。

13.1 .Ack

变量报警是否被应答。

变量类型：离散、整数、实数
域的类型：离散(读写)
有 效 值：0,1

13.2 .Alarm

变量是否有报警。

变量类型：离散、整数、实数
域的类型：离散(只读)
有 效 值：0,1
注　　释：1：报警；0：取消报警

13.3 .AlarmEnabled

变量是否允许报警

变量类型：离散、整数、实数

域的类型：离散(读写)
有 效 值：0，1

13.4 .ChartLength

历史趋势曲线的时间长度

变量类型：历史曲线
域的类型：整数(读写)
有 效 值：正整数
注　　释：单位：秒

13.5 .ChartStart

历史趋势曲线的起始时间

变量类型：历史曲线
域的类型：整数(读写)
有 效 值：正整数
注　　释：1970 年 1 月 1 日至今的秒数

13.6 .Comment

变量字典中变量的注释

变量类型：离散、模拟、信息、历史曲线、报警窗口

域的类型：信息(读写)
有 效 值：字符串

13.7 .DevTarget

大小偏差报警的目标值

变量类型：整数、实数
域的类型：模拟(读写)
有 效 值：在变量定义的范围内

13.8 .DisplayMode

历史趋势曲线显示数值方式

变量类型：历史曲线
域的类型：整数(读写)
有 效 值：0，1
注　　释：0 -- 最大值/最小值方式
1 -- 平均值方式

13.9 .EngUnits

工程单位

变量类型：整数、实数

域的类型：字符串(只读)
有 效 值：最大 15 个字符

13.10 .Group

变量所属的报警组名

变量类型：离散、整数、实数类型变量
域的类型：信息(读写)
有 效 值：字符串

13.11 .HiHiLimit

变量高高报警限

变量类型：整数、实数
域的类型：模拟(读写)
有 效 值：在变量定义的范围内

13.12 .HiHiStatus

变量高高报警状态

变量类型：整数、实数
域的类型：离散(只读)
有 效 值：0，1

注　　释：0 – 无高高报警
1 – 有高高报警

13.13 .HiLimit

变量高报警限

变量类型：整数、实数
域的类型：模拟(读写)
有 效 值：在变量定义的范围内

13.14 .HiStatus

变量高报警状态

变量类型：离散、整数、实数、组变量
域的类型：离散(只读)
有 效 值：0，1
注　　释：0 – 无高报警
1 – 有高报警

13.15 .LoLimit

变量低报警限

变量类型：整数、实数
域的类型：实数(读写)

有 效 值：在变量定义的范围内

13.16 .LoLoLimit

变量低低报警限

变量类型：整数、实数
域的类型：模拟(读写)
有 效 值：在变量定义的范围内

13.17 .LoLoStatus

变量低低报警状态

变量类型：整数、实数
域的类型：离散(只读)
有 效 值：0，1
注　　释：0 - 无低低报警
1 - 有低低报警

13.18 .LoStatus

变量低报警状态

变量类型：整数、实数
域的类型：离散(只读)

有 效 值：0，1
注　　释：0 - 无低报警
1 - 有低报警

13.19 .MajorDevPct

大偏差报警百分数

变量类型：整数、实数
域的类型：整数(读写)
有 效 值：0 ～ 100%

13.20 .MajorDevStatus

大偏差报警状态

变量类型：整数、实数
域的类型：离散(只读)
有 效 值：0，1
注　　释：0 - 无大偏差报警
1 - 有大偏差报警

13.21 .MaxRaw

I/O 变量的最大原始值

变量类型：I/O 变量
域的类型：实数、整数(只读)
有 效 值：在变量义的范围内

13.22 .MaxRange

历史趋势曲线显示的最大值百分数

变量类型：历史曲线
域的类型：实数(读写)
有 效 值：0. ～ 100

13.23 .MaxValue

变量最大值

变量类型：整数、实数
域的类型：实数(只读)
有 效 值：取决于变量类型

13.24 .MinorDevPct

小偏差报警百分数

变量类型：整数、实数
域的类型：整数(读写)
有 效 值：0 ～ 100%

13.25 .MinorDevStatus

小偏差报警状态

变量类型：整数、实数
域的类型：离散(只读)
有 效 值：0，1
注　　释：0 - 无小偏差报警
1 - 有小偏差报警

13.26 .MinRaw

I/O 变量的最小原始值

变量类型：I/O 变量
域的类型：实数、整数(只读)
有 效 值：在变量定义的范围内

13.27 .MinRange

历史趋势曲线显示的最小值百分数

变量类型：历史曲线
域的类型：实数(读写)
有 效 值：0. ～ 100

13.28 .MinValue

变量最小值

变量类型：整数、实数
域的类型：实数(只读)
有 效 值：取决于变量类型

13.29 .Name

以信息类型描述变量名

变量类型：离散、整数、信息、实数、历史曲线
域的类型：信息(只读)

13.30 .NextPage

报警窗口后一页

变量类型：报警窗口变量（仅对历史报警窗口有效）
域的类型：离散(只写)
有 效 值：0，1
注　　释：0 -- 自动赋值
1 -- 翻页

13.31 .OffMsg

离散值为关时信息

变量类型：离散
域的类型：信息(读写)
有 效 值：字符串

13.32 .OnMsg

离散值为开时信息

变量类型：离散
域的类型：信息(读写)
有 效 值：字符串

13.33 .Pen1 .. Pen8

历史趋势曲线的笔

变量类型：历史曲线
域的类型：信息(读写)

13.34 .PrevPage

报警窗口前一页

变量类型：报警窗口变量（仅对历史报警窗口有效）
域的类型：离散(只写)
有 效 值：1，0
注　　释：0 -- 自动赋值
1 -- 翻页

13.35 .PriFrom

报警窗口显示的最小报警优先级

变量类型：报警窗口变量
域的类型：整数(读写)
有 效 值：0 ～ 999

13.36 .Priority

报警优先级

变量类型：离散、整数、实数
域的类型：整数(读写)
有 效 值：0 ～ 999

13.37 .PriTo

报警窗口显示的最大报警优先级

变量类型：报警窗口变量
域的类型：整数(读写)
有 效 值：0 ～ 999

13.38 .RocPct

报警变化率

变量类型：整数、实数
域的类型：整数(读写)
有 效 值：0 ～ 100

13.39 .RocStatus

报警变化率状态

变量类型：整数、实数
域的类型：离散(只读)
有 效 值：0，1
注　　释：0 - 无变化率报警
1 - 有变化率报警

13.40 .ScooterPosLeft

左指示器位置

变量类型：历史曲线
域的类型：实数(读写)
有 效 值：0.0 ～ 1.0

13.41 .ScooterPosRight

右指示器位置

变量类型：历史曲线
域的类型：实数(读写)
有 效 值：0.0 ～ 1.0

13.42 .TotalPage

报警窗口最大页数

变量类型：报警窗口变量
域的类型：整数(只读)
注　　释：报警窗口最大页数=总报警数量/当前窗口最多可显示的报警数量。

第十四章 函数

14.1 Abs( )

计算绝对值

语 法： Result=Abs(Number)
参 数： Number -- 数字、整数、实数变量
返 回： Result -- 计算结果
注 释： 无
示 例：

Abs(14); 返回值为 14
Abs(-7.5); 返回值为 7.5

14.2 Ack( )

应答指定变量的报警

语 法： Ack("Tagname")
参 数： Tagname -- 报警变量、报警组、或组变量
返 回： 无
注 释： 此函数常和按钮连接，当发生报警时，用此函数进行报警确认，它将产生应答报警事件。
示 例：

Ack("温度");
Ack("第一车间");

14.3 AckAll( )

应答所有未应答的报警

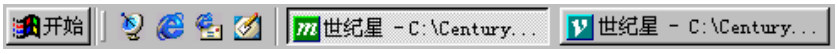
语　法： AckAll( )
参　数： 无
注　释： 参见 ACK 函数
返　回： 无
说　明： 无

14.4 ActivateApp( )

激活正在运行的窗口应用程序

语　法： ActivateApp(TaskName)
参　数： taskName -- 要激活的任务名,字符串或信息类型变量。
返　回： 无
注　释： 在 Win9X 中，所有的任务名都在"任务栏"中，此函数用于激活正在运行的窗口应用程序，使之获得输入焦点。该函数主要用于配合函数 SendKeys 的使用。
示　例：

在 Win 98 中，“任务栏”如下：



要激活 Microsoft Word，则调用： ActivateApp("Microsoft Word")；

14.5 AlmMoveWindow( )

滚动报警显示窗口（适用于历史报警窗口）

语　法：AlmMoveWindow(ObjectName,Option,Repeat)
参　数：ObjectName -- 报警窗口名称，报警窗口变更
Option -- "PageDown": 向下滚动一页
"PageUp": 向上滚动一页
"LineDown": 向下滚一行
"LineUp": 向上滚一行
"Bottom": 到最后一条报警
"Top": 到第一条报警
Repeat -- 执行操作次数
返　回：无
注　释：无
示　例：AlmMoveWindow("报警窗口 1","PageDown",4)
表示报警窗口 1 向下滚动 4 页。

14.6 ArcCos( )

计算反余玄值

语　法：Result=ArcCos(Number)
参　数：Number -- 数字、整数、实数变量，取值范围：-1 ～ 1
返　回：Result -- 计算结果(0 ～ 180 度)
注　释：无
示　例：
ArcCos(1); 返回值为 0
ArcCos(-1); 返回值为 180

14.7 ArcSin( )

计算反正玄值

语　法：Result=ArcSin(Number)
参　数：Number -- 数字、整数、实数变量，取值范围：-1 ～ 1
返　回：Result -- 计算结果(-90 ～ 90 度)
注　释：无
示　例：

ArcSin(1);返回值为 90
ArcSin(-1);返回值为-90

14.8 ArcTan( )

计算反正切值

语　法：Result=ArcTan(Number)
参　数：Number -- 数字、整数、实数变量
返　回：Result -- 计算结果(-90 ～ 90 度)
注　释：无
示　例：

ArcTan(0); 返回值为 0
ArcTan(1); 返回值为 45

14.9 Beep( )

发出蜂鸣声

语　法：Beep(Tagname)
参　数：Tagname--整数或整数变量，表示蜂鸣声次数。
返　回：无
注　释：无
示　例：

Beep（5）;
表示发出 5 声蜂鸣声。

14.10　Bit( )

获得整数变量某位的值

语　法：DiscResult= Bit(Tagname, BitNo)
参　数：Tagname -- 整数变量名
BitNo -- 位号（0 ～ 31）
返　回：DiscResult - 0,1
注　释：无
示　例：

变频器运行状态=8；BitNo(变频器运行状态，3)的值为 1

14.11　BitSet( )

设置整数变量某位的值

语　法：BitSet("Tagname", BitNo, Discrete_Value)
参　数：Tagname -- 整数变量名
BitNo -- 位号（0 ～ 31）

Discrete_Value -- 离散变量或离散值(0,1)

返　回：无

注　释：只能设置整数的某位

示　例：

变频器运行状态=8；BitSet(变频器运行状态，2，1)，则变频器运行状态为 12

14.12　ChangePassword( )

显示改变密码对话框，使登录用户改变其密码

语　法：ChangePassword( )

参　数：无

返　回：无

注　释：无

示　例：

为按扭设置命令语言连接：ChangePassword( );

运行时按下按扭，弹出“修改密码”对话框，用户输入密码后，按确定按钮，则用户的密码被设置为新值。

14.13　CloseWindow( )

关闭打开的画面

语　法：CloseWindow(WindowName)

参　数：WindowName -- 打开的画面名

返　回：无

注　释：无

示　例：

CloseWindow（"历史趋势曲线"）；　将关闭“历史趋势曲线”画面。

14.14　ConvertDateTime( )

将时间转换成自 1970 年 1 月 1 日 0 时 0 分 0 秒起至年、月、日、时、分、秒表示的时间的秒数。

语　法：ConvertDateTime(int Year,int Month,int Day,int Hour,int Minute,int Second)

参　数：Year:　年，此值必须介于 1970 和 2019 之间

Month:　月，此值必须介于 1 和 12 之间

Day:　日，此值必须介于 1 和 31 之间

Hour:　时，此值必须介于 0 和 23 之间

Minute:分，此值必须介于 0 和 59 之间

Second:秒，此值必须介于 0 和 59 之间

返　回：返回自 1970 年 1 月 1 日(00:00:00)即 UCT 起到该时刻所经过的秒数

注　释：无

示　例：无

14.15　CopyScreenToPrint( )

将当前屏幕上的内容用硬拷贝方式输出到打印机

语　法：CopyScreenToPrint( )

参　数：无

返　回：无

注　释：本函数通常用于按钮的命令语言连接

示　例：无

14.16　Cos( )

计算余弦值

语　法：Result=Cos(Number)
参　数：Number -- 数字、整数、实数变量
返　回：Result -- 计算结果
注　释：无
示　例：

Cos(90)；返回值为 0
Cos(0)；　返回值为 1

14.17　Counter(　)

计数器函数

语 法:

Counter("CounterControlTag",TriggerCondition,ControlCondition,StartValue);

参 数:

CounterControlTag -- 计数器控制变量，离散类型
TriggerCondition --　触发条件
0：上升沿计数，
1：下降沿计数，
2：脉冲计数
ControlCondition -- 计数控制条件

0：正常计数，
1：暂停计数，
2：计数器赋值

StartValue -- 计数器初始值

返 回：-1：计数器控制变量没找到或数据类型错误
-2：计数控制条件错误
-3：触发条件错误
-0：计数结果

注　释：无

示　例：无

14.18　Delay( )

延时函数

语　法：Delay(MilliSecond)

参　数：MilliSecond -- 整数、整数变量

返　回：无

注　释：延时以毫秒为单位，建议取不大于 1000 的整数

示　例：无

14.19　DialogStringEntry( )

显示对话框，输入字符串

语　法：

Result=DialogStringEntry("MessageTag_Text",UserPrompt_Text)

参　数：MessageTag_Text -- 接收输入字符串的变量或变量的域

UserPrompt_Text -- 用户输入提示，字符串或信息变量。

返　回： 0 -- 用户按取消按钮
1 -- 用户按确定按钮
-1 -- 内部错误
-2 -- 不能初始化
-3 -- 变量未定义
-4 -- 变量不是信息类型
-5 -- 写失败

注　释： 无

示　例： 无

14.20　DialogValueEntry( )

显示对话框，输入离散、整数、实数值

语　法：

```
Result=DialogValueEntry("ValueTag_Text",LowLimit,HighLimit
                                ,UserPrompt_Text)
```

参　数： ValueTag_Text　-- 接收输入值的变量或变量的域
LowLimit　-- 用户输入值的下限
HighLimit　-- 用户输入值的上限
UserPrompt_Text -- 用户输入提示

返　回： 0 -- 用户按取消按钮
1 -- 用户按确定按钮
-1 -- HightLimit<=LowLimit
-2 -- 不能初始化
-3 -- 写失败

注　释： 无

示　例： 无

14.21 DirectoryCreat( )

创建目录

语　法：DirectoryCreate(Directory)
参　数：Directory -- 带路径的目录名，字符串或信息变量。
返　回：无
注　释：无
示　例：无

14.22 DirectoryRemove( )

删除目录

语　法：DirectoryRemove(Directory)
参　数：Directory -- 带路径的目录名，字符串或信息变量。
返　回：无
注　释：无
示　例：无

14.23 DText( )

由一个离散变量动态修改信息变量

语　法：

```
MessageTag=DText("Discrere_Tag",OnMessage,OffMessage)
```

参　数：Discrete_Tag -- 离散类型变量
OnMessage -- Discrete_Tag 为 1(True,On,Yes)时的信息

OffMessage -- Discrete_Tag 为 0(False,Off,No)时的信息

返　回：MessageTag -- 信息类型变量

注　释：无

示　例：

开关信息 = Dtext("开关变量", "打开","关闭");

当开关变量=1 时，开关信息的值为"打开"

当开关变量=0 时，开关信息的值为"关闭"。

14.24 EventsLogQuery( )

调用事件查询对话框

语　法：EventsLogQuery( )

参　数：无

返　回：无

注　释：调用事件查询对话框，可参见帮助中“系统配置\运行系统配置\事件记录配置”和运行系统的“报表打印\事件查询报表”。

示　例：无

14.25 Exit( )

退出运行环境

语　法：Exit(Options)

参　数：Options - 0：退出运行系统，返回到 Windows

1：关机

2：重新启动计算机

返　回：无

注　释：无

示　例：无

14.26 Exp( )

计算指数函数 e 的幂

语 法： Result=Exp(Number)
参 数： Number -- 数字、整数、实数变量
返 回： Result -- 计算结果(-88.72 ～ 88.72)
注 释： 无
示 例：

Exp(1)；返回 e^1 的计算值 2.718 .....

14.27 FileCopy( )

拷贝文件

语 法： FileCopy(SourceFile,DestFile)
参 数： SourceFile -- 原文件名(带路径)，字符串或信息类型变量
DestFile -- 目标文件名(带路径)或目录，字符串或信息类型变量
返 回： 0 -- 成功
1 -- 失败
注 释： 此函数复制源文件到目的文件，它与 Windows 文件管 理器中的 Copy 功能相似。
示 例：

CopyStatus=FileCopy("C:\CSWORK*.WIN", "C:\BACK");
复制 C:\CSWORK\目录中的所有扩展名为.WIN 的文件到 C:\BACK 目录中。如复制成功，则 CopyStatus=0，否则 CopyStatus=1;

14.28 FileDelete( )

删除文件

语　法： FileDelete(FileName)
参　数： FileName -- 要删除的文件，字符串或信息类型变量
返　回： 0 -- 成功
1 -- 失败
注　释： 无
示　例：

DeleteStatus= FileDelete("C:\BACK");
删除 C:\BACK 目录中的所有文件。
如删除成功，则 DeleteStatus=0，否则 DeleteStatus=1;

14.29 FileLineDelete( )

CSV 文件行删除

函　数： Result=FileLineDelete(FileName, StartLineNo, LineNumber);
参　数： FileName -- 文件路径名，字符串或信息变量
StartLineNo -- 删除起始行，从 1 开始
LineNumber -- 删除行数
返　回： 0 -- 操作正常
1 -- 文件路径名错误或文件不存在
2 -- 参数输入错误
注　释： CSV 文件的行删除，一般用于文本窗口控件中。
示　例： 无

14.30 FileMove( )

移动文件

语 法： FileMove(SourceFile,DestFile)

参 数： SourceFile -- 原文件名(带路径) ，字符串或信息类型变量
DestFile -- 目标文件名(带路径)或目录，字符串或信息类型变量

返 回： 0 -- 成功
1 -- 失败

示 例：

MoveStatus= FileMove("C:\BACK1","C:\BACK2");
将 C:\BACK1 目录中的所有文件移到 C:\BACK2 目录中。
如移动成功，则 MoveStatus=0，否则 MoveStatus=1;

14.31 FileReadFields( )

从文件中读出变量记录，文件中的变量记录由逗号(,)分隔

语 法：

BytePosition=FileReadFields(FileName,FileOffset,"StartTag"
,NumberOfFields)

参 数： FileName -- 文件名
FileOffset -- 文件中读变量的起始位置
StartTag -- 接收从文件中读出变量的起始变量名，该变量名必须由一个数字字符结束(如 MyTag1)，此参数必须是一个表明变量名的字符串，而不是变量本身。
NumberOfFields -- 要读的字段数目(文件的每条记录中以逗号分隔的字段数目)

返 回： BytePosition -- 读取下一个记录的起始位置

注　释：如果 StartTag="MyTag1"，NumberOfField=3，则从文件中读出的字段将存到 MyTag1，MyTag2，MyTag3。这些具有连续名的变量必须先创建，且可以为不同的　　数据类型(如整数、信息等)

示　例：

如 C:\DATA\FILE.CSV 的第一行为："This is text, 3.1416, 5,"
则函数

NextLinePosition=FileReadFields ("C:\DATA\FILE.CSV", 1, "MyTag1", 3);

将读出此行，并把"This is text"保存在 MyTag1 中，3.1416 保存在 MyTag2 中，5 保存在 MyTag3 中。

此函数在读出之后返回新的字节位置，可以在下次读时使用此返回值作为 FileOffset 的值，如：

FileReadFields("C:\DATA\FILE.CSV"NextLinePosition,"MyTag1",3);

14.32　FileReadMessage( )

从文件中读几个或一行字符

语　法：BytePosition=FileReadMessage(FileName,FileOffset, "Message_Tag",CharsToRead)

参　数：FileName　　-- 文件名
FileOffset　-- 文件中读字符的起始位置
Message_Tag -- 接收从文件中读出字符的变量名
CharsToRead -- 要读的字符数目，为 0，则读一行字符

返　回：BytePosition --读取下一个字符的起始位置

注　释：无

示　例：

NextLinePosition=FileReadMessage("C:\DATA\FILE.CSV", 1, "MyTag1", 0);

文件"C：\DATA\FILE. CSV"的第一行将被读出并保存到 MsgTag1 中。

此函数在读出之后返回新的字节位置，可以在下次读时使用此返回值作为 FileOffset 的值，如：

```
FileReadMessage("C:\DATA\FILE.CSV",NextLinePosition,
                "MyTag1",0);
```

14.33 FileReadWriteFields( )

非标准文件的成组读写

语　法： BytePosition=FileReadWriteFields(FileName, FileOffset, "StartTag",NumberOfFields,Options)

参　数：

FileName ---- 文件名

FileOffset ---- 文件中读变量的起始位置，从 0 开始，写数据时，-1 表示追 加。

StartTag ---- 接收从文件中读出变量的起始变量名，该变量名必须 由一个数字字符结束(如 MyTag1),此参数必须是一个表明变量名的字符串，而不是变量本身。

NumberOfFields -- 要读的字段数目(文件的每条记录中以逗号分隔的字段数目)

Options ---- 读或写变量，0 表示读，1 表示写

返　回： BytePosition ---- 读下一个记录的起始位置

注　释： 如果 StartTag="MyTag1"，NumberOfField=3，则从文件中读出的字段将存到 MyTag1，MyTag2，MyTag3。这些具有连续名的变量必须先创建，且可以为不同的数据类型(如整数、信息等)

示　例：

如 C:\DATA\FILE.CSV 的第一行为："This is text, 3.1416, 5"，则函数

```
NextLinePosition=FileReadFields("C:\DATA\FILE.CSV",0,
```

"MyTag1",3，0);

将读出此行，并把“This is text"保存在 MyTag1 中，3.141 保存 MyTag2 中，5 保存在 MyTag3 中。

14.34 FileReadWriteTag( )

非标准文件的单个读写

语　法：FileReadWriteTag(FileName, FileOffset, "Tagname", Options);

参　数：FileName　-- 文件名，带全路径

FileOffset -- 文件中读写字符的起始位置，从 0 开始计数，可以是整数 或整数变量，若是写变量，“-1”表示接上一次位置写文件

Tagname -- 文件中读出或写入变量的变量名

Options -- 读或写变量，0 表示读，1 表示写

返　回：无

注　释：无

示　例：无

14.35 FileSelectDialog( )

文件选择对话框

语 法：MessageResult=FileSelectDialog(FilePath, ExtName);

参 数：FilePath -- 要选择文件所在的目录，字符串或信息变量

ExtName -- 要选择文件的扩展名，字符串或信息变量

返　回：选中文件的包含路径的文件名，信息类型

注　释：无

示　例：MessageResult=FileSelectDialog("C:\TEST", "TBL");

弹出文件选择对话框，选择后返回包含路径和扩展名的文件名到 MessageResult 中，如：C:\TEST\20000101000000.TBL

14.36　FileWriteFields( )

写变量记录到文件中，写到文件中的变量由逗号(,)分隔

语　法：BytePosition=FileWriteFields(FileName,FileOffset,
"StartTag",NumberOfFields)

参　数：FileName　-- 文件名,如文件不存在，则创建该文件

FileOffset -- 写变量到文件中的起始位置,为-1 时,则写到文件尾

StartTag　-- 写变量到文件中的起始变量名，该变量名必须由一个数字字符结束(如 MyTag1)，此参数必须是一个表明变量名的字符串，而不是变量本身。

NumberOfFields -- 要写的字段数目(文件的每条记录中以逗号分隔的字段数目)

返　回：BytePosition　-- 写下一个记录的起始位置

注　释：如果 StartTag="MyTag1"，NumberOfField=3，则从文件中读出的字段将存到 MyTag1，MyTag2，MyTag3。这些具有连续名的变量必须先创建，且可以为不同的 数据类型(如整数、信息等)

示　例：

将一行“This is text,3.1416,5”写到文件 C：\DATA\FILE.CSV 的第一行中。“This is text”是 MyTag1 的当前值，3.1416 是 MyTag2 的当前值，5 是 MyTag3 的当前值，则：

FileWriteFields("C:\DATA\FILE.CSV",1,"MyTag1",3);

若将文本串 MyTag1 写到 C：\DATA\FILE.CSV 的末尾，则：

```
FileWriteFields("C:\DATA\FILE.CSV", -1, "MyTag1",3);
```

14.37 FileWriteMessage( )

写几个或一行字符到文件中

语　法：

```
BytePosition=FileWriteMessage(FileName,FileOffset,"Message
                                    _Tag",LineFeed)
```

参　数： FileName　-- 文件名,如文件不存在，则创建该文件
FileOffset -- 写字符到文件中的起始位置
Message_Tag-- 包含要写到文件中的字符的变量名
LineFeed　-- 写操作后，是否写回车、换行符
如为文本文件，则设 LineFeed=1

返　回： BytePosition -- 读下一个字符的起始位置

示　例：

将信息变量 MsgTag1 写入文件 C：DATA\FILE.CSV 的末尾，则：

```
FileWriteMessage ("C:\DATA\FILE.CSV", 0, "MsgTag1",1);
```

14.38 FileWriteTags( )

语　法：

```
BytePosition=FileWriteTags(FileName,FileOffset,"TagName1",
..."TagName10")
```

参　数： FileName　-- 文件名,如文件不存在，则创建该文件
FileOffset -- 写变量到文件中的起始位置，为-1.则写到文件尾
"TagName1"..."TagName10"--如为变量名，则写变量的值到 CSV 格式(FileName)文件中，否则作为字符串写入到 CSV 格式(FileName)

文件中。

返　回：BytePosition--写下一个记录的起始位置

注　释："TagName1"..."TagName10"中任何一个参数若为""时，表示不写此变量。

示　例：无

14.39 FindFileOrDirectory( )

查找文件或文件夹是否存在

语　法：FindFileOrDirectory(FileName, Mode)

参　数：FileName -- 文件或文件夹，带全路径

Mode -- 文件或文件夹，0 表示文件，1 表示文件或文件夹

返　回：0 表示存在，1 表示不存在

注　释：用于写文件前查找文件或文件夹是否存在，以免写文件错误

示　例：无

14.40 FloatFromFileTable( )

查表函数

语　法：Result=FloatFromFileTable(FileName, IndexData)

参　数：FileName -- 文件名，要求带全路径及扩展名，格式为 SCV

IndexData -- 索引号

返　回：Result -- 返回的值

注　释：此表主要用于一些要补偿的设备，如热电阻。下面为有关文件的说明及示例。

C 盘 CenturyStar 目录下创建一文件 data1.csv, 格式为.SCV,文件内

容示例如下：
1.0,2.365
2.0,2.437
3.0,2.563
4.0,2.634
5.0,2.781
其中逗号前表示索引号，逗号后表示返回的查表值，

示 例： FloatFromFileTable("c:\centurystar\data1.csv",1.0)
返回值为 2.365;
FloatFromFileTable("c:\centurystar\data1.csv",2.0)
返回值为 2.437;

14.41 FormAutoSave( )

由组态报表模板生成报表文件

语 法： IntResult=FormAutoSave(ModuleName, FilePath, PreName);
参 数： ModuleName -- 模板名称，字符串或信息变量
FilePath -- 生成的报表文件存储路径，字符串或信息变量
PreName -- 自动生成文件的前缀，字符串或信息变量
返 回： Result: 0 -- 操作成功， 1 -- 操作失败
注 释： 报表文件名是自动生成的，其方式如下：
前缀 YYYYMMDDHHMMSS.TBL
其中：YYYY -- 年，MM -- 月，DD -- 日，
HH -- 时，MM -- 分，SS -- 秒
示 例： IntResult=FormAutoSave("报表 1", "C:\TEST","日报");将工程中的组态报表模板"报表 1"，自动存储到"C:\TEST"目录中，如当前日期时间为 2000 年 1 月 1 日 0 时 0 分 0 秒，则自动生成文件名为：日报 20000101000000.TBL

14.42　GetHistDataValue( )

获取历史数据变量的值

语　法：GetHistDataValue("Tagname", Year, Month, Day, Hour, Minute, Second, TimeLength, ValType_Text);

参　数：Tagname -- 变量名，已经定义的带历史数据记录的变量

Year, Month, Day, Hour, Minute,Second -- 年，月，日，时，分，秒；整数或整数类型变量

TimeLength -- 时间长度；整数或整数类型变量

ValType_Text-- 要求返回值类型，字符串，具体参数及意义如下：

"Value" -- 变量在指定时刻的值

"AverageValue"-- 变量在 TimeLength 时间范围内的平均值

"MaxValue" ---- 变量在 TimeLength 时间范围内的最大值

"MinValue" ---- 变量在 TimeLength 时间范 围 内的最小值

返　回：无

注　释：参数 ValType_Text 要求的字符串必须与如上所列的四个字符之一相同，且必须加引号。

示　例：无

14.43　GetWindowMilliseconds( )

读取 Windows 启动时到当前的系统毫秒时间

语　法：Result=GetWindowMilliseconds( )

参　数：无

返　回：Windows 启动时到当前的系统毫秒时间，整数变量，最大可显示到 22 天的毫秒时间

注　释：该函数返回值受整数取值范围的影响，系统运行超过 23 天，返回值

就不正确，只有重新启动计算机后，函数才能继续运行。

示　例：无

14.44　HideWindow( )

隐藏打开的画面窗口

语　法：HideWindow(Window)

参　数：Window -- 已打开的画面窗口,字符串或信息变量。

返　回：无

注　释：无

示　例：HideWindow("历史趋势曲线"); 隐藏"历史趋势曲线"画面窗口

14.45　HTGetPenName( )

获得历史趋势曲线中正在使用笔的变量名

语　法：MessageResult=HTGetPenName(Hist_Tag,PenNum)

参　数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型

PenNum　-- 笔号(1 -- 8), 整数类型

返　回：MessageResult -- 笔的变量名，信息类型

注　释：无

示　例：

在历史曲线 Hist_Tag 中查找第 2 根笔的变量名，并将结果放在信息 TrendPen 中,则：

```
TrendPen=HTGetPenName(Trend1,2);
```

14.46 HTGetTimeAtScooter( )

获得历史趋势曲线中指示器位置的时间

语 法： InterResult=HTGetTimeAtScooter(Hist_Tag,ScooterNum)

参 数： Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型
ScooterNum -- 指示器号(1--左指示器，2--右指示器)，整数类型

返 回： 历史趋势曲线中指示器位置的时间。该时间为相对于 GMT(格林尼治时间)1970 年 1 月 1 日 00 时 00 分 00 秒的相对时间，单位：秒，整数类型

注 释： 无

示 例：

历史趋势曲线的变量名 Trend1，则曲线左指示器位置的时间为：

```
LeftScooterTime=HTGetTimeAtScooter(Trend1,1);
```

14.47 HTGetTimeStringAtScooter( )

获得历史趋势曲线中指示器位置的日期/时间信息串

语 法：

```
MessageResult=HTGetTimeStringAtScooter(Hist_Tag,ScooterNum,
                                       Format_Text)
```

参 数： Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型
ScooterNum -- 指示器号(1--左指示器，2--右指示器)，整数类型
Format_Text-- 时间信息串格式
"Date" -- YYYY 年 MM 月 DD 日
"Time" -- HH 时 MM 分 SS 秒
"DateTime"-- YYYY 年 MM 月 DD 日 HH 时 MM 分 SS 秒

返 回： 历史趋势曲线中指示器位置的日期/时间信息串, 信息类型

注　释：无

示　例：

历史趋势曲线的变量名 Trend1，则曲线左指示器位置的时间信息为：

```
LeftScooterTimeMessage=HTGetTimeStringAtScooter
                    (Trend1,1,"Time")
```

14.48　HTGetValue( )

获得历史趋势曲线笔的值

语　法：

```
RealResult=HTGetValue(Hist_Tag,PenNum,ValType_Text)
```

参　数：Hist_Tag　-- 历史趋势曲线的变量名，历史曲线类型

PenNum　　-- 笔号(1 ～ 8)，整数类型

ValType_Text -- 要求返回值类型

"PenAverageValue" -- 历史趋势曲线的平均值

"PenMaxValue" -- 历史趋势曲线的最大值

"PenMinValue" -- 历史趋势曲线的最小值

返　回：历史趋势曲线指定笔的值，内存实数类型

注　释：无

示　例：要得到趋势 Trend1 的 Pen1 处数据的最大值，则：

```
TrendPen1MaxValue=HTGetValue(Trend1,1,"PenMaxValue");
```

14.49　HTGetValueAtScooter( )

获得历史趋势曲线指示器位置笔的值

语　法：

RealResult=HTGetValueAtScooter(Hist_Tag,ScooterNum,PenNum,ValType_Text)

参 数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型
ScooterNum -- 指示器号(1--左指示器，2--右指示器)，整数类型
PenNum -- 笔号(1 ～ 8)，整数类型
ValType_Text -- 要求返回值类型
"PenValue" -- 在指示器位置笔的值，返回实数值
"PenValid" -- 判断获得的值是否有效，返回离散值
0 -- 无效，1 -- 有效。

返 回：历史趋势曲线指定笔的值，类型由参数 ValType_Text 决定

注 释：无

示 例：

采集趋势曲线 Trend1 的笔 Pen8 在左指示器的当前值:
ValidStatus=HTGetValueAtScooter(Trend1,1,8,"PenValid"); 如 ValidStatus=0；则返回值无效，否则有效。

14.50 HTGetValueAtZone( )

获得历史趋势曲线中左右指示器之间的值

语 法：

RealResult=HTGetValueAtZone(Hist_Tag,PenNum,ValType_Text)

参 数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型
PenNum -- 笔号(1 ～ 8)，整数类型
ValType_Text -- 要求返回值类型
"PenAverageValue" -- 历史趋势曲线的平均值
"PenMaxValue" -- 历史趋势曲线的最大值
"PenMinValue" -- 历史趋势曲线的最小值

返 回：历史趋势曲线中左右指示器之间的值，内存实数类型

注　释：无
示　例：
要取得趋势曲线“Trend1”的笔 Pen1 在左右指示器之间的平均值，并把结果存入内存实数变量 AverageValue 中，则：
AverageValue=HTGetValueAtZone(Trend1, 1, "PenAverageValue");

14.51　HTScrollLeft( )

按给定百分数左移历史趋势曲线，此百分数是相对于当前曲线的时间宽度

语　法：HTScrollLeft(Hist_Tag,Percent)
参　数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型
Percent -- 曲线左移百分数，实数类型，取值范围：0.0 ～ 100.0
返　回：无
注　释：无
示　例：将趋势曲线 Trend1 的时间轴向左滚动 50%，则：
HTScrollLeft(Trend1,50.0);
若当前显示起始于下午 12:00:00，而且显示宽度为 1 小时。在函数执行后，新的趋势曲线将起始于上午 11:30:00，时间长度为 1 小时 30 分钟。

14.52　HTScrollRight( )

按给定百分数右移历史趋势曲线，此百分数是相对于当前曲线的时间宽度

语　法：HTScrollRight(Hist_Tag,Percent)
参　数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型

Percent　-- 曲线右移百分数，实数类型，取值范围：0.0 ～ 100.0

返　回：无

注　释：无

示　例：

将趋势曲线 Trend1 的时间轴向右滚动 50%，则：

HTScrollRight(Trend1,50.0);

若当前显示起始于下午 12:00:00，而且显示宽度为 1 小时。在函数执行后，新的趋势曲线将起始于上午 12:30:00，时间长度为 30 分钟。

14.53　HTSelectTag( )

显示选择变量对话框，给指定历史曲线选择变量名。

语　法：HTSelectTag(Hist_Tag)

参　数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型

返　回：无

注　释：无

示　例：无

14.54　HTSetPenName( )

给指定笔指定一个变量

语　法：HTSetPenName(Hist_Tag,PenNum, "Tagname")

参　数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型

PenNum　-- 笔号(1 ～ 8)，整数类型

Tagname　-- 给指定笔的新变量名，信息类型

返　回：无
注　释：无
示　例：将趋势曲线 Trend1 的 Pen2 的变量设置为温度，则：
HTSetPenName(Trend1,2,"温度");

14.55　HTUpdateToCurrentTime( )

设置历史趋势曲线的终止时间为当前时间，曲线的时间轴宽度保持不变

语　法：HTUpdateToCurrentTime(Hist_Tag)
参　数：Hist_Tag -- 历史趋势曲线的变量名，历史曲线类型
返　回：无
注　释：用于查看最新数据
示　例：显示历史趋势曲线“Trend1”在当前时间的最新数据，则：
HTUpdateToCurrentTime(Trend1);若现在是下午 12:00:00 且曲线时间长度为 1 小时，则新的终止时间将为下午 12:00:00，新的起始时间将为 11:00:00。

14.56　HTZoomIn( )

历史趋势曲线在时间轴方向曲线放大一倍，即曲线的时间轴宽度缩小一倍

语　法：HTZoomIn(Hist_Tag,LockString)
参　数：Hist_Tag　-- 历史趋势曲线的变量名，历史曲线类型
LockString -- 缩放类型
"StartTime" -- 时间轴起始时间不变
"Center"　-- 时间轴中间时间不变
"EndTime"　-- 时间轴终止时间不变

返　回：无

注　释：通过缩短时间轴长度，以使趋势曲线局部放大。

如果左右指示器不在曲线两端，则放大左右指示器之间的曲线，此时 LockString 无效。

示　例：如保持趋势变量“Trend1”的起始时间不变，且 Trend1.ScoeterPosRight 等于 0.0，起始时间为下午 1:00:00，曲线长度为 30 秒，要放大曲线起始部分，则：

HTZoomIn(Trend1,"StartTime");执行后，起始时间仍为 1:00:00，而曲线长度为 15 秒。

14.57　HTZoomOut( )

历史趋势曲线在时间轴方向曲线缩小一倍，即曲线的时间轴宽度放大一倍

语　法：HTZoomOut(Hist_Tag,LockString)

参　数：Hist_Tag　-- 历史趋势曲线的变量名，历史曲线类型

LockString -- 缩放类型

"StartTime" -- 时间轴起始时间不变

"Center"　-- 时间轴中间时间不变

"EndTime"　-- 时间轴终止时间不变

返　回：无

注　释：指示器位置对缩小没有影响，但曲线缩小后将设置

.ScooterPosLeft=0.0 和 .ScooterPosRight=1.0

示　例：如保持趋势变量“Trend1”的时间轴的中间时间不变，起始时间为下午 1:00:00，曲线长度为 30 秒，且 Trend1.ScoeterPosRight=0.9，要缩小曲线，则：

```
HTZoomIn(Trend1,"StartTime");
```

执行后，起始时间为 1:00:15，而曲线长度为 1 分钟，右指示器值 Trend1.ScoeterPosRight=1.0。

14.58 In( )

读指定 I/O 地址的端口的值。

语　法： IntResult=In(HexPortAddre);

参　数： HexPortAddre—I/O 端口地址，十六进制字符串，范围："0"—"3FF"。

返　回： -1,表示读操作失败；其它，I/O 端口值，其范围为 0～255。

注　释： 无

示　例： PortValue=In("200");

14.59 InfoAppActive( )

测试应用程序是否为活动的

语　法： DiscreteResult=InfoAppActive(AppName)

参　数： AppName -- 应用程序名,字符串或信息类型变量

返　回： DiscreteResult -- 1：应用程序正在运行
0：应用程序未运行

注　释： 为得到应用程序的标题，可使用 InfoAppTitle( )函数。

示　例：

InfoAppActive("Microsoft Word");
返回 1，表明 Word 程序正在运行；
返回 0，表明 Word 程序未运行。

14.60 InfoAppTitle( )

获得应用程序的标题或正在运行的 Window 任务列表名中指定的程序名

语　法： MessageResult=InfoAppTitle(ProgramExeName)
参　数： ProgramexeName -- 应用程序名,字符串或信息类型变量
返　回： MessageResult -- 应用程序的标题或正在运行的 Window 任务列表名中指定的程序名
注　释： 无
示　例：

InfoAppTitle("calc") 将返回 "Calculator"
InfoAppTitle("Word") 将返回 "Microsoft Word"

14.61 InfoDisk( )

获得本地或网络磁盘驱动器信息

语　法： IntergerResult=InfoDisk(Drive,InfoType,Trigger)
参　数： Drive -- 包含驱动器号的变量，信息类型
InfoType -- 要获取的信息类型
1：磁盘驱动器的总空间数(以字节计)
2：磁盘驱动器上的空闲空间数(以字节计)
Trigger -- 每当此值改变时,执行本函数,本参数可为任何变量(不只是系统变量)
返　回： IntergerResult -- 本地或网络磁盘驱动器信息
注　释： 无
示　例：

要每分钟得到磁盘的容量大小：
InfoDisk("C", 1, $分) 将返回 875236855（磁盘总空间数），
InfoDisk("C", 2, $分) 将返回 127323835（磁盘空闲空间数）。

14.62 InfoFile( )

获得文件或目录的信息

语　法：IntergerResult=InfoFile(Filename,InfoType,Trigger)
参　数：Filename -- 文件名信息
InfoType -- 要获取的信息类型
1：文件是否存在，存在 --返回 1，不存在 --返回 0
2：文件大小(字节数)
3：文件日期/时间(自 1970 年 1 月 1 日 0 时 0 分 0 秒起的相对秒数)
Trigger -- 每当此值改变时，执行本函数，本参数可为任何变量(不只是系统变量)
返　回：IntergerResult -- 文件或目录的信息，整数类型
注　释：文件名必须包含文件的完整路径，可以包含通配符(*，?)
示　例：
要每分钟知道文件是否存在：
InfoFile("C:\CSWORK\历史曲线.WIN",1,$分);
如返回 1，则：文件找到。
要每分钟知道文件大小：InfoFile("c:.exe", 2, $分);
要每分钟知道文件保存时间：InfoFile("c:.exe", 3, $分);

14.63 InfoAppDir( )

获得当前应用程序目录

语　法：MessageResult=InfoAppDir( )
参　数：无

返　回：当前应用程序目录，信息类型
注　释：无
示　例：

要得到当前应用程序目录，则：
AppDir=InfoAppDir( );
返回到 AppDir 中的值为"C:\CenturyStar".

14.64　InfoResource( )

获得系统资源值

语　法：IntergerResult=InfoREsource(ResourceType,Trigger)
参　数：ResourceType -- 要获取的系统资源类型
1：GDI 资源可用空闲空间的百分比
2：USER 资源可用空闲空间的百分比
3：当前内存中空闲空间的字节数
4：当前正在运行的任务数
Trigger　-- 每当此值改变时，执行本函数，本参数可为任何变量(不只是系统变量)
返　回：IntergerResult -- 系统资源值，整数类型
注　释：无
示　例：InfoResource(1, $分); 返回 54（空闲百分比）
InfoResource(2, $分); 返回 36（空闲百分比）
InfoResource(3, $分); 返回 11524093（字节数）
InfoResource(4, $分); 返回 14（任务数）

14.65　Int( )

返回小于或等于指定数值的最大整数

语　法：IntergerResult=Int(Number)
参　数：Number -- 数字、整数、实数变量
返　回：IntergerResult -- 整数值
注　释：无
示　例：

Int(4.7);　返回 4
Int(-4.7); 返回 -5
IOSetAppTopic( )改变 DDE 会话的应用程序名和话题名

14.66　IOSetAppTopic( )

改变 DDE 会话的应用程序名和话题名

语　法：SetDDEAppTopic(DDEDeviceName, AppName,TopicName)
参　数：DDEDeviceName -- DDE 设备名
AppName　　-- DDE 应用程序名
TopicName　　-- DDE 话题名
返　回：无
注　释：无
示　例：无

14.67　IOSetItem(　)

改变 DDE 会话的项目名，仅用于 IO 类型变量

语　法：IOSetItem("Tagname",Item)
参　数：Tagname -- IO 类型变量名

Item -- DDE 项目名

返 回：无

注 释：无

示 例：无

14.68 Log( )

返回指定数值的自然对数

语 法：RealResult=Log(Number)

参 数：Number -- 数字、整数、实数变量

返 回：RealResult -- 自然对数值

注 释：无

示 例：

Log(100); 返回 4.605

Log(1); 返回 0

14.69 Log10( )

返回以 10 为底的 x 的对数

语 法：Result=LogN(Number)

参 数：Number -- 数字、整数、实数变量

返 回：Result -- 对数值

注 释：无

示 例：

Log10(100); 返回 2

Log10(1); 返回

14.70 LogOff( )

退出登录

语　法：LogOff( )
参　数：无
返　回：无
注　释：无
示　例：无

14.71 LogOn( )

显示登录对话框，输入用户名和密码

语　法：LogOn( )
参　数：无
返　回：无
注　释：无
示　例：无

14.72 Out( )

写给定值到指定的 I/O 端口。

语　法：Out(HexPortAdder,Number)
参　数：HexPortAdder—I/O 端口地址，十六进制字符串，范围："0"—"3FF"。

Number　　—十进制数，范围：0～255。

返　回：无。写给定值到指定的 I/O 端口。

注　释：无

示　例：

Out("200",255)

表示把数 255 写到地址为 200H 的端口。

14.73　PI( )

返回圆周率值

语　法：RealResult=PI( )

参　数：无

返　回：RealResult -- 3.1415926...

注　释：无

示　例：无

14.74　PlaySound( )

通过 Windows 的声音设备(如果系统已安装)播放声音文件(.wav)

语　法：PlaySound(SoundName,Flags)

参　数：SoundName -- 要播放的声音文件(*.wav)，带路径。

Flags　　-- 为以下参数之一

0：停止播放

1：同步播放

2：异步播放

3：重复播放直到下次调用该函数

返　回：无

注　释：声音需要在安装了 wave 形式的音频设备上播放。
示　例：同步播放 C:\HORNS.WAV:
PlaySound ("C:\HORNS.WAV",1);

14.75　PrintEventsQueryForm( )

打印事件查询报表

语　法：PrintEventsQueryForm( )
参　数：无
返　回：无
注　释：在按钮或其他命令语言中用此函数时，可打印事件查询报表。
示　例：无

14.76　PrintFormTable( )

以组态报表格式调出文件，用于打印报表

语　法：PrintFormTable(FileName)
参　数：FileName -- 可以是“.TXT”“.CSV”“.MDL”等三种类型的文件名或信息变量类型，要求文件名为带路径的全名。
返　回：无
注　释：不是“.TXT”“.CSV”“.MDL”等三种类型的文件名或信息变量类型，调用不成功。本功能主要用来打印文件。
示　例：无

14.77 PrintHistAlarm( )

历史报警数据报表打印

语　法：PrintHistAlarm( )
参　数：无
返　回：无
注　释：在按钮或其他命令语言中用此函数时，可打印历史报警数据报表。
示　例：无

14.78 PrintHistData( )

历史数据报表打印

语　法:

```
PrintHistData(Tagname1,Tagname2,Tagname3,Tagname4,Tagname5,
Tagname6,StartDateTime,EndDateTime,IntervalTime,Options)
```

参　数：Options: 0-- 不设参数选择。
1-- 打印报表
2-- 打印曲线
3-- 打印报表、曲线
4-- 设置参数到对话框，由用户选择

返　回：无
注　释：无
示　例：无

14.79 PrintRealAlarm( )

实时报警数据报表打印

语　法：PrintRealAlarm( )
参　数：无
返　回：无
注　释：在按钮或其他命令语言中用此函数时，可打印实时报警数据报表。
示　例：无

14.80 PrintRealMessage( )

按行打印字符串

语　法：PrintRealMessage(PortNo, "CommandCode", Message);
参　数：PortNo -- 端口号
CommandCode -- 打印机控制代码,十进制,逗号分割, "EPSON1600, 27, 64, 3, 30"
Message -- 打印信息，字符串或信息变量
返　回： 0 -- 打印正常
1 -- 打印机初始化错误
2 -- 打印机端口错误
3 -- 打印机超时
注　释： 可根据需要在按钮或其他命令语言中用此函数时，实现按行打印字符串信息。
示　例：无

14.81　PrintWindow( )

打印指定窗口

语　法：PrintWindow(Window,Left,Top,Width,Height,Options)

参　数：Window -- 要打印的窗口名,字符串或信息类型变量

Left　-- 窗口左边框到打印页面左边的距离(单位：毫米)，数字、整数或实数变量

Right　-- 窗口右边框到打印页面右边的距离(单位：　毫米)，数字、 整数或实数变量

Width　-- 窗口在打印页面上的宽度(单位：毫米)，数字、整数或实数变量，0：最大匹配宽度

Height -- 窗口在打印页面上的高度(单位：毫米)，数字、整数或实数变量，0：最大匹配高度

Option -- 离散值(0 或 1)，仅用于 Width 和 Height 都为 0 时

1：窗口在最大纵横比下，以窗口尺寸的整数倍数打印

0：窗口以页面的最大纵横比打印

返　回：无

注　释：1：如果窗口中包含位图，应设置 Option=1，以免位图被拉长。

2：许多报表可通过本函数队列打印。建议用 PrintWindow( ) 函数取代 PrintHT( ) 函数，在每次打印屏幕而不是打印历史趋势曲线时，应从历史趋势曲线运行对话框对打印进行初始化。

3：如果希望窗口中的文字能被正确打印，建议将所有要被打印的窗口中的文本设成 "True Type" 字体。

4：打印窗口中的按钮时，按钮上的文本可能看不见，其原因是按钮上的文本为"System" 字体，而不是"True Type"字体。另外，"System"字体在打印机上与在屏幕上的效果略有不同，这时，可试着将按钮扩大。

示　例：

每天上午 8:30 打印三页报表，使用命令语言：

```
if ($时 == 8) && ($分 == 30) THEN
```

```
PrintWindow("1st Shift Summary",0,0,0);
PrintWindow("2nd Shift Summary",0,0,0);
PrintWindow("3rd Shift Summary",0,0,0);
ENDIF;
```

14.82 pvAddRealPoint( )

在指定的温控曲线控件中增加一个采样实时值。

语　法： pvAddRealPoint("ControlName", TimeOffset, Value);

参　数： ControlName -- 温控曲线控件名称。

TimeOffset -- 相对前一采样点的时间偏移量(即距前一值的时间间隔值)，第一个值取 0，单位为秒。

Value -- 温度的采样值，实数或实数变量。

示　例：

pvAddRealPoint("预热炉温控曲线", 5, 85,);

此语句在预热炉温控曲线控件中增加一个 85 度的温度采样实时值。此采样实时值距前一值的时间间隔值为 5 秒。

14.83 pvAddSetPoint( )

在指定的温控曲线控件中增加一段温度设定曲线。

语　法：

pvAddSetPoint("ControlName", TimeOffset, Value);

参　数： ControlName -- 温控曲线控件名称。

TimeOffset　-- 相对前一采样点的时间偏移量(即距前一值的时间间隔值)，第一个值取 0，单位为秒。

Value　-- 温度的设定值，实数或实数变量。

注　释：此函数仅用于自由设定方式。

示　例：

pvAddSetPoint("预热炉温控曲线", 5, 85);在预热炉温控曲线控件中增加一段温度设定曲线，其设定温度为 85 度，此设定值距前一值的时间间隔值为 5 秒。

14.84　pvClear( )

在指定的温控曲线控件中删除温控实时曲线或温控设定曲线。

语　法：pvClear("ControlName", CurveMode);

参　数：ControlName--温控曲线控件名称。

CurveMode--曲线选择参数，整数或整型变量。

1：清除温控实时曲线

0：清除温控设定曲线

示　例：

pvClear("预热炉温控曲线", 0);

此语句清除预热炉温控曲线控件中的温度设定曲线。

pvClear("预热炉温控曲线", 1);

此语句清除预热炉温控曲线控件中的温度实时曲线。

14.85 pvGetValue( )

在指定的温控曲线控件中获取指定时刻的温度设定值或指定时刻的温度实时值，若指定时刻无采样，则返回该时刻前最近的一次采样值。

语　法：

pvGetValue("ControlName",TimeOffset,"Tagname",Option);

参　数：

ControlName--温控曲线控件名称。

TimeOffset --相对时间坐标原点的时间偏移量（也就是绝对时间坐标），第一个值取 0，单位为秒。

TagName --实数或实数变量。

Option --曲线选择参数，整数或整型变量。

1：温度实时值

0：温度设定值

示　例：

pvGetValue("预热炉温控曲线",5, "预热炉实时温度",1);表示从预热炉温控曲线控件中获取指定时刻的温度实时值，该值距坐标原点的时间间隔为 5 秒，并将该值存放到变量“预热炉实时温度”中。

pvGetValue("预热炉温控曲线",5, "预热炉设定温度",0);表示从预热炉温控曲线控件中获取指定时刻的温度设定值，该值距坐标原点的时间间隔为 5 秒，并将该值存放到变量“预热炉设定温度”中。

14.86 pvInitSetCurve( )

初始化设定曲线。

语 法：

pvInitSetCurve("ControlName", "fileName");

参 数：ControlName--温控曲线控件名称。

fileName--fileName 文件以.csv 格式按曲线方式、曲线段数、初始温度、各段升温速率、设定温度、保温时间依次存放设定曲线信息。

示 例：

设文件：C:\Test.csv”存放数据如下:

0,

3,

25.5,

10,80,20,

15,50,10,

20,60,5,

注 释：

第一行：曲线设定方式。1：自由设定方式；0：升温保护方式。上例为升温保护方式。

第二行：曲线段数，上例为 3 段设定曲线。

第三行：初始温度，上例为 25.5 度

第四至六行：依次为三段设定曲线的升温速率、设定温度、保温时间。上例中的第四行表示升温速率为 10、设定温度为 80 度、保温时间为 20 秒。

14.87 pvLoadData( )

从指定的文件中读取温控设定曲线或温控实时曲线的采样历史数据值，文件名后缀必须为.csv。

语　法：

pvLoadData("ControlName", FileName, Option);

参　数：

ControlName--温控曲线控件的名称。

FileName--FileName 文件以.csv 格式按曲线方式、曲线段数、初始温度、各段升温速率、设定温度、保温时间依次存放设定曲线信息。

Option--曲线选择参数，整数或整型变量。

0：读取温控设定曲线

1：读取温控实时曲线的采样历史数据值

示　例：

“C:\test.csv”存放数据为：

```
0,
3,
25.5,
10,80,20,
15,50,10,
20,60,5,
```

```
pvLoadData( "预热炉温控曲线","c:\test.csv",0);
```

读取温控设定曲线并传送到"预热炉温控曲线"控件中显示出来。

```
pvLoadData( "预热炉温控曲线","c:\test.csv",1 );
```

读取温控实时曲线的采样历史数据值并传送到"预热炉温控曲线"控件中显示出来。

14.88 pvModifySetCurve( )

在指定的温控曲线控件中修改某段温控设定曲线。

语 法： pvModifySetCurve("ControlName", Index, RiseRate, SetValue, HoldTime);

参 数：

ControlName--温控曲线控件名称。

Index --温控曲线段索引编号。

RiseRate --设置温控曲线段的升温速率

SetValue --设置温控曲线段的设定温度

HoldTime --设置温控曲线段的保温时间

返 回： 无

注 释： 无

示 例：

pvModifySetCurve("预热炉温控曲线",2, 20,80,25);

将预热炉温控曲线控件中的第二段温控设定曲线设置为：升温速率 20；设定温度 80；保温时间 25

14.89 pvMoveSlide( )

在指定的温控曲线控件中设置游标左移或右移。

语　法：

pvMoveSlide("ControlName", Direction, PointNumber);

参　数：

ControlName--温控曲线控件名称。

Direction　--设置游标的移动方向

0：向右移动，越界则不动作。

1：向左移动，越界则不动作。

PointNumber--设置游标移动的采样点数

注　释：无

示 例：

pvMoveSlide("预热炉温控曲线", 1, 3);

在“预热炉温控曲线”控件中将右游标左移动 3 个采样点。

14.90　pvSaveData( )

保存温控曲线数据

语　法：

pvSaveData("ControlName", FileName, Option);

参　数：

ControlName--温控曲线控件名称。

FileName--FileName 文件以.csv 格式按曲线方式、曲线段数、初始温度、各段升温速率、设定温度、保温时间依次存放设定曲线信息。

Option--曲线选择参数，整数或整型变量。

0：读取温控设定曲线

1：读取温控实时曲线的采样历史数据值

返　回：无

注　释：无

示　例：

pvSaveData("预热炉温控曲线","c:\test.csv",0);

将"预热炉温控曲线"控件中的温控设定曲线以.csv 格式存放到文件"c:\test.csv"中。

pvSaveData("预热炉温控曲线","c:\test.csv",1);

将"预热炉温控曲线"控件中的温控实时曲线的采样历史数据值以.csv 格式存放到文件"c:\test.csv"中。

14.91　Rand( )

随机函数

语　法：RealResult=Rand()

参　数：无

返　回：随机数，范围 0～1，实数类型

注　释：无

示　例：

RealResult=100.*Rand()

以上例子得到一个 0～100 之间的实数。

14.92 RecipeDelete( )

此函数用于删除指定工程配方模板文件中当前指定的配方。

语　法：RecipeDelete(FileName, RecipeName)

参　数：FileName -- 工程配方名称。

RecipeName -- 工程配方模板文件中某一个配方的名字。该工程配方将被本函数从 FielName 指定的工程配方模板文件中删除。

返　回：无

注　释：指定的工程配方如果在工程配方模板文件中不存在，则不做任何操作。

示　例：

```
FileName="建材厂配方";
RecipeName="水泥管";
RecipeDelete (FileName, RecipeName);
```

或

```
RecipeDelete ("建材厂配方","水泥管");
```

以上两种方式均可将工程配方模板文件"建材厂配方" 中的名为"水泥管"的配方删除。

3.93 RecipeLoad( )

将指定工程配方模板文件中指定的工程配方调入到工程配方模板所定义的内部数据变量中。

语　法：RecipeLoad (FileName, RecipeName);

参　数：FileName -- 工程配名称

RecipeName -- 工程配方模板文件中某一个工程配方的名字。该工程配方将被本函数调入到工程配方模板所定义的内部数据变量中。

返　回：无

注　释：指定的工程配方如果在工程配方模板文件中不存在，则不做任何操作。

示　例：

```
FileName="建材厂配方";
RecipeName="水泥管";
RecipeLoad (FileName,RecipeName);
```

或

```
RecipeLoad ("建材厂配方","水泥管");
```

以上两种方式均可将工程配方模板文件"建材厂配方" 中的名为"水泥管"的配方调入到内部数据变量中。

14.94 RecipeSave( )

将内部数据变量以工程配方的形式写进工程配方模板文件。

语　法：RecipeSave (FileName,RecipeName);

参　数：FileName：工程配方名称。

RecipeName：工程配方模板文件中某一个工程配方的名字。该工程配方的值将由本函数从工程配方模板所定义的内部数据变量中调入。

返　回：无

注　释：指定的工程配方如果在工程配方模板文件中不存在，本函数将在最后一列创建一个新的。如果已经存在，则直接将该工程配方原有的值覆盖。

示　例：

```
FileName="建材厂配方";
RecipeName="水泥管";
RecipeSave (FileName,RecipeName);
```

或

```
RecipeSave ("建材厂配方","水泥管");
```

以上两种方式均可将工程配方模板文件："建材厂配方" 中的名为："水泥管" 的配方的值依据内部数据变量进行更新。

14.95 RecipeSelectNextRecipe( )

得到指定工程配方模板文件中的指定工程配方的下一个工程配方名。

语　法：

```
MessageTag=RecipeSelectNextRecipe(FileName, RecipeName)
```

参　数： FileName -- 工程配方名称，字符串或信息变量。

RecipeName -- 向本函数提交当前的工程配方名。

返　回： MessageTag -- 字符串或信息变量，当前工程配方的下一个工程配方的名字。

注　释： 1：如果指定的工程配方是指定的工程配方模板文件里的最后一个工程配方，则 RecipeName 的值保持不变。

2：如果指定的工程配方在指定的工程配方模板文件里并不存在，则返回最后一个工程配方的名字。

示　例：

```
FileName="建材厂配方";
RecipeName="水泥管";
MessageTag=RecipeSelectNextRecipe (FileName,RecipeName);
```

或

```
MessageTag =RecipeSelectNextRecipe ("建材厂配方","水泥管");
```

以上两种方式均可将工程配方模板文件"建材厂配方" 中的名为"水泥管" 的配方的下一个工程配方名"预制板"。

14.96 RecipeSelectRecipe( )

弹出配方选择对话框。该对话框列出指定的工程配方模板文件中的所有配方名。用户可以从中选择一个配方名或输入一个新的配方名。

语 法：

MessageTag=RecipeSelectRecipe (FileName, PromptMessage);

参 数：FileName -- 工程配方名称，字符串或信息变量。

PromptMessage -- 用户输入提示,字符串或信息变量。

返 回：MessageTag -- 出口参数，保存用户从配方选择对话框中得到的工程配方名，字符串或信息变量。

注 释：该配方名由别的工程配方管理函数使用，如：

RecipeDelete，RecipeLoad，RecipeSave，

RecipeSelectPreviousRecipe，RecipeSelectNextRecipe

示 例：无

14.97 RecipeSelectPreviousRecipe( )

得到指定工程配方模板文件中的指定工程配方的上一个工程配方名。

语 法：

MessageTag=RecipeSelectPreviousRecipe(FileName, RecipeName)

参 数：FileName -- 工程配方名称。

RecipeName -- 既是入口参数又是出口参数。向本函数提交当前的工程配方名，并被本函数重新赋值为当前工程配方的上一个工程配方的名字。

返 回：无

注 释：1：如果指定的工程配方是指定的工程配方模板文件里的第一个工程配方，则 RecipeName 的值保持不变。

2：如果指定的工程配方在指定的工程配方模板文件里并不存在，则

返回第一个工程配方的名字。

示　例：

```
FileName="建材厂配方";
RecipeName="水泥管";
MessageTag=RecipeSelectPreviousRecipe (FileName,RecipeName);
```

或

```
MessageTag=RecipeSelectPreviousRecipe ("建材厂配方","水泥管");
```

以上两种方式均可将工程配方模板文件"建材厂配方" 中的名为"水泥管"的配方的上一个工程配方名"电线管"。

14.98　ReDDELink( )

重新建立 DDE 连接

语　法：ReDDELink( )
参　数：无
返　回：无
注　释：无
示　例：无

14.99　ReFailDDELink( )

重新建立未成功的 DDE 连接

语　法：ReFailDDELink ( )
参　数：无
返　回：无
注　释：无
示　例：无

14.100 ReStarAlarmLog( )

重新启动报警记录

语　法：ReDDELink( )
参　数：无
返　回：无
注　释：无
示　例：无

14.101 ReStartHistoryDataLog( )

重新启动历史数据记录

语　法：ReStartHistoryDataLog( )
参　数：无
返　回：无
注　释：无
示　例：无

14.102 Round( )

按指定精度返回值

语　法：RealResult=Round(Number,Precision)
参　数：Number　　-- 数字、整数、实数变量
　　Precision -- 指定精度，数字、整数或实数变量
返　回：RealResult -- 指定精度的值
注　释：无
示　例：

Round(3.14159,2)；返回 3.14

14.103 SendKeys( )

发送键到当前获得输入焦点的应用程序。对于接受键值的应用程序来说，此键相当于从键盘输入。

语　法：SendKeys(KeyT)
参　数：KeyT -- 键代码或信息类型变量

Key	Code	Key	Code
BACKSPACE	{BACKSPACE} 或 {BS}	HOME	{HOME}
BREAK	{BREAK}	INSERT	{INSERT}
CAPSLOCK	{CAPSLOCK}	LEFT	{LEFT}
DELETE	{DELETE} 或 {DEL}	NUMLOCK	{NUMLOCK}
DOWN	{DOWN}	PAGEDOWN	{PAGEDOWN}或{PGDN}
END	{END}	PAGEUP	{PAGEUP}或{PGUP}
ENTER	{ENTER} 或 ~	PRTSC	{PRTSC}
ESCAPE	{ESCAPE} 或 {ESC}	RIGHT	{RIGHT}
TAB	{TAB}	UP	{UP}
F1	{F1}	F2	{F2}
F3	{F3}	F4	{F4}
F5	{F5}	F6	{F6}
F7	{F7}	F8	{F8}
F9	{F9}	F10	{F10}
F11	{F11}	F12	{F12}
a～z	{a}～{z}	0～9	{0}～{9}
SPACE	{SPACE}		

组合键(SHIFT、CTRL 和 ALT)的代码如下

Key	Code

SHIFT +
CTRL ^
ALT %

返 回：无
注 释：在调用本函数前，必须使接受键值的应用程序获得输入焦点。
本函数可用于向应用程序输入数据，或发送命令到应用程序。
示 例：

```
SendKeys("^{F1}"); 发送 Ctrl+F1
```

14.104 Sgn( )

获得数值的符号(正数、零或负数)

语 法：IntergerResult=Sgn(Number)
参 数：Number -- 数字、整数、实数变量
返 回：IntergerResult -- 1：正数，0：零，-1：负数
注 释：无
示 例：

```
Sgn(12.3); 返回 1
Sgn(0);    返回 0
Sgn(-123); 返回 -1
```

14.105 ShowAt( )

用垂直和水平像素坐标重新定位正在显示的窗口的中心点

语　法：ShowAt(Window,Horiz,Vert)

参　数：Window -- 窗口名,字符串或信息类型变量

Horiz　-- 垂直像素，数字、整数或实数变量

Vert　-- 水平像素，数字、整数或实数变量

返　回：无

注　释：窗口打开时，由垂直和水平像素定义窗口的中心点。当窗口超出屏幕范围时，将自动调整垂直和(或)水平像素坐标，使窗口完全显示在当前屏幕中

示　例：显示“历史曲线”画面窗口，其中心点在：(200,100)

ShowAt(″历史曲线″,200,100);

14.106　ShowTopLeftAt( )

用垂直和水平像素坐标重新定位正在显示的窗口的左上角

语　法：

ShowTopLeftAt(Window,Horiz,Vert)

参　数：Window -- 窗口名,字符串或信息类型变量

Horiz　-- 垂直像素，数字、整数或实数变量

Vert　-- 水平像素，数字、整数或实数变量

返　回：无

注　释：窗口打开时，由垂直和水平像素定义窗口的中心点。当窗口超出屏幕范围时，将自动调整垂直和(或)水平像素坐标，使窗口完全显示在当前屏幕中

示　例：显示“历史曲线”画面窗口，其左上角坐标为：(200,100)

ShowTopLeft(″历史曲线″,200,100);

14.107 ShowWindow( )

显示画面窗口

语 法：ShowWindow(WindowName)
参 数：WindowName -- 画面窗口名,字符串或信息类型变量
返 回：无
注 释：无
示 例：显示“历史曲线”画面窗口： ShowWindow(″历史曲线″);

14.108 Sin( )

计算正弦值

语 法：Result=Sin(AngleNumber)
参 数：AngleNumber -- 度数，数字、整数或实数变量
返 回：Result -- AngleNumber 的正弦值
注 释：无
示 例：

Sin(90); 返回 1
Sin(0); 返回 0]

14.109 SQLAppendStatement( )

使用 SQLStatement 的内容继续一个 SQL 语句

语 法：

Result=SQLAppendStatement(ConnectionID,szSQLStatement);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.110　SQLClearStatement ( )

释放和 SQLHandle 指定的语句相关的资源

语　法：

Result=SQLClearStatement(ConnectionID, SQLHandle);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值

SQLHandle -- 当使用高级函数时，SQL 返回 SQLHandle，供内部使用

返　回：Result -- 整数，具体意义参见错误信息报告释放和 SQLHandle 指定的语句相关的资源

注　释：无

示　例：无

14.111　SQLClearTable( )

清空表格

语　法：Result=SQLClearStatement(ConnectionID, SQLHandle);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()

函数为每个数据库连接分配的一个数值

SQLHandle -- 当使用高级函数时，SQL 返回 SQLHandle，供内部使用

返　回： Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.112 SQLCommit( )

定义一组 transaction 命令的结尾

语　法：Result=SQLCommit(ConnectionID);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.113 SQLConnect( )

连接世纪星到 szconnectstring 指定的数据库中

语　法：ConnectionID=SQLConnect(szConnectString);

参　数：szConnectString -- 指示数据库以及任何附加登录信息的字符串，如"DSN=dsn;UID=uid;PWD=pwd"

返　回：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值,是一个较大的整数，在 SQL 函数中特别重要,具体意义参见错误信息报告

注　释：无
示　例：无

14.114　SQLCreateTable( )

创建新表

语　法：

```
Result=SQLCreateTable(ConnectionID,TableName,
TemplateName );
```

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
TableName -- 数据库中表名
TemplateName -- 表格模板名
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.115　SQLDelete( )

删除一个或多个记录

语　法：Result=SQLDelete(ConnectionID, TableName, WhereExpr);
参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
TableName -- 数据库中表名
WhereExpr -- 定义一个条件，此条件对表格中所有的行或为真，或为假。命令只对条件为真的行起作用。格式：

ColumnNamecomparison_operatorexpression

注　意：如果列为字符串类型，表达式必须带单引号。下例将选择所有 name 列为 centurystar 的行：

name=’ centurystar’

下例选择 agg 列界于 20 到 30 之间的行：

agg>=20 and agg<30

返　回：Result -- 整数，具体意义参见"错误信息报告"

14.116　SQLDisconnect( )

断开和数据库的连接

语　法：Result=SQLDisconnect(ConnectionID);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.117　SQLDropTable( )

破坏一个表格

语　法：Result=SQLDropTable(ConnectionID, TableName);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值

TableName --　数据库中表名

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无
示　例：无

14.118　SQLEndSelect( )

在 SQLSelect()后使用本函数来释放资源

语　法：Result=SQLEndSelect(SQLHandle);
参　数：SQLHandle -- 当使用高级函数时，SQL 返回 SQLHandle，供内部使用
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.119　SQLExecute( )

执行一个 SQL 语句

语　法：
Result=SQLExecute(ConnectionID,SQLHandle, szSQLStatement);
参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
SQLHandle -- 当使用高级函数时，SQL 返回 SQLHandle，供内部使用
SzSQLStatement -- 实际语句
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.120 SQLFirst( )

选择由 SQLSelect()选择的表格中的首项记录

语 法：Result=SQLFirst(SQLHandle);
参 数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
返 回：Result -- 整数，具体意义参见错误信息报告
注 释：无
示 例：无

14.121 SQLGetRecord( )

从当前选择缓存区返回由 RecordNumber 指定的记录

语 法：Result=SQLGetRecord(SQLHandle, RecordNumber);
参 数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
RecordNumber -- 返回的实际记录个数
返 回：Result -- 整数，具体意义参见错误信息报告
注 释：无
示 例：无

14.122 SQLInsert( )

使用捆绑表中指定的变量中的值在表格中插入一个新记录。捆绑表中的参数定义了世纪星中变量和数据库表格列的对应关系。

语 法：

Result=SQLInsert(ConnectionID,TableName, BindList);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数为每个数据库连接分配的一个数值

TableName-- 数据库中表名

BindList -- SQL.DEF 文件中定义的捆绑列表名

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.123　SQLInsertEnd( )

释放插入语句

语　法：Result=SQLInsertEnd(SQLHandle);

参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.124　SQLInsertExecute( )

执行已经准备的语句

语　法：Result=SQLInsertExecute(SQLHandle);

参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.125 SQLInsertPrepare( )

准备一个插入语句

语 法：

Result=SQLInsertPrepare(ConnectionID, TableName, BindList);

参 数： ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函数 为每个数据库连接分配的一个数值

TableName-- 数据库中表名

BindList -- SQL.DEF 文件中定义的捆绑列表名

返 回： Result -- 整数，具体意义参见错误信息报告

注 释： 无

示 例： 无

14.126 SQLLast( )

选择 SQLSelect()指定表格中的末项

语 法： Result=SQLLast(SQLHandle);

参 数： SQLHandle -- 当使用高级函数时，SQL 返回 SQLHandle，供内部使用

返 回： Result -- 整数，具体意义参见错误信息报告

注 释： 无

示 例： 无

14.127 SQLLoadStatement( )

准备一个插入语句

语　法：Result=SQLLoadStatement(ConnectionID, szFileName);
参　数：ConnectionID --用户创建的内存整形变量，用来保存 SQLConnec()
函数为每个数据库连接分配的一个数值
szFileName -- 包含信息的文件名
返　回：Result -- 整数，具体意义参见错误信息报告

14.128 SQLNext( )

选择表中的下一条记录

语　法：Result=SQLNext(SQLHandle);
参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.129 SQLNemRows( )

指出有多少条记录符合上一次 SQLSeclect()的指定

语　法：Result=SQLNumRows(SQLHandle);
参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无
示　例：无

14.130 SQLPrepareStatement( )

本语句为 SQLSetParam() 准备一个 SQL 语句。该语句可以由 SQLSetStatement(),或 SQLLoadStatement()创建

语　法： Result=SQLPrepareStatement(ConnectionID);
参　数： ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec() 函 数为每个数据库连接分配的一个数值
返　回： Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.131 SQLPrev( )

选择表中的上一条记录

语　法：Result=SQLPrev(SQLHandle);
参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.132 SQLRollback( )

撤消最近的位于 SQLTransact()后的尚未“提交”的指令

语　法：Result=SQLRollback(ConnectionID);
参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.133 SQLSelect( )

访问一个数据库并返回一个表的信息，此信息可以被 SQLFirst(),SQLLast(),SQLNext()和 SQLPrev().

语　法：

Result=SQLSelect(ConnectionID,TableName,BindList, hereExpr, OrderByExpr);

参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
TableName -- 数据库中表名
BindList -- SQL.DEF 文件中定义的捆绑列表名
WhereExpr -- 定义一个条件，此条件对表格中所有的行或为真，或为假。命令只对条件为真的行起作用。格式：
ColumnNamecomparison_operatorexpression

注　意：如果列为字符串类型，表达式必须带单引号。
下例将选择所有 name 列为 centurystar 的行：
name=’centurystar’
下例选择 agg 列界于 20 到 30 之间的行：

agg>=20 and agg<30

OrderByExpr--定义排序的列和方向。只有列可以进行排序。格式：ColumnName[ASC|DESC]。

例：为名为 manager 的列进行升排序，“manager ASC”。为多列排序，格式：

ColumnName[ASC|DESC],ColumnName[ASC|DESC]

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.134　SQLSetParamChar( )

将指定的参数设成特定的字符串。本函数可以调用多次，以设置参数值

语　法：

Result=SQLSetParamChar(SQLHandle,ParameterNumber,
ParameterValue, MaxLen);

参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用

ParameterNumber-- 语句中实际的参数个数

ParameterValue-- 设定的实际值

MaxLen -- 列的最大容量。这一设定决定参数是字符串型还是长字符串型。如果 MaxLen 小于数据库允许的最大长度，参数是字符串型，否则是长字符串型

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.135 SQLSetParamDate( )

将指定的日期参数设置成特定的字符串

语　法：

```
Result=SQLSetParamDate(SQLHandle,ParameterNumber,
                       szDataString)
```

参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
ParameterNumber-- 语句中实际的参数个数
szDataString -- 日期格式的字符串

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.136 SQLSetParamFloat( )

将指定的参数设置成特定的值

语　法：

```
Result=SQLSetParamFloat( SQLHandle, ParameterNumber,
                         ParameterValue );
```

参　数：SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
ParameterNumber-- 语句中实际的参数个数
ParameterValue -- 设定的实际值

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.137 SQLSetParamInt( )

将指定的参数设置成特定的值

语 法：

```
Result=SQLSetParamInt( SQLHandle, ParamterNumber,
                       ParameterValue );
```

参 数： SQLHandle--当使用高级函数时，SQL 返回 SQLHandle，供内部使用
ParameterNumber-- 语句中实际的参数个数
ParameterValue -- 设定的实际值

返 回： Result -- 整数，具体意义参见错误信息报告

注 释： 无

示 例： 无

14.138 SQLStatement( )

启动一个语句缓存区，此语句缓存区由 SQLStatement()使用

语 法：

```
Result=SQLSetStatement(ConnectionID,szSQLStatement );
```

参 数： ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()
函数为每个数据库连接分配的一个数值
SzSQLStatement -- 实际语句

返 回： Result -- 整数，具体意义参见错误信息报告

注 释： 无

示 例： 无

14.139 SQLTransact( )

定义了一组访问指令的开始。在 SQLTransact()指令和 SQLCommit()指令之间的一组指令称为一个访问组。一个访问组可以象单个访问一样操作。在 SQLTransact()指令执行后，所有接下的操作都不委托给数据库，直到 SQLCommit()指令执行

语　法：Result=SQLTransact(ConnectionID);
参　数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：无
示　例：无

14.140 SQLUpdate( )

用当前的世纪星变量更新数据库中的记录
语　法：
Result=SQLUpdate(ConnectionID, TableName, BindList, WhereExpr);
参 数：ConnectionID -- 用户创建的内存整形变量，用来保存 SQLConnec()函数为每个数据库连接分配的一个数值
TableName-- 数据库中表名
BindList -- SQL.DEF 文件中定义的捆绑列表名
WhereExpr -- 定义一个条件，此条件对表格中所有的行或为真，或为假。命令只对条件为真的行起作用。格式：
ColumnName comparison_operator expression
返　回：Result -- 整数，具体意义参见错误信息报告
注　释：如果列为字符串类型，表达式必须带单引号。

下例将选择所有 name 列为 centurystar 的行：

name=’ centurystar’

下例选择 agg 列界于 20 到 30 之间的行：

agg>=20 and agg<30

示　例：无

14.141　SQLUpdateCurrent( )

更新数据库的当前行记录

语　法：Result=SQLUpdateCurrent( );

参　数：无

返　回：Result -- 整数，具体意义参见错误信息报告

注　释：无

示　例：无

14.142　Sqrt( )

计算平方根

语　法：RealResult=Sqrt(Number)

参　数：Number　　-- 数字、整数、实数变量

返　回：RealResult -- Number 的平方根

注　释：无

示　例：Sqrt(16); 返回 4

14.143 StartApp( )

启动指定窗口应用程序

语　法：StartApp(AppName)
参　数：AppName -- 要启动的应用程序名
返　回：无
注　释：在应用程序名称前，使用全路径，由于本函数不支持长文件名，在使用路径名和文件名　时，用 DOS 环境下显示的路径名和文件名。
示　例：

```
StartApp("C:\Winword\Word") ;
```

启动 Microsoft Word。

14.144 StopHistoryDatLog( )

停止历史数据记录

语　法：StopHistoryDatLog( )
参　数：无
返　回：无
注　释：无
示　例：无

14.145 StringASCII( )

获得指定信息类型变量首字符的 ASCII 值

语　法：ASCIIResult=StringASCII(Message)

参　数：Message -- 字符或信息类型变量，整数类型
返　回：ASCIIResult -- 指定信息类型变量首字符的 ASCII 值
注　释：无
示　例：

StringASCII("A"); 返回 65
StringASCII("About"); 返回 65
StringASCII("about"); 返回 97

14.146　StringChar( )

获得指定 ASCII 码所对应的字符

语　法：MessageResult=StringChar(ASCII)
参　数：ASCII -- ASCII 码或整数类型变量
返　回：MessageResult -- 指定 ASCII 码所对应的字符
注　释：本函数的一个用处是可以不用键盘给信息类型变量添加字符，还可以插入 ASCII 码在 32-126 范围之外的字符。
示　例：

OneLineMessage=MessageTag+StringChar(13)+StringChar(10); 在信息变量 MessageTag 后将回车（CR）和换（LF）符加到 MessageTag 的末尾，结果存到信息变量 OneLineMessage。

14.147　StringCompare( )

字符串比较

语　法：DiscResult= StringCompare(Message1,Message2)
参　数：Message1 - 字符串或信息类型变量
Message2 - 字符串或信息类型变量
返　回：DiscResult - 1：字符串相同，0：字符串不同

注　释：无
示　例：无

14.148　StringFromChartLength( )

将历史曲线长度转换为字符串

语　法：
　　MessageResult=StringFromChartLength(Hist_ChartLength)
参　数：Hist_ChartLength - 历史曲线长度
返　回：MessageResult - 信息变量
注　释：无
示　例：
　　历史曲线长度：Hist.ChartLength=10;则
　　StringFromChartLength（Hist.ChartLength）　返回"00小时10分00秒"

14.149　StringFromFileTable( )

根据索引值从文件中查找字符串

语　法：
　　MassageResult=StringFromFileTable(FileName,IndexData)
参　数：FileName -- 文件名（带路径），字符串或信息变量
　　IndexData -- 索引值，数字或实数变量
返　回：MessageResult　　-- 信息变量
注　释：CSV 文件格式如下：
　　1.1,1.1 号设备故障
　　1.2,1.2 号设备故障
　　.....
示　例：

文件名为：C:\TEST\ALARMMESS.CSV;则
MassageResult=StringFromFileTable("C:\TEST\ALARMMESS.CSV", 1.1) 返回 "1.1 号设备故障"

14.150 StringFromIntg( )

将整数值转换为其他进制的字符串

语　法：MessageResult=StringFromIntg(Number,Base)
参　数：Number -- 数字、整数变量
返　回：MessageResult -- 其他进制的字符串
注　释：无
示　例：

```
StringFromIntg (18,2);  返回 "10010"
StringFromIntg (18,8);  返回 "22"
StringFromIntg (18,10); 返回 "18"
StringFromIntg (18,16); 返回 "12"
```

14.151 StringFromReal( )

将实数值转换为字符串，该字符串以浮点格式表示，或以指数格式表示

语　法：MessageResult=StringFromReal(Real,Precision,Type)
参　数：Number -- 数字、实数类型变量
Precision -- 小数点后的位数
Type -- 显示方式
"f"：浮点数格式
"e"：小写"e"的指数格式

"E"：大写"E"的指数格式

返　回：MessageResult -- 指定格式的字符串

注　释：无

示　例：

StringFromReal(123.4567,2,"f"); 返回 "123.45"
StringFromReal(123.4567,2,"e"); 返回 "1.23e2"
StringFromReal(123.4567,1,"E"); 返回 "1.2E2"

14.152　StringFromTime( )

将时间值(从 1970 年 1 月 1 日起，以秒为单位)转换为字符串

语　法：

MessageResult=StrintFromTime(SecsSince1-1-70,StringType)

参　数：SecsSince1-1-70 -- 时间值(从 1970 年 1 月 1 日起至 2038 年 1 月 18 日，以秒为单位)

StringType　　-- 显示方式

1："YY 年 MM 月 DD 日" 格式
2："HH 时 MM 分 SS 秒" 格式
3："YY 年 MM 月 DD 日 HH 时 MM 分 SS 秒" 格式

返　回：MessageResult -- 指定格式的字符串

注　释：无

示　例：

StrintFromTime(86400,1) 返回 "70 年 01 月 02 日"
StrintFromTime(86400,2) 返回 "12 时 00 分 00 秒"
StrintFromTime(86400,3) 返回 "70 年 01 月 02 日 12 时 00 分 00 秒"

14.153 StringInString( )

查找一字符串在另一字符串中第一次出现的位置

语 法：

IntergerResult=StringInString(Text,SearchFor,StartPos,CaseSens)

参 数： Text -- 要查找的字符串，字符串或信息类型变量
SearchFor -- 被查找的字符串，字符串或信息类型变量
StartPos -- 开始查找的起始位置，数字或整数类型变量
CaseSens -- 是否区分大小写(1：区分，0：不区分)，数字或整数类型变量

返 回： IntergerResult -- 查找到的位置

注 释： 返回 0, 表示未找到。

示 例：

StringInString ("Today is Sunday","is",1,0) 返回 7
StringInString ("Today is Sunday","day",1,0) 返回 3
StringInString ("Todayis Sunday","day",8,0)返回 13
StringInString ("Today is Sunday","day",1,1) 返回 0
StringInString ("Today is Veteran's Day","Day",1,1) 返回 20

14.154 StringLeft( )

返回指定字符串中左边指定字符个数的字符串

语 法： MessageResult=StringLeft(Text,Chars)

参 数： Text -- 字符串或信息类型变量
Chars -- 字符个数，数字或整数类型变量

返 回： MessgaeResult -- 字符串

注 释： 如果 Chars=0, 则返回整个字符串

示　例：

```
StringLeft("Today is Sunday",5);  返回 "Today"
StringLeft("Today is Sunday",8);  返回 "Today is"
StringLeft("Today is Sunday",30); 返回 "Today is Sunday"
StringLeft("Today is Sunday",0);  返回 "Today is Sunday"
```

14.155　StringLen( )

返回字符串的长度

语　法：IntergerResult=StringLen(Text)
参　数：Text -- 字符串或信息类型变量
返　回：IntergerResult -- 字符串的长度
注　释：无
示　例：

```
StringLen("Today is Sunday"); 返回 15
StringLen("12.345"); 返回 6
```

14.156　StringLower( )

将指定字符串中所有大写字母转换为小写字母

语　法：MessageResult=StringLower(Text)
参　数：Text -- 字符串或信息类型变量
返　回：IntergerResult -- 转换后的字符串
注　释：无
示　例：

```
StringLower("Today is Sunday"); 返回 "today is sunday"
```

StringLower("TODAY IS SUNDAY"); 返回 "today is sunday"

14.157　StringMid( )

从字符串中指定位置开始，返回指定个数的字符

语　法： MessageResult=StringMid(Text,StartChar,Chars)
参　数： Text　　-- 字符串或信息类型变量
StartChar -- 字符在字符串中的位置，数字或整数类型变量
Chars　　-- 字符个数，数字或整数类型变量
返　回： MessgaeResult -- 字符串
注　释： 无
示　例： StringMid("Today is Sunday",1,5);　返回"Today"
StringMid("Today is Sunday",7,2);　返回 "is"
StringMid("Today is Sunday",10,6); 返回 "Sunday"

14.158　StringReplace( )

替换或改变字符串中的指定部分

语　法：

MessageResult=StringReplace(Text,SearchFor,ReplaceWith,CaseSens,NumToReplace, MatchWholeReplace)

参　数： Text　　-- 字符串或信息类型变量
SearchFor　-- 要替换或改变的字符串，字符串或信息类型变量
ReplaceWith -- 替换字符串，字符串或信息类型变量
CaseSens　-- 是否区分大小写(1：区分，0：不区分)，数字或整数类型变量

NumToReplace -- 要替换的次数，0：全部；<0：不替换，数字或整数类型变量

MatchWholeReplace -- 是否需要全字匹配(1：需要，0：不需要)，数字或整数类型变量

返　回： IntergerResult -- 替换或改变后的字符串

注　释： 使用此函数能获取信息类型变量，并替换成字符、单词或短语，函数不能识别特殊字符，如 @#$%&*()等，函数将它们视为分隔符。

示　例：

```
StringReplace("In From Within","In","Out",0,1,0);
返回 "Out From Within"

StringReplace("In From Within","In","Out",0,0,0);
返回 "Out From WithOut"

StringReplace("In From Within","In","Out",1,0,0);
返回 "Out  From Within"

StringReplace("In From Within","In","Out",0,0,1);
返回 "Out From Within"
```

14.159 StringRight( )

返回指定字符串中右边指定字符个数的字符串

语　法： MessageResult=StringRight(Text,Chars)

参　数： Text -- 字符串或信息类型变量

Chars -- 字符个数，数字或整数类型变量

返　回： MessgaeResult -- 字符串

注　释： 如果 Chars=0，则返回整个字符串

示　例：

```
StringRight ("Today is Sunday",6);  返回 "Sunday"
StringRight ("Today is Sunday",9); 返回 "is Sunday"
```

StringRight ("Today is Sunday",30); 返回 "Today is Sunday"
StringRight ("Today is Sunday",0); 返回 "Today is Sunday"

14.160 StringSpace( )

在信息类型变量或表达式中产生一个空格串

语　法：MessageResult=StringSpace(NumSpaces)
参　数：NumSpaces　　-- 空格数，数字或整数类型变量
返　回：MessgaeResult -- 字符串
示　例：

StringSpace(4); 返回 "　　"
"Today"+StringSpace(1)+"is"+ StringSpace(2)+"Sunday";
返回 "Today is Sunday"

14.161 StringTest( )

检测字符串首字符，确定是否为指定类型

语　法：DiscreteResult=StringTest(Text,TestType)
参　数：Text　　-- 字符串或信息类型变量
TestType -- 字符类型
1：字母、数字('A'-'Z','a'-'z',和'0'-'9')
2：数字('0'-'9')
3：字母('A'-'Z','a'-'z')
4：大写字母('A'-'Z')
5：小写字母('a'-'z')
6：标点符号(0x21-0x2F)
7：ASCII 字符(0x00-0x7F)

8：十六进制字符('A'-'F','a'-'f',和'0'-'9')
9：可打印字符(0x20-0x7E)
10：控制字符(0x00-0x1F 或 0x7F)
11：空白字符(0x09-0x0D 或 0x20)

返　回：DiscreteResult -- 1：是指定类型，0：不是指定类型；若参数不在 1～11 之间时，返回结果都为 0。

注　释：无

示　例：

```
StringTest("About",3);  返回 1
StringTest("About",10); 返回 0
StringTest("123.45",2); 返回 1
```

14.162　StringToIntg( )

将由数字字符组成的字符串转换成整数数值

语　法：IntergerResult=StringToIntg(Text)

参　数：Text -- 字符串或信息类型变量

返　回：IntergerResult -- 整数数值

注　释：转换时，如果首字符不是一个数字(忽略空格)，则字符串的值=0；如果首字符是　一个数字，则继续读后续的字符，直到遇到一个非数字值为止

示　例：

```
StringToIntg("About"); 返回 0
StringToIntg("Today is Sunday"); 返回 0
StringToIntg("123.45"); 返回 123
StringToIntg("3About"); 返回 3
```

14.163　StringToReal( )

将由数字字符组成的字符串转换成实数值

语　法：IntergerResult=StringToReal(Text)
参　数：Text -- 字符串或信息类型变量
返　回：IntergerResult -- 实数值
注　释：转换时，如果首字符不是一个数字(忽略空格)，则字符串的值=0；如果首字符是 一个数字，则继续读后续的字符，直到遇到一个非数字值为止
示　例：

```
StringToReal("About"); 返回 0.0
StringToReal("Today is Sunday"); 返回 0.0
StringToReal("123.45"); 返回 123.45
StringToReal("3About"); 返回 3.0
```

14.164　StringTrim( )

删除字符串中的空格

语　法：MessageResult=StringTrim(Text,TrimType)
参　数：Text　　-- 字符串或信息类型变量
TrimType -- 删除方式
1：删除前边空格(第一个非空格字符的左边)
2：删除后边空格(最后一个非空格字符的右边)
3：删除单词间单个空格外的多余空格
返　回：MessgaeResult -- 处理后的字符串
注　释：无
示　例：

```
StringTrim(" Today is Sunday ",1);  返回 " Today is Sunday "
StringTrim(" Today is Sunday ",2); 返回 " Today is Sunday"
StringTrim("Today is Sunday",3); 返回"Today is Sunday"
```

14.165 StringUpper( )

将指定字符串中所有小写字母转换为大写字母

语　法：MessageResult=StringUpper(Text)
参　数：Text -- 字符串或信息类型变量
返　回：IntergerResult -- 转换后的字符串
注　释：无
示　例：

StringUpper("Today is Sunday"); 返回 "TODAY IS SUNDAY"
StringUpper("today is sunday"); 返回 "TODAY IS SUNDAY"

14.166 tagReadDisc( )

读离散变量的值

语　法：DiscResult=tagReadDisc("Tagname");
参　数：Tagname--变量名或字符串
返　回：DiscResult--离散变量
注　释：无
示　例：

离散 1=tagReadDisc("dd"+"2");
读离散变量"dd2"的值到变量"离散 1"中

14.167 tagReadInt( )

读整数变量的值

语　法：IntResult=tagReadInt(″Tagname″);
参　数：Tagname--变量名或字符串
返　回：IntResult--整数变量
注　释：无
示　例：

整数 1=tagReadInt(″aa″+″2″);
读整数变量"aa2"的值到变量"整数 1"中

14.168　tagReadFloat( )

读实数变量的值

语　法：FloatResult=tagReadFloat(″Tagname″);
参　数：Tagname--变量名或字符串
返　回：FloatResult--实数变量
注　释：无
示　例：

实数 1=tagReadFloat(″cc″+″2″);
读实数变量″cc2"的值到变量"实数 1"中

14.169　tagReadMess( )

读信息变量的值

语　法：MessResult=tagReadMess(″Tagname″);
参　数：Tagname--变量名或字符串
返　回：MessResult--信息变量

注　释：无

示　例：

信息 1=tagReadMess("mm"+"2");

读信息变量"mm2"的值到变量"信息 1"中

14.170　tagWriteDisc( )

写离散值到离散变量中

语　法：DiscResult=tagWriteDisc("Tagname", DiscValue)

参　数：Tagname--变量名或字符串

DiscValue--离散值 0、1 或离散变量

返　回：DiscResult--离散变量；0：成功；1：失败

示　例：

FORIndex=1TO10tagWriteDisc("dd"+StringFromIntg(Index, 10), 1);NEXT;将离散值"1"写到离散变量"dd1"--"dd10"中

14.171　tagWriteFloat( )

写实数值到实数变量中

语　法：

FloatResult=tagWriteFloat("Tagname", FloatValue)

参　数：Tagname--变量名或字符串

FloatValue --实数或实数变量

返　回：DiscResult--离散变量；0：成功；1：失败

注　释：无

示　例：

FOR Index=1 TO 10

tagWriteFloat("aa"+StringFromIntg(Index, 10), 12.52);NEXT;
将实数值"12.52"写到实数变量"aa1"--"aa10"中

14.172 tagWriteInt( )

写整数值到整数变量中

语　法： IntResult=tagWriteInt("Tagname", IntValue);
参　数： Tagname--变量名或字符串
IntValue--整数或整数变量
返　回： DiscResult--离散变量；0：成功；1：失败
注　释： 无
示　例：
FORIndex=1TO10tagWriteInt("aa"+StringFromIntg(Index, 10), 12);NEXT;将整数值"12"写到整数变量"aa1"--"aa10"中

14.173 tagWriteMess( )

写实数值到实数变量中

语　法：
MessResult=tagWriteMess("Tagname", String);
参　数： Tagname--变量名或字符串
String --字符串或信息变量
返　回： DiscResult--离散变量；0：成功；1：失败
注　释： 无
示　例：
FORIndex=1TO10tagWriteMess("aa"+StringFromIntg(Index, 10),

"ABCDEF");NEXT;将字符串"ABCDEF"写到信息变量"aa1"--"aa10"中

14.174 Tan( )

计算正切值

语　法： Result=Tan(AngleNumber)
参　数： AngleNumber -- 数字、整数、实数变量
返　回： Result -- 计算结果
注　释： 无
示　例：

Tan(45); 返回 1
Tan(0); 返回 0

14.175 Text( )

按指定格式转换模拟(整数、实数)变量成字符串

语　法： MessageResult=Text(Analog_Tag,Format_Text)
参　数： Analog_Text -- 数字、整数、实数变量
Format_Text -- 转换格式，如 "#0.00"
返　回： MessageResult -- 转换后的字符串
注　释： 无
示　例：

Text(1234,"#.00"); 返回 1234.00
Text(123.4,"#.00"); 返回 123.40
Text(45.678,"#.00"); 返回 45.68

14.176　Timer( )

定时器函数。

语　法：Timer("TimerControlTag",TimeLength)

参　数：TimerControlTag—整数、离散变量，0 表示复位定时器；1 表示启动定时器.

TimeLength　—整数变量或数字，单位为秒，表示定时器定时时间长度。

返　回：复位时总返回 0 ；启动定时器后，定时器的定时时间到时返回 1 ，直到复位前一直保持 1 。

注　释：

在完成一个定时器操作后要再次启动定时器，应对定时器进行复位。例如：当 TimerFlag=1，执行了 Timer("TimerFlag",10）函数，表示定时 10 秒，10 秒后，函数返回 1；要再次使用定时器函数，应先让 TimerFlag=0，执行 Timer("TimerFlag",10)，定时器复位；注意启动定时器和复位定时器的定时长度应相同，复位后的定时器才能重新启动。

示　例：

Timer("TimerFlag",100)；由变量 TimerFlag 控制定时器操作，定时器的时间长度为 100 秒。

14.177　Trunc( )

截取指定实数(浮点数)，去掉小数点右边部分

语　法：ResultNumericTag=Trunc(Number)

参　数：Number -- 数字、整数、实数变量

返　回：ResultNumericTag -- 整数类型值

注　释：此函数相当于将一个实数类型变量赋给一个整数类型变量

示　例：

Trunc(123.45)；返回 123

Trunc(-123.4)；返回 –123

14.178　txAddMessage( )

此函数用于在指定的文本显示框控件中给指定文本显示框添加一个字符串信息。

语　法：

txAddMessage("ControlName",FileOffset,Message_Tag,LineFeed);

参　数 ：

ControlName：用户定义的文本显示框控件名称。

FileOffset：设置要加的字符串信息的超始位置，可以是整数、实数、整型或实型变量，-1 表示加在末尾。

Message_Tag：字符串或信息变量

LineFeed：加完字符信息后是否要换行，离散变量，0 或 1，0 表示不换行，1 表示换行。

示　例 ：

txAddMessage("文本显示框", -1, “abcdefg", 1)

该语句执行后在名为"文本显示框"控件末尾加入一行字符“abcdefg”，并自动换行。

14.179　txClear( )

此函数用于清除指定的文本显示框控件中的内容。

语　法：txClear("ControlName");

参　数：ControlName：用户定义的文本显示框控件名称，可以为中文名或英文名。

返　回：无

注　释：无

示　例：

txClear("文本显示框");

该语句执行后，清除文本显示框中所有内容。

14.180　txDisplayMessage( )

此函数用于控制文本显示框控件显示文本的起行

语　法：txDisplayMessage("ControlName", StartLine);

参　数：ControlName：用户定义的文本显示框控件名称，可以为中文名或英文名。

StartLine:显示文本的起行，整数或整数变量，最小值为 1。

返　回：无

注　释：无

示　例：

txDisplayMessage("文本显示框", 2);该语句执行后，文本显示框控件从第二行开始显示有关内容。

14.181　txLoadFile( )

调入已有的文本显示框内容。

语　法：txLoadFile("ControlName", FileName);

参　数：

"ControlName"—文本显示框控件名；

FileName— 包含文本显示框内容的文件名，字符串或信息变量；

返　回：

1— 调入成功

1— 调入不成功

注　释：FileName 为该文件保存的文件名，要带全路径。

示　例：

txLoadFile("文本显示框","c:\演示工程\文本显示框.CSV")调入“c:\演示工程\文本显示框.CSV”文件到"文本显示框"控件中。

14.182　txSaveFile( )

保存指定的文本显示框数据。

语　法：

txSaveFile("ControlName", FileName);

参　数：

"ControlName"—文本显示框控件名；

FileName— 包含文本显示框数据的文件名，字符串或信息变量；

返　回：

1— 调入成功

1— 调入不成功

注　释：FileName 为该文件保存的文件名，要带全路径。

示　例：

txSaveFile("文本显示框","c:\演示工程\文本显示框.CSV")保存“文本显示框”的数据到该工程目录下的“c:\演示工程\文本显示框.CSV”文件中

14.183 UpdateRealChart( )

语 法：UpdateRealChart("RealChartName")
参 数：RealChartName：实时曲线名称，可以为中文名或英文名。
返 回：无
注 释：UpdateRealChart("RealChartName")函数用于刷新实时曲线
示 例：无

14.184 UserDll( )

调用用户自定义函数

语 法：UserDll(FunctionNo)
参 数：FunctionNo -- 动态库中的函数编号。
返 回：无
注 释：用户自定义函数，即由用户自己编制动态库 DLL, 在命令语言中，通过本函数调用。需要编制动态库的方法及示例程序，请与本公司直接联系。
示 例：无

14.185 WWControl( )

激活、最小化、最大化、关闭应用程序

语 法：WWControl(AppTitle,ControlType)
参 数：AppTitle -- 应用程序名，实际字符串或信息类型变量
ControlType -- 控制类型

"Restore"：激活并显示应用程序窗口
"Minimize"：最小化应用程序窗口
"Maximize"：最大化应用程序窗口
"Close"：关闭应用程序窗口

返　回：无
注　释：无
示　例：

```
WWControl("Calculator","Restore");
WWControl("Microsoft Excel","Close");
WWControl(InfoAppTitle("Word"), "Close");
```

14.186　xxClear( )

清除棒图控件中的指定棒图

语　法：xxClear("ControlName", Index);
参　数：

ControlName：用户定义的棒图控件名称，可以为中文名或英文名。
Index：给出棒图控件中的棒图索引号，取值范围 1-8，当取值为-1 时，则清除所有棒图。

返　回：无
注　释：无
示　例：

xxClear("发电量棒图", 1);清除发电量棒图中的第 1 个棒图。
xxClear("发电量棒图", -1);　清除发电量棒图中的所有棒图。

14.187 xbSetHisPara( )

设置带图历史数据参数

语 法： xbSetHistPara(ControlName, Year,Month, Day, Hour, Minute, Second);

参 数： ControlName--表示带图名;

Year,Month, Day, Hour, Minute, Second---分别表示历史带图的起 始时间的年、月、日、时、分、秒，可以是整数或整数变量;

返 回： 无

注 释： 无

示 例： 无

14.188 xbSetParameter( )

设置带图参数

语 法：

xbSetParameter(ControlName, AminValue, AHValue, ALValue);

参 数： ControlName：用户定义的带图控件名称，可以为中文名或英文名;

AminValue: Amin 的值，可以是整数或整数变量;

AHValue: AH 的值，可以是整数或整数变量;

ALValue: AL 的值，可以是整数或整数变量;

返 回： 无

注 释： 无

示 例： 无

14.189 xbSetTagname( )

设置带图变量名

语　法：xbSetTagname(ControlName, Tagname);
参　数：ControlName：用户定义的带图控件名称，可以为中文名或英文名；
Tagname: 带图所反映的变量名，字符串或信息变量；
返　回：无
注　释：无
示　例：无

14.190 xbSetTimeLength( )

设置带图显示时间长度

语　法：xbSetTimeLength(ControlName, TimeLength);
参　数：
ControlName：用户定义的带图控件名称，可以为中文名或英文名；
TimeLength: 设置带图显示时间长度，单位：秒
返　回：无
注　释：无
示　例：无

14.191 xxClear( )

清除棒图控件中的指定棒图

语 法：

xxClear("ControlName", Index);

参 数：

ControlName：用户定义的棒图控件名称，可以为中文名或英文名。

Index：给出棒图控件中的棒图索引号，取值范围 1-8，当取值为-1时， 则清除所有棒图。

返 回：无

注 释：无

示 例：

xxClear("发电量棒图", 1);清除发电量棒图中的第 1 个棒图。

xxClear("发电量棒图", -1);清除发电量棒图中的所有棒图。

14.192 xxModify( )

修改棒图控件中的指定棒图，

语 法：xxModify("ControlName", "Tagname", Index);

参 数：

ControlName：用户定义的棒图控件名称，可以为中文名或英文名。

Tagname：用来修改指定棒图所表示的变量。

Index：给出棒图控件中的棒图索引号，取值范围 1-8，当取值为-1时，则清除所有棒图。

返 回：无

注 释：无

示 例：

xxModify("发电量棒图","一号机发电量",1);

用“一号机发电量”修改发电量棒图中的第 1 个棒图。

14.193 xyAddPoint( )

此函数用于在指定的 XY 曲线控件中给指定曲线添加一个数据点。

语　法:xyAddPoint ("ControlName", X, Y, Index);

参　数：

ControlName：用户定义的 XY 曲线控件名称。

X：设置数据点的 x 轴坐标值，可以是整数、实数、整型或实型变量

Y：设置数据点的 y 轴坐标值，可以是整数、实数、整型或实型变量

Index：给出 XY 曲线控件中的曲线索引号，取值范围 1-8。

返　回：无

注　释：无

示　例：

xyAddPoint ("动力-速度曲线", 50, 30, 1);

xyAddPoint ("动力-速度曲线", 动力, 速度, 2);

第一个例子的语句执行后在"动力-速度曲线"控件中索引号为 1 的曲线上添加一个数据点，该点的坐标值为(50，30)，表示该点的动力是 50，速度是 30。

第二个例子的语句执行后在"动力-速度曲线"控件中索引号为 1 的曲线上添加一个数据点，该点的坐标值是动力和速度的实时变量数据库中的值。

14.194 xyClear( )

此函数用于在指定的 XY 曲线控件中清除指定曲线。

语　法：xyClear("ControlName",Index);

参　数：

ControlName：用户定义的 XY 曲线控件名，可以为中文名或英文名。

Index：给出 XY 曲线控件中的曲线索引号，取值范围 1-8，当取值为 -1 时，则清除所有曲线。

示　例：

```
xyAddNewPoint ( "动力-速度曲线", 1 );
xyAddNewPoint ( "动力-速度曲线", -1 );
```

第一句语句执行后在"动力-速度曲线"中清除索引号为 1 的指定曲线。

第二句语句执行后在"动力-速度曲线"中清除所有曲线。

14.195　xyGetValue( )

获得指定 xy 曲线指定点的值。

语　法：

```
xyGetValue( "ControlName","xName","yName",Index,PointNumber )
```

参　数：

"ControlName" -- xy 曲线控件名；

"xName"-- 指定的 x 返回值，整数或实数类型变量；

"yName"-- 指定的 y 返回值，整数或实数类型变量；

Index -- xy 曲线的索引号，取值范围为 1～8；

PointNumber -- xy 曲线中指定曲线点的位置，从 1 开始计数，小于等于最大采样数。

返　回：无

示　例：

```
xyGetValue("动力-速度曲线","动力","速度",1,2)
```

14.196　xyLoadFile( )

调入已有的 xy 曲线数据。

语　法：xyLoadFile("ControlName",FileName)

参　数：

"ControlName"—xy 曲线控件名；

FileName— 包含 xy 曲线数据的文件名，字符串或信息变量；

返　回：

1— 调入成功

0— 调入不成功

说　明：FileName 为该文件保存的文件名，要带全路径。

示　例：

xyLoadFile("动力-速度曲线","c:\演示工程\动力速度.CSV")调入“c:\演示工程\动力速度.CSV”文件到"动力-速度曲线"控件中。

14.197　xySaveFile( )

保存指定的 xy 曲线数据。

语　法：xySaveFile("ControlName",FileName)

参　数：ControlName -- xy 曲线控件名；

FileName -- 包含 xy 曲线数据的文件名，字符串或信息变量；

返　回：

1— 保存成功

0— 保存不成功

说　明：FileName 为该文件保存的文件名，要带全路径。

示　例：

xySaveFile("动力-速度曲线","c:\演示工程\动力速度.CSV")保存“动力-速度曲线”的数据到该工程目录下的“c:\演示工程\动力速度.CSV”文件中。

14.198 xySelectPen( )

选择 xy 曲线的某一曲线。

语 法： xySelectPen("ControlName",Index)

参 数：

ControlName — XY 曲线控件名；

Index -- 给出 XY 曲线控件中的曲线索引号，取值范围 1-8；-1 表示选择全部曲线。

说 明： 选择 XY 曲线中指定笔的曲线

示 例：

xySelectPen("动力-速度曲线",1)

选择索引号为 1 的曲线到“动力-速度曲线”中

14.199 xySetMinMaxValue( )

动态设定 xy 曲线 X 轴和 Y 轴的最大最小值。

语 法：

xySetMinMaxValue("ControlName",XMinValue,XMaxValue, YMinValue, YMaxValue);

参 数：

ControlName — xy 曲线控件名；

XMinValue — X 轴最小值；

YMinValue — Y 轴最小值；

XMaxValue — X 轴最大值；

YMaxValue — Y 轴最大值；

说 明： 无

示 例：

xySetMinMaxValue("动力-速度曲线",-100，200，-200，500)设定“动力-速度曲线”曲线 X 轴和 Y 轴的最小最大值分别为-100，200，-200，500。

第十五章　图形控件

15.1　图形控件概述

世纪星为用户提供了十多种专业图形控件，每种图形控件中有多个图形控件单元。这些图形控件单元均可实现无限缩放，为了不改变一些专用设备如时钟、仪表等图形形状，对这些图形控件单元进行了按比例放大或缩小处理。图形控件中每个单元都具有专用参数输入对话框，在这些对话框中，开发人员输入少量参数，本系统自动处理这些参数，生成图形控件单元的属性连接和动画连接。

世纪星的图形控件具有以下特点：

一、是系统开发人员的向导。

开发人员绘制图形控件单元，建立属性和动画连接后，用系统的“分裂单元”功能，对图形控件单元进行分解，然后分析每个图素，从中可以学习画面组合、3D 图形制作、动画连接等功能的用法。

二、提高系统开发人员的开发效率。

开发人员利用图形控件单元的组合，可以快速生成工作画面。还可以利用系统的“分裂单元”功能，借用部分或全部分裂后的图素，组合成所需图形。

15.2　图形控件使用方法

在绘图工具栏中，选择图形控件按钮，便可以弹出如图 15-1 窗口：

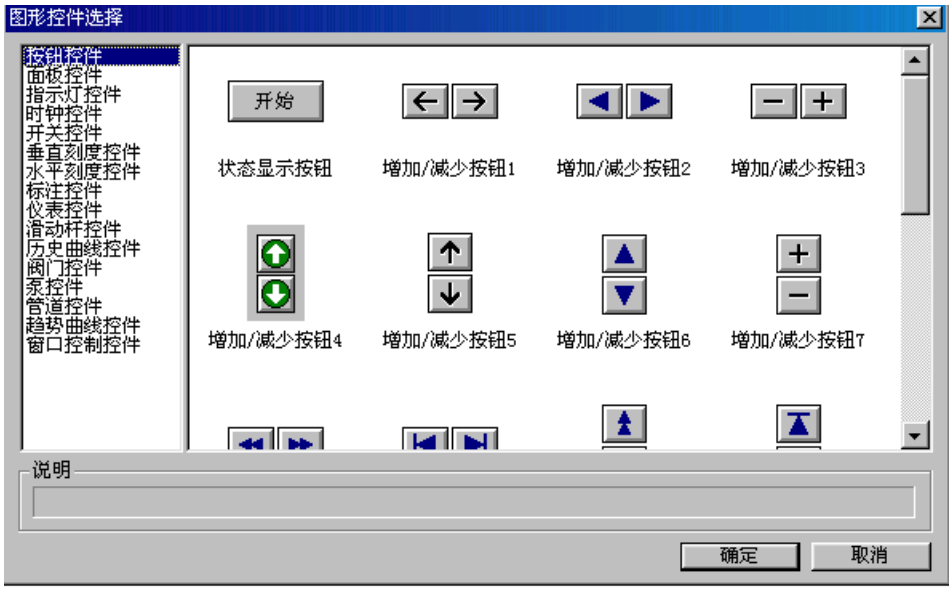


图 15-1

上述对话框的左面窗口为图形控件类型窗口，共包含十四种专业图形控件种类：按钮控件、面板控件、指示灯控件、时钟控件、开关控件、垂直刻度控件、水平刻度控件、标注控件、仪表控件、滑动杆控件、历史曲线控件、阀门控件、泵控件及管道控件。

选择图形控件类型有两种方法：一是用鼠标左键选择；二是用键盘的光标键选择，用回车键选中。

对话框的右面窗口为图形控件单元选择窗口，其中图形控件单元是与图形控件类型相对应的图形控件单元。

选中图形控件类型后，即可用鼠标左键双击图形控件单元或选中所需图形控

件单元后，按对话框的确定键选择。

15.3　按钮类型

本图形控件类型包含二十二种图形控件单元，单元图形及其属性连接和动画连接如下：

一、状态显示按钮：

图15-2

图15-3

本按钮为启动/停止按钮，并且在按钮上显示当前按钮所代表的功能状态。用户可用此按钮来控制离散设备，如电机的启动和停止等。

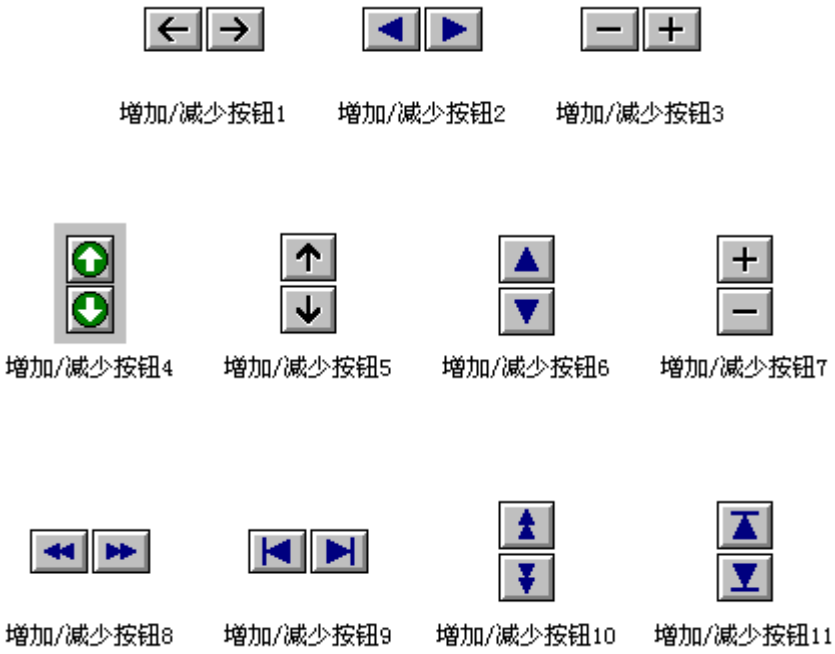
系统运行时，本按钮为触敏按钮，按钮弹起时显示图15-2 形状，按钮按下时显示图15-3 形状。

本按钮可以通过图形控件按钮连接对话框来连接到一离散变量。还可以改变按钮在两种离散状态下显示的文本，以及按钮按下时按钮上的填充颜色。

状态显示按钮的连接对话框如图15-4所示：

图15-4

二、增加/减少按钮：



以上按钮为水平方向和垂直方向上的增加/减少按钮。用户可用此按钮作为系统模拟量的输入，如控制步进电机的转速等。

系统运行时，本按钮为触敏按钮，用户每按动按钮，对应的变量值按固定步长增加或减少。

本按钮可以通过图形控件按钮连接对话框来连接到一整数或实数变量。还可以设定按钮每次按下时变量增加或减少的步长。

增加/减少按钮的连接对话框如图15-5 ：

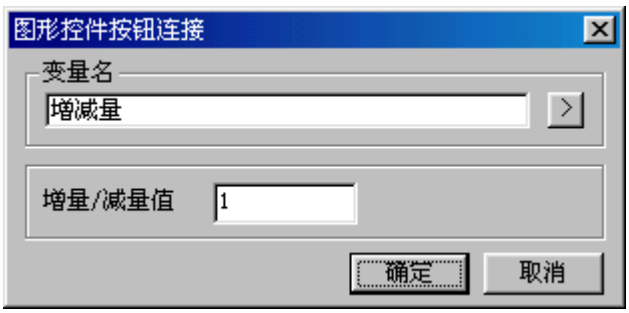


图15-5

对话框参数说明如下：

变　量　名：按钮和变量字典连接的变量名，整数或实数类型。按“>”按钮可以选择变量字典中已经定义的整数或实数变量。

增量/减量值：按钮每次按下时，变量值增加或减少的步长。

三、指示灯按钮：

系统运行时，本按钮为触敏按钮，按钮弹起时和按下时指示灯颜色显示如图15-6。用户可用此按钮作为离散设备的输入。

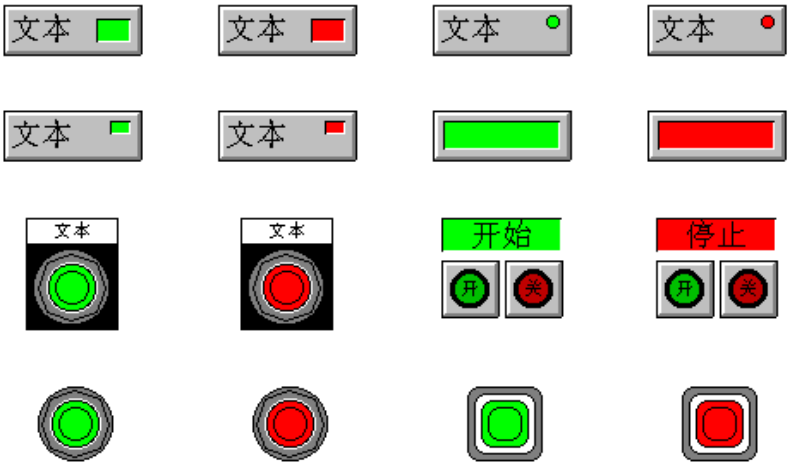


图15-6

本按钮可以通过图形控件按钮连接对话框来连接到一离散变量。可以改变按钮弹起和按下时按钮上的填充颜色。

指示灯按钮的连接对话框如图15-7 ：

图15-7

对话框参数说明如下：

变 量 名：按钮和变量字典连接的变量名，离散类型。按“>”按钮可以选择变量字典中已经定义的离散变量。
按钮文本：按钮上显示的文本。
颜色属性：按钮按下和弹起时的填充颜色。
激 活 键：定义激活本按钮的热键，当 Ctrl 和/或 Shift 选择时有效。

四、离散下按式按钮：

图15-8为离散下按式按钮。用户可用此按钮作为离散设备的输入。系统运行时，本按钮为触敏按钮。

图15-8

本按钮可以通过图形控件按钮连接对话框来连接到一离散变量。可以改变按钮和文本的颜色。

离散下按式按钮的连接对话框如图15-9 ：

对话框参数说明如下：

变 量 名：按钮和变量字典连接的变量名，离散类型。按“>”按钮可以选择变量字典中已经定义的离散变量。
按钮文本：按钮上显示的文本。
颜色属性：按钮和文本的颜色设置。

图15-9

五、命令语言下按按钮：

图15-10

图15-10为命令语言下按按钮。用户可用此按钮作为带命令语言的离散设备的输入。系统运行时，本按钮为触敏按钮。

本按钮可以通过图形控件按钮连接对话框来连接到一离散变量。可以改变按钮和文本的颜色。

命令语言下按按钮的连接对话框如 图15-11 ：

对话框参数说明如下：

按钮按下表达式：按钮下按的条件表达式。按“>”按钮可以选择变量字典中已经定义变量。

按钮文本：按钮上显示的文本。

颜色属性：按钮和文本的颜色设置。

命令语言：输入按钮按下、按住和弹起时命令语言。

图15-11

15.4　面板类型

本图形控件类型包含十种图形控件单元，单元图形及其属性连接和动画连接如下：

一、单框面板：

以上面板为单框3D显示面板。用户可以用它们作为控制面板的底框。

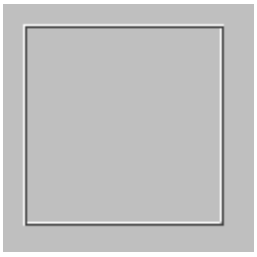
单框面板的连接对话框如图15-12 ：

对话框参数说明如下：

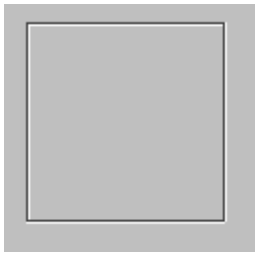
高亮颜色：面板3D显示时高亮部分颜色属性。

阴影颜色：面板3D显示时阴影部分颜色属性。

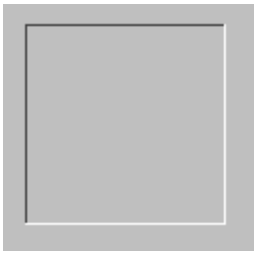
填充颜色：面板填充区域颜色属性设置。



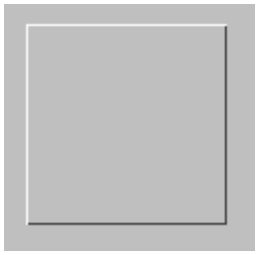
单凸形



单凹形



凹形



凸形

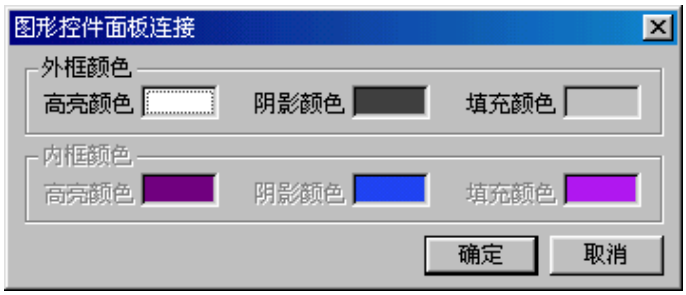
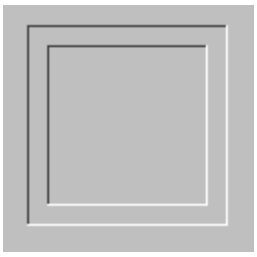
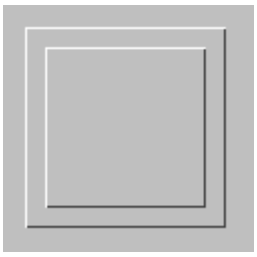


图15-12

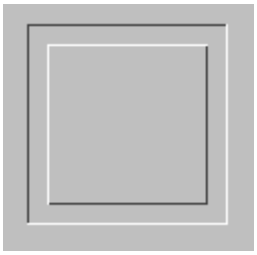
二、双框面板：



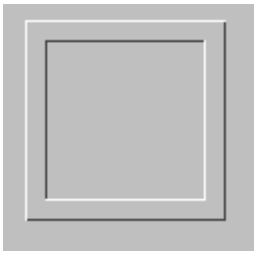
双凹形



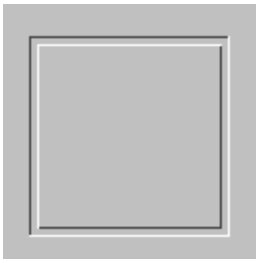
双凸形



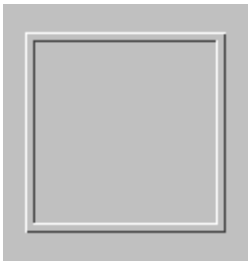
内凸外凹形1



内凹外凸形1



内凸外凹形2



内凹外凸形2

以上面板为双框3D显示面板。用户可以用它们作为控制面板的底框。双框面

板的连接对话框如图15-13 ：

图15-13

对话框参数说明如下：

高亮颜色：面板3D显示时高亮部分颜色属性。
阴影颜色：面板3D显示时阴影部分颜色属性。
填充颜色：面板填充区域颜色属性设置。

15.5 指示灯类型

本图形控件类型包含六种图形控件单元，单元图形及其属性连接和动画连接如下：

管形指示灯

圆形指示灯

园顶指示灯

平顶指示灯

面板指示灯

矩形指示灯

上述指示灯可以作为系统离散量的输出指示，如系统报警等。指示灯可以由指定的表达式控制是否闪烁。指示灯的连接对话框如图15-14、图15-15 ：

图15-14

对话框参数说明如下：

变　量　名：按钮和变量字典连接的变量名，离散类型。按“>”按钮可以选择变量字典中已经定义的离散变量。

指示灯颜色：指示灯在开和关两种状态下显示的颜色。

允 许 闪 烁：定义指示灯是否具有闪烁功能。

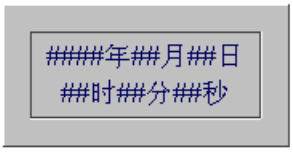
闪烁表达式：控制指示灯闪烁的表达式。表达式为1时，指示灯闪烁。

闪 烁 速 度：选择指示灯闪烁的频率。闪烁频率在运行系统配置菜单中设置。

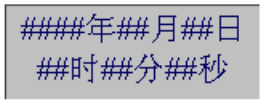
图15-15

15.6 时钟类型

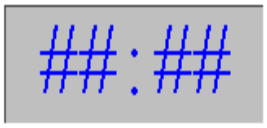
本图形控件类型包含六种图形控件单元，单元图形及其属性连接和动画连接如下：



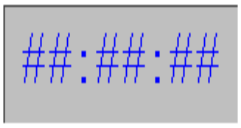
面板时钟1



面板时钟2



时间时钟1



时间时钟2

模拟时钟1

模拟时钟2

上述时钟用于显示系统当前的日期、时间。

时钟控件的连接对话框如图15-16 ：

图15-16

对话框参数说明如下：

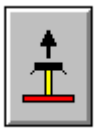
边框颜色：时钟边框的显示颜色。
背景颜色：时钟背景的显示颜色。
文本颜色：时钟文本的显示颜色。

15.7 开关类型

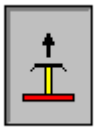
本图形控件类型包含四种图形控件单元，单元图形及其属性连接和动画连接如下：

推杆开关(打开状态)

推杆开关(闭合状态)



推动开关(打开状态)



推动开关(闭合状态)

摇轴开关(打开状态)

摇轴开关(闭合状态)

拨动开关(打开状态)

拨动开关(闭合状态)

以上开关可以用来控制离散设备，如电机的启动和停止等。可以通过图形控件开关连接对话框来连接到一离散变量。开关的连接对话框如图15-17 ：

对话框参数说明如下：

变 量 名：开关和变量字典连接的变量名，离散类型。按“>”按钮可以选择变量字典中已经定义的离散变量。

激 活 键：定义激活本开关的热键，当 Ctrl 和/或 Shift 选择时有效。

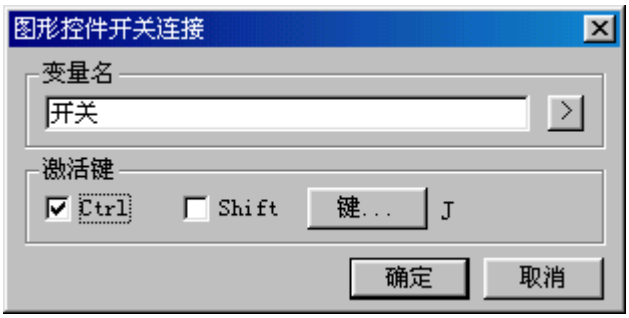


图15-17

15.8　垂直刻度类型

本图形控件类型包含十二种图形控件单元，单元图形及其属性连接和动画连接如图15-18：

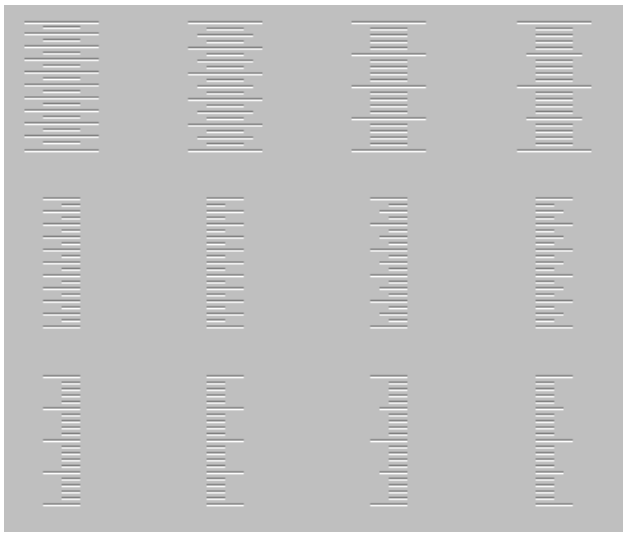


图15-18

以上垂直刻度可以和标注一起指示物体垂直方向上的尺寸大小。如滑动杆输入、温度显示等。

垂直刻度的连接对话框如图15-19 ：

图15-19

对话框参数说明如下：

高亮颜色：刻度3D显示时高亮部分颜色属性。
阴影颜色：刻度3D显示时阴影部分颜色属性。

15.9 水平刻度类型

本图形控件类型包含十二种图形控件单元，单元图形及其属性连接和动画连接如图15-20：

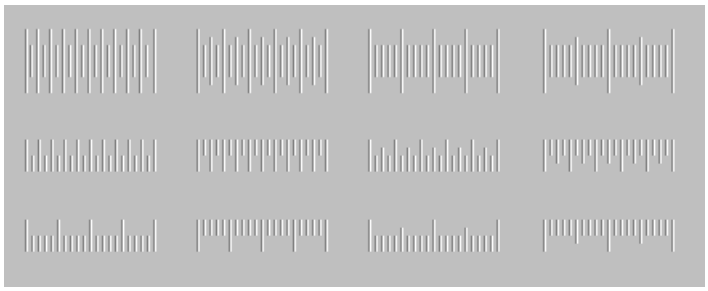


图15-20

以上水平刻度可以和标注一起指示物体水平方向上的尺寸大小。如滑动杆输入、温度显示等。水平刻度的连接对话框如图15-21 ：

图15-21

对话框参数说明如下：

高亮颜色：刻度3D显示时高亮部分颜色属性。
阴影颜色：刻度3D显示时阴影部分颜色属性。

15.10　标注类型

本图形控件类型包含十二种图形控件单元，单元图形及其属性连接和动画连接如下：

图15-22为标注控件，可以和水平/垂直刻度一起指示物体的尺寸大小。

通过标注连接对话框可以设定标注的范围、标注方向、标注颜色、小数点显示位数、标注显示位置及标注显示的字体。

标注的连接对话框如图15-23 ：

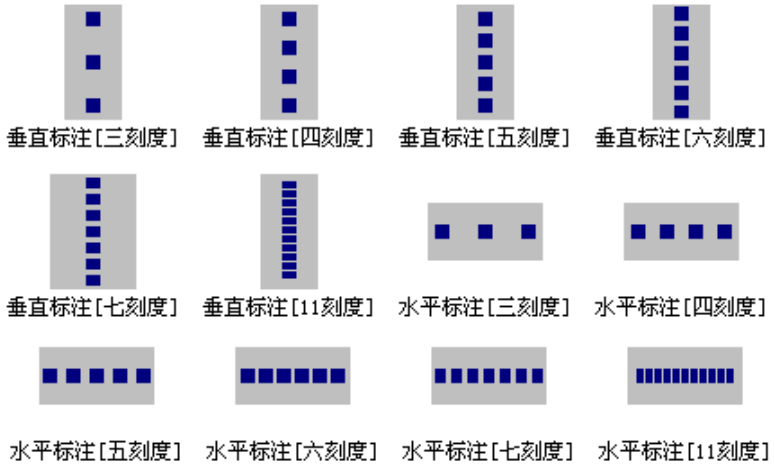

垂直标注[三刻度]
垂直标注[四刻度]
垂直标注[五刻度]
垂直标注[六刻度]
垂直标注[七刻度]
垂直标注[11刻度]
水平标注[三刻度]
水平标注[四刻度]
水平标注[五刻度]
水平标注[六刻度]
水平标注[七刻度]
水平标注[11刻度]

图15-22

图形控件标注连接
标注值范围
最 小 值
0
最 大 值
100
标注递增方向
从上[左]往下[右]
从下[右]往上[左]
标注属性
标注颜色
小数点位数
0
标注位置
居左
居中
居右
标注字体
字体
Times New Roman
确定
取消

图15-23

对话框参数说明如下：

最 小 值：　标注显示的最小值。
最 大 值：　标注显示的最大值。
标注方向：　标注从最小值到最大值的方向。
标注颜色：　标注显示的颜色属性。
小数点位数：标注显示的小数点位数。
标注位置：　标注显示时的位置。
标注字体：　标注显示的字体。

15.11　仪表类型

本图形控件类型包含六种图形控件单元，单元图形及其属性连接和动画连接如下：

一、　速度表

速度表可以用来显示电机转速等，如图15-24。

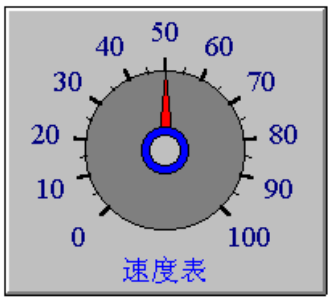


图15-24

本速度表可以通过图形控件仪表连接对话框来连接到一整数或实数变量。速

度表的连接对话框如图15-25。

图15-25

对话框参数说明如下：

仪表名称： 显示在速度表上的名称。

变 量 名： 速度表和变量字典连接的变量名，整数或实数类型。按“>”按钮可以选择变量字典中已经定义的整数或实数变量。

最 小 值： 速度表显示的最小值。

最 大 值： 速度表显示的最大值。

名称颜色： 速度表上显示的名称颜色属性。

标注颜色： 速度表的标注显示的颜色属性。

背景颜色： 速度表的背景显示的颜色属性。

小数点位数： 速度表的标注显示的小数点位数。

二、 面板仪表

面板仪表用于在控制面板上显示模拟量数值的变化。如图15-26

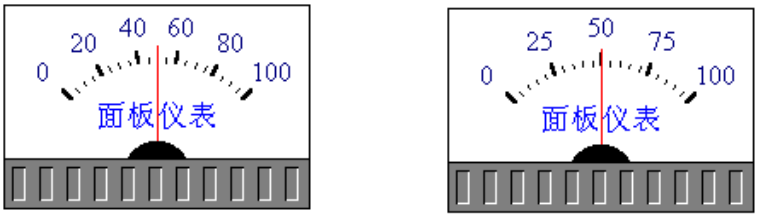


图15-26

面板仪表可以通过图形控件仪表连接对话框来连接到一整数或实数变量。面板仪表的连接对话框如图15-27所示：

图15-27

对话框参数说明如下：

仪表名称：　显示在面板仪表上的名称。

变 量 名：　面板仪表和变量字典连接的变量名，整数或实数类型。按“>”

按钮可以选择变量字典中已经定义的整数或实数变量。

最 小 值： 面板仪表显示的最小值。

最 大 值： 面板仪表显示的最大值。

名称颜色： 面板仪表上显示的名称颜色属性。

标注颜色： 面板仪表的标注显示的颜色属性。

背景颜色： 面板仪表的背景显示的颜色属性。

小数点位数： 面板仪表的标注显示的小数点位数。

三、 指示灯仪表

指示灯仪表用于数据变化的进程。如图15-28所示:

图15-28

指示灯仪表可以通过图形控件仪表连接对话框来连接到一整数或实数变量。

指示灯仪表的连接对话框如图15-29所示：

对话框参数说明如下：

仪表名称： 显示在指示灯仪表上的名称。

变 量 名： 指示灯仪表和变量字典连接的变量名，整数或实数类型。按“>”按钮可以选择变量字典中已经定义的整数或实数变量。

最 小 值： 指示灯仪表显示的最小值。

最 大 值： 指示灯仪表显示的最大值。

名称颜色： 指示灯仪表上显示的名称颜色属性。

填充颜色： 指示灯仪表上进程填充的颜色属性。

四、　温度仪表

温度仪表用于显示物体的温度。如图15-30所示:

图15-29

图15-30

温度仪表可以通过图形控件仪表连接对话框来连接到一整数或实数变量。

温度仪表的连接对话框如图15-31所示:

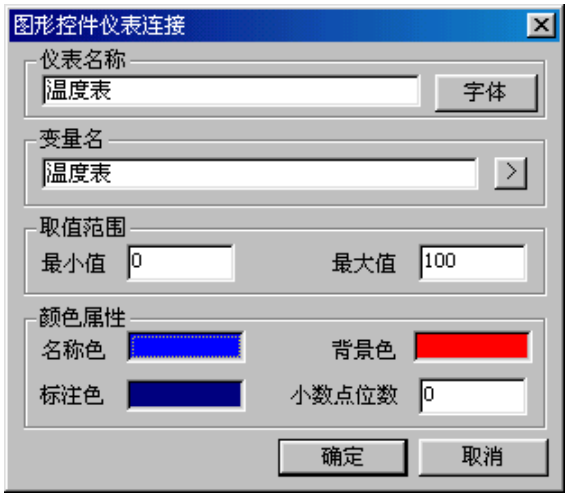


图15-31

对话框参数说明如下：

仪表名称： 显示在温度仪表上的名称。

变 量 名： 温度仪表和变量字典连接的变量名，整数或实数类型。按“>”按钮可以选择变量字典中已经定义的整数或实数变量。

最 小 值： 温度仪表显示的最小值。

最 大 值： 温度仪表显示的最大值。

名称颜色： 温度仪表上显示的名称颜色属性。

标注颜色： 温度仪表的标注显示的颜色属性。

背景颜色： 温度仪表的背景显示的颜色属性。

小数点位数： 温度仪表的标注显示的小数点位数。

15.12 滑动杆类型

本图形控件类型包含八种图形控件单元，单元图形及其属性连接和动画连接

如下图15-32、 图15-33、图15-34所示：

图15-32

图15-33

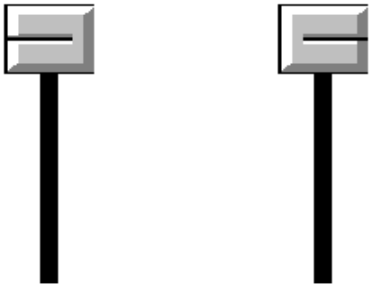


图15-34

滑动块和滑动杆用于模拟变量的输入。系统运行时，滑动块为触敏元件，当其获得焦点时即可输入数据。

滑动块、滑动杆的连接参考动画连接中的滑动杆输入连接。

15.13 历史曲线类型

本图形控件类型包含两种图形控件单元，单元图形及其属性连接和动画连接如下：

一、 左右指示器

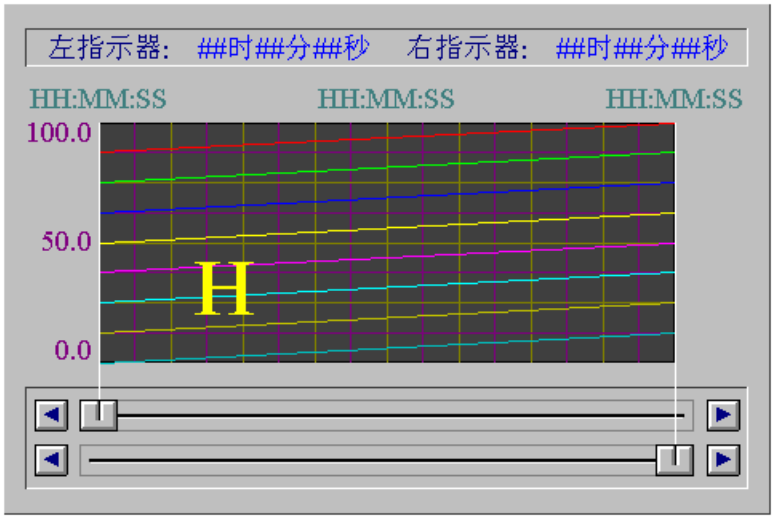


图15-35

图15-35为历史趋势曲线的左右指示器输入图形控件单元。

左右指示器输入控制方式有两种：一种方式是用鼠标在滑动杆上滑动输入，另一种方式是由指示器两端的按钮控制输入。

左右指示器的值对应历史曲线变量中的.ScooterPosLeft和.ScooterPosRight域值，其值变化范围在0.0和1.0之间，其中0.0对应历史趋势曲线的最左边，1.0对应历史趋势曲线的最右边。

指示器两端的按钮每次按下时的增/减量为0.02。

左右指示器的连接参考动画连接中的历史趋势曲线的连接。

示例：

历史趋势曲线动画连接时，输入的历史曲线变量为：Hist1，则左指示器的值为：

Hist1.ScooterLeftPos，右指示器的值为：Hist1.ScooterRightPos。

在历史趋势曲线建立动画连接后，系统自动为左右指示器两端的移动按钮建立命令语言连接。

左指示器向左移动：

```
IF hist1.ScooterPosLeft-0.02>=0.0 THEN
   hist1.ScooterPosLeft=hist1.ScooterPosLeft-0.002;
ELSE
   hist1.ScooterPosLeft=0.0;
ENDIF;
```

左指示器向右移动：

```
IF hist1.ScooterPosLeft+0.02<=1.0 THEN
   hist1.ScooterPosLeft=hist1.ScooterPosLeft+0.002;
ELSE
   hist1.ScooterPosLeft=1.0;
ENDIF;
```

右指示器向左移动：

```
IF hist1.ScooterPosRight-0.02>=0 THEN
   hist1.ScooterPosRight=hist1. ScooterPosRight-0.002;
```

```
    ELSE
        hist1.ScooterPosRight=0.0;
    ENDIF;
右指示器向右移动：
    IF hist1.ScooterPosRight+0.02<=1.0 THEN
        hist1.ScooterPosRight=hist1. ScooterPosRight+0.002;
    ELSE
        hist1.ScooterPosRight=1.0;
    ENDIF;
```

注：用户可以采用类似的方法建立自己的左右指示器输入控制按钮。

二、时间轴设定

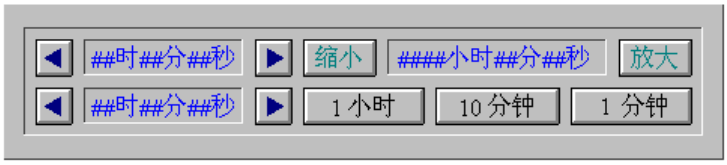


图15-36

图15-36为历史趋势曲线的时间轴控制输入图形控件单元。

图中左部分为历史趋势曲线的起始时间（左上）输入控制及历史趋势曲线的终止时间（左下）输入控制；右下部分为历史趋势曲线的起始时间及终止时间增/减值的卷动量。

图中右上部分为历史趋势曲线的放大和缩小控制部分，其功能参阅历史趋势曲线的缩放函数功能。

历史趋势曲线的时间轴控制连接对话框如图15-37 ：

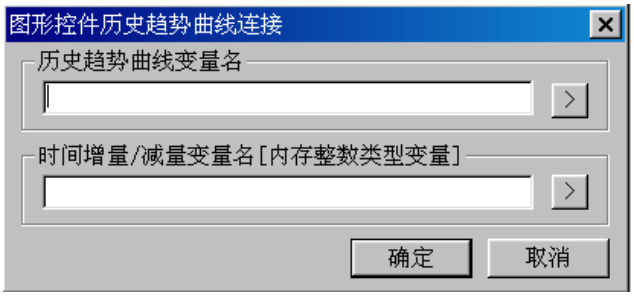


图15-37

对话框参数说明如下：

历史趋势曲线变量名： 时间轴连接的历史趋势曲线对应于变量字典中历史曲线变量名，历史曲线类型。按“>”按钮可以选择变量字典中已经定义的历史曲线变量。

时间增量/减量变量名： 曲线时间轴输入卷动量对应于变量字典中整数变量名，整数类型。按“>”按钮可以选择变量字典中已经定义的整数变量。

示例：

历史趋势曲线动画连接时，输入的历史曲线变量为：Hist1，则历史趋势曲线的起始时间为：Hist1.ChartStart，曲线显示的时间长度为：Hist1.ChartLength。

时间增量/减量变量名（即输入卷动量）为：ScrollTime。

历史趋势曲线起始时间、终止时间左右按钮输入控制，其功能是改变趋势曲线左端或右端的时间值，命令语言连接程序如下：

时间轴左端向左卷动：（起始时间左按钮）

```
hist1.ChartStart=hist1.ChartStart-ScrollTime;
hist1.ChartLength=hist1.ChartLength+ScrollTime;
```

将时间轴左端左移 ScrollTime，时间长度增加 ScrollTime，时间轴右端保持不变。

时间轴左端向右卷动：（起始时间右按钮）

```
hist1.ChartStart=hist1.ChartStart+ScrollTime;
hist1.ChartLength=hist1.ChartLength-ScrollTime;
```

将时间轴左端右移 ScrollTime，时间长度减少 ScrollTime，时间轴右端保持不变。

时间轴右端向左卷动：（终止时间左按钮）

```
hist1.ChartLength=hist1.ChartLength-ScrollTime;
```

将时间轴左端保持不变，时间长度减少 ScrollTime，时间轴右端左移 ScrollTime。

时间轴右端向右卷动：（终止时间右按钮）

```
hist1.ChartLength=hist1.ChartLength+ScrollTime;
```

将时间轴左端保持不变，时间长度增加 ScrollTime，时间轴右端右移 ScrollTime。

历史趋势曲线起始缩放按钮输入控制，其功能是为了便于查看数据的变化，而对曲线进行放大和缩小。命令语言连接程序如下：

曲线放大按钮（曲线在时间轴方向曲线放大一倍，即曲线的时间轴宽度缩小一倍）：

```
HTZoomIn(hist1, "Center" );
```

曲线缩小按钮（曲线在时间轴方向曲线缩小一倍，即曲线的时间轴宽度放大一倍）：

```
HTZoomOut(hist1, "Center" );
```

15.14　阀门类型

本图形控件类型包含十二种图形控件单元，单元图形及其属性连接和动画连接如下：

一、　二通阀门（手动控制）

图15-38

图15-38为二通阀门手动控制。系统运行时，本阀门为触敏元件，阀门弹起时和按下时控制阀门的开通和关闭。本阀门可以通过图形控件阀门连接对话框来连接到一离散变量。

阀门的连接对话框如图15-39 ：

图15-39

对话框参数说明如下：

变 量 名：阀门和变量字典连接的变量名，离散类型。按“>”按钮可以选择

变量字典中已经定义的离散变量。

填充颜色：阀门开通和关闭的填充颜色。

二、 二通阀门（电动控制）

图15-40为二通阀门电动控制。

图15-40

系统运行时，本阀门由表达式控制阀门的开通和关闭。

阀门的连接对话框如图15-41 ：

图15-41

对话框参数说明如下：

表 达 式：阀门开通和关闭的控制表达式。表达式为1时，阀门开通。

填充颜色：阀门开通和关闭的填充颜色。

三、　三通阀门（手动控制）

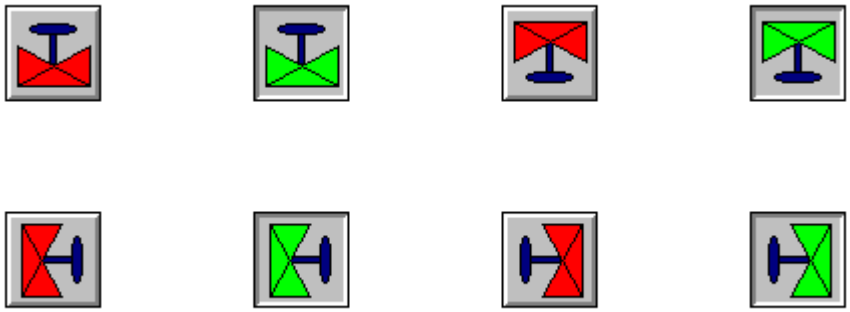


图15-42

图15-42为三通阀门手动控制。系统运行时，本阀门为触敏元件，阀门弹起时和按下时控制阀门的开通和关闭。

本阀门可以通过图形控件阀门连接对话框来连接到一离散变量。阀门的连接对话框如图15-43所示：

图15-43

对话框参数说明如下：

变 量 名：阀门和变量字典连接的变量名，离散类型。按“>”按钮可以选择变量字典中已经定义的离散变量。

填充颜色：阀门开通和关闭的填充颜色。

四、 三通阀门（电动控制）

图15-44

图15-44为三通阀门电动控制。

系统运行时，本阀门由表达式控制阀门的开通和关闭。

阀门的连接对话框如图15-45所示：

图15-45

对话框参数说明如下：

表 达 式：阀门开通和关闭的控制表达式。表达式为1时，阀门开通。
填充颜色：阀门开通和关闭的填充颜色。

15.15　泵类型

本图形控件类型包含十二种图形控件单元，单元图形及其属性连接和动画连接如下：

一、　手动控制泵

图15-46为手动控制的泵。

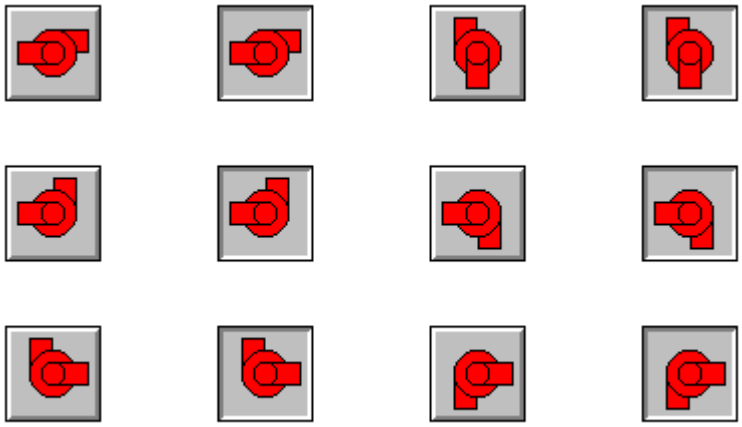


图15-46

系统运行时，本泵为触敏元件，泵弹起时和按下时控制泵的运行和停止。
可以通过图形控件泵连接对话框来连接到一离散变量。

泵的连接对话框如下图15-47所示：

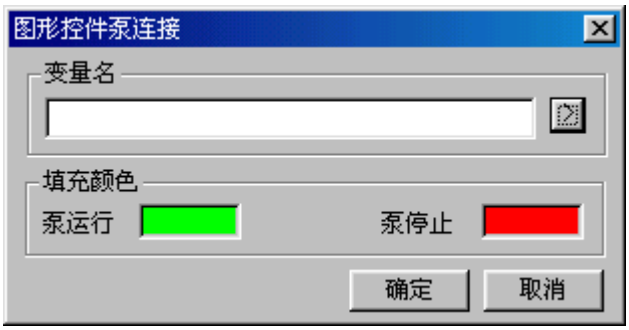


图15-47

对话框参数说明如下：

变 量 名：泵和变量字典连接的变量名，离散类型。按“>”按钮可以选择变量字典中已经定义的离散变量。

填充颜色：泵运行和停止的填充颜色。

二、 自动控制泵

图15-48为自动控制的泵。

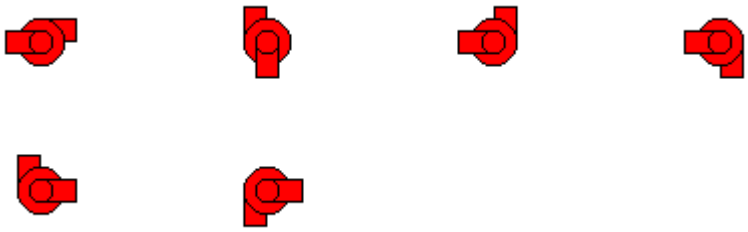


图15-48

系统运行时，本泵由表达式控制泵的运行和停止。

泵的连接对话框如下图15-49所示：

图15-49

对话框参数说明如下：

表 达 式：泵运行和停止的控制表达式。表达式为1时，泵运行。
填充颜色：泵运行和停止的填充颜色。

15.16　管道类型

本图形控件类型包含十四种图形控件单元，其中，有一个水平流动管道、一个垂直流动管道、一个立体管道，前面11种单元图形及其属性连接和动画连接

如下图15-50为自动控制的管道。系统运行时，管道由表达式控制流通。

管道的连接对话框如图15-51所示：

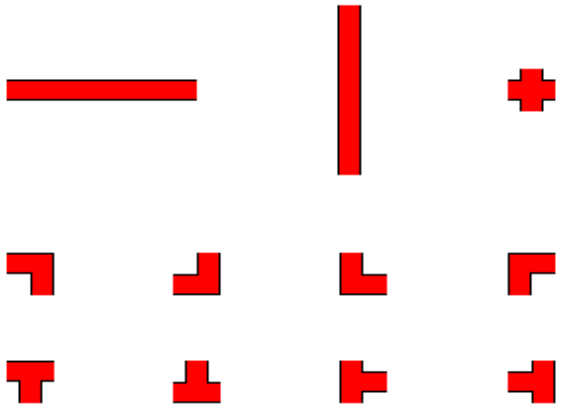


图15-50

图15-51

对话框参数说明如下：

表 达 式：管道流通的控制表达式。表达式值为1时，管道流通。
填充颜色：管道流通和不通的填充颜色。

水平流动管道如图15-52所示：

图15-52

管道的连接对话框如图15-53所示：

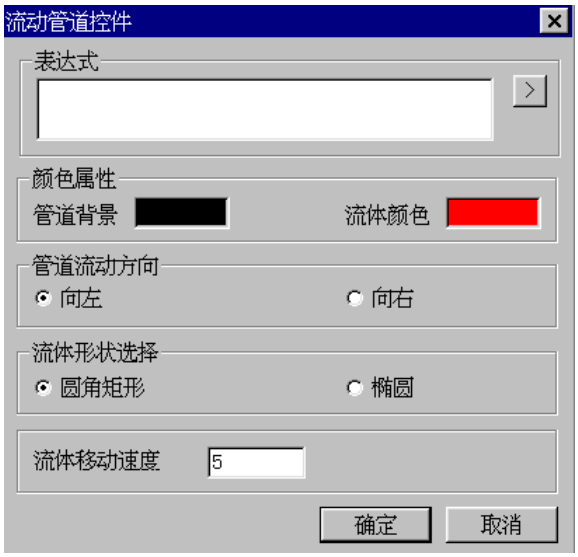


图15-53

表 达 式--结果为离散量（0、1）的表达式，是否流动就根据该表达式的结果颜色属性--管道背景和流体颜色可选择管道流动方向--向左、向右可选择流体形状选择--圆角矩形和椭圆形可选流体移动速度--1～10可选择输入.

垂直流动管道如图15-54所示：

管道的连接对话框如图15-55所示：

表 达 式--结果为离散量（0、1）的表达式，是否流动就根据该表达式的结果颜色属性--管道背景和流体颜色可选择管道流动方向--向上、向下可选择流体形状选择--圆角矩形和椭圆形可选流体移动速度--1～10可选择输入.

图15-54

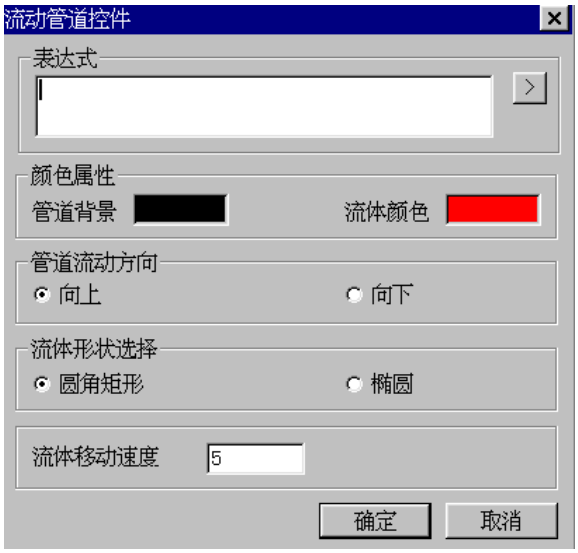

流动管道控件
表达式
颜色属性
管道背景
流体颜色
管道流动方向
向上
向下
流体形状选择
圆角矩形
椭圆
流体移动速度
5
确定
取消

图15-55

立体管道如图15-56所示：

图15-56

立体管道的连接对话框如图15-57所示：

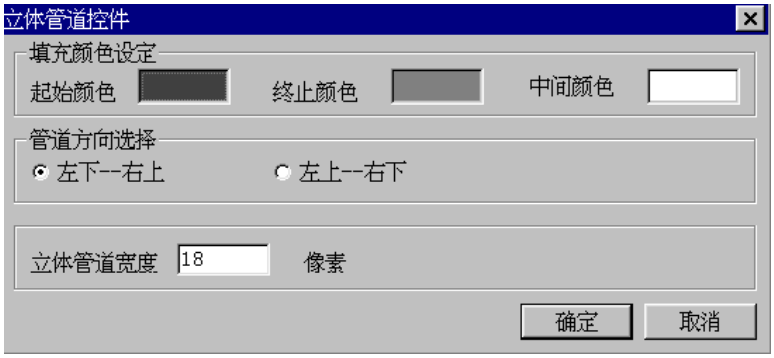


图15-57

填充颜色设定--起始颜色、终止颜色、中间颜色等三种颜色都可选择

管道方向选择：左下—右上、左上—右下两种方向可选择
立体管道宽度：可自由输入，数字越大宽度越宽

15.17 温控曲线

温控曲线控件概述

在温控曲线中，纵轴代表温度值，横轴对应时间的变化，同时将每一个温度采样点显示在曲线中，另外还提供一个游标，当用户把游标放在某一个温度的采样点上时，该采样点的注释值就可以显示出来。

在画面上放置温控曲线

操作方法如下：

在开发系统中，选择“图形控件\趋势曲线控件\温控曲线”，在画面上选择好您要放温控曲线的位置，左键单击即可。

在画面上放置的温控曲线控件如下图15-58所示：

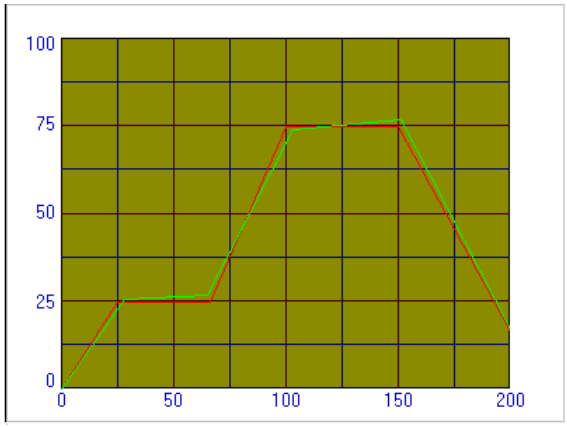


图15-58

温控曲线属性设置

画面上用鼠标左键双击温控曲线控件，则弹出温控曲线属性对话框（如图15-59所示）。在对话框中设置温控曲线的属性，具体属性设置简介如下：

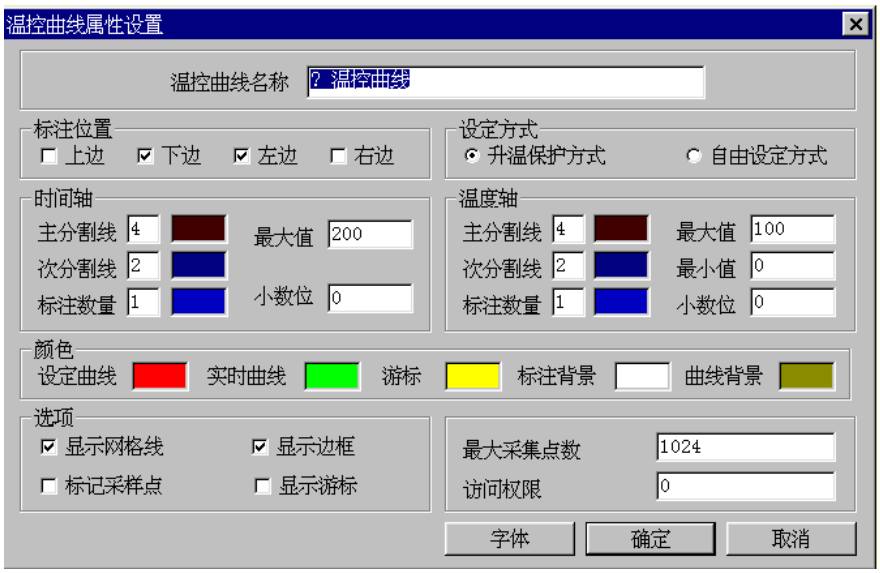


图15-59

在温控曲线中，纵轴代表温度变量，而纵轴坐标则代表温度的大小和高低。

温控曲线名称：温控曲线的命名，可以为中文名或英文名，缺省为“? 温控曲线”
标 注 位 置：上边、下边、左边、右边等四个位置可选。
设 定 方 式：升温保护方式和自由设定方式可选。

时间轴
主分割线：用于设定时间轴主分割线的数量及颜色
次分割线：用于设定时间轴次分割线的数量及颜色
标注数量：用于设定时间轴标注数量的数量及颜色
最 大 值：用于设定时间轴标注的最大值
小 数 位：用于设定时间轴标注的小数位数

温度轴
主分割线：用于设定温度轴主分割线的数量及颜色
次分割线：用于设定温度轴次分割线的数量及颜色
标注数量：用于设定温度轴标注数量的数量及颜色
最 大 值：用于设定温度轴标注的最大值

最 小 值：用于设定温度轴标注的最小值
小 数 位：用于设定温度轴标注的小数位数

颜色（32位真彩色可供颜色设定）
设定曲线颜色：用于设置“设定曲线”的显示颜色。
实时曲线颜色：用于设置“实时曲线”的显示颜色。
游 标 颜 色：用于设置游标颜色
标注背景颜色：用于设置标注背景颜色
曲线背景颜色：用于设置温控曲线背景颜色

选项（以下四个选都是可选的）
显示网格线：是否显示网格线
显 示 边框：是否显示边框
标记采样点：是否标记采样点
显 示 游标：是否显示游标

最大采集点：允许的最大采集点，范围为1～2147483647
访 问 权 限：范围为0～9999

“温控曲线控件”设定

温控曲线的设定方式有两种

1、自由设定方式
2、升温保护方式

程序应用中，使用升温保护方式来设置温控曲线方便适用，每一段温控曲线都有升温曲线和保温曲线组成，在自由设定方式中，用户可以任意设置温控曲线，两种方式的区别如上图所示。其中，控件函数pvAddSetPoint()用于温控曲线的自由设定方式；pvInitSetCurve()，pvLoadData()，pvModifySetCurve()用于温控曲线的升温保护方式，而pvAddRealPoint()则可用于自由设定和升温保护方式设定这两种方式。

温控曲线函数

温控曲线函数的具体使用请参见函数一章

15.18　棒图控件

在画面上放置棒图控件

在画面上放置棒图控件的方法类似放置历史趋势曲线，具体操作步骤如下：

单击图形控件按钮，在图形控件选择对话框中选中趋势曲线控件，在该对话框中右边双击棒图控件，即可在画面中加棒图控件。在画面上放置的棒图控件如下图15-60所示：

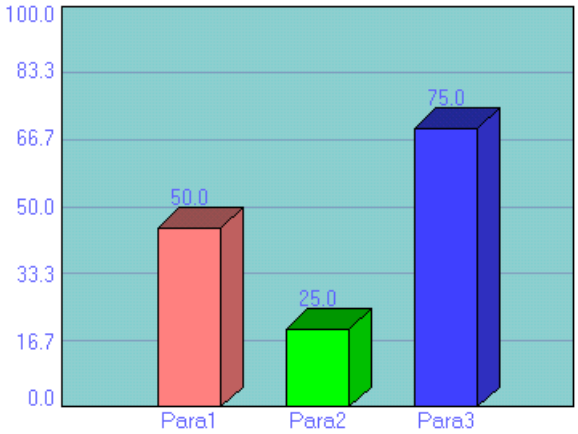


图15-60

在此控件中最多用柱状图（平面或立体）方式反映8个模拟变量的变化趋势。

棒图控件属性设置

在画面中，用鼠标双击棒图控件，弹出棒图控件属性对话框图15-61。在对话框中设置棒图控件的属性。

此对话框中各项的含义分别介绍如下：

棒图控件名称：棒图控件的命名，可以为中文名或英文名，缺省为“？棒图控件”
标注位置：上边、下边、左边、右边等四个位置可选

棒图类型：平面棒图、立体棒图可选
颜色
标注背景：用于设置棒图控件的标注显示颜色。
棒图背景：用于设置棒图控件的背景显示颜色。
网格颜色：用于设置棒图控件的网格颜色。
文本颜色：用于设置棒图控件的文本颜色。

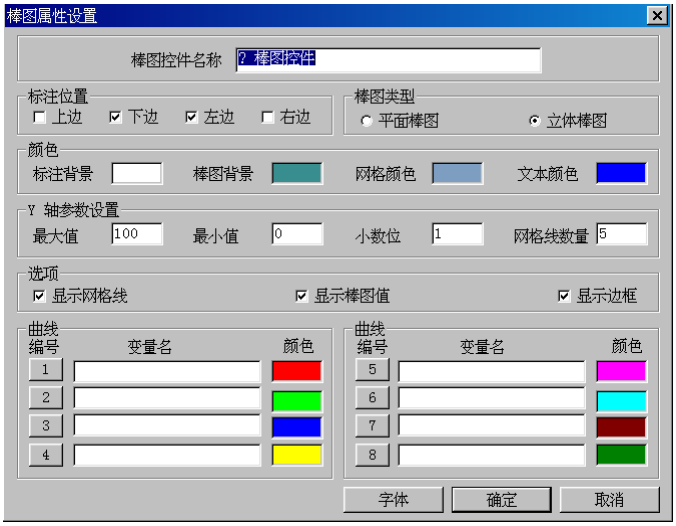


图15-61

Y轴参数设置
最大值：Y轴的最大值设置
最小值：Y轴的最小值设置
小数位：Y轴的小数点位数
网格线数量：Y轴的网格线数量
选项
显示网络线、显示棒图值、显示边框等三种属性都是可选择的。
棒图
棒图1～8，共8个棒图名、颜色,都是可选择设置的。

“棒图控件”函数

参见函数一章

15.19 XY曲线

XY曲线概述

在画面上放置XY曲线

在画面上放置XY曲线的方法类似放置历史趋势曲线，具体操作步骤如下：
单击图形控件按钮，在图形控件选择对话框中选中趋势曲线控件，在该对话框中右边双击XY曲线，即可在画面中加入XY曲线。在画面上放置的XY曲线控件如下图15-62所示：

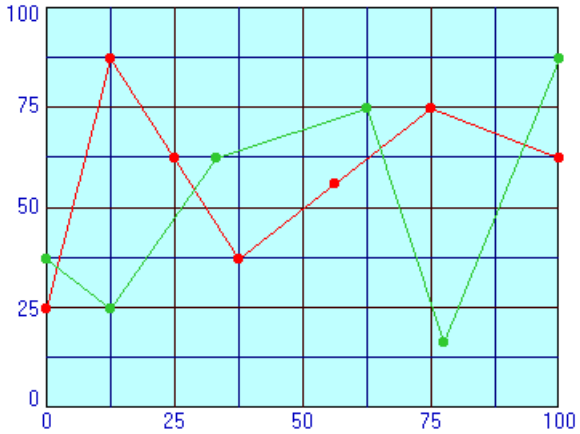


图15-62

在此控件中X轴和Y轴变量由用户任意设定，因此，XY曲线能用曲线方式反应任意两个变量之间的函数关系。

XY曲线属性设置

在画面中，用鼠标双击XY曲线，弹出图15-63XY曲线属性对话框。在对话框中设置XY曲线的属性。

图15-63

此对话框中各项的含义分别介绍如下：
XY 曲线名称：曲线的命名，可以为中文名或英文名，缺省为“? XY曲线”

标注位置：上边、下边、左边、右边等四个位置可选颜色
标注背景：此按钮用于设置XY曲线的标注显示颜色。
曲线背景：此按钮用于设置XY曲线的背景显示颜色。

X轴
主分割线：主分割线的数量及颜色
次分割线：次分割线的数量及颜色
标注数量：标注数量的数量及颜色
最 大 值：标注的最大值
最 小 值：标注的最小值
小 数 位：标注的小数位数

Y轴 （所有参数类似 X轴）

曲线颜色
曲线1～8，共8根曲线的颜色都是可选的。

选项（以下四个选都是可选的）

显示网格线：是否显示网格线
显示边框：是否显示边框
标记采样点：是否标记采样点
连接采样点：是否连接采样点
最大采集点：允许的最大采集点，范围为1～2147483647
访问权限：范围为0～9999

“XY曲线控件”函数

参见函数一章

15.20 带图控件

带图控件概述

在画面上放置带图控件

在画面上放置带图控件的方法类似放置历史趋势曲线，具体操作步骤如下：
单击图形控件按钮，在图形控件选择对话框中选中趋势曲线控件，在该对话框中右边双击带图控件，即可在画面中加入带图控件。在画面上放置的带图控件如下图15-64所示：

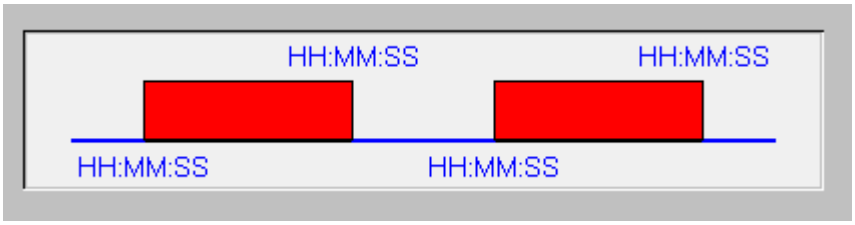


图15-64

带图控件主要用于表示泵的开关状态及通讯状态。

带图控件属性设置

在画面中，用鼠标双击带图控件，弹出图15-65带图控件属性对话框。在对话框中设置带图控件的属性。

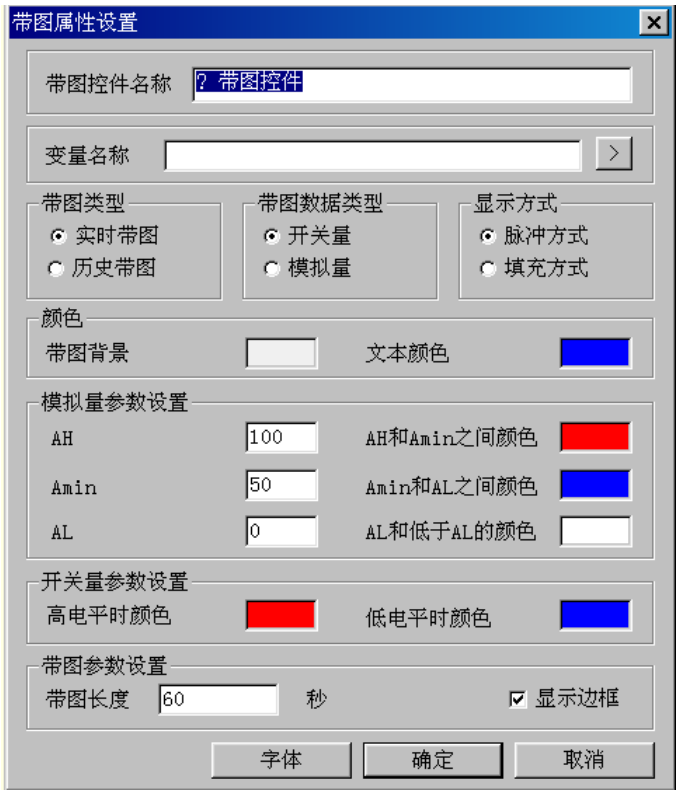


图15-65

此对话框中各项的含义分别介绍如下：

带图控件名称：带图控件的命名，可以为中文名或英文名，缺省为“? 带图控件”，无需预先定义变量。

变量名称：带图控件所反应的世纪星中的变量名，可以是模拟量和开关量。

带图类型：可以是历史带图和实时带图中的一种。

带图数据类型：可以是模拟量和开关量中的一种。

显示方式：可以是脉冲方式和填充方式中的一种。

颜色：文本颜色和填充颜色都可自己设置。

模拟量参数设置：设置AH、Amin、AL等三个参数的值及其颜色。

开关量参数设置：设置高电平及低电平的颜色。
带图参数设置：可设置带图长度及选择是否显示边框。

带图控件函数（具体使用参见相关函数）

15.21　窗口控制控件

文本显示框

在画面上放置文本显示框

在画面上放置文本显示框的方法类似放其他控件，具体操作步骤如下：
单击图形控件按钮，在图形控件选择对话框中选中窗口控制控件，在该对话框中右边双击文本显示框，即可在画面中加入文本显示框。
在画面上放置的文本显示框如下图15-66所示：

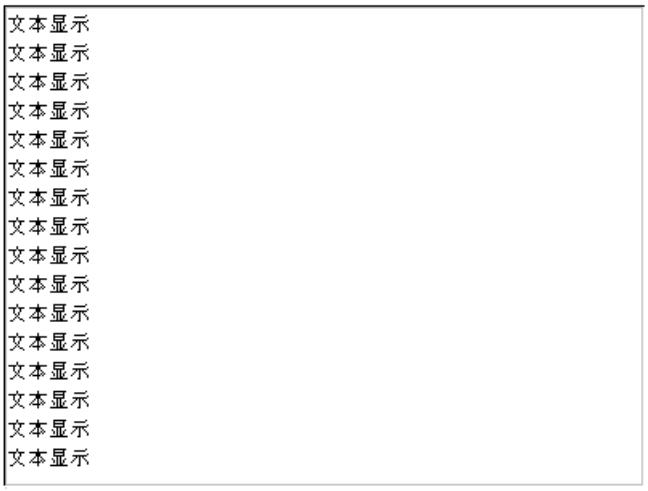


图15-66

文本显示框属性设置

在画面中，用鼠标双击文本显示框，弹出图15-67文本显示框属性对话框。在对话框中设置文本显示框的属性。

图15-67

此对话框中各项的含义分别介绍如下：

文本显示框名称：文本显示框的命名，可以为中文名或英文名，缺省为“？文本显示框”

文本颜色：此按钮用于设置文本显示框的背景显示颜色。

背景颜色：此按钮用于设置文本显示框的背景显示颜色。

显示边框：是否显示边框

访问权限：范围为0～9999

字　　体：可设置字符和字体、字体样式、大小等到属性

文本显示框控件函数

参见函数一章

附录 A　数据库数据转换及实用程序 DBExcel

世纪星支持动态数据交换（DDE），能够和其他支持动态数据交换的应用程序方便的交换数据。如 Microsoft Excel 通过 DDE 从世纪星的实时数据库中读取数据，世纪星也可以通过 DDE 从 Microsoft Excel 中读取数据，从而完成数据采集、报表打印等功能。

DBExcel 是本公司开发的用于世纪星和 Microsoft Excel 交换数据的实用程序，DBExcel 可以将世纪星的数据库转换成 .CSV 格式的文本文件，此格式的文件可以在 Microsoft Excel 或 Notepad 中打开。DBExcel 还可以将用户修改后的 .CSV 文件转换成世纪星格式的数据库文件。

一、　设置应用程序目录

选择“设置/设置应用程序目录”弹出如图 A-1 所示对话框：

图 A-1

在设置应用程序目录对话框中可以输入应用程序目录，或选择“浏览”按钮，进行目录选择，如图 A-2 所示：

二、　世纪星的数据库格式文件转换成 .CSV 格式的文本文件

选择“选项/变量数据库 → Excel (.CSV)”即可将世纪星的数据库格式文件转换成 .CSV 格式的文本文件。

图 A-2

三、 .CSV 格式的文本文件转换成世纪星的数据库文件

选择“选项/Excel (.CSV) → 变量数据库”即可将.CSV 格式的文本文件转换成世纪星的数据库格式文件。

转换前系统将会弹出如图 A-3 所示提示对话框：

选择“是”将进行数据转换，并且覆盖原有数据库文件；选择“否”将不进行数据转换。

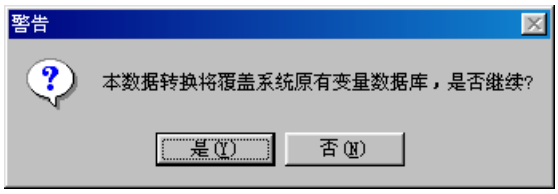


图 A-3

四、 DBExcel 数据转换目的

主要目的在两个方面：一是数据库格式文件转换成 Microsoft Excel 的 .CSV 格式文件，这样就可以在 Microsoft Excel 中对变量数据库的变量进行报表打印；二是当数据库中的变量数据在 Microsoft Excel 中修改后可以转换回数据库，从而达到在开发应用程序的同时，设计变量数据库。

附录 B　历史数据记录及实用程序 HDExcel

一、世纪星进行历史数据记录的方式

世纪星根据记录变化灵敏度对要求记录的变量自动进行历史数据记录。世纪星每次刷新实时数据库时，便对要求历史数据记录的变量进行变化灵敏度判断，决定是否对此变量数据进行记录。因此，在“数据字典”中选择“是否记录”的同时，还需定义“记录变化灵敏度”，该灵敏度的缺省值为 1，用户应避免将其设为 0，除非该变量的刷新频率很低。世纪星根据“记录变化灵敏度”对变量进行判断，只有当变量数据变化的幅度（相对上一次历史数据记录点）大于等于“记录变化灵敏度”值时，才对此变量作一次记录。

“记录变化灵敏度”是一个很重要的参数，此值若规定太大，会造成记录准确性下降，此值若太小，则会造成记录点过多，用户应根据实际情况，选择此值大小。

二、历史数据在硬盘上的存储方式

世纪星按照用户在历史数据记录配置中设置的参数进行历史数据存储，所有变量的历史数据记录按照时间顺序存储到硬盘上的带有时间信息的文件中。如：96081500.LGH 文件。

三、查询历史数据记录

世纪星提供动态数据交换文件：HistData.DLL 及其调用函数，用于查询历史记录数据。用户可以在自己的程序中使用这些函数，查询或处理历史数据。

函数的调用方法如下：

1. UINT ConvertTime(int Year,int Month,int Day,int Hour,int Minute,int Second);

函数功能：将时间转换成自 1970 年 1 月 1 日 0 时 0 分 0 秒起至年、月、日、时、分、秒表示的时间的秒数。

入口参数：Year：　年，此值必须介于 1970 和 2019 之间
Month：　月，此值必须介于 1 和 12 之间
Day：　日，此值必须介于 1 和 31 之间
Hour：　小时，此值必须介于 0 和 23 之间
Minute：　分钟，此值必须介于 0 和 59 之间
Second：　秒，此值必须介于 0 和 59 之间

返 回 值：返回自 1970 年 1 月 1 日（00:00:00）即 UCT 起到该时刻所经过的秒数

2. BOOL GetHistValueAtTime(char* Tagname,UINT ChartTime,float* fValue);

函数功能：得到变量某一时刻的值。

入口参数：Tagname：　变量在变量数据库中的名称。
ChartTime：　自 1970 年 1 月 1 日（00:00:00）即 UCT 起到该时刻所经过的秒数
fValue：　保存变量的实数值，为统一格式，HistData.DLL 将变量的离散、整数、实数值统一返回成浮点值。

返 回 值：TRUE – 成功，FALSE – 失败。

3. BOOL GetHistoryData(char* Tagname,UINT ChartStartTime,UINT ChartEndTime,
UINT IntervalTime,float* DataArray,int SizeOfArray);

函数功能：获得变量在某一段时间范围内的值。

入口参数：Tagname：　变量在变量数据库中的名称。
ChartStartTime：起始时刻，该时间为自 1970 年 1 月 1 日（00:00:00）即 UCT 起到该时刻所经过的秒数。

ChartEndTime：结束时刻，该时间为自 1970 年 1 月 1 日（00:00:00）即 UCT 起到该时刻所经过的秒数。

IntervalTime：采集时间间隔，单位：秒。

DataArray：指向一个double 型数组的首地址，该数组必须存在。

SizeOfArray： DataArray 数组的大小。

返 回 值：TRUE -- 成功找到数据，FALSE -- 失败。

四、历史数据转换实用程序 HDExcel

HDExcel 是本公司利用 HistData.DLL 及其调用函数开发的实用程序，用于将变量在某一段时间内的历史数据转换成 .CSV 格式的文本文件，本文件可以由 Microsoft Excel 或平共处 Notepad 读取，主要用于历史数据的数据分析和报表打印。

HDExcel 转换过程如下：

1. 设置工作路径

选择“文件/设置工作路径”弹出如图 B-1 所示对话框：

图 B-1

设置工作路径对话框中，历史数据文件目录为世纪星运行时保存历史数据的

目录；CSV 文件目录是保存历史数据转换成 .CSV 格式文件的目录。选择浏览可以选择硬盘中已经存在的目录，如图 B-2 所示：

图 B-2

2. 转换参数设置

选择“操作/设置转换参数”弹出如图 B-3 所示对话框：

历史数据转换参数说明：

文 件 名：保存历史数据转换成 .CSV 格式文件的文件名。
变 量 名：要转换的历史数据变量在系统变量数据库中变量名称。
起始时间：要转换的历史数据变量的起始时间。
终止时间：要转换的历史数据变量的终止时间。
时间间隔：读取历史数据的间隔。
转换精度：转换后实数数据保留小数点后数据的位数。

图 B-3

3. 开始转换

在设置了工作路径和历史数据转换参数后，“操作/开始转换”菜单有效，选择此项菜单，即可完成历史数据的转换工作。

附录 C DDEDriver ToolKit 驱动系统

一、DDE 概念

DDE（动态数据交换）是 Microsoft 公司设计的一种基于 Windows 消息驱动机制的动态数据交换协议。应用程序采用三层标识系统，即服务程序名、话题名和项目名，在两个或多个应用程序之间动态传送数据。DDE 数据交换时，提供数据的应用程序为服务应用程序，接收数据的应用程序为客户应用程序。

世纪星就是采用 DDE 协议同下位机进行数据交换的。例如：将世纪星作为服务器，EXCEL 作为客户程序，EXCEL 要从世纪星取得数据，则可在 EXCEL 的某单元格中规定“＝CSVIEWER|TAGNAME!ITEM”，其中 CSVIEWER 是世纪星的服务应用程序名，TAGNAME 是话题名，ITEM 是变量的项目名，当世纪星中该变量发生变化时，EXCEL 中相应的单元格的数据就会改变。

二、DDEDriver ToolKit 驱动模块开发步骤

DDEDriver 是本公司开发的通用 DDE 协议软件开发包，用于世纪星系统和下位机之间数据传输。

DDEDriver 以动态链接库的形式向用户提供八个函数用来完成系统初始化、数据库创建、数据传输等功能。用户使用这些函数开发 DDE 服务程序（即 I/O SERVER）。

1. 系统初始化

I/O SERVER 利用 DDE_INITIAL() 函数完成 DDEDriver 的初始化工作。初始化内容包括：声明本 I/O SERVER 的服务程序名和话题名，登记 I/O SERVER

的主窗口句柄。其中，服务程序名和话题名用来唯一标识本 I/O SERVER，当同时运行多个 I/O SERVER 时，两个 I/O SERVER 的服务程序名或话题名不能重名，要具有唯一性。

2. 创建数据库

DDEDriver 为每一个 I/O SERVER 维护一个数据库。所有需要传送的数据变量都要登记入库，每个数据变量的项目名、数据类型、读写属性等必须和世纪星数据库中定义的数据变量相同。

DDEDriver 开发包的 CreateDataBaseItem(LPCSTR lpItemName,int nType，int nAttrib,DDETagValue InitValue) 函数用来完成数据库的入库工作。其中的参数说明如下：

lpItemName – IO 变量的项目名，用来唯一标识 I/O SERVER 的数据变量，是一个以 NULL 结束的字符串。

nType – IO 变量类型，共有四种类型：离散类型(DISC_TYPE)、整数类型(INT_TYPE)、实数类型(REAL_TYPE)、信息类型(MESSAGE_TYPE)。

nAttrib – IO 变量的读写属性，0：读写属性，1：只写属性；DDEDriver 用来标明当世纪星监控组态软件系统中的该变量变化时， DDEDriver 是否将数据发送到 I/O SERVER。“读写”属性表明传送，“只写”表示不传送。

DDETagValue 是 DDEDriver 的变量数据联合，数据格式如下：

```
typedef union
{
        BOOL bValue;            // 离散类型变量
        int    nValue;          // 整数类型变量
        float fValue;           // 实数类型变量
        char szValue[132];      // 信息类型变量
```

} DDETagnameValue;

DDETagnameValue 数据联合适用于所有 DDE 变量，例如，采用如下方式定义整数变量：

DDETagnameValue Value;

Value.nValue 表示变量具有 DDETagnameValue 结构的整数值。

InitValue – IO 变量的初始值，是 DDEDriver 创建数据库时，变量的初始值。

3. 主窗口消息传输

系统运行时，DDEDriver 通过 I/O SERVER 的主窗口接收 WINDOWS 消息，并将其过滤处理，以达到 DDE 初始化、接收和发送数据、终止 DDE 协议等目的。

I/O SERVER 通过重载函数 LRESULT CMainFrame::WindowProc(UINT message, WPARAM wParam, LPARAM lParam) 接收 WINDOWS 消息，由 DDE_MESSAGE(UINT message, WPARAM wParam, LPARAM lParam) 函数直接传送到 DDEDriver 模块，DDEDriver 模块对这些消息进行过滤处理，成功处理 DDE 消息，返回 TRUE，否则返回 FALSE。

4. 向世纪星发送变化的数据

I/O SERVER 成功完成初始化和创建数据库后，就可以和世纪星进行数据交换了，I/O SERVER 不断将变化了的数据传送给世纪星。

5. 接收世纪星传送来的数据

I/O SERVER 和世纪星之间的数据交换是双向的，若规定数据点的属性为 READWRITE“读写”，则当用户在利用命令语言连接或值输入连接修改该变量的值时，世纪星会将此数据下发给 I/O SERVER。

I/O SERVER 通过调用 LPCTSTR GetReceivedItemName() 函数，判断世纪星是否有下发数据。当函数的返回值为 NULL 时，表明没有下发数据。否则，函数返回下发数据的项目名。

I/O SERVER 通过调用 DDETagValue GetReceivedData(LPCSTR lpItemName) 获得世纪星下发数据。

6. I/O SERVER 系统资源释放

系统运行结束时，DDEDriver 自动释放数据库所占用的系统资源。

在运行过程中，I/O SERVER 想释放某变量所占用的系统资源，可以调用 BOOL FreeDataBaseItem(LPCSTR lpItemName) 函数，此函数删除数据库中一个已存在的数据。并释放其占用的系统资源。

7. 终止 I/O SERVER 和世纪星的数据交换

系统运行结束时，DDEDriver 自动调用 BOOL DDE_TERMINATE() 函数终止 I/O SERVER 和世纪星的数据交换。

在运行过程中，I/O SERVER 想终止 I/O SERVER 和世纪星的数据交换。可以随时调用 BOOL DDE_TERMINATE() 函数终止 I/O SERVER 和世纪星系统的数据交换。

三、DDEDriver ToolKit 驱动模块函数库

1. BOOL DDE_INITIATE(HWND hWnd,LPCSTR lpAppName,LPCSTR lpTopicName)

功能：本函数用以完成 DDEDriver 数据交换的一些初始化工作，在调用其他的 DDE 函数之前必须首先调用此函数，若此函数调用失败，则应退出应用程序。

参数：hWnd：I/O Server 主窗口句柄。

LpAppName： I/O Server 服务程序名，是以 NULL 结束的字符串缓冲区，最长 32 个字符。

LpTopicName：I/O Server 服务程序的话题名，是以 NULL 结束的字符串缓冲区，用以存放本程序，最长 32 个字符。

返回：TRUE ：函数调用成功

FALSE：函数调用失败

示例：

I/O Server 的服务程序名为“IOServer”，话题名为“Tagname”。

```
BOOL CTESTApp::InitInstance()
{
        ……
        HWND hWnd=AfxGetMainWnd()->GetSafeHwnd();// 主窗口句柄

        if(DDE_INITIATE(hWnd," IOServer"," Tagname")==FALSE)
                return FALSE;                    // 初始化失败。
        else                                     // 初始化成功。
        {
                                                 // 在此加入其他初始化程序
        }
        ……
        return TRUE
}
```

2. BOOL DDE_TERMINATE()

功能：I/O Server 服务程序退出时，调用此函数，结束 DDE 会话，并释放 DDEDriver 数据交换所占用的系统资源。

参数：无

返回：TRUE -- 调用成功

FALSE -- 调用失败

示例：

```
int CTESTApp::ExitInstance()
{
……
if(DDE_TERMINATE()==FALSE)                    // 调用失败
    return    0;
……                                  // 调用成功
return CWinApp::ExitInstance();
}
```

3. BOOL CreateDataBaseItem(LPCSTR lpItemName,int nType, int nAttrib,DDETagValue InitValue)

功能：I/O Server 服务程序调用此函数建立数据库。

参数：lpItemName – IO 变量的项目名

nType – IO 变量类型

nAttrib – IO 变量的读写属性，0：读写属性，1：只写属性

InitValue – IO 变量的初始值，数据格式为 DDETagValue 数据联合。

返回：TRUE -- 调用成功

FALSE -- 调用失败

注释：以上各参数的项目名、数据类型、读写属性等必须和世纪星监控组态软件系统数据库中定义的数据变量相同。

示例：

```
int CMainFrame::OnCreate(LPCREATESTRUCT lpCreateStruct)
{
……
DDETagValue InitValue;              // 定义数据结构

InitValue.bValue=TRUE;
CreateDataBaseItem("离散变量",DISC_TYPE,READWRITE,InitValue);

InitValue.nValue=10;
CreateDataBaseItem("整数变量",INT_TYPE,READWRITE,InitValue);

InitValue.fValue=10.0f;
CreateDataBaseItem("实数变量",REAL_TYPE,READWRITE,InitValue);

Strcpy(InitValue.szValue, "世纪星监控组态软件系统");
CreateDataBaseItem("信息变量",MESSAGE_TYPE,READWRITE,InitValue);
……
return 0;
}
```

4. BOOL FreeDataBaseItem(LPCSTR lpItemName)

功能：删除数据库中一个已存在的数据。

参数：lpItemName – IO 变量的项目名

返回：TRUE -- 调用成功

FALSE -- 调用失败

示例：

```
void CMainFrame::OnClose()
{
……
FreeDataBaseItem("离散变量");
```

```
FreeDataBaseItem("整数变量");
FreeDataBaseItem("实数变量");
FreeDataBaseItem("信息变量");
……
CFrameWnd::OnClose();
    }
```

5. BOOL DDE_MESSAGE(UINT message, WPARAM wParam, LPARAM lParam)

功能：将主窗口收到接受的消息传送到 DDEDriver，便于处理 DDE 消息

参数：message，wParam, lParam – 主窗口收到的消息参数

返回：TRUE -- DDEDriver 已经正确处理 DDE 消息

FALSE – 不是 DDE 消息，或 DDEDriver 处理 DDE 消息失败

示例：

```
LRESULT CMainFrame::WindowProc(UINT message, WPARAM wParam,
LPARAM lParam)
{
……
if(DDE_MESSAGE(message, wParam, lParam))// 传送消息到 DDEDriver
{                                   // 为 TRUE,成功处理 DDE 消息
    LPCTSTR szItemName=GetReceivedItemName();
                                        // 检测是否有接收变量
if(szItemName)          // 有数据接收
DDETagValue Value=GetReceivedData(szItemName);

return TRUE;
}
……
return CFrameWnd::WindowProc(message, wParam, lParam);
```

}

6. BOOL DDE_SENDDATA(LPCSTR lpItemName,int nType,DDETagValue Value)

功能：I/O Server 服务程序调用此函数，将数据库中的数据发送给世纪星。

参数：lpItemName – IO 变量的项目名

nType – IO 变量类型

Value – IO 变量的当前值，数据格式为 DDETagnameValue 联合类型。

返回：TRUE -- 调用成功

FALSE -- 调用失败

示例：

```
void CMainFrame::OnTimer(UINT nIDEvent)
{
    ……
    DDETagValue Value;

    Value.nValue=10;
    if(DDE_SENDDATA("整数变量",INT_TYPE,Value)==TRUE)
      {
                          // 传送成功
      }
      else
      {
                          // 传送失败
    }
      ……
    CFrameWnd::OnTimer(nIDEvent);
}
```

7. LPCTSTR GetReceivedItemName()

功能：在函数 DDE_MESSAGE() 返回值为 TRUE 时调用，当世纪星数据发生变化时，返回变量的项目名。
参数：无
返回：当世纪星数据发生变化时，返回变量的项目名，否则为 NULL。

示例参见五。

8. DDETagValue GetReceivedData(LPCSTR lpItemName)

功能：I/O Server 调用此函数，当世纪星数据发生变化时，得到世纪星传送来的变量值。
参数：lpItemName – 变量的项目名。
返回：世纪星传送来的变量值。
注释：此函数在函数 GetReceivedItemName() 返回不是 NULL 时调用。

示例参见五。

附录 D　世纪星文件格式说明

当世纪星建立应用程序目录时，就建立一组数据文件供世纪星开发系统(CSMAKER)和世纪星运行系统(CSVIEMER)使用，建立的每个应用程序必须在单独的目录中。除非特别说明，否则，在任何情况下，绝不允许编辑该目录下的这些文件。以下简单介绍世纪星所建工程目录下的基本文件的具体含义：

*.WIN

每一个画面对应一个后缀为 .WIN 的文件，它包含画面中所有图素的信息（包括动画连接信息）,此类文件只能在开发系统中使用，此类文件一定不能删除。

*.WVW

当某画面存盘后，开发系统 CSMAKER 将 *.WIN 编译成 *.WVW，*.WVW 文件为运行系统 CSVIEMER 专用。若以后在 CSMAKER 中不再编辑 *.WIN(*.WIN 文件不被更改)，则 CSVIEMER 每次装入的画面数据都来自 *.WVW，而不是 *.WIN。此类文件在必要时可删除，但在运行该工程前必须打开所有画面，并保存所有画面，系统会自动生成此类文件。

ALARMLOG.DAT

报警记录数据文件，必须时可删除。

CSALARM.CFG

此文件包含报警配置信息，此类文件一定不能删除。

CSMAKER.CFG

此文件包含开发系统配置信息，此类文件一定不能删除。

CSVIEWER.CFG

此文件包含运行系统配置信息，此类文件一定不能删除。

HISTDATA.CFG

此文件包含历史数据配置信息，此类文件一定不能删除。

NETWARE.CFG
此文件包含网络配置信息，此类文件一定不能删除。

ALARMLOG.DAT
此文件包含所有的报警记录信息。

COMPORT.DAT
此文件包含所有的通讯口参数信息。

CSUSERS.DAT
此文件包含用户配置的信息。

FILTER.DAT
此文件包含数据变量过滤信息。

FORMPRN.DAT
此文件包含所有的报表打印信息。

GROUP.DAT
此文件包含报警组名的所有信息。

RECIPE.DAT
此文件包含配方管理的所有信息。

WINDOWS.DAT
此文件包含所有画面的属性信息。

APPLY.SPT
此文件包含应用程序命令语言的信息。

DATACHG.SPT
此文件包含数据变化命令语言的信息。

EVENTS.SPT
此文件包含事件命令语言的信息。

HOTKEY.SPT
此文件包含热键命令语言的信息。

TAGNAME.DB
此文件包含数据变量的所有信息。

DEVICE.DRV
此文件包含 I/O 设备的所有信息。

PID.DAT
此文件包含 PID 控制的信息

附录 E　常见问题解答

一、如何创建新工程

创建新工程

用世纪星开发您的实际工程时，她是以“工程”为单位来管理的，所以，当您第一次运用世纪星来开发系统或作练习时，您应为自己的设计新建一个工程。新建工程主要是指定您的工程的名称和所在的目录。

创建新工程很简单，您用鼠标单击开发系统菜单栏中的“文件”，待菜单弹出后选中“工程项目管理器”并单击，系统就会弹出如图 E-1 所示的对话框：

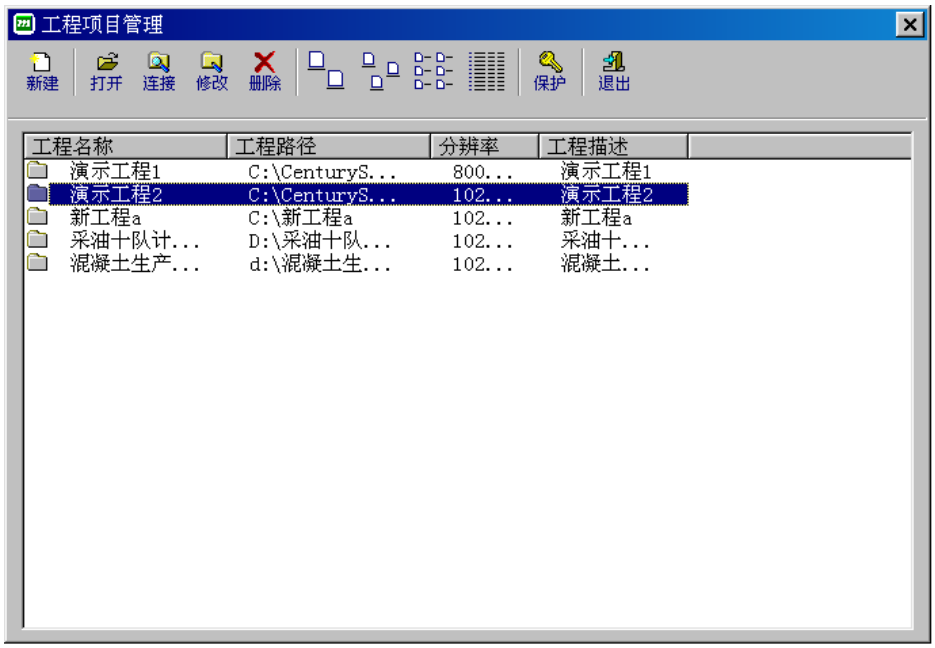


图 E-1

这时，您需要做的事情就是为自己的工程项目新建一个工程项目名。您可以在上面这个对话框中选“新建”，一个新的对话框又出现了，它提示您新的工程

名称为“新工程”；所在目录为“C:\新工程”；工程描述为“新工程”。您如果不满意可以修改它，这里给您一个建议“最好使您的工程的三个描述名称一制”。做完了吗？那么按“确定”就可以了。此时系统会提示您所选的目录不存在，是否创建，您当然选“确定”。就这样，您创建了自己的一个工程。下面的图 E-2 是新建工程时的对话框。

图 E-2

工程项目名称：用户在此输入框中输入代表一个工程项目的名称，这个名称应象征性地代表了工程项目的主要内容。
工程项目路径：编辑输入与工程名称相对应的工程目录。
工程项目描述：具体地说明项目的用途，描述性字符。

您只要按照您的需要输入或选择上述有关项，单击“确定”即可，此时，您已经建立了一个新工程。

二、如何连接已有工程

连接已有工程

当您重装世纪星或对其升级后，若您想再次使用以前的工程的话，就有必要使用“连接已有工程”功能。在“图 E-3”所示的工程项目管理器对话框中，单击“连接”按钮，系统弹出“图 E-4”所示的对话框。

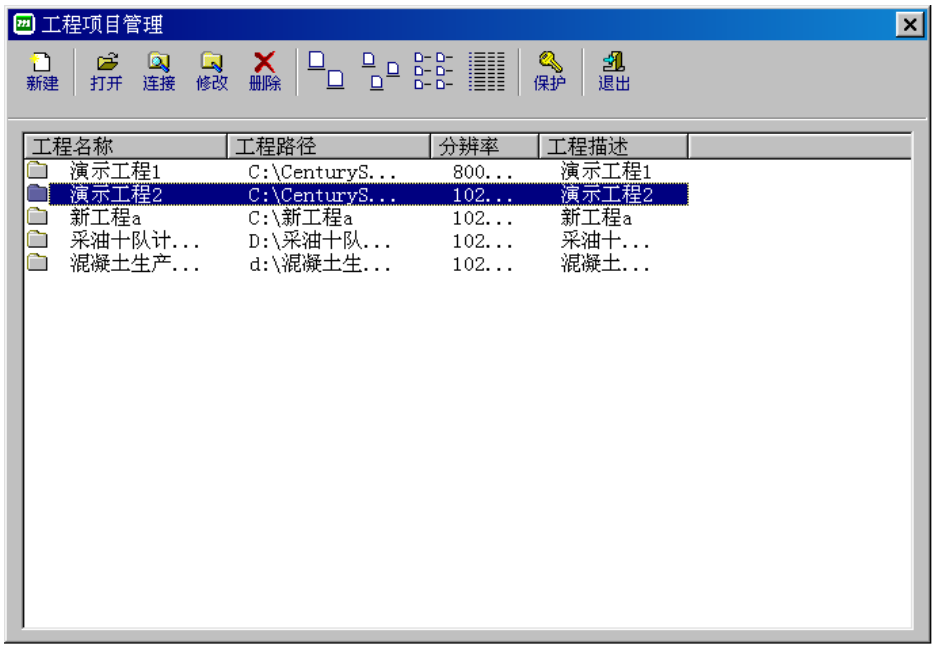


图 E-3

此时，单击“浏览”，选中您想要连接的工程，单击“确定”，此时，您就连接成功。在项目管理器中您能看到刚连接的工程。

图 E-4

三、如何创建新画面

创建新画面

当您第一次进入开发系统并新建了一个工程，然而您却发现您什么也做不了。这是因为，您的工程还是一个真正的空白，甚至连一张白纸都没有。在常用工具区中又一个“白纸按钮”（正确名称叫“新建”），用鼠标点一下(或单击“文件/新画面”)，系统弹出了如图 E-5 对话框。它要求您为这个空白页取一个名字，就像下面这个图示，同时对话框中还有许多属性，您可以尝试着改变它们。这里的改变不会对下面的练习产生影响，除非您将画面的宽度或高度调得太小了。

图 E-5

四、如何使用历史趋势曲线查看历史数据

使用历史趋势曲线查看历史数据

趋势分析是组态软件不可缺少的重要功能，它有实时趋势曲线和历史趋势曲线两种。实时趋势曲线用于实时显示数据的变化情况，曲线会自动卷动，便于观察变量的变化趋势，不需专门定义实时趋势曲线变量；每个实时趋势曲线可反应四个由变量及运算符组成的表达式的变化趋势，而一个画面可定义数量不限的趋势曲线。历史趋势曲线用于查看历史数据，曲线一般不自动卷动，与功能按钮一起工作，利用历史趋势曲线变量的域或使用与历史趋势曲线有关的函数可以完成对历史趋势曲线的控制，如翻页、启动/停止记录、打印曲线等功能；使用前必须先定义历史趋势曲线变量，每个历史趋势曲线可反映 8 个已选择记录属性的变量（而不能是表达式）的历史数据变化趋势。

要想历史趋势曲线能显示历史曲线情况，必须满足以下三个条件：

1、要显示的变量必须是已选择“记录”性的模拟变量；

2、在历史数据记录配置中必须选择“允许历史数据记录”属性；
3、要显示历史趋势曲线的时间段里必须有历史数据。

五、如何拷贝工程

拷贝工程

有下列几种情况需要拷贝工程：

1、因有些工程是在办公室开发的，要想把工程带到工程现场，或想把现场工程拷贝到公司做备份时，必须将工程拷贝到软盘或光盘中；

2、当一个工程在开发过程中或开发完成后，工程商或工程开发人员因有些问题需要与厂商（世纪长秋公司）沟通，厂商需要看到工程，此时需要用 E-Mail 把工程发到世纪长秋公司。

用世纪星开发的工程中，有下列几类文件占用空间较多，一般在拷贝工程时可以删除：

1、扩展名为“.WVW”文件，此类文件为运行系统下的画面文件；
2、文件名为数字开头的，如:2001.....等，这类文件是历史数据文件和历史数据索引文件；
3、Alarmlog.dat 报警记录文件（如有的话）也可删除；

拷贝工程应注意以下几个问题：

1、删除文件前最好做备份；
2、如果历史数据很重要，以数字开头的文件不要删除；
3、当删除以上介绍的两类文件后，用 WinZip 把工程目录压缩成一个文件；
4、工程中没有整幅点位图，压缩好的工程应为几百 K 左右，一张软盘能装完；
5、解压工程后，在世纪星的项目管理器中用“连接”功能连接工程上，“打开所有画面”后“保存所有画面”即可。

六、如何使用新设备驱动

使用新设备驱动

世纪星提供免费开发驱动功能，有些用户所用的采集设备就是以前所没有做的，如一些非标设备的驱动。有关设备的驱动需要两个文件：“Device.idx”和“*.dll”，当你收到世纪长秋公司发来的以上两个文件时，应做以下工作：

1、将“Device.idx”拷贝到你安装世纪星所在目录下，如“C:\CenturyStar”;将“*.dll”文件拷贝到世纪星目录下的“Driver”目录下。

2、如果该设备第一次做，需要重新注册，方法如下：

在安装世纪星所在目录下，运行“Register.exe”文件(该文件图标为一个红色的大写的“R”)，弹出如图 E-6 对话框：

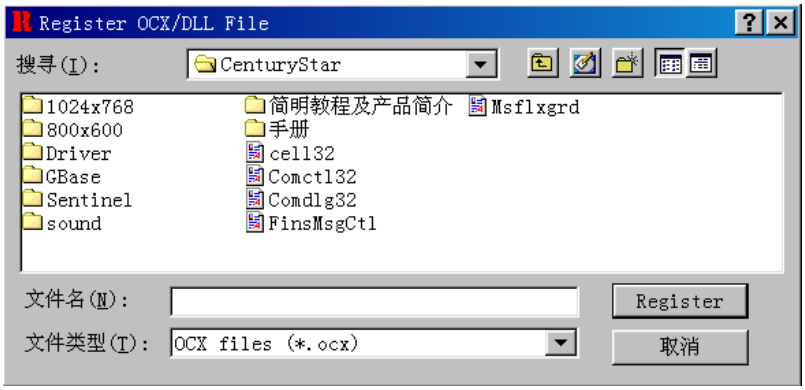


图 E-6

在此对话框中选择文件类型为“DLL Fiels(*.dll)”后，双击“Driver”目录，双击以上所述的 DLL 文件，或选择以上所述的 DLL 文件后按“Register”按钮即可。

结 束 语

本手册全面、系统地介绍了世纪星通用监控组态软件，其内容包括系统命令及其使用、操作步骤、用户编程方法及演示示例等。构架力求开放、透明度高、逻辑性强，目的是能够让用户运用自如、灵活操作。在此，世纪长秋全体同仁感谢您使用世纪星通用监控组态软件，以及您对我们工作的大力支持，并致以崇高的敬意。

“科学技术是第一生产力”。知识经济的典型产品是计算机软件，每次产业革命曾有一些相对落后的国家一跃成为世界的强国，对于我们来说，知识经济时代，是一个绝佳的机会，同时也面临着挑战，对国家是如此，对于企业更是如此。世纪长秋同仁本着“博学、奋斗”的精神，研制出一代又一代的工控组态软件产品，她的卓越表现及强大的性能正是世纪长秋人始终不渝的奋斗结晶。在开发过程中，我们注入了大量的研制资金，并且引入了目前最先进的编程技术和系统构造模型，率先引用COM组件接口技术、网络结构层次、监控模型、网络设备等全新概念，拓展了世纪星的应用范围，使得其功能更加完备。

实践证明我们付出的努力是值得的，经过了一些行业专家的评定，世纪星经的起任何的精挑细审，尤其是一些创新技术简直可以独占鳌头，但是，世纪长秋人并不会因为取得初步的成绩而自满。我们相信在您的一直信赖和关注下，世纪长秋科技开发有限公司经过不断的努力一定会开发出技术更新、功能更强的工业自动化监控组态软件，服务于社会。