

HMI Designer 教学手册

版本号: F202407HT-CN



目 录

1. 前言	15
1.1. 修改记录	15
1.2. 关于亿达	15
2. B 系列人机软件概述	16
2.1. 软件概述	16
2.2. 界面简介	16
2.3. 关于安装	17
3. 基本操作	18
3.1. 文件	18
3.1.1. 新建工程	18
3.1.2. 打开工程	18
3.1.3. 最近打开	18
3.1.4. 保存和另存为	18
3.1.5. 打印	18
3.1.6. 查看备份	18
3.2. 编辑	19
3.3. 画面	19
3.3.1. 新建画面	19
3.3.2. 查看画面	20
3.3.3. 预览	21
3.3.4. 保存图像	21
3.3.5. 导入	21
3.3.6. 窗体设定	22
3.4. 视图	23
3.5. 工具	23
3.5.1. 运行工程	23
3.5.2. 下载到设备	23

3.5.3. 生成安装包	23
3.5.4. 脚本编辑器	23
3.5.5. MacroC 编辑器	23
3.5.6. 组件属性编辑器	23
3.5.7. 查看变数	23
3.5.8. 操作记录设置	24
3.5.9. 清理资源	25
3.5.10. 正则表达式编辑器	26
3.5.11. 共享文件设置	26
3.5.12. 资料备份和更新	28
3.5.13. 组件固有文字翻译	28
3.5.14. 导出翻译	28
3.5.15. 导入翻译	28
3.5.16. 词库设定	28
3.6. 设置	28
3.6.1. 软件语言	28
3.6.2. 语言设定	28
3.6.3. 工程设置	28
3.6.4. 通讯设置	34
3.6.5. 物联网设置	34
3.6.6. 属性	41
3.7. 窗口	41
3.8. 帮助	41
4. 公共属性	42
4.1. 对象名称	42
4.2. 几何尺寸	42
4.3. 字体	43
4.4. 光标	45
4.5. 聚焦策略	45

4.6. 布局方向	46
4.7. 自动填充背景	46
4.8. 使用	46
4.9. 调色板	46
4.10. 样式表	48
4.11. 框架	48
4.12. 对齐	50
4.13. 操作权限	51
4.14. 线类型	52
4.15. 画刷风格	53
4.16. 模态	54
4.17. 窗体类型	54
4.17.1. MainForm	54
4.17.2. SubForm	54
4.17.3. FramelessSubForm	55
4.17.4. KeypadForm	55
4.17.5. EmbeddedForm	55
5. 公共功能	57
5.1. 软键盘	57
5.1.1. 软键盘属性编辑	57
5.1.2. 系统软键盘	60
5.1.3. 自定义软键盘	62
5.2. 多国语言编辑	63
5.3. 变数编辑弹窗	63
5.4. 图片资源	63
6. 组件	68
6.1. 水平滚动条	68
6.2. 垂直滚动条	68
6.3. 水平滑动条	68

6.4. 垂直滑动条	68
6.5. 时间编辑器	68
6.5.1. 组件属性	68
6.5.2. QDateTimeEdit	68
6.5.3. 物联网	69
6.5.4. 组件宏接口	71
6.5.5. 组件宏属性	73
6.5.6. 组件信号	75
6.6. 输入区块	75
6.7. 数值	75
6.7.1. 功能介绍	75
6.7.2. 组件结构	76
6.7.3. 属性设置弹窗	76
6.7.4. 属性编辑器	83
6.7.5. 动作编辑器	85
6.7.6. 脚本宏	85
6.8. 整数、小数输入框	85
6.8.1. 功能介绍	85
6.8.2. 组件结构	86
6.8.3. 属性设置弹窗	86
6.8.4. 属性编辑器	89
6.8.5. 动作编辑器	94
6.8.6. 脚本宏	94
6.9. 圆弧	94
6.9.1. 功能介绍	94
6.9.2. 组件结构	94
6.9.3. 属性设置弹窗	95
6.9.4. 属性编辑器	98
6.9.5. 动作编辑器	98

6.9.6. 脚本宏	98
6.10. 椭圆	98
6.10.1. 功能介绍	98
6.10.2. 组件结构	99
6.10.3. 属性设置弹窗	99
6.10.4. 属性编辑器	101
6.10.5. 动作编辑器	101
6.10.6. 脚本宏	101
6.11. 直线	102
6.11.1. 功能介绍	102
6.11.2. 组件结构	102
6.11.3. 属性设置弹窗	102
6.11.4. 属性编辑器	105
6.11.5. 动作编辑器	105
6.11.6. 脚本宏	105
6.12. 多边形	105
6.12.1. 功能介绍	105
6.12.2. 组件结构	106
6.12.3. 编辑方式	106
6.12.4. 属性设置弹窗	107
6.12.5. 属性编辑器	111
6.12.6. 动作编辑器	111
6.12.7. 脚本宏	111
6.13. 矩形	111
6.13.1. 功能介绍	111
6.13.2. 组件结构	111
6.13.3. 属性设置弹窗	112
6.13.4. 属性编辑器	115
6.13.5. 动作编辑器	115

6.13.6. 脚本宏	115
6.14. 位按钮	115
6.14.1. 功能介绍	115
6.14.2. 组件结构	115
6.14.3. 属性设置弹窗	116
6.14.4. 属性编辑器	130
6.14.5. 动作编辑器	133
6.14.6. 脚本宏	133
6.15. 按钮组	134
6.16. 复选框	134
6.17. 功能按钮	134
6.17.1. 功能介绍	134
6.17.2. 组件结构	134
6.17.3. 属性设置弹窗	134
6.17.4. 属性编辑器	149
6.17.5. 动作编辑器	150
6.17.6. 脚本宏	150
6.18. 图片按钮	150
6.18.1. 功能介绍	150
6.18.2. 组件结构	151
6.18.3. 属性设置弹窗	151
6.18.4. 动作编辑器	166
6.18.5. 属性编辑器	166
6.18.6. 脚本宏	167
6.19. 单选按钮	167
6.20. CadWidget (CAD)	167
6.21. 柱状图	168
6.21.1. 一般属性	168
6.21.2. 外观形状	168

6.21.3. 坐标轴	178
6.21.4. 柱状图	189
6.21.5. 数据采集	193
6.21.6. 物联网	196
6.21.7. 组件宏接口	200
6.21.8. 组件宏属性	208
6.22. 变量示波器	216
6.22.1. 功能介绍	216
6.22.2. 组件结构	217
6.22.3. 属性编辑器	217
6.22.4. 参数设置弹窗	220
6.22.5. 脚本宏	224
6.23. 绘图	224
6.23.1. 功能介绍	224
6.23.2. 组件结构	225
6.23.3. 属性设置弹窗	226
6.23.4. 属性编辑器	235
6.23.5. 脚本宏	238
6.24. 折线图	238
6.25. 示波器	238
6.25.1. 功能介绍	238
6.25.2. 组件结构	238
6.25.3. 属性编辑弹窗	239
6.25.4. 脚本宏	246
6.26. 饼状图	246
6.27. 自定义绘图	246
6.27.1. 功能介绍	246
6.27.2. 组件结构	246
6.27.3. 属性编辑器	247

6.27.4. 脚本宏	249
6.28. 计算器	249
6.29. 权限管理	249
6.29.1. 组件外观	249
6.29.2. 简介	249
6.29.3. 属性编辑弹窗	249
6.29.4. 外观	251
6.29.5. 属性编辑器	251
6.29.6. 系统多国语言	252
6.29.7. 功能详述	252
6.29.8. 权限设置	253
6.29.9. 登录权限	255
6.29.10. 重置密码	255
6.29.11. 关闭窗口体	255
6.30. 万年历	255
6.31. 无线管理	256
6.31.1. 组件外观	256
6.31.2. 简介	256
6.31.3. 属性编辑弹窗	256
6.31.4. 属性编辑器	256
6.31.5. 功能详述	257
6.31.6. 插件函数表	257
6.32. 代码显示器	258
6.32.1. 组件外观	258
6.32.2. 简介	258
6.32.3. 属性编辑弹窗	259
6.32.4. 属性编辑器	260
6.32.5. 属性接口	261
6.32.6. 接口函数	262

6.33. 代码编辑器	262
6.33.1. 简介	263
6.33.2. 属性编辑弹窗	263
6.33.3. 属性编辑器	270
6.33.4. 系统多国语言	271
6.33.5. 功能详述	272
6.33.6. 插件属性	275
6.33.7. 插件函数表	279
6.34. 文本框	292
6.35. 多文本框	292
6.36. 下拉框	292
6.37. 表格集	292
6.37.1. 组件外观	292
6.37.2. 简介	292
6.37.3. 属性编辑弹窗	293
6.37.4. 属性编辑器	298
6.37.5. 系统多国语言	298
6.37.6. 功能详述	298
6.37.7. 属性接口	298
6.37.8. 宏函数	304
6.38. 树状图	311
6.38.1. 组件外观	311
6.38.2. 简介	311
6.38.3. 属性编辑弹窗	311
6.38.4. 属性编辑器	315
6.38.5. 系统多国语言	315
6.38.6. 功能详述	315
6.38.7. 属性	316
6.38.8. 函数接口	316

6.38.9. 信号	320
6.39. 列表框	322
6.40. 日期时钟	322
6.41. 动态图片	322
6.42. 动态文本	322
6.43. 坐标显示器	322
6.44. 历史报警显示器	322
6.45. 当前执行显示器	322
6.46. 操作记录	322
6.46.1. 功能介绍	322
6.46.2. 组件结构	324
6.46.3. 属性编辑器	324
6.46.4. 脚本宏	325
6.47. 进度条	326
6.48. HVisionView (视觉插件)	326
6.49. 标签	326
6.50. 指示灯	326
6.51. 框架	326
6.52. 网格分割器	326
6.53. 选项组	326
6.54. 切换卡	326
6.55. HCam2D	326
6.56. 五面钻	326
6.57. CAM 插件	326
6.58. 凸轮磨边	326
6.59. SpringEditorPlugins (3D 弹簧)	326
6.59.1. 弹簧属性	327
6.59.2. 交互操作	328
6.59.3. HMI 接口	329

6.60. 程序传输	330
6.60.1. 功能介绍	330
6.60.2. 组件结构	331
6.60.3. 属性编辑器	332
6.60.4. 动作编辑器	337
6.60.5. 脚本宏	337
6.61. 文件选择器	337
6.61.1. 功能介绍	337
6.61.2. 组件结构	338
6.61.3. 属性编辑器	339
6.61.4. 动作编辑器	349
6.61.5. 脚本宏	349
6.62. 参数监控表格	350
6.63. 自定义程式跳转	350
6.64. 伺服参数	350
6.65. 键盘容器	350
6.66. 键盘按键	350
6.67. 键盘显示	350
7. 基本宏函数	350
7.1. 宏编辑器	350
7.2. System 函数群	350
7.2.1. getData (获取变量的值)	350
7.2.2. setData (设定变量的值)	351
7.2.3. getBit (获取变量位的值)	352
7.2.4. setBit (设置通道数据位为 0 或者 1)	353
7.2.5. saveData (保存数据)	353
7.2.6. loadData (读取数据)	354
7.2.7. colorFromString (将字符转换成颜色值)	354
7.2.8. loadExeProgramFile (载入执行程序)	355

7.2.9. saveVar (存储指定变数)	355
7.2.10. loadVar (加载指定通道的变数)	356
7.2.11. initVar (恢复变数为出厂值)	357
7.2.12. getProgramDir (获取指定通道的程式存储路径)	357
7.2.13. getExePrgFileMain (获取指定通道当前正在执行主程式文件名)	358
7.2.14. getExePrgFileSub (获取指定通道当前正在执行副程式文件名)	358
7.2.15. getMDItext (获得 MDI 内容)	358
7.2.16. exeMDI (执行 MDI)	359
7.2.17. setMDItext (设置 MDI 内容)	359
7.2.18. sleep (延时函数)	360
7.2.19. execute (运行外部程序)	360
7.2.20. resetCNC (重置控制器指定通道)	360
7.2.21. getDdatas (获得连续多个变数值)	361
7.2.22. setDdatas (设置连续 多个变数值)	362
7.2.23. getBits (获得连续多个 bit 位的值)	363
7.2.24. setBits (设置连续多个 bit 位的值)	364
7.2.25. getUsbDevicePath (获得 U 盘目录)	365
7.2.26. getPrePrgFileSub (获取预解子程序文件名)	365
7.2.27. getPrePrgFileMain (获取预解主程序文件名)	365
7.2.28. setCurScreenSize (设置当前屏幕大小)	366
7.2.29. hPrint (在终端打印信息)	366
7.2.30. keyToString (将按键值转化为按键文本)	366
7.2.31. stringToKey (将按键文本转化为按键值)	367
7.2.32. convertCompressCnc (将文件转化为 CNC 格式)	367
7.2.33. simulateHandle (模拟触摸屏操作)	367
7.2.34. setInputMethod (设置切换中英文输入, 迷你软键盘)	368
7.3. HMI 函数群	369
7.4. Math 函数群	369
7.5. String 函数群	369

7.6. Number 函数群	369
7.7. HInputDialog 函数群	369
7.8. HMessageBox 函数群	369
7.9. Hfile 函数群	369
7.10. HFileEdit 函数群	369
7.11. HFileInfo 函数群	369
7.12. HFileVar 函数群	369
7.13. HAuthorityManagement 函数群	369
7.14. HErrorDialog 函数群	369
7.15. HProperty 函数群	369
7.16. HSharedFile 函数群	369
7.17. HCamera 函数群	369
7.18. HVisionModule 函数群	370
8. 其他	370

1. 前言

1.1. 修改记录

修改时间	备注
20240507	初次修订手册

1.2. 关于亿达

广州亿达科技有限公司于 2018 年建立，旨在打造优秀的“指尖数控”理念，让自动化开发和机床自动化触手可及。作为中国高性能控制器制造商之一，亿达科技在机床和自动化领域具有较高的国际竞争力和影响力。在中国多个地区，均建立了完整，专业的销售和服务渠道。

亿达科技一直专注于数控系统、运动控制器、边缘计算控制器、OpenCNC 开发平台、CAD/CAM 技术、机器视觉技术、机械手控制技术、工业物联网技术的研发和生产。尤其是性能优异的开放式二次开发平台，让机器设备客制化开发变得更简单，并为客户创造专属的产品价值。亿达科技提出建立 6 个核心技术内嵌（运动控制、HMI、PLC、机器视觉、CAD/CAM、物联网）作为我们产品的核心关键，为客户提供完善的自动化解决方案。此外我们已经在车铣加工中心、磨床、弹簧机、刀具机、木工机械、绕线机、旋压机、弯管机、3C 电子等行业，积累了丰富的产品经验和客户基础。

在高速高精领域，亿达科技拥有多种高性能运动控制算法，广泛适用于不同行业需求，尤其是在多轴联动插补、RTCP 五轴联动插补、多轴（40 轴）控制、多通道（16 通道）控制、电子凸轮、卷绕、张力等复合应用场景下，为客户提供了可选择的解决方案。

专注需求、关注变化、卓越创新、共享共担是亿达科技创立以来所秉持的经营理念和企业价值。一直以来，我们坚持初心，砥砺前行，不断研发好用、可靠的自动化产品，并将

OpenCNC 理念做到客户终端，成就客户专属价值。

2. B 系列人机软件概述

2.1. 软件概述

FINGER HMI Designer是亿达新一代B系列自动控制系统人机界面开发工具。

2.2. 界面简介

FINGER HMI Designer软件的主页面如图 2.2-1 所示：

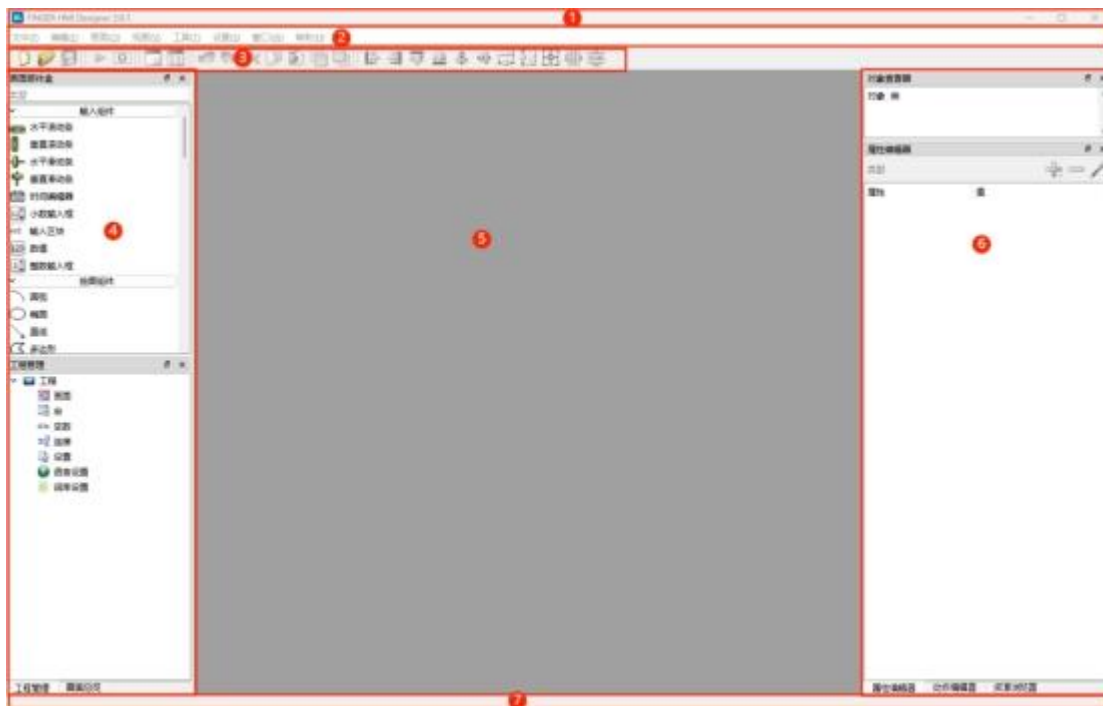


图 2.2-1 软件主页面图示

各区域介绍如下：

1. 标题栏：显示软件版本、文件路径等信息。
2. 菜单栏：为软件的大多数功能提供入口，含有文件、编辑、画面、视图、工具、设置、窗口、帮助等按钮选项。
3. 工具栏：提供各种编辑工具，方便用户使用，提高窗口设计效率。

4. 辅助窗口：用户可在菜单栏->视图中选择需要显示的辅助窗口，可通过拖拽窗口标题栏移动该辅助窗口位置（可在其他辅助窗口上方或下方），包含画面部件盒、工程管理、画面总览。
 - 1) 画面部件盒包含所有插件，可拖拽添加插件到用户编辑区域；
 - 2) 工程管理可查看画面、宏、变数等设置；
 - 3) 画面总览可预览所有工程画面内容。
5. 用户区域：用户编辑的主要区域，以窗口为基本载体，用户可向窗口中添加各类插件。
6. 辅助窗口，包含对象查看器、动作编辑器、资源浏览器
 - 1) 对象查看器可查看当前画面添加的所有插件类和对象名称，点击对象名称可快速定位具体的插件；
 - 2) 动作编辑器可查看对应插件的事件和宏；
 - 3) 资源浏览器可将保存的插件和对应的事件及宏添加到其他工程。
7. 状态栏：显示临时信息。

2.3. 关于安装

3. 基本操作

3.1. 文件



图 3.1-1 文件功能选单

3.1.1. 新建工程

3.1.2. 打开工程

3.1.3. 最近打开

3.1.4. 保存和另存为

3.1.5. 打印

3.1.6. 查看备份

1. 软件会帮助用户自动备份正在编辑的工程文件，若用户需要使用备份文件，可通过【文

件】选单中的【查看备份】选项打开备份文件。

- 单击此项，会弹出备份文件对话框，用户可通过该对话框查看软件自动备份的文件，
点击想要编辑的文件，然后再点击【打开】，软件会弹出一个对话框提示用户是否保存
当前正在编辑文件，用户选择【保存】或【抛弃】都会关闭正在编辑文件并打开选中的
备份文件，打开选中文件后即可对选中文件进行编辑操作。

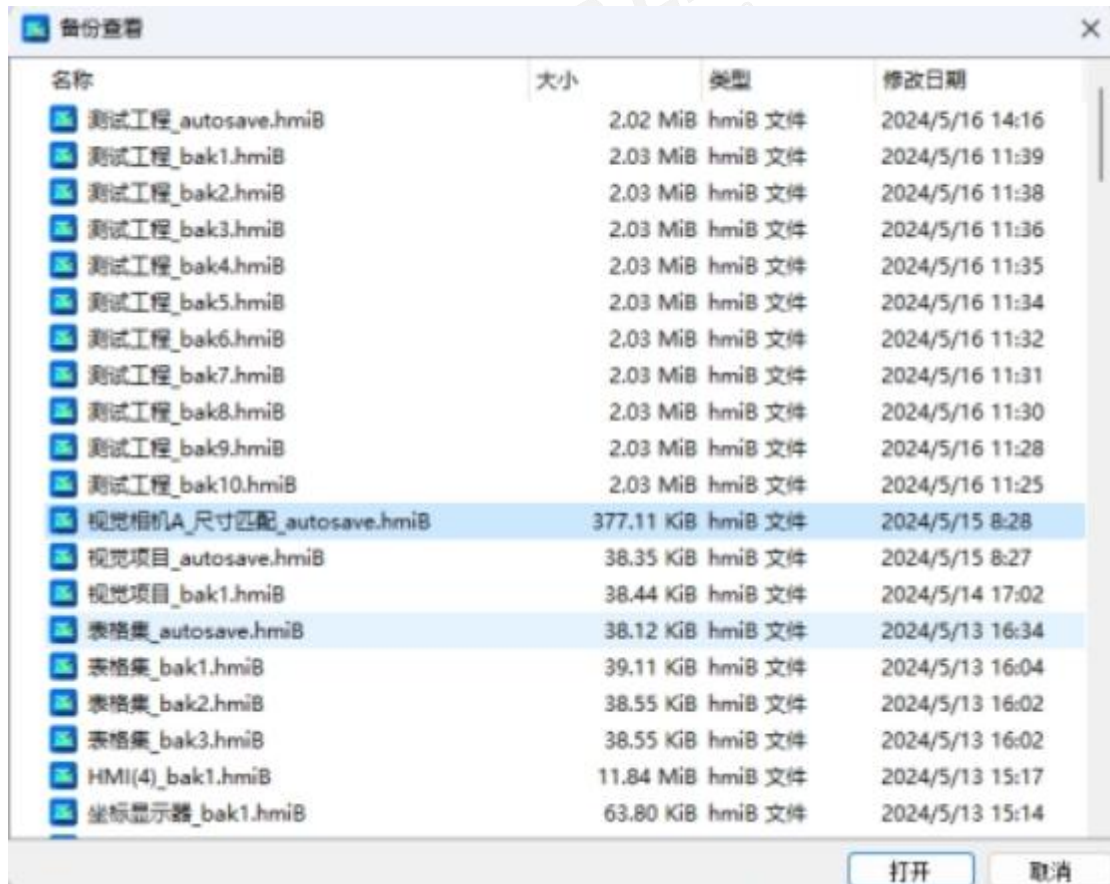


图 3.1.6-1 备份查看

3.2. 编辑

3.3. 画面

3.3.1. 新建画面

用户需要在当前工程中新建一个画面时可使用该功能，在【画面】菜单中点击【新建】选项时，当前工程下会弹出一个新的空画面，默认名称为Form，用户可在属性设置框中设置该画

面的属性。

3.3.2. 查看画面

点击该选项后，软件中会弹出【查看窗体】对话框，对话框中会列举出当前工程内所有的画面名称。

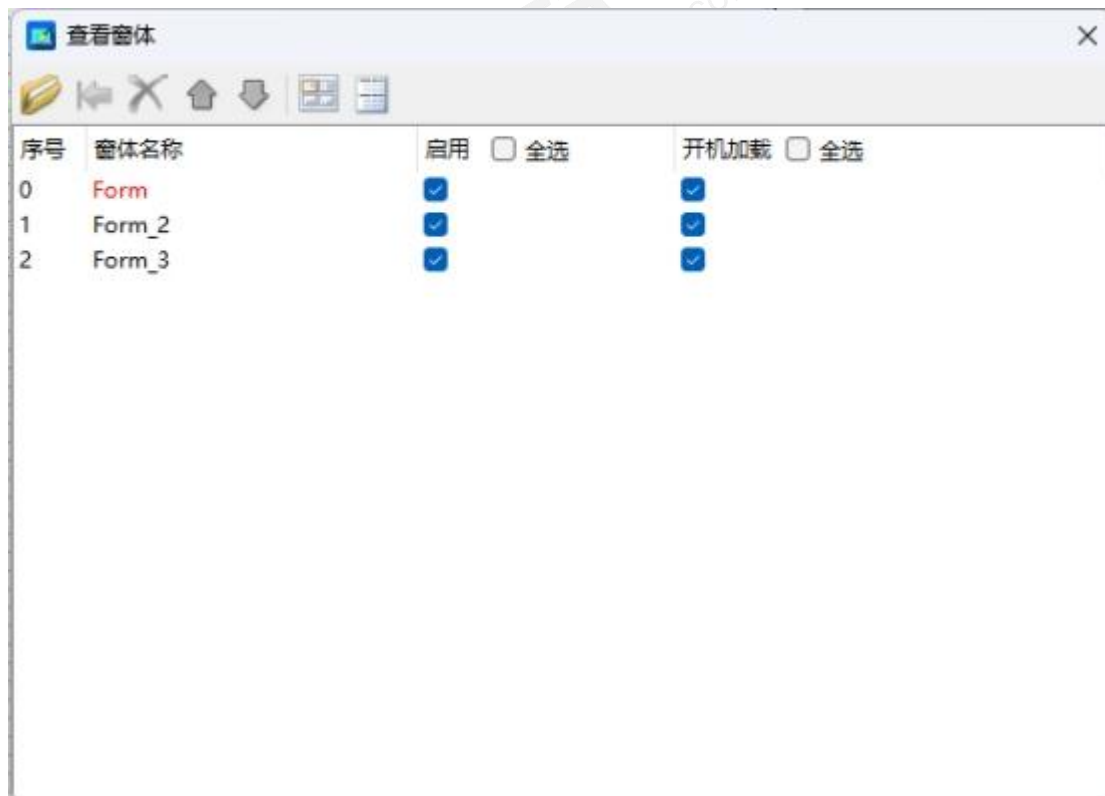


图 3.3.2-1 查看窗体对话框

1. 打开对话框时，用户需要先点击列表中的窗体条目，然后再进行相应设置操作。将鼠标悬停在按钮图形上，指针下方将显示相应按钮中文说明。
2. 列表中含有【开机加载】和【启用】选项，取消勾选【开机加载】，将工程上传至控制器后，初始化工程时不会加载此画面，相较于加载窗体，用户加载工程耗时更少，速度更快、更为流畅，能更快进去起始窗体画面。
3. 用户可将画面设置为【启用】状态，若未勾选【启用】时，该画面上传至控制器后无法在控制器中使用。
4. 点击列表中窗体名称，然后点击  按钮，可将选中的窗体设置为起始窗体，即为开机之后显示的第一个画面，起始窗体在列表中标记为红色，以使用户区分。

3.3.3. 预览

在主窗口中点击选中窗体，然后点击【画面】菜单下的【预览】选项，即可预览该窗体上传控制器后的效果。

3.3.4. 保存图像

选中窗体后点击【保存图像】选项，软件将会生成一张显示内容为选中窗体的png图片。

3.3.5. 导入

此功能可以将另一份工程中的画面导入当前工程中，点击该按钮后，弹出【选择工程】对话框，选择好工程文件后点击【确定】，会弹出导入对象对话框，对话框中有【画面】和【宏】切换卡，点击选中需要导入的画面和宏（按住【Ctrl】可进行多选操作），点击【导入】，即可将所选宏和窗体导入当前工程文件中。

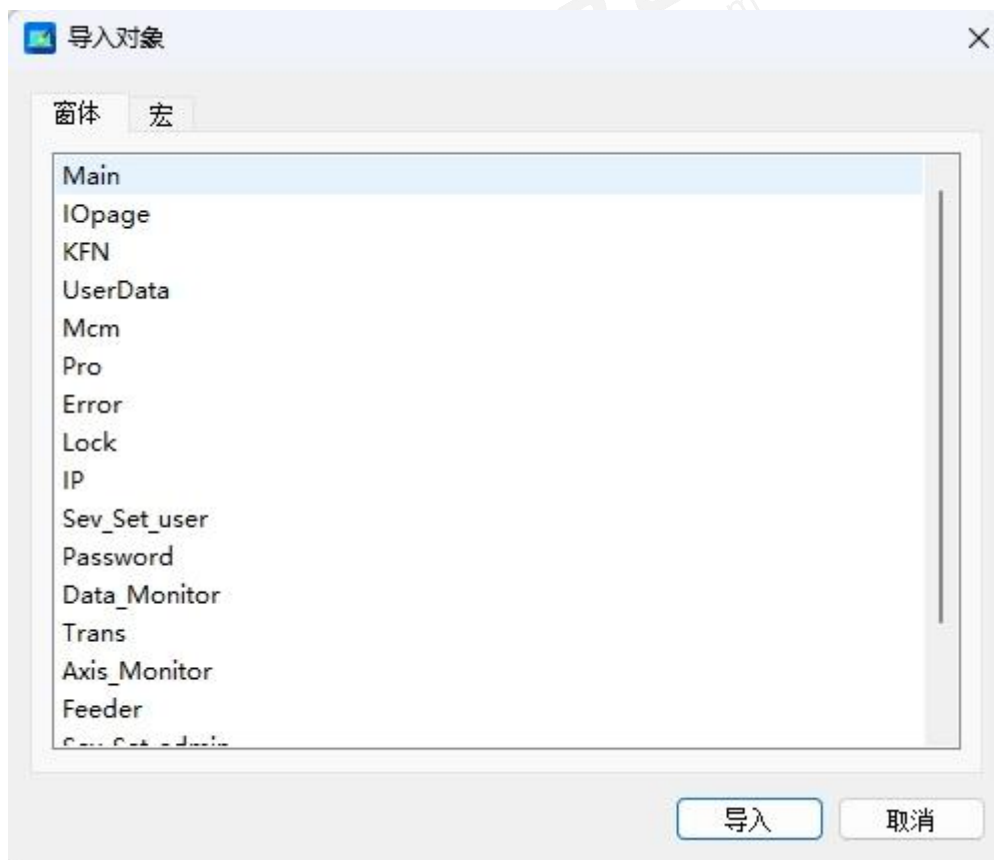


图 3.3.5 选择导入内容

3.3.6. 窗体设定



图 3.3.6 窗体设定

1. 在【窗体设定】对话框中，用户可设定作者名称、栅格、布局……等选项，取消勾选【栅格】时，用户不可对栅格内容进行设置，选中的画面在windows端软件默认显示栅格，选中【栅格】时，用户可设定【是否可见】，若取消勾选是否可见，则当前工程选中画面将不再显示栅格，右侧可对栅格大小进行设置，X代表水平方向，Y代表竖直方向，设置好后，可见画面中像素点间距发生改变。
2. 【窗体设定】中包含布局选项，用户可对【默认布局】或【布局功能】进行设置。

3.4. 视图

3.5. 工具

3.5.1. 运行工程

3.5.2. 下载到设备

3.5.3. 生成安装包

3.5.4. 脚本编辑器

3.5.5. MacroC 编辑器

3.5.6. 组件属性编辑器

3.5.7. 查看变数

用户选择【查看变数】后，软件会弹出一个【查看变数弹窗】，设置好相关项后，下方表格中可显示当前工程文件中已经使用过的系统变数。

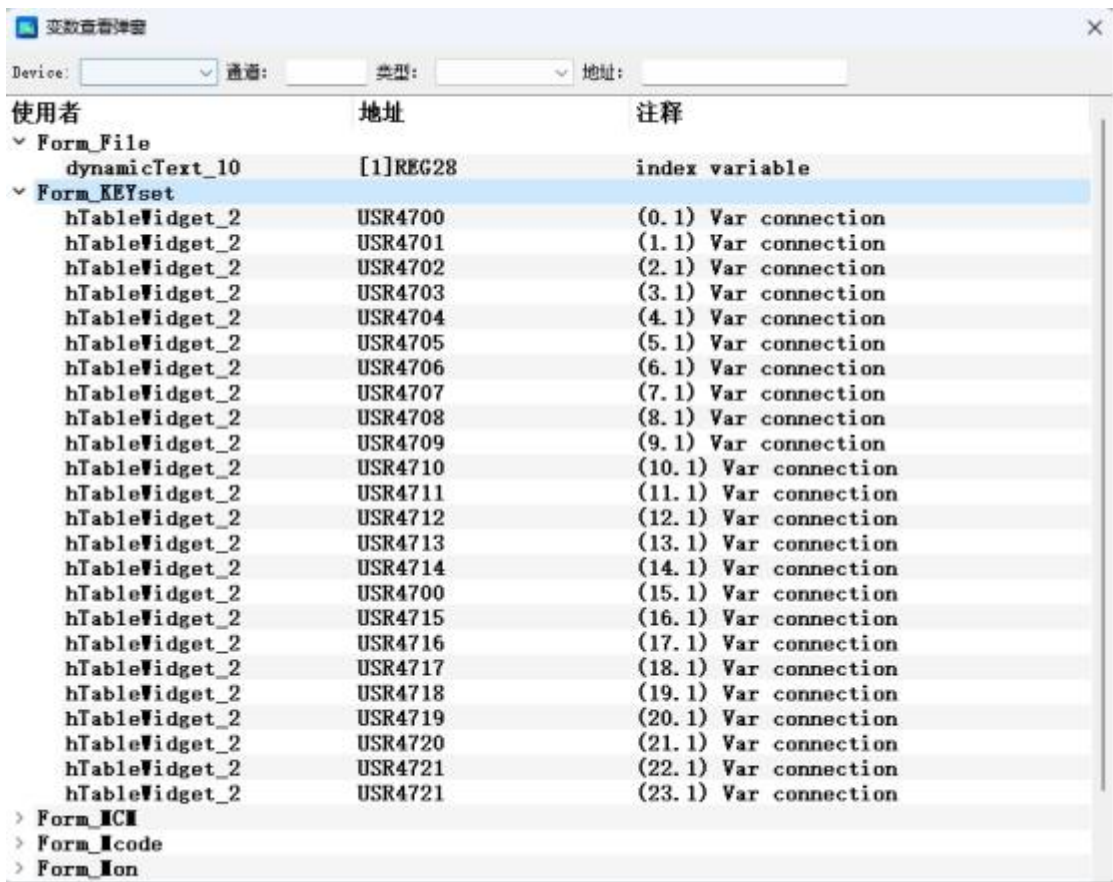


图 3.5.7 变数查看弹窗

也可在标题下方输入通道号（范围 1~16）、类型和地址进行过滤，输入上述三项后，表格中显示内容仅为满足条件的系统变数。

3.5.8. 操作记录设置

设置工程画面中的各插件是否开启操作记录，勾选后该插件的操作触发能在操作记录插件中查看。

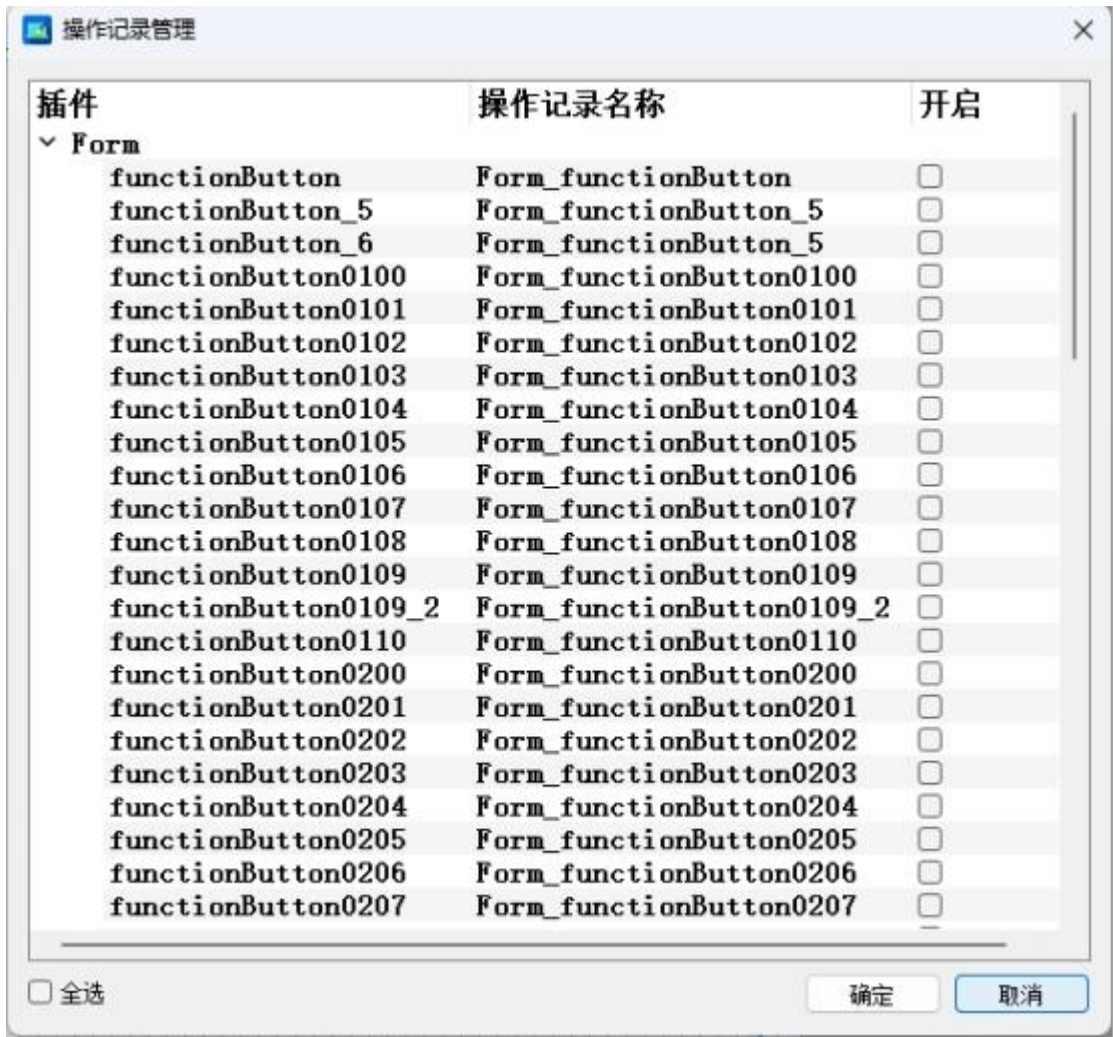


图 3.5.8 操作记录管理对话框

3.5.9. 清理资源

通过图片按钮、动态图片等插件添加的图片资源在按钮移除、图片替换等操作后，未正在使用在资源可以在【工具】—【清理资源】处查看，正在使用的资源不显示，清理未使用的资源可减少工程大小。

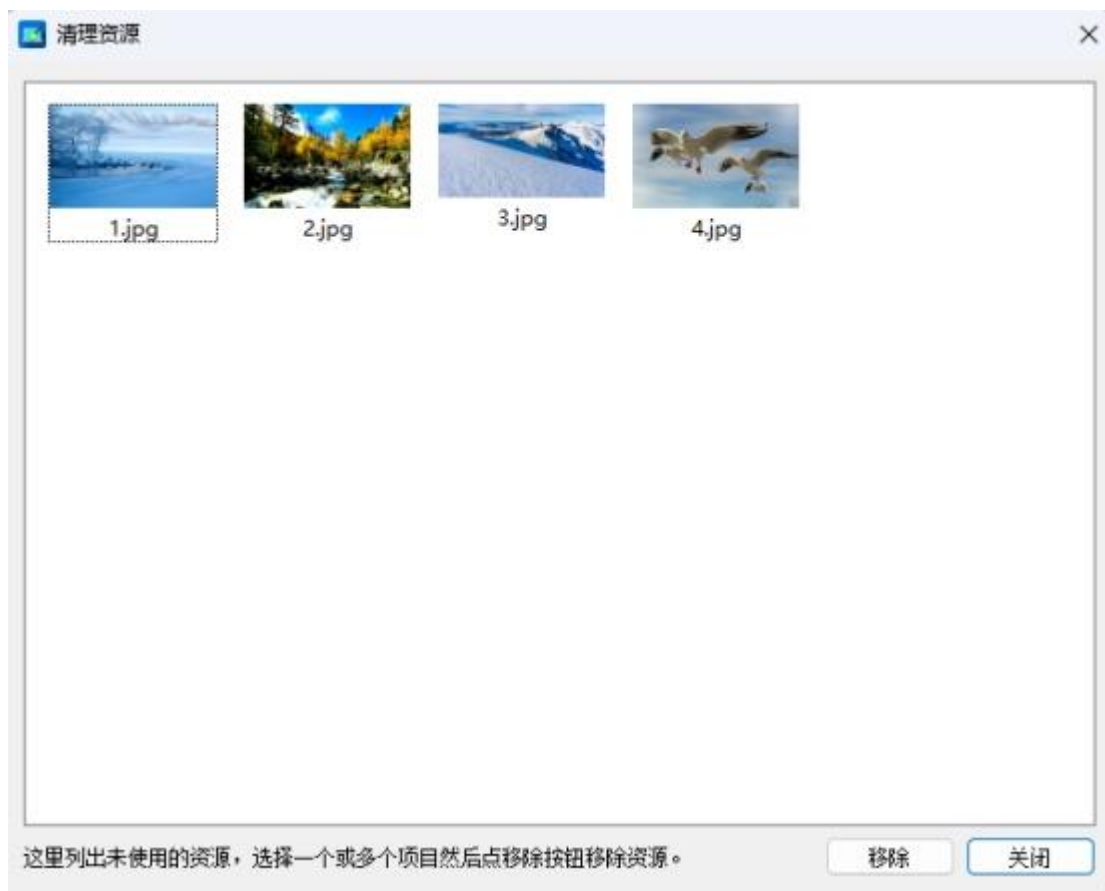
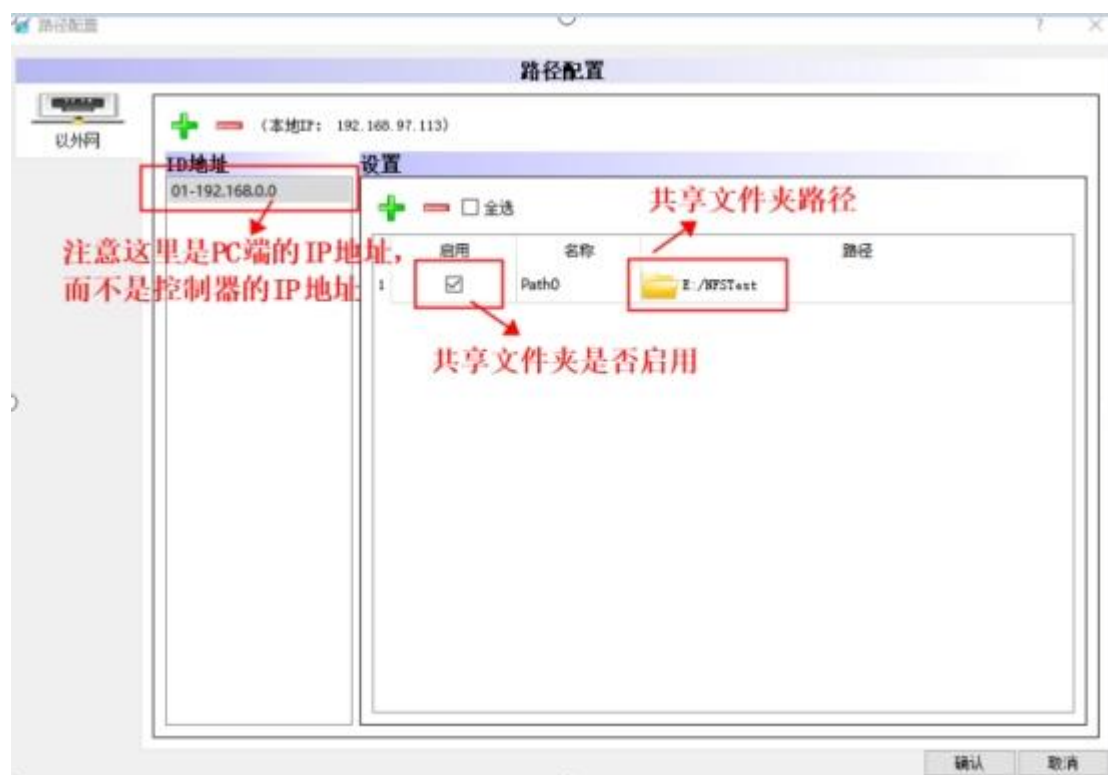


图 3.5.9 清理资源对话框

3.5.10. 正则表达式编辑器

3.5.11. 共享文件设置

共享文件能够使控制器加载PC端的磁盘文件，配合文件选择器插件一起使用，使用需PC端开启NFS服务、搭建NFS服务器、更新资料“Nfs_Server.axBin”、HMI Designer 设置共享文件路径。具体流程可查看《共享文件夹教程》，里面有详细配置说明、需要更新的系统资料、NFS服务器软件等。



结合宏向导HSharedFile函数群使用，可通过宏设置电脑IP、查询远程设备数量、获取远程设备信息、重新挂载文件夹



3.5.12. 资料备份和更新

3.5.13. 组件固有文字翻译

3.5.14. 导出翻译

3.5.15. 导入翻译

3.5.16. 词库设定

3.6. 设置

3.6.1. 软件语言

3.6.2. 语言设定

3.6.3. 工程设置

用户可在新建工程时进行工程属性相关设置，也可新建工程完毕后修改其属性。点击菜单栏

【设置】选项，在选单中点击【工程设置】，软件会弹出“工程设置”对话框，如图 3.6.3-1 所示



图 3.6.3-1 工程设置对话框

“工程设置”对话框中有【常规】、【客制】、【权限设置】、【屏幕尺寸调整】、【字体】多个选项卡，用户可根据实际需要设置工程属性。

【常规选项卡】

设备轴数：有 40 轴和 64 轴可选，默认工程 40 轴。创建对应轴数的工程后需搭配对应轴数的系统资料使用，默认的资料资料搭配 40 轴工程使用，后缀带 64 轴的系统资料搭配 64 轴工程使用，可通过Trans软件右下角查看控制器工程轴数。

屏幕大小：用户可在该下拉框中选择画面分辨率，默认为 800*600，建议用户根据控制器实

际大小来设置画面分辨率。

默认位置大小：该功能选中后，开机第二个进度条为默认显示，即图 3.6.3-2 所示进度条样式。



图 3.6.3-2 默认进度条位置大小

X：选择开机第二个进度条左上角X轴（横向）的位置

Y：选择开机第二个进度条左上角Y轴（纵向）的位置

宽度：设置开机第二个进度条的宽度

高度：设置开机第二个进度条的高度

颜色：设置开机第二个进度条的颜色，默认为红色

【客制】

Formview标题：设置Formview标题

Formview图标：设置Formview图标

开机画面：可以更改默认开机画面图片

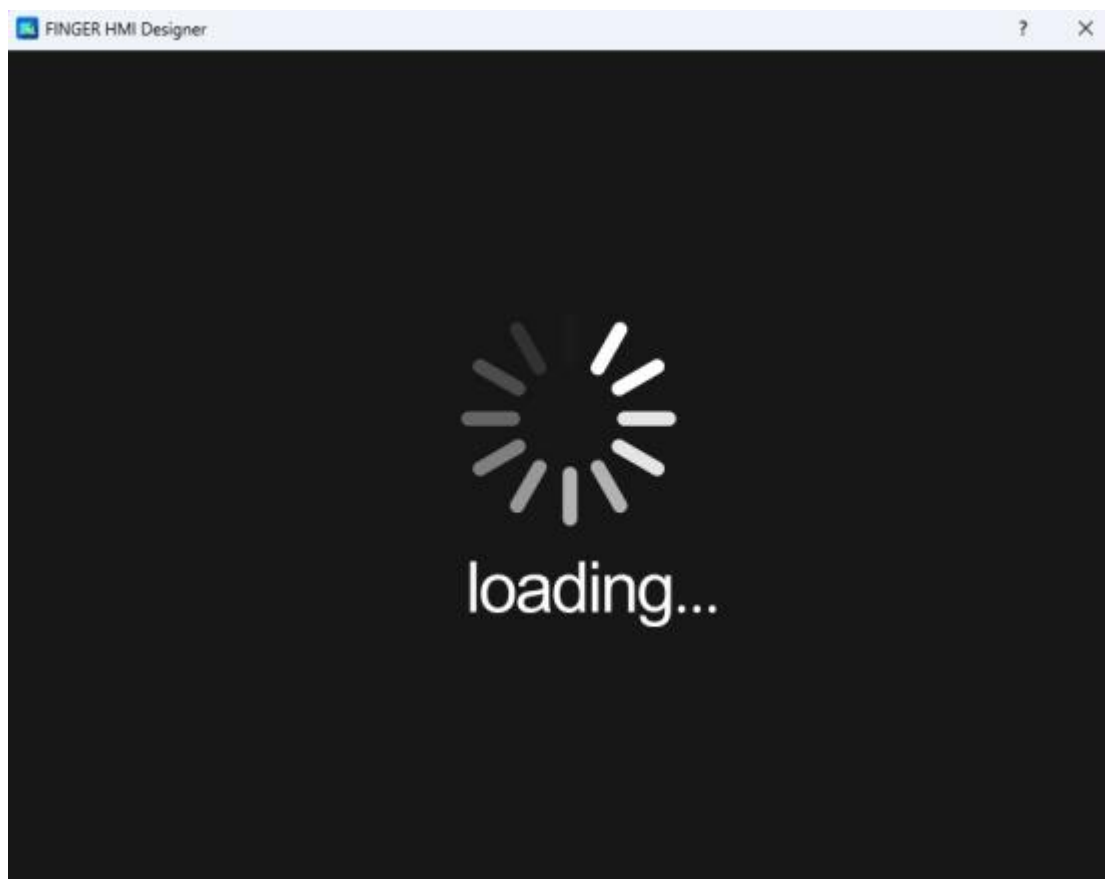


图 3.6.3-3 默认开机画面

【权限设置】

设置控制器的权限，搭配权限管理插件使用，在此处设定等级，通过权限管理插件来登录/切换权限。

重置权限等级及密码：勾选后，传入控制器会移除之前并使用新设置的权限等级及密码

开机恢复默认等级：勾选后，选取的等级为每次开机后默认的等级，等级越小优先级越高，等级 0 代表无权限设置

设置等级数量及对应密码：设置后在权限管理插件可下拉查看设定等级

添加和移除等级：先添加等级栏，再去设置等级名称和密码

【屏幕尺寸调整】

用户工程中的屏幕尺寸需要根据控制器屏幕大小来进行设置，若需要对建立工程时设置好的画面尺寸进行调整，可在菜单栏【设置】选单下的【工程设置】对话框中的【屏幕尺寸调整】选项卡中进行修改。

用户可直接修改当前工程中的画面尺寸，也可通过“另存为新的工程”进行修改。若选择“修

改当前工程”，点击确定后，当前工程画面尺寸将会被设置会修改后的尺寸；若选择“另存为新的工程”，则当前正在编辑工程画面大小不会修改，软件将弹出一个对话框，设置好名称和存放路径后，该路径下会生成一份与正在编辑工程相同的新的工程文件，新文件画面尺寸即为修改后的尺寸。

【字体】

【工程设置】对话框中的【字体】选项卡可设置工程中的字体，用户可在其中设置 5 种字体。双击“Enable”列下的表格框，该表格框会变为下拉框样式，用户可单击下拉框来选择字体，如图 3.6.3-4 所示

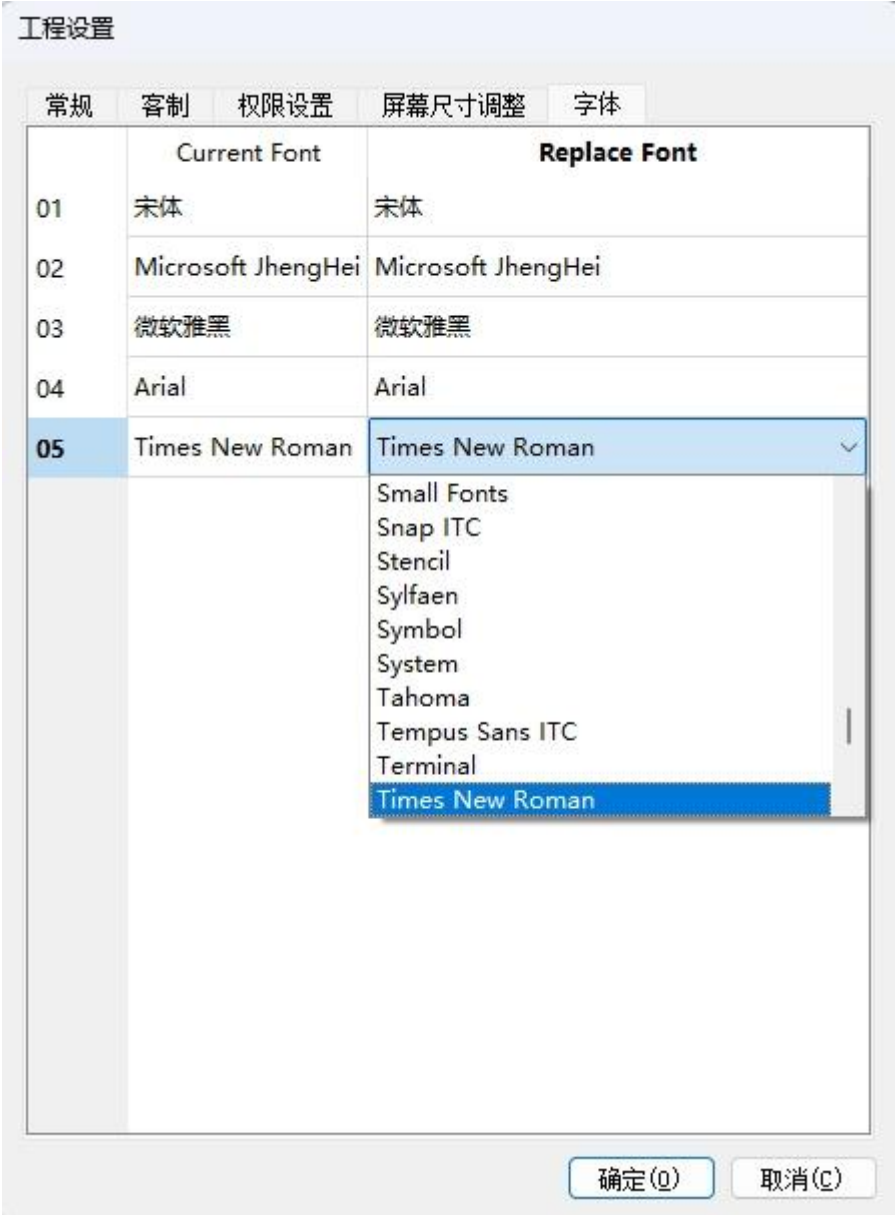


图 3.6.3-4 设置工程字体族

设置好工程字体后，即可看到控件属性中的字体选项中添加了工程字体中的选项，如图 3.6.3-5 所示。

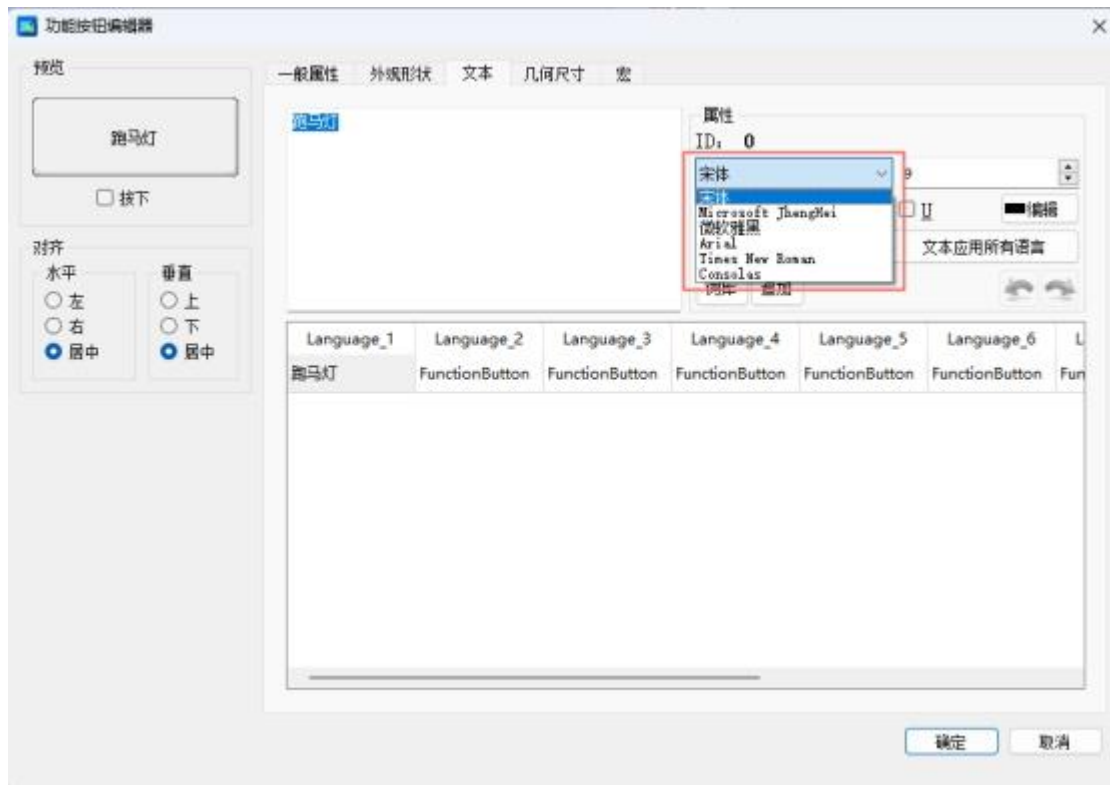


图 3.6.3-5 功能按钮插件中的字体属性

3.6.4. 通讯设置



通讯设置可以将控制器上的文件挂载到PC端，结合PCbase使用可以在PC端访问控制器的文件。

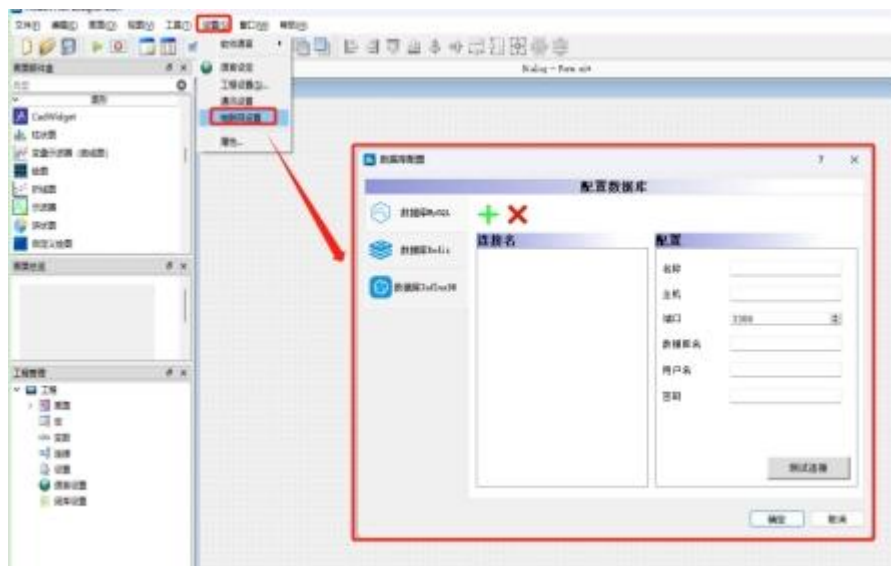
IP：控制器的IP，需要PC端和控制器IP在同一网段

端口：控制器设置的端口一致，一般为 60000

挂载盘符：设定为PC端未使用的盘符

3.6.5. 物联网设置

1. 入口

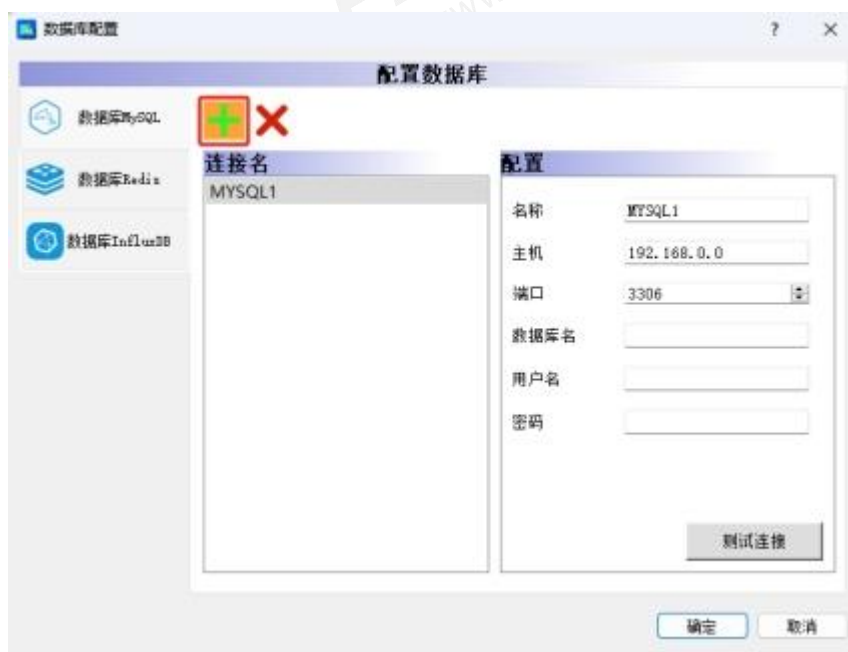


2. 功能概述

此设置功能主要是在启用物联网功能情况下，用于设置物联网的数据来源，此处用于配置数据库数据源，支持MySQL、Redis和InfluxDB，目前真正正在使用的仅有MySQL。

3. 数据库配置

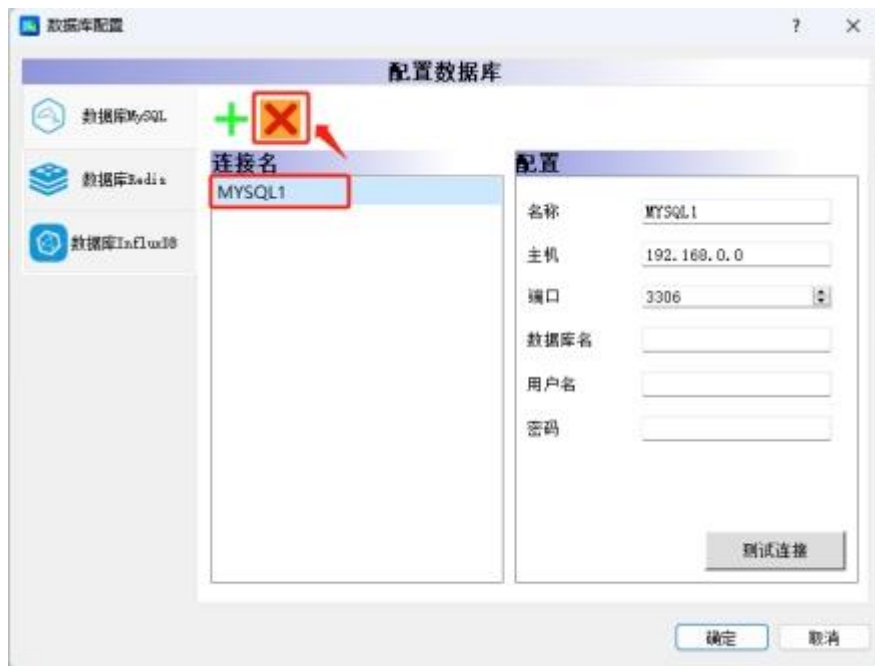
1) 创建连接



如上图所示，点击  按钮进行创建一个数据库连接。创建后将在连接名栏目下添加一条

连接。

2) 删除连接

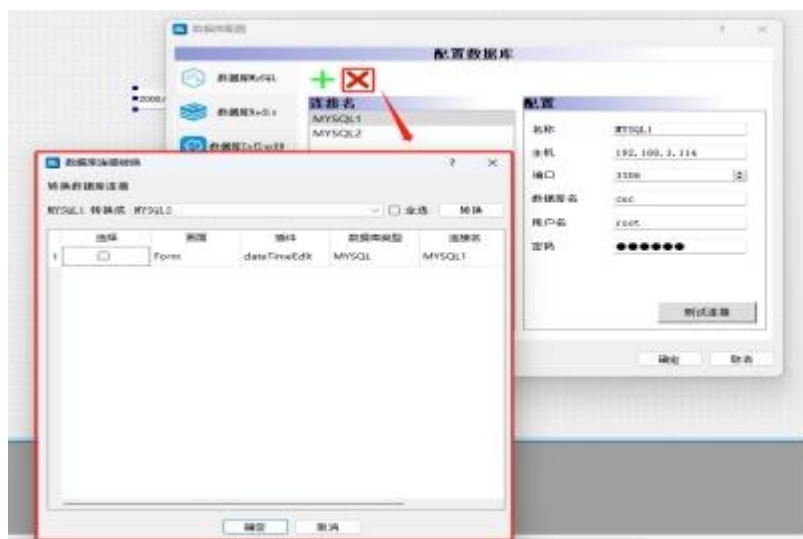


如上图所示，在连接名栏目下选择一条连接，然后点击  按钮进行删除连接。注意：在删除连接时，将分情况进行删除：

- 当前需删除连接未被使用

若当前所要删除的连接在HMI中未被使用，那么将可以直接删除。

- 当前需删除连接已被使用



如上图所示，当所要删除的数据库连接已在HMI中被使用，那么删除时将跳转到“转换数据库连接”设置窗口进行转换设置。



如上图所示，该窗口将罗列出所要删除的数据库连接已被哪些组件所关联，然后用户可以选择将已关联此数据库连接的组件替换成其他数据库连接，从而实现转换。

3) 配置

● 名称

表示数据库连接的名称。

● 主机

表示数据库的主机地址，例如：192.168.0.1

● 端口

表示数据库的端口，例如：3306

● 数据库名

表示数据库的库名，例如：cnc

● 用户名

表示数据库的用户名，例如：root

● 密码

表示数据库连接的密码，例如：123456

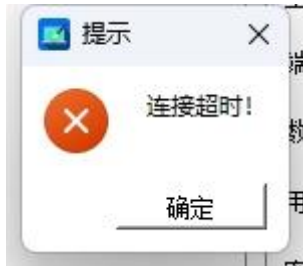
- 测试连接

表示对当前所配置的数据库连接进行连通测试。测试连接可能的提示：

- 连接成功

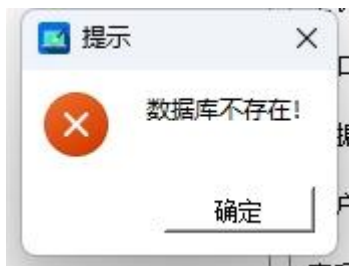


- 连接超时



在网络不通情况或主机地址不对情况下的提示。

- 数据库名不对



- 用户名或密码不对



4) 配置说明



如上图所示，数据库配置画面是由tab标签、新增连接按钮、删除连接按钮、列表区、连接配置区和按钮组组成。

a. tab 标签：

通过点击tab标签切换到不同的数据库，然后展示该数据库已配置的连接信息列表区、配置区、新增连接按钮和删除连接按钮。

b. 新增连接按钮

通过点击按钮进行连接的新增，点击后将预生成一个连接，并且该连接的初始化信息将在配置区展示，且可编辑修改。最后未点击按钮组的确认按钮，则本次的新增不生效。

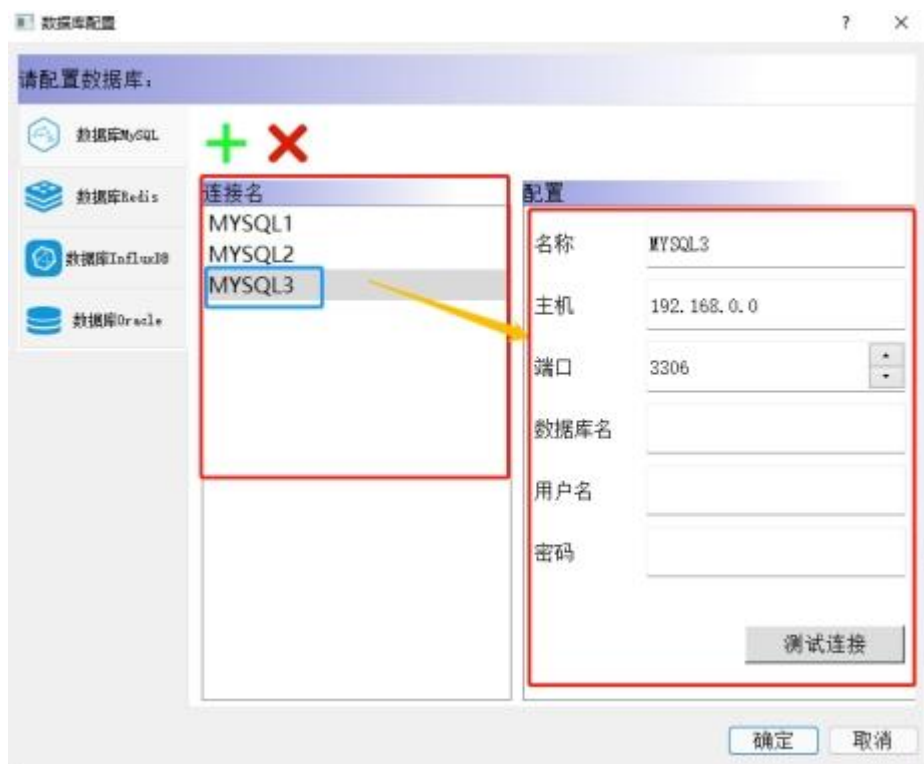
c. 删除连接按钮

通过点击按钮进行连接的删除，点击后先会判断连接在工程画面中是否已被关联，如果已被关联，则将弹出如下画面进行插件已配置的连接转换操作：



下拉框中可以选中已配置的连接，当勾选目标行后，点击转换，那么列表中的目标行的连接名列将被改成所选择的连接名。最后点击确认按钮后将真正进行转换，否则转换不生效。同时数据库配置画面的列表区将删掉此转换的连接，未点击按钮组的确认按钮，则本次的修改及删除不生效。

d. 列表区



如上图，连接信息列表区中显示已经配置好的连接信息，点击新增按钮后生成的连接也在列表中追加展示，然后选中的连接的具体信息将在配置区展示，可修改，且可通过测试按钮进行连通测试。

e. 连接配置区

通过选中列表区的指定连接，即可在连接配置区对该连接的具体信息进行修改，在列表中切换选项时进行保存到缓存中，只有最后点击按钮组的确认按钮才生效，否则操作无效。

f. 按钮组

按钮组有确认跟取消，当配置修改或者新增或者删除连接后，点击确认按钮本次操作才生效，否则点击取消按钮或者直接关闭窗口，本次操作均不生效。

3.6.6. 属性

3.7. 窗口

3.8. 帮助

4. 公共属性

4.1. 对象名称

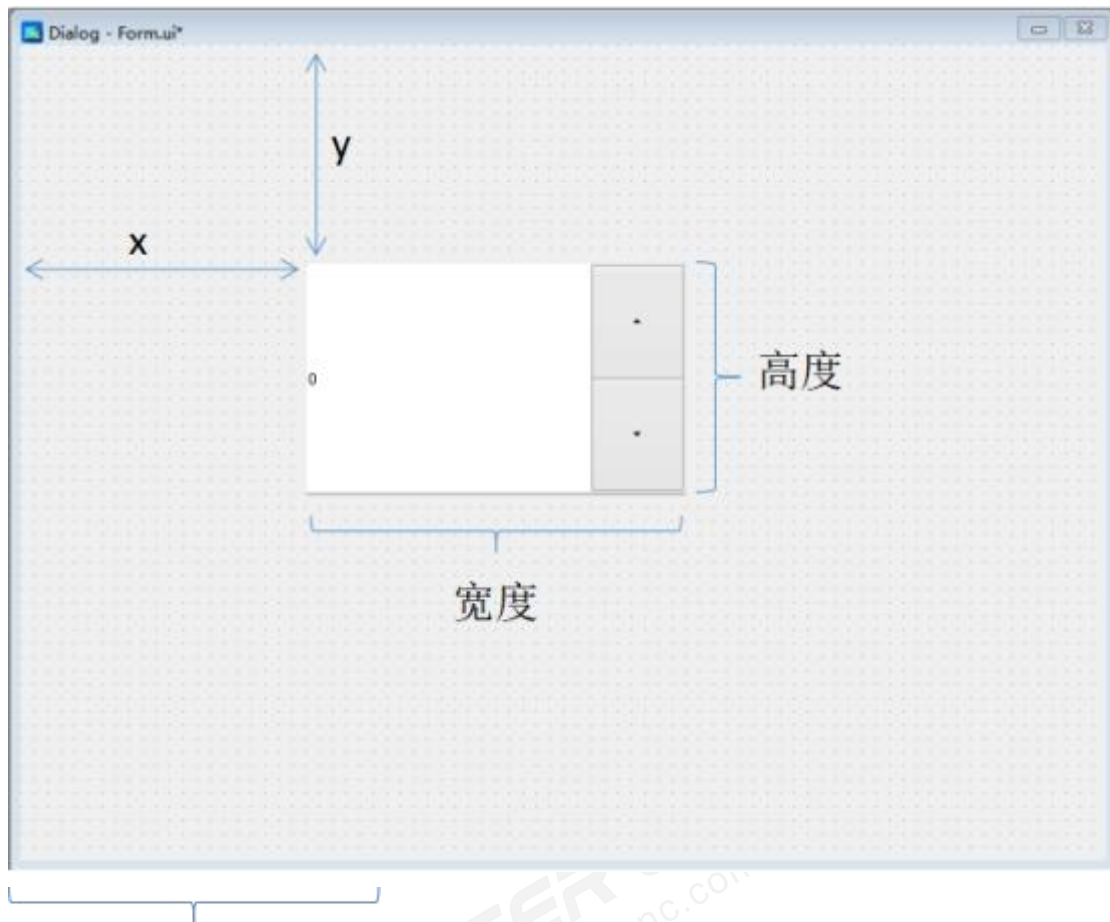
对象名称只能以字母或下划线作为开头，由字母、数字和下划线组成的命名标识符，避免使用关键字作为对象名称使用。每个对象都可以通过设置对象名称（Object Name）来标识和访问，用于在代码中引用和操作对象。

QObject	
对象名称	spinBox

4.2. 几何尺寸

几何尺寸由四个属性组成：x、y、width 和 height。其中，x 和 y 表示控件左上角相对于父窗口左上角的偏移量，width 和 height 表示控件的宽度和高度。

几何尺寸	[(210, 160), 281 x 171]
X	210
Y	160
宽度	281
高度	171



4.3. 字体

▼ 字体	A [宋体, 9]
字体族	宋体
点大小	9
粗体	☐
斜体	☐
下划线	☐
删除线	☐

- 字体族

在工程可用的字体族中选择一种字体，可用字体可以在工程设置中设定。（[跳转到工程设置的字体设置说明](#)）。

- 点大小

设置字体大小。

- 粗体

设置字体是否使用粗体



- 斜体

设置字体是否使用斜体



- 下划线

设置字体是否使用下划线



- 删除线

设置字体是否使用删除线



【字距调整】

勾选时有效利用字符形状间的间隙，使字形之间的空间更加相似。

【反锯齿】

即抗锯齿，用于消除[显示器](#)输出的[画面](#)中图物边缘出现凹凸锯齿，默认为开启反锯齿，图为反锯齿对比：



图 4.3-1 反锯齿对比

4.4. 光标

将鼠标指针移动至组件时的指针显示形态。

4.5. 聚焦策略

聚焦策略（Focus Policy）决定了部件（Widget）如何获取焦点和处理焦点事件。插件提供了不同的焦点策略，以适应不同的用户交互需求。

策略	处理方式
NoFocus	部件不会获取焦点，并且不会处理焦点事件。这通常用于展示性的部件或不需要用户交互的部件。
TabFocus	部件可以使用 Tab 键来获取焦点。这是大多数可接受用户输入的部件的常见策略。
ClickFocus	部件可以在鼠标单击时获取焦点。这对于单击即可编辑的部件非常有用。
StrongFocus	部件可以通过 Tab 键或鼠标单击来获取焦点。这是一种更灵活的焦点策略，通常用于自定义用户界面和特殊交互需求。
WheelFocus	部件可以通过 Tab 键、鼠标单击、鼠标滚轮事件来获取焦点。这对于需要滚动控制的部件很有用。

4.6. 布局方向

策略	处理方式
LeftToRight	从左至右布局
RightToLeft	从右至左布局
LayoutDirectionAuto	自动布局（默认为从左至右布局）



图 4.6-1 布局对比

4.7. 自动填充背景

勾选后能使背景颜色生效，不勾选时背景颜色不生效或组件背景色将跟随父部件背景色。

4.8. 使用

默认勾选，勾选时插件为启用状态，不勾选时插件不启用。

4.9. 调色板

默认选项为继承，即保持父部件的调色板属性。单击“继承”，显示内容变为“改变调色板”，单击此项，弹出调色板编辑窗口：

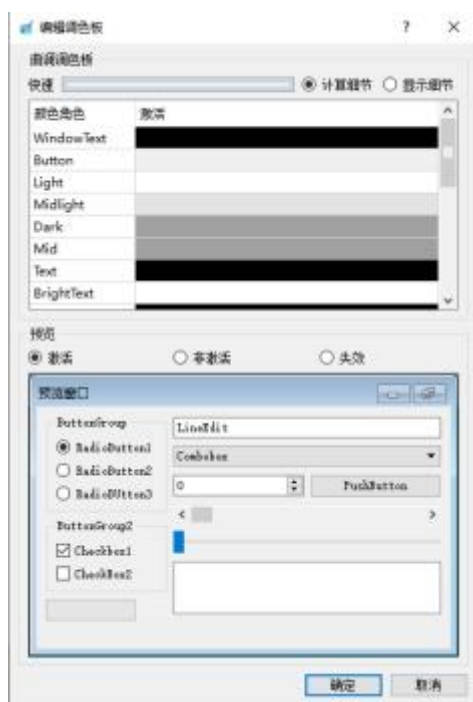


图 5.1-2 编辑调色板

快速：单击标题后的白框，进行颜色选择，计算机将根据所选择的颜色（以数据表示则为 RGB 值）调整所有对话框激活的颜色，快速选择绿色（120，255，255）效果：

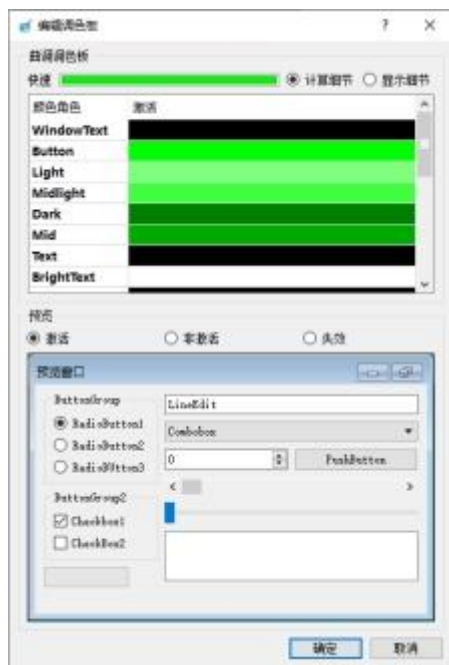


图 5.1-3 快速选择绿色

1. 计算细节：计算机自动调整组件非激活和失效两种状态下的颜色
2. 显示细节：用户自行分别调整组件激活、非激活、失效三种状态下的颜色
3. 设置好调色板后，用户可选择激活、非激活、失效来查看三种状态下的预览效果，其中

激活指的是当前窗口显示，非激活指的是窗口后台但是没有最小化，失效表示该窗口不能编辑。

4.10. 样式表

- 1. 样式表可对插件属性样式进行定制处理，在父部件样式表中设置的属性，子部件相同属性也会被修改，此项功能不建议用户使用。
- 2. 例如修改水平滚动条的滑动块颜色为红色，可添加样式表

```
QScrollBar:horizontal {  
height:30px;background:rgb(240, 240, 240);  
margin:0px, 0px, 0px, 0px;  
padding-left:30px;  
padding-right:30px;  
}  
QScrollBar::handle:horizontal {  
background-color: red;  
min-height: 30px;  
}
```



图 4.10-1 水平滚动条添加样式表

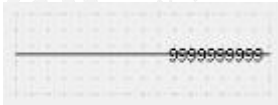
4.11. 框架

QFrame	
边框形状	Box
边框阴影	Raised
线宽	8
中线宽度	9

框架的线宽、中线宽度参数示意图如下



● 边框形状

选项	说明
NoFrame	无框架形状，即不显示框架
Box	矩形框架形状，部件被一个简单的矩形边框包围 
Panel	面板形状，与矩形框架形状类似，呈现凸起或凹陷效果。 
WinPanel	Windows 风格矩形面板，凸起或凹陷效果。 
HLine	没有框架的水平线，可以作为分离器。 
VLine	垂直线形状，呈现为一条垂直直线。

	
StyledPanel	矩形面板，依赖于当前 GUI 样式，凸起或凹陷效果。 

● 边框阴影

Plain	无阴影效果，显示一个普通的面板或框架。 
Raised	凸起阴影效果，看起来浮在背景上。 
Sunken	凹陷阴影效果，看起来沉入背景。 

● 线宽

设置边框的线条宽度。对 Box, Panel, HLine, VLine 框架类型生效。对 NoFrame, WinPanel, StyledPanel 设置线宽不会产生视觉上的效果。

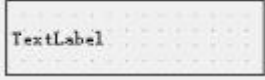
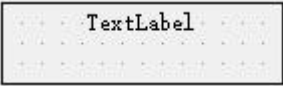
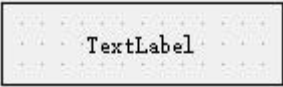
● 中线宽度

设置边框的中线宽度。对 Raised 和 Sunken 的阴影效果生效。

4.12. 对齐

文本的对齐策略决定了文本在容器的对齐方式。

▼ 对齐	AlignLeft, AlignVCenter
水平的	AlignLeft
垂直的	AlignVCenter

方向	策略	效果
水平	AlignLeft(左对齐)	文本左边缘与容器或页面的左边缘对齐。 
	AlignHCenter(水平居中对齐)	文本在容器或页面中水平居中对齐。 
	AlignRight(右对齐)	文本右边缘与容器或页面的右边缘对齐。 
	AlignJustify(两端对齐)	文本在容器或页面中两端对齐，即首尾行左右两端对齐，中间通过调整字间距使文本整体宽度与容器或页面宽度相等。 
垂直	AlignTop(顶部对齐)	文本上边缘与容器或页面的顶部对齐。 
	AlignVCenter(垂直居中对齐)	文本在容器或页面中垂直居中对齐。 
	AlignBottom(底部对齐)	文本下边缘与容器或页面的底部对齐。 

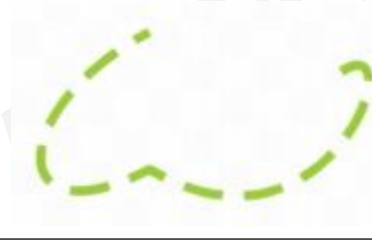
4.13. 操作权限

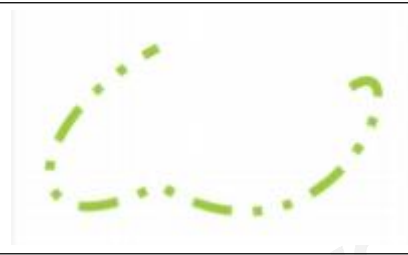
- 1. 输入类型的插件（数值插件、代码编辑器、表格等等）能够设置操作权限，插件默认权

限为 0，即不设权限。

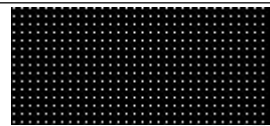
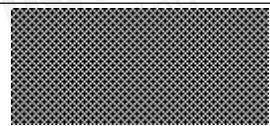
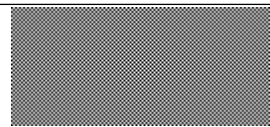
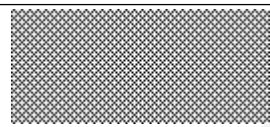
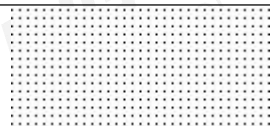
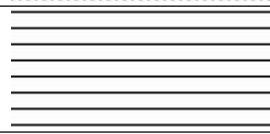
2. 控制机器人权限在“设置”-->“工程设置”-->“权限设置”页面设置，配合权限管理插件使用。
3. 当控制器的人机权限数值大于插件权限数值时，权限不足。（操作插件所需权限为 1，控制器权限为 2 时权限不足无法操作插件。）

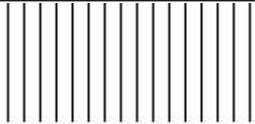
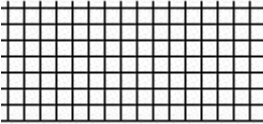
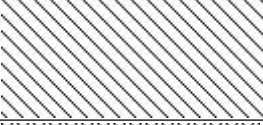
4.14. 线类型

选项	样式
NoPen(无线条)	不绘制
SolidLine(实线)	
DashLine(虚线)	
DotLine(点线)	
DashDotLine(一点一划线)	

DashDotDotLine (两点一划线)	
------------------------	--

4.15. 画刷风格

选项	样式
NoBrush (无画刷)	
SolidPattern (实心)	
Dense1Pattern (稠密 1)	
Dense2Pattern (稠密 2)	
Dense3Pattern (稠密 3)	
Dense4Pattern (稠密 4)	
Dense5Pattern (稠密 5)	
Dense6Pattern (稠密 6)	
Dense7Pattern (稠密 7)	
HorPattern (水平)	

VerPattern(垂直)	
CrossPattern(交叉)	
BDiagPattern(反向斜线)	
FDiagPattern(正向斜线)	
DiagCrossPattern(交叉斜线)	

4.16. 模态

模态与非模态的区别

模态窗口：窗口弹出后，没关闭之前，不可以对同一应用程序的其他窗口进行操作；

非模态窗口：窗口弹出后，没关闭之前，可以对同一应用程序的其他窗口进行操作，可以同时和多个窗口进行交互；

4.17. 窗体类型

4.17.1. MainForm

主窗体类型，最为常用的窗体类型之一，主要用来编辑和存放插件，起始画面必须是主窗体类型。

4.17.2. SubForm

有边框子窗体类型，有边框子窗体一般需要依赖主窗体使用，有边框子窗体与无边框子窗体

的区别在于边框显示，边框标题可通过属性和宏进行编辑。

4.17.3. FramelessSubForm

无边框子窗体类型，无边框子窗体与有边框子窗体的区别在于边框显示。

4.17.4. KeypadForm

键盘窗体类型，主要用于编辑自定义软键盘

4.17.5. EmbeddedForm

嵌入式子画面类型

配合Frame（框架）插件使用

1. Frame 增加属性 `referenceFormEnable`，是否使用画面引用功能
2. Frame 插件增加属性 `autoZoomFormEnable`，控制嵌入式画面是否自适应大小。
 - 1) 开启了 `autoZoomFormEnable` 属性，则根据 Frame 的几何尺寸，自动调整嵌入式画面的分辨率，令其恰好铺满。
 - 2) 未开启 `autoZoomFormEnable` 属性，则嵌入式画面比 Frame 几何尺寸大时，使用滚动条调整显示位置。嵌入式画面比 Frame 几何尺寸小时，剩余位置显示空白。
3. 禁止嵌套使用 Frame 的引用功能

编辑时，存在嵌套使用的情况下，使用弹窗提示用户。

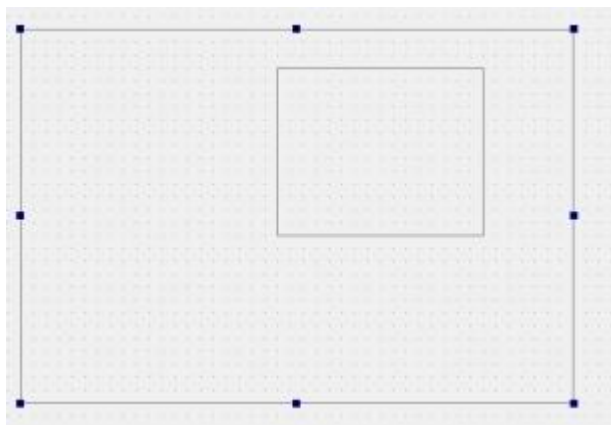


图 4.17.5-1 不支持嵌套 Frame

4. 多个 Frame 关联同一个嵌入式子画面时，只有一个 Frame 画面可正常显示



图 4.17.5-2 嵌入式子画面不支持复用

5. 嵌入式子画面中禁止使用 Frame 的画面引用功能，避免循环嵌套 Frame 的引用功能。例如，子画面 A 的 Frame 插件引用子画面 B，子画面 B 的 Frame 插件又引用子画面 A，导致循环嵌套。

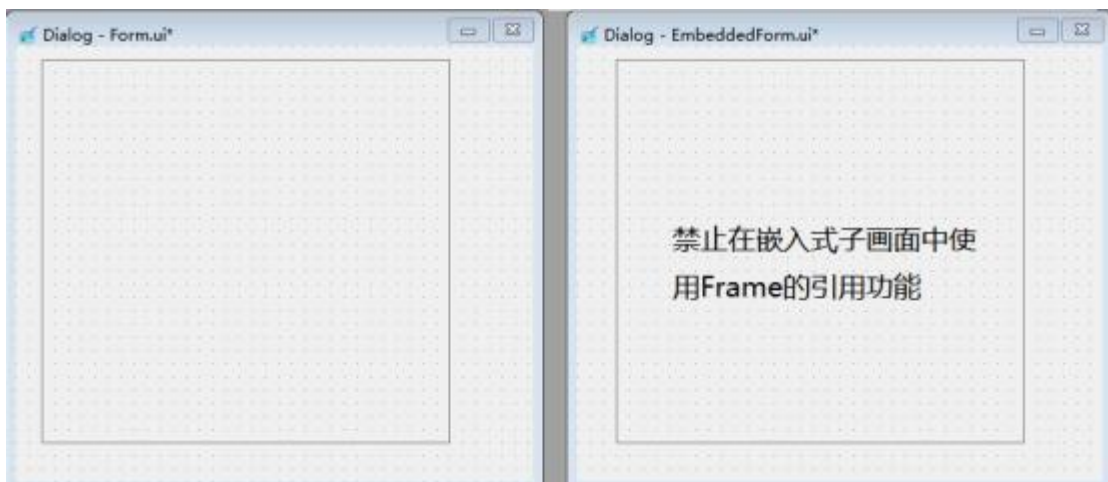


图 4.17.5-3 嵌入式子画面禁止使用 Frame 的画面引用功能

6. 增加接口，可以使用宏切换引用的子画面。

//切换引用画面

//name: 画面名

```
bool switchReferenceForm(QString name);
```

示例：Form.frame.switchReferenceForm("Form_2");

5. 公共功能

5.1. 软键盘

5.1.1. 软键盘属性编辑

软键盘

基础

软键盘类型：

自动

软键盘画面：

☒ 自动隐藏

☒ 可移动

位置策略：

自动

自定义几何尺寸

X：

0

宽度：

0

Y：

0

高度：

0

● 软键盘类型

软键盘类型	处理方式
无	不使用软键盘
自动	使用系统软键盘
自定义	使用自定义软键盘

● 软键盘画面

软键盘类型选择自定义软键盘时，需要选择具体的软键盘画面，以确定使用哪个自定义软键盘。（软键盘画面是指窗体类型为 KeypadForm 的画面）

● 自动隐藏

57

开启“自动隐藏”，点击屏幕上软键盘区域外的位置，取消输入并关闭软键盘。

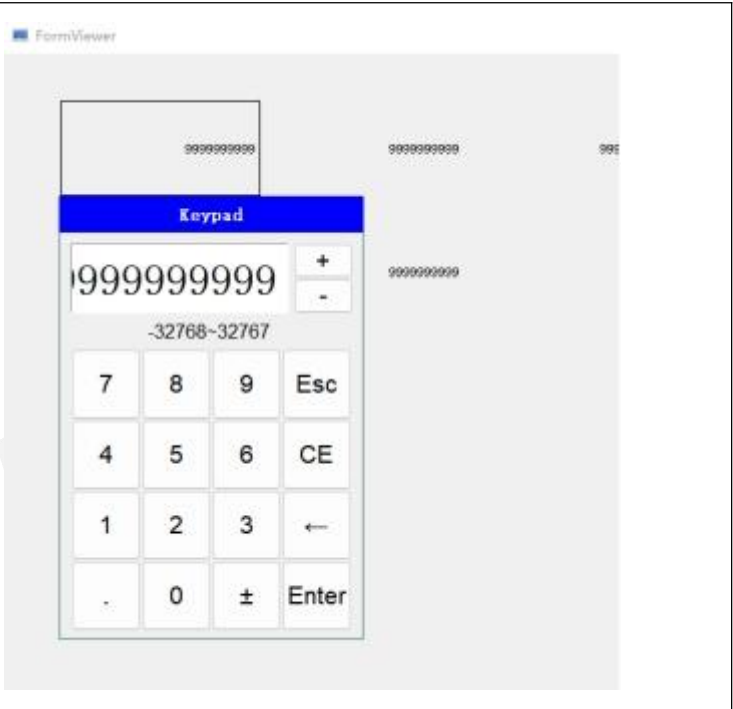


● 可移动

开启“可移动”选项，软件盘位置可以使用拖拽标题栏的方式移动。

● 位置策略

策略	具体
自动	自动计算软键盘的位置，默认显示在插件的下方。下方空间不足时，尝试放在上、右、左侧。

	
居中	软键盘居中显示 
自定义	根据自定义几何尺寸的 x, y 坐标，显示软键盘。

- 自定义几何尺寸
- X: 软键盘左上角的横坐标
- Y: 软键盘左上角的纵坐标
- 宽度: 软键盘宽度
- 高度: 软键盘高度
- (注: 宽度和高度任意一项设置为 0 时, 使用默认的软键盘几何尺寸)

5.1.2. 系统软键盘

系统软键盘如下：

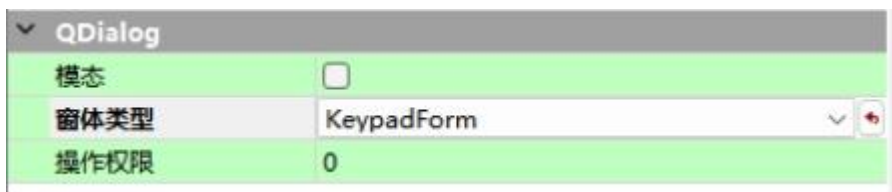
软键盘类型	样式
二进制	
八进制	
十进制	

十六进制	
数字密码输入键盘	
字符密码输入键盘	
中英文键盘	

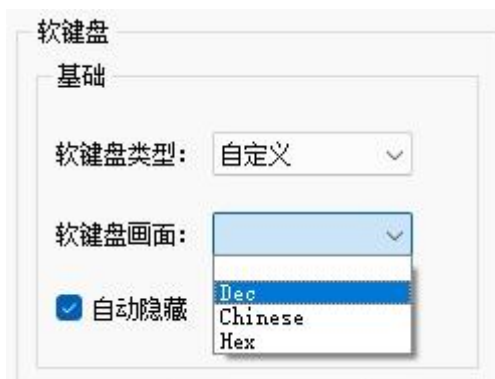
5.1.3. 自定义软键盘

1. 设计方式

新增一个画面，在属性编辑器中设置画面的窗体类型为“KeypadForm”，画面的对象名称即为键盘名称。



完成以上步骤，即可使用“软键盘画面”下拉框选择已有的键盘画面。



自定义软键盘由用户设计，提供了“键盘容器”，“键盘按键”，“键盘显示”三种组件供设计使用。（[跳转到三种组件说明](#)）

2. 自定义软键盘范例



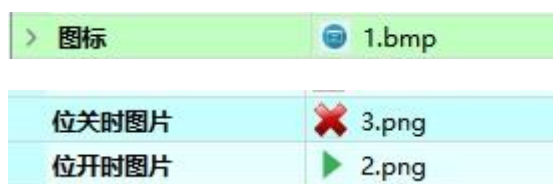
5.2. 多国语言编辑

5.3. 变数编辑弹窗

5.4. 图片资源

在图标、图片类型属性中，可以使用“图片资源编辑器”添加图片资源到工程中。

1. 图标、图片类型属性

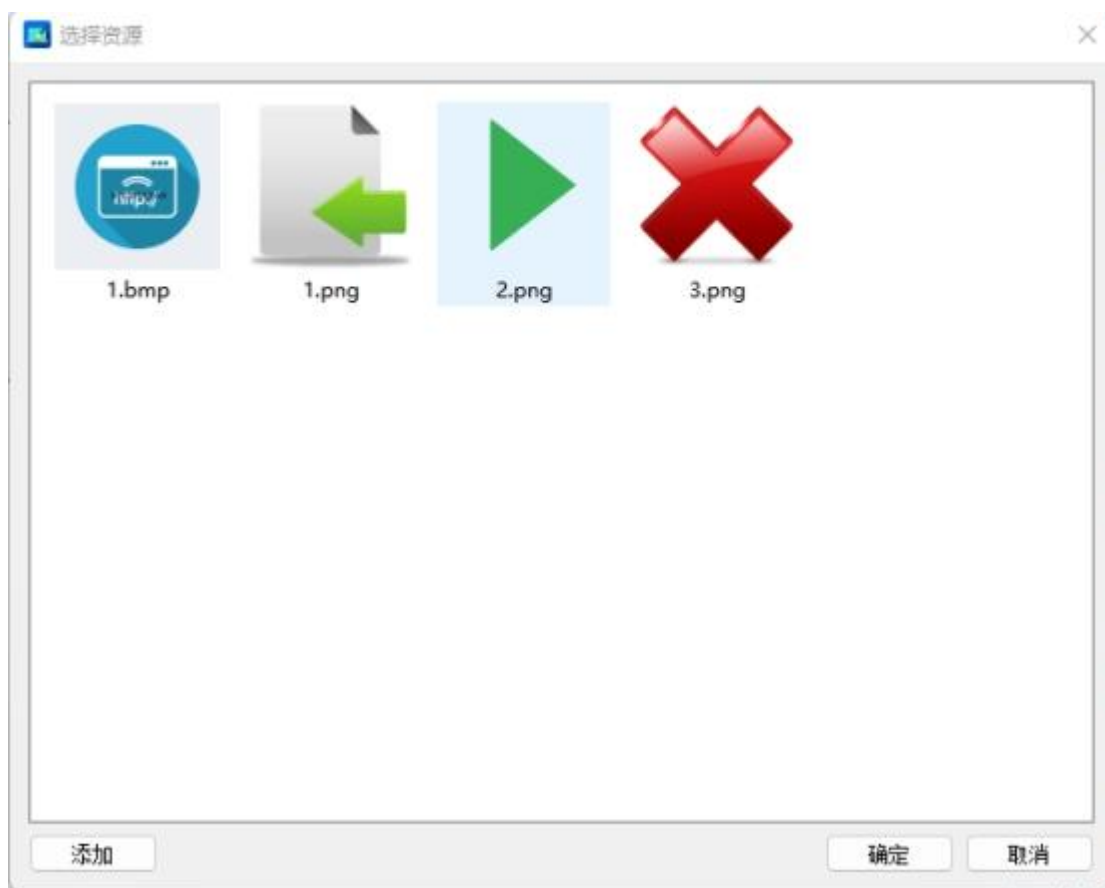


2. 图片资源编辑器

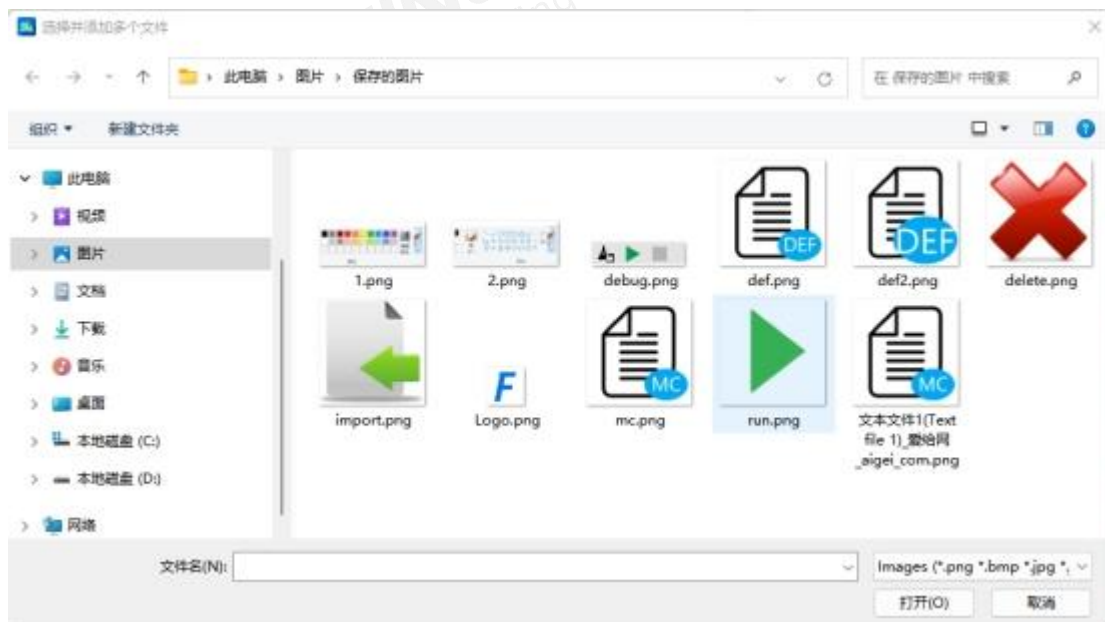
选中属性，点击右侧“...”按钮



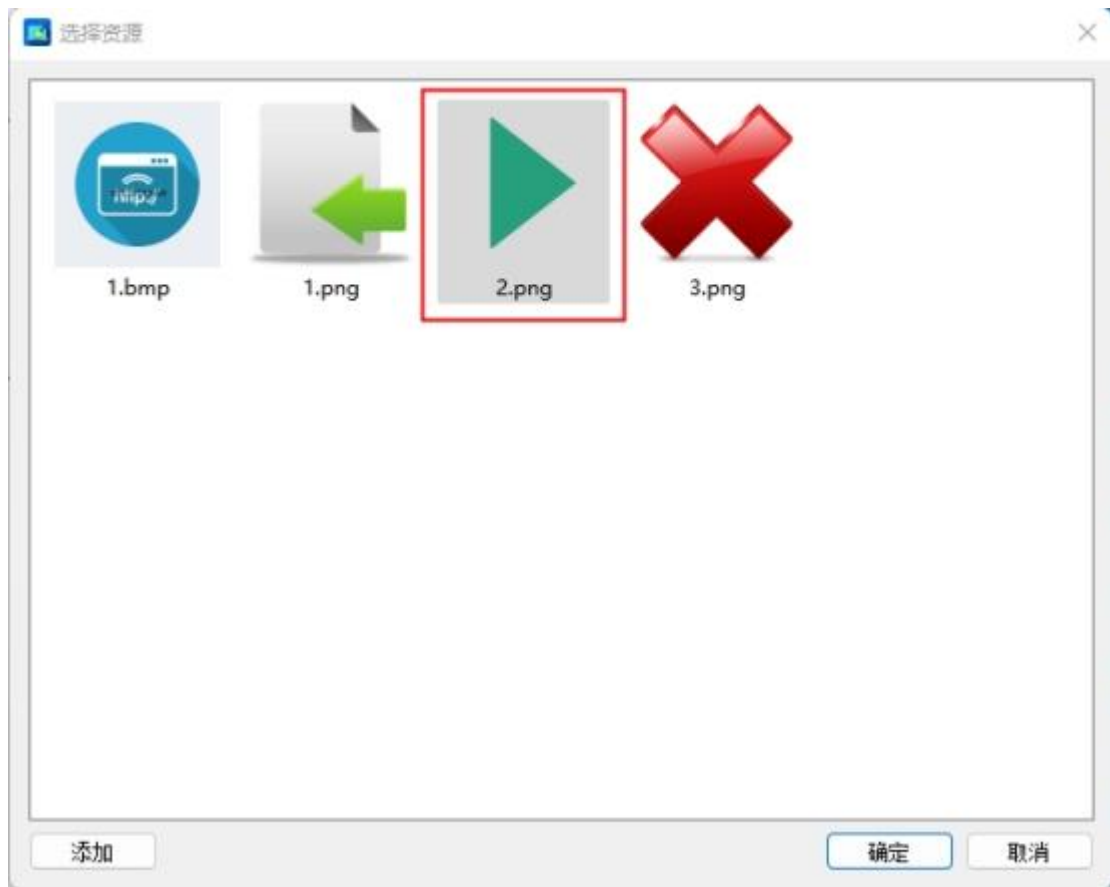
打开图片资源编辑器



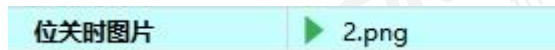
点击“添加”按钮，在本地文件中选择需要添加的图片。



完成添加后，选中所需的图片



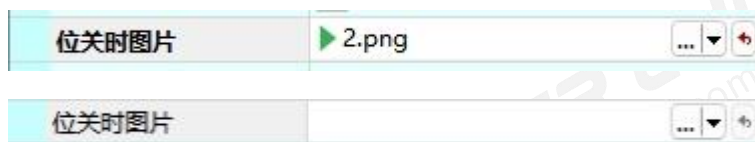
点击“确定”，即可设置到相应的属性中。



3. 重置属性关联图片

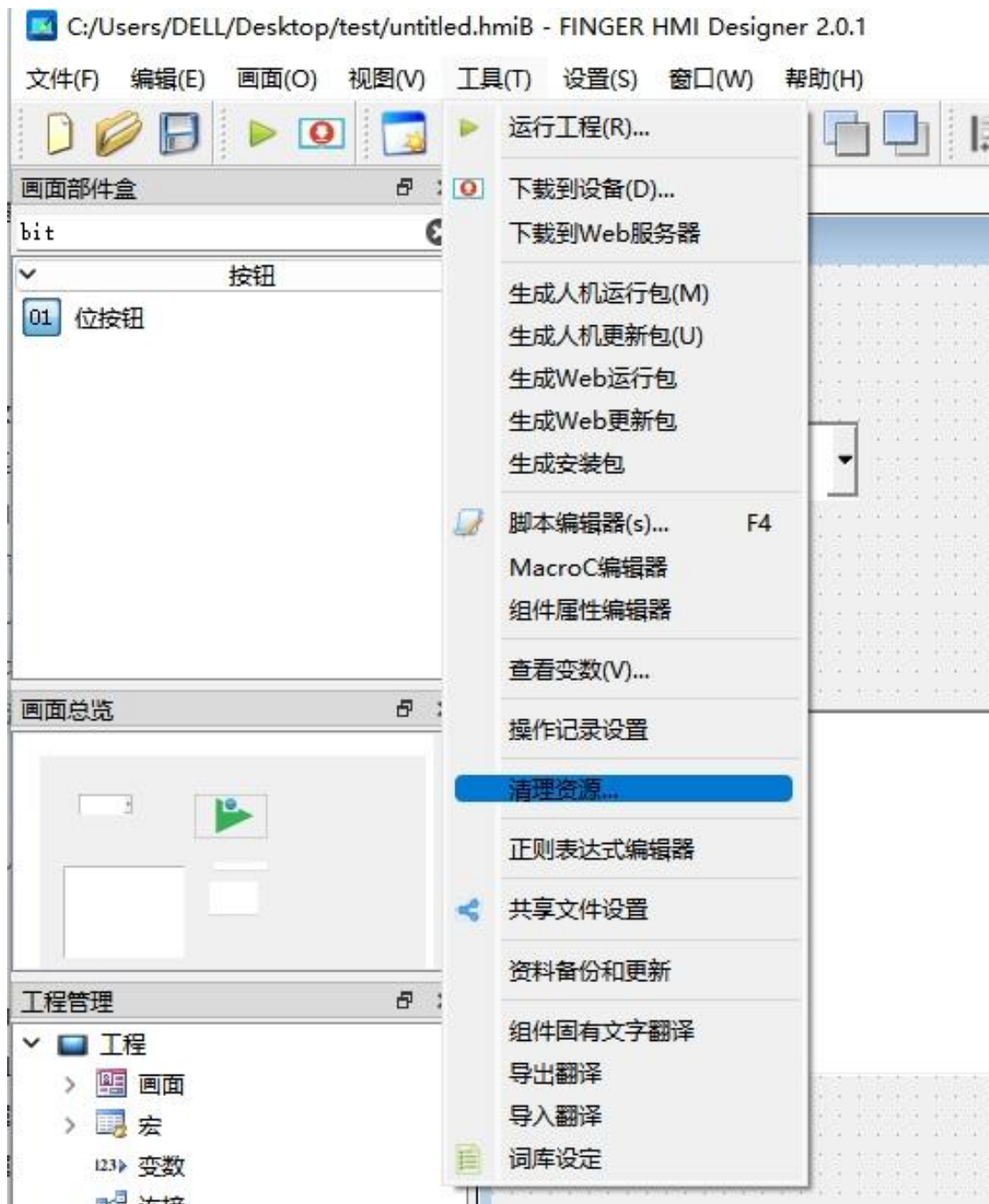


选中属性，点击右侧“”按钮，即重置属性关联图片。



4. 图片资源的清理

“图片资源清理器”可在 HMI 软件的工具栏中打开。



在“图片资源清理器”中选中在工程中未使用的图片，点击“移除”，即可将该图片从工程中移除。



注意，未在工程中使用的图片才会显示在“图片资源清理器”中，并且可移除。若工程中已使用过图片（某个属性关联该图片），则无法移除。

6. 组件

6.1. 水平滚动条

6.2. 垂直滚动条

6.3. 水平滑动条

6.4. 垂直滑动条

6.5. 时间编辑器

时间编辑器主要为用户提供了一种直观和方便的方式来选择和输入时间值, 亦或用来显示时间, 从而在各种应用场景中提供简单的输入操作, 并可通过在确定输入时间时发送时间改变信号, 然后通过外部监听信号以执行各种数据的获取与处理动作, 可用于需通过选择时间进行获取数据的项目中。

6.5.1. 组件属性

【对象名称】

默认名称: dateTimeEdit

【几何尺寸】【调色板】【字体】【光标】【聚焦策略】【布局方向】【自动填充背景】【使用】

【对齐】同公共属性说明

6.5.2. QDateTimeEdit

【最大日期】

表示设置组件的最大可选日期限制。

【最小日期】

表示设置组件的最小可选日期限制。

【格式】

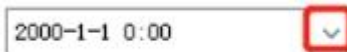
表示设置显示的时间字符串格式，例如：yyyy/M/d H:m:s

【日期时间】

表示设置组件的当前日期时间。

【弹出日历选择时间模式】

表示设置组件的时间设置模式。勾选表示在HMI运行时点击



按

钮时将弹出日历模式供选择日期时间；否则将以



模式进行设置日期

时间。

6.5.3. 物联网

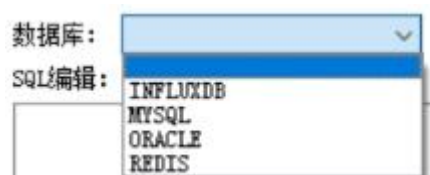
此栏目主要用于设置与物联网相关的属性。

【数据源类型】

用于设置物联网的数据来源。

【启用】

表示是否启用物联网功能。

【数据库】

通过下来列表选择要使用的数据库类型。目前仅支持MySQL数据库。仅限上述“数据源类型”

选择“数据库”选项时显示设置项。

【连接】

通过下拉列表选择数据库连接，下拉列表数据项来源于物联网设置中的数据库连接配置。仅限上述“数据源类型”选择“数据库”选项时显示设置项。

【SQL编辑】

用于编辑SQL查询语句，例如：`select col,col2 from table`。仅限上述“数据源类型”选择“数据库”选项时显示设置项。

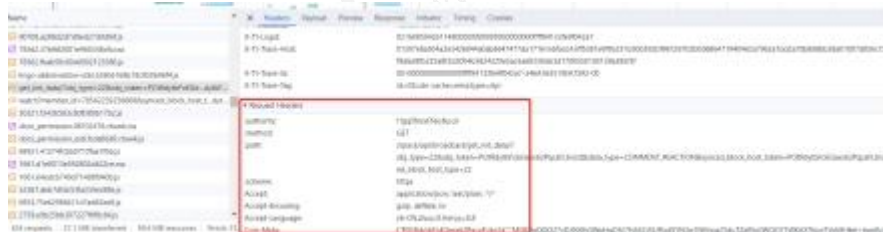
用于设置API请求的请求方式。仅限上述“数据源类型”选择“API”选项时显示设置项。

【URL】

用于设置API请求地址，例如：`https://www.baidu.com`。仅限上述“数据源类型”选择“API”选项时显示设置项。

【请求数据头】

用于设置API请求的数据头，以键（Key）-值（value）形式设置，如：



通过点击  按钮进行添加，点击  按钮进行删除。

仅限上述“数据源类型”选择“API”选项时显示设置项。

【启用映射】

设置是否启用数据字段映射。若未启用映射，则组件的数据要求字段必须为date（参考下述映射表），否则数据未映射，将不会在组件中设置成功。

【映射表】

在启用映射后，通过数据库查询得到的数据字段不是date情况下，用户可以使用查询到的数据字段映射到date字段即可。

【数据转换器】

通过下拉列表选择需要使用的数据转换器。下拉列表中的数据项来源于下述“编辑”中所创建的数据转换器。

【编辑】

通过点击编辑按钮进行管理数据转换器。

【预览】

通过点击预览按钮打开数据预览窗口。

【自动更新】

表示组件是否启用定时获取物联网数据进行自动刷新。

【更新频率】

表示组件数据定时刷新的周期，单位为秒（s）。

6.5.4. 组件宏接口

1 selectDate（选择日期）

```
/**
 * @描述 获取时间选择器时间值改变时的日期
 * @返回值 日期格式字符串[yyyy/MM/dd]
 */
```

示例：Form.dateTimeEdit.selectDate()

2.2 selectDateTime（选择日期时间）

```
/**
 * @描述 获取时间选择器时间值改变时的日期时间
 * @返回值 日期时间格式字符串[yyyy/MM/dd HH:mm:ss]
 */
```

示例：Form.dateTimeEdit.selectDateTime()

2.3 selectTime（选择时间）

```
/**
 * @描述 获取时间选择器时间值改变时的时间
 * @返回值 时间格式字符串[HH:mm:ss]
 */
```

示例：Form.dateTimeEdit.selectTime()

2.4 setSelectDate（设置所选日期）

```
/**  
* @描述 设置时间选择器的日期  
* @参数 date 日期格式字符串[yyyy/MM/dd]  
* 无返回值
```

示例: Form.dateTimeEdit.setSelectDate("2022/06/08")

2.5 setSelectDateTime (设置所选日期时间)

```
/**  
* @描述 设置时间选择器的日期时间  
* @参数 dateTime 日期时间格式字符串[yyyy/MM/dd  
HH:mm:ss]  
* 无返回值
```

示例: Form.dateTimeEdit.setSelectDateTime("2022/06/08 13:13:01")

2.6 setSelectTime (设置所选时间)

```
/**  
* @描述 设置时间选择器的时间 (设置时间时默认日期为当前  
日期)  
* @参数 time 时间格式字符串[HH:mm:ss]  
* 无返回值
```

示例: Form.dateTimeEdit.setSelectTime("13:13:01")

2.7 updateIOTData (更新物联网数据)

```
/**  
* @描述 手动获取物联网数据刷新  
*/
```

示例: Form.dateTimeEdit.updateIOTData()

6.5.5. 组件宏属性

属性	类型	描述	示例
date	QDate	选中日期， 参 考 JavaScript t 中的 Date 的 时 间 设 置	Form.dateTimeEdit.date=new Date()
dateTime	QDateTime	选 中 日 期 时 间，参 考 JavaScript t 中的 Date 的 时 间 设 置	Form.dateTimeEdit.dateTime=new Date()
displayFormat	QString	格 式 化 时 间 格 式	Form.dateTimeEdit.displayFormat="yyyy/M/d H:mm"
enabled	bool	插 件 有 效 状 态	Form.dateTimeEdit.enabled=true
focusPolicy	FocusPolicy	焦点策略， NoFocus=0 , TabFocus= 1 , ClickFocu s=2 , StrongFoc us=11 ,	Form.dateTimeEdit.focusPolicy=1

		WheelFocus=15	
font	QFont	字体	<pre> var fontMap={}; fontMap["weight"]=12; fontMap["pointSize"]=12; fontMap["family"]="宋体"; fontMap["italic"]=true; fontMap["underline"]=true; Form.dateTimeEdit.font = HProperty.setQFont(fontMap) </pre>
geometry	QRect	几何尺寸	<pre> var geoMap={}; geoMap["x"]=12; geoMap["y"]=12; geoMap["width"]=12; geoMap["height"]=12; Form.dateTimeEdit.geometry = HProperty.setQRect(geoMap); </pre>
maximumDate	QDateTime	最大日期, 参 考 JavaScript 中的 Date 的时间设置	<pre> Form.dateTimeEdit.maximumDate=new Date(7999, 11, 31); </pre>
minimumDate	QDateTime	最小日期, 参 考 JavaScript	<pre> Form.dateTimeEdit.minimumDate=new Date(1752, 8, 14); </pre>

		t 中的 Date 的 时 间 设 置	
popupCalendar	Bool	弹 出 日 历 选 择 模 式	Form.dateTimeEdit.popupCalendar=true
visible	bool	插 件 的 可 见 性	Form.dateTimeEdit.visible=false

6.5.6. 组件信号

selectDateTimeChanged（日期时间选择改变信号）

```
/**
 * @描述 时间选择器时间值改变信号
 * @参数 dateTime 日期时间格式字符串
 */
```

表示在组件日期时间发生改变时将发送此信号。

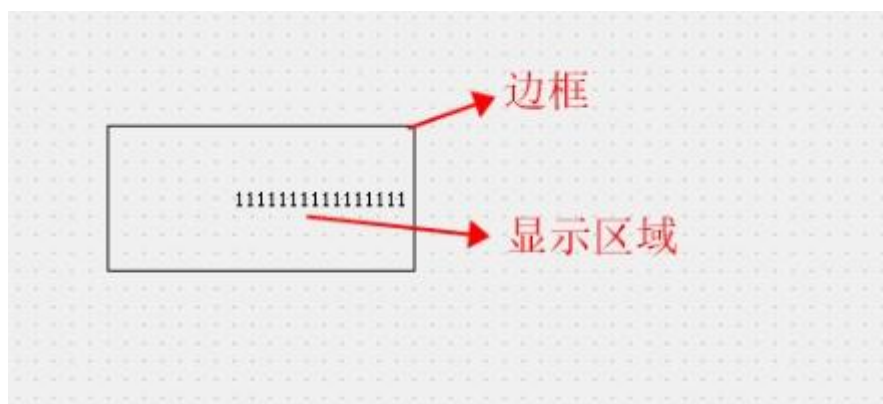
6.6. 输入区块

6.7. 数值

6.7.1. 功能介绍

可通过该组件输入数值到设定的地址，也可设定地址，读取并且将数值显示于在插件上。

6.7.2. 组件结构



数值组件构成简单，由边框和显示区域构成。

6.7.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“一般属性”、“外观形状”两页。

6.7.3.1. 一般属性



1. 一般属性

1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2) [焦点策略](#) ([跳转到聚焦策略说明](#))

焦点策略 (Focus Policy) 决定了部件 (Widget) 如何获取焦点和处理焦点事件。

3) 只读

勾选该选项，组件处于只读模式，用户无法编辑数值。

2. [软键盘](#)

使用触控的方式输入时，可以使用软键盘。（[跳转到软键盘说明](#)）

3. 监视的变数

1) 关联变数

设置插件关联的变数，可使用变数编辑弹窗输入变数。（[跳转到变数编辑弹窗的介绍](#)）

2) 数值单位

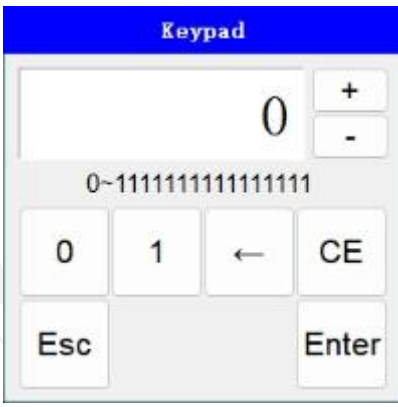
单位	长度
Word	字 (16Bit)
DWord	双字 (32Bit)

3) 数值类型

数值类型	说明
Floating	浮点型
Signed Decimal	有符号十进制
Unsigned Decimal	无符号十进制
Signed BCD	有符号 BCD
BCD	无符号 BCD
Hexadecimal	十六进制
Binary	二进制

以下是数值类型对应的软键盘类型。

数值类型	软键盘类型
Floating Signed Decimal Unsigned Decimal Signed BCD BCD	十进制软键盘

	
Hexadecimal	十六进制软键盘 
Binary	二进制软键盘 

- 4) 整数位数
设置整数位数。
- 5) 小数位数

数值类型使用浮点类型时，可以设置小数位数。

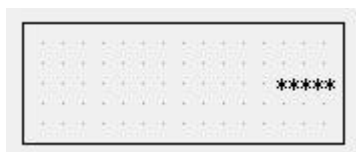
6) 最小/最大值

数值输入合法范围

数值输入		
Word	BCD	0~9999
	Signed BCD	-999~9999
	Signed Decimal	-32768~32767
	Unsigned Decimal	0~65535
	Hex	0~0xFFFF
	Binary	0~0xFFFF
DWord	BCD	0~99999999
	Signed BCD	-99999999~99999999
	Signed Decimal	-2147483648~2147483647
	Unsigned Decimal	0~4294967295
	Hex	0~0xFFFFFFFF
	Binary	0~0xFFFFFFFF

7) 显示为*号

将数值显示为*号



8) 前向补齐方式

补齐方式	具体
不补齐	不进行补齐
以 0 补齐	用 0 字符补全不足的整数位数。 
以空格补齐	用空格补全不足的整数位数。



9) 无效值启用

启用该项，输入空值时，将无效值“0xBBBBBB”写入到变数。

10) 可见位

关联变数，设置插件是否可见。

11) 位开可见

状态	具体
勾选	“可见位”位变数为 1 时，插件可见。否则隐藏。
不勾选	“可见位”位变数为 0 时，插件可见。否则隐藏。

12) 最大步长变数

最大步长关联变数。

输入方式	结果
软键盘	当数值变化超过最大步长时，当次输入无效。
输入区块（ 跳转到输入区块介绍 ）	当数值变化超过最大步长时，在“输入区块插件”显示“invalid inputs!(abs(输入值 - 当前值) > 步长)”，并且当次输入无效。

6.7.3.2. 外观形状



1. 框架

设置框架的外观形状。（[跳转到框架说明](#)）

2. 预览

预览外观形状的设置效果。

3. 字体

设置显示文本的字体。（[跳转到字体说明](#)）

4. 对齐

设置文本对齐策略。(跳转到[对齐说明](#))

5. 颜色

1) 自动填充背景颜色

当自动填充背景颜色功能启用时，插件将根据当前的颜色方案自动填充背景颜色。

2) 文本颜色

设置文本颜色。



3) 背景颜色

设置背景颜色。



(注：启用“自动填充背景颜色”选项，背景颜色生效)

4) 聚焦时颜色

设置插件拥有焦点时的背景颜色。

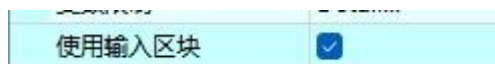
6. 工具提示

配合标签插件使用，设置聚焦和光标悬停在插件时，标签组件的“工具提示”功能显示的内容(跳转到[标签插件的“工具提示”功能说明](#))。

6.7.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。属性编辑器里大部分的属性设置，功能与属性设置弹窗一致。这里只介绍属性设置弹窗里面未介绍的功能。

1. 使用输入区块



勾选“使用输入区块”时，输入内容在“输入区块”插件中显示。点击“Enter”按钮后，再将内容写入到数值插件。（[跳转到输入区块插件的介绍](#)）



2. 操作权限

操作权限	0
------	---

设置组件的操作权限。如果权限不足，数值插件将无法编辑。（[跳转到权限说明](#)）

3. 操作记录

操作记录	☐
------	--------------------------

组件开启操作记录功能。

“关联变数”，“可见位”，“最大步长变数”发生变化时，将变化记录到操作记录中。（[跳转到操作记录说明](#)）

	时间	记录者	操作	详情
1	2023-08-29 16:58:30.738	Form_hNumber_2	变数改变	[0]USERD, 13->90
2	2023-08-29 16:56:50.382	System	0.8.1.254	初始化过程中出现错误或者超时，初始化停止
3	2023-08-29 16:56:50.333	System	0.8.1.3	获得从站数量过程中
4	2023-08-29 16:56:50.301	System	0.8.1.2	设定通讯周期过程中
5	2023-08-29 16:56:50.268	System	0.8.1.1	创建FRAME过程中
6	2023-08-29 16:56:50.199	System	-1.7.7	保存公共变数 COM (MCM) 完成
7	2023-08-29 16:56:50.198	System	2.6.1	通道2 控制器错误
8	2023-08-29 16:56:50.186	System	0.2.1	公共通道 Reset
9	2023-08-29 16:56:50.142	System	-1.7.2	保存-1 MCM变数完成
10	2023-08-29 16:56:50.114	System	-1.7.2	保存-1 MCM变数完成
11	2023-08-29 16:56:50.047	System	0.8.1.0	开始初始化
12	2023-08-29 16:56:50.046	System	0.3.0	切换到画面0
13	2023-08-29 16:56:50.025	System	1.2.1	通道1 Reset
14	2023-08-29 16:56:50.022	System	2.2.1	通道2 Reset

4. 操作记录名称

> 操作记录名称	Form_hNumber_2
----------	----------------

设置组件在操作记录表格中的显示名称。

6.7.5. 动作编辑器

动作编辑器	
事件	宏
值改变前	
编辑完成	

编辑过程中，数值改变前，触发该事件。

（注：“关联变数”的改变不会触发该事件）

1. 值改变前

2. 编辑完成

编辑过程中，数值编辑完成，触发该事件。

（注：“关联变数”的改变不会触发该事件）

6.7.6. 脚本宏

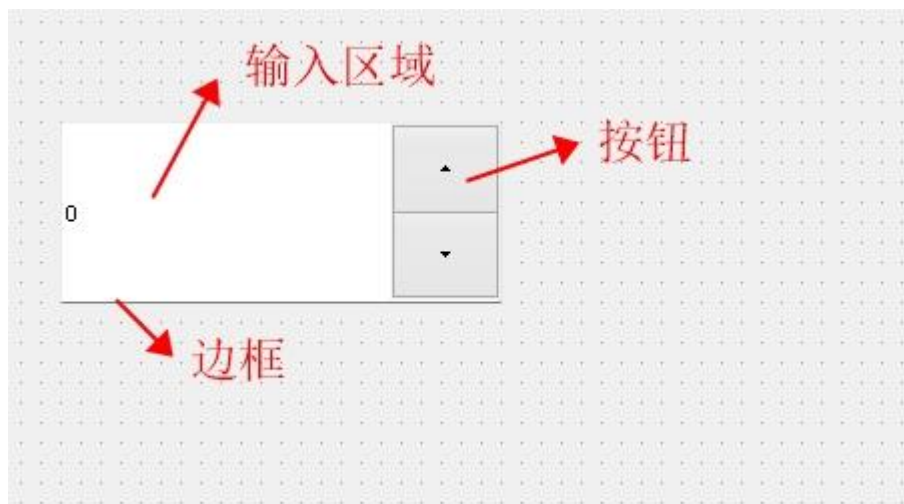
具体可见 macroWizard 文件夹的“Number_Designer_CN.xml”。

6.8. 整数、小数输入框

6.8.1. 功能介绍

1. 整数输入框提供了一个带有自增和自减按钮的整数选择框。用户可以通过点击按钮或手动输入来调整整数值。
2. 小数输入框提供了一个带有自增和自减按钮的小数选择框，用户可以通过点击按钮或手动输入来调整小数值。

6.8.2. 组件结构

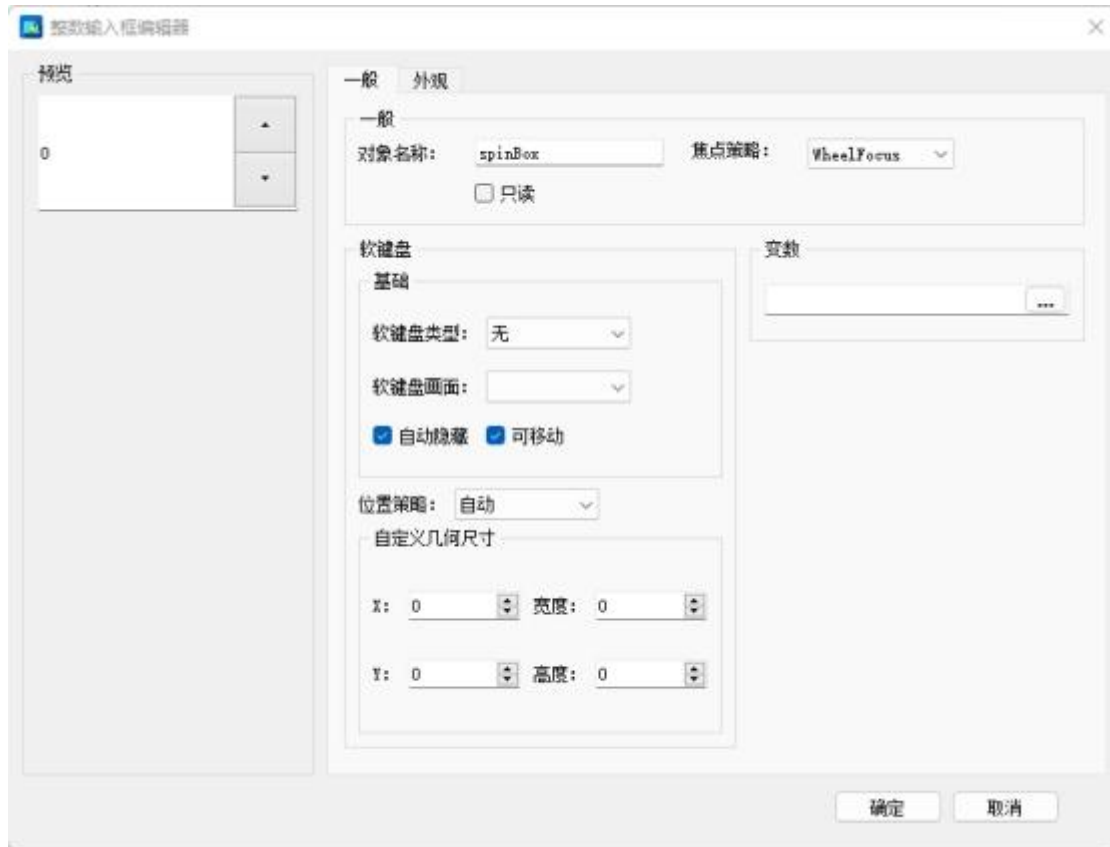


整数、小数输入框组件构成简单，由边框、输入区域、按钮构成。

6.8.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“一般属性”页。

6.8.3.1. 一般属性



1. 一般

1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2) [焦点策略](#) ([跳转到聚焦策略说明](#))

焦点策略 (Focus Policy) 决定了部件 (Widget) 如何获取焦点和处理焦点事件。

3) 只读

勾选该选项，组件处于只读模式，用户无法编辑整数值。

2. [软键盘](#)

使用触控的方式输入时，可以使用软键盘。([跳转到软键盘说明](#))

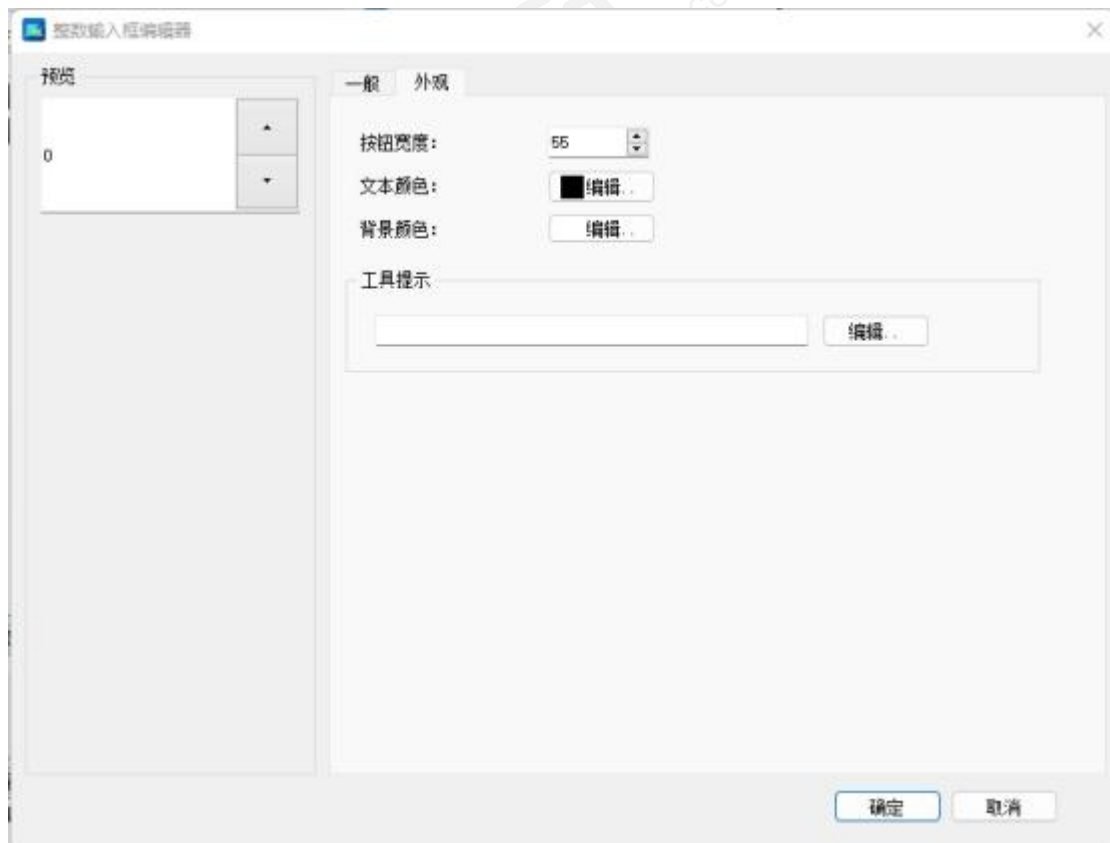
3. 变数

设置组件关联的变数，可使用变数编辑弹窗输入变数。（[跳转到变数编辑弹窗的介绍](#)）

关联变数后，显示的数值从对应地址中读取。

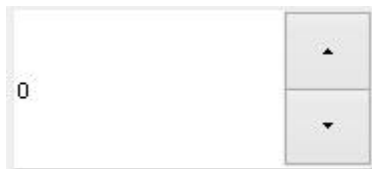
修改值后，变数的值也跟着改变。

6.8.3.2. 外观



1. 按钮宽度

设置加、减按钮的宽度。



2. 文本颜色

设置文本颜色



3. 背景颜色

设置背景颜色



4. 工具提示

设置聚焦和光标悬停在插件时，标签组件的“工具提示”功能显示的内容（[跳转到标签插件的“工具提示”功能说明](#)）。

6.8.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。属性编辑器里大部分的属性设置，功能与属性设置弹窗一致。这里只介绍属性设置弹窗里面未介绍的功能。

1. 整数输入框独有属性

1) 显示整数基数

displayIntegerBase	10
--------------------	----

整数基数定义了显示整数的进制。默认情况下，displayIntegerBase 的值为 10，即以十进制方式显示整数。选择整数基数 2、8、10 和 16，分别对应于二进制、八进制、十进制和十六进制。

2. 小数输入框独有属性

小数位数

小数位数	2
------	---

用于设置显示和输入的浮点数值的小数位。

3. 整数、小数输入框共有的属性

2) 是否循环

设置到达最大最小值时，继续增大或减小数值（使用鼠标滚轮或点击按钮），数值是否循环。

范例：最小值 0，最大值 100。当数值到达 100 时，继续增大数值，此时数值变为 0。

3) 边框

设置组件是否使用边框。

有边框：



无边框：



4) 按钮符号

按钮符号	NoButtons
------	-----------

选项	样式
UpDownArrows (上下箭头)	
PlusMinus (加减号)	

NoButtons (无按钮)	0
-----------------	--------------

5) 特别值文本

> 特别值文本

设置数值小于等于最小值时，显示的文本。

示例：设置特别值文本为“Invalid!”，最小值为0。当数值小于等于0，则显示特别值文本。

Invalid

6) 加速

加速 ☐

默认情况下，递增和递减速度是固定的，无论用户按下箭头键或鼠标滚轮多长时间，每次递增或递减的值都是相同的。

启用加速模式后，在用户连续按住箭头键或滚动鼠标滚轮时，在递增或递减的过程中加速递增或递减的值，从而更快地改变值。在大多数情况下，加速模式可以提高用户体验。

7) 千位分隔符

showGroupSepara... ☒

千位分隔符是用于将较大数字以可读的方式进行分组的符号，通常是逗号或空格。例如，将数字 1000000 显示为 “1,000,000”。

8) 后缀

> 后缀

后缀字符串是在显示值的末尾添加的额外文本。它可以用来表示单位、符号或任何其他相关信息。

9) 前缀

> 前缀	
------	--

前缀字符串是在显示值的开头添加的额外文本。它可以用来表示单位、符号或任何其他相关信息。

10) 最小值

最小值	0
-----	---

最小值定义了可以设置的最小有效值。用户无法通过直接输入或通过增减按钮将值设置为小于最小值的值。

11) 最大值

最大值	100
-----	-----

最大值定义了可以设置的最大有效值。用户无法通过直接输入或通过增减按钮将值设置为大于最大值的值。

12) 单击步长

单击步长	1
------	---

单步增量定义了用户通过单击上下箭头按钮或使用键盘上的上下箭头键时，值的增加或减少的幅度。

13) 步进类型

stepType	DefaultStepType
----------	-----------------

选项	说明
DefaultStepType	默认步进类型，使用单步增量（singleStep）
AdaptiveDecimalStepType	自适应十进制步进类型，在输入值时根据当前值的数量级自动调整步进大小。如果当前值为 1000，步进大小会调整为 100。如果当前值为 100，步进大小会调整为 10，这样

	可以确保用户在输入时能够更精确地调整值，并且步进大小与当前值的数量级相匹配。
--	--

14) 数值

数值	100
----	-----

设置当前值。

15) 按钮宽度

按钮宽度	38
------	----

调整按钮的宽度。

16) 操作权限

操作权限	0
------	---

设置组件的操作权限。如果权限不足，数值插件将无法编辑。（[跳转到权限说明](#)）

17) 使用

使用	☒
----	-------------------------------------

启用或禁用组件。禁用组件后，用户将无法与控件进行交互，并且不能更改其值。

18) 操作记录

操作记录	☐
------	--------------------------

组件开启操作记录功能。

关联的变数地址、值发生变化时，将变化记录到操作记录中。（[跳转到操作记录说明](#)）

19) 操作记录名称

操作记录名称	Form_spinBox
--------	--------------

设置组件在操作记录表格中的显示名称。

6.8.5. 动作编辑器

动作编辑器	
事件	宏
值改变	
编辑完成	

1. 值改变

值发生改变时触发该事件。

2. 编辑完成

编辑完成时发出。当用户完成编辑（例如按下 Enter 键或离开组件）时，该事件将被触发。

6.8.6. 脚本宏

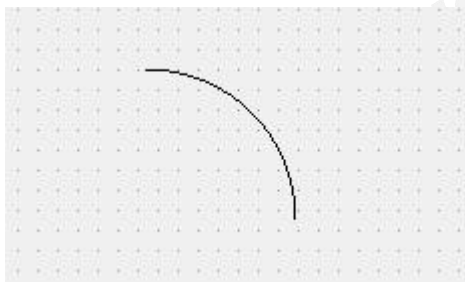
具体可见 macroWizard 文件夹的 “SpinBox_Designer_CN.xml”，
“DoubleSpinBox_Designer_CN.xml”。

6.9. 圆弧

6.9.1. 功能介绍

绘制一段圆弧。

6.9.2. 组件结构



圆弧组件仅由一圆弧组成。

6.9.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“外观形状”页。



1. 一般属性

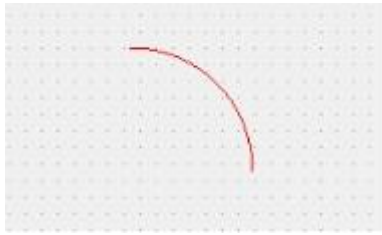
1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 边框

2) 线颜色

设置线颜色。



3) 线条宽度

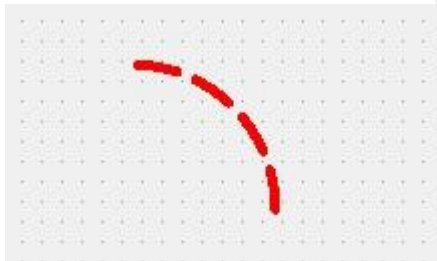
设置线条宽度。



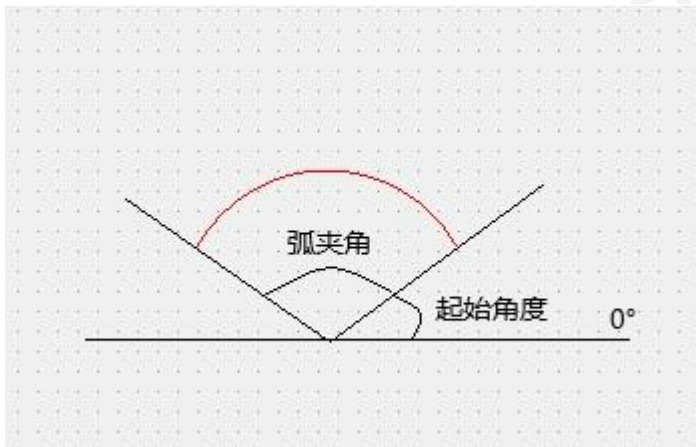
3. [线类型](#) ([跳转到线类型说明](#))

设置线型。

设置为虚线线型时的圆弧样式如下，



4. 起始角度、结束角度（弧夹角）



角度的正值表示逆时针方向，负值表示顺时针方向。零度在 3 点钟方位。

5. 圆弧形状

选项	说明
Arc (圆弧)	
Pie (饼)	
Chord (弦)	

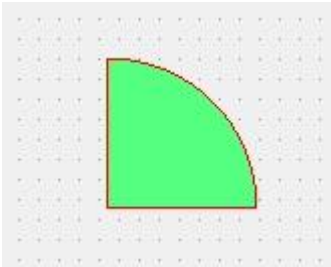
6. 预览

预览外观设置的效果。

7. 背景

1) 背景颜色

设置封闭图形的填充颜色。



(注：圆弧形状选择 Pie 或者 Chord 时，画刷的填充类型不等于“无画刷”背景设置生效。)

2) [填充类型](#)

设置画刷的风格。



（注：圆弧形状选择 Pie 或者 Chord 时，画刷的填充类型生效。）

8. [几何尺寸](#)

该属性控制组件在画面中的位置与大小。

6.9.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。圆弧组件属性编辑器里的属性设置，功能与属性设置弹窗一致。可参照属性设置弹窗了解每项功能。

6.9.5. 动作编辑器

圆弧组件无事件触发。

6.9.6. 脚本宏

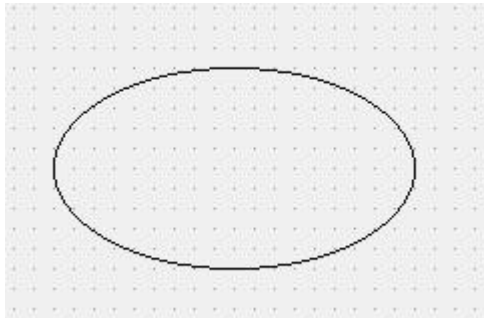
具体可见 macroWizard 文件夹的 “Arc_Designer_CN.xml”。

6.10. 椭圆

6.10.1. 功能介绍

绘制一个椭圆弧。

6.10.2. 组件结构



椭圆组件仅由一个椭圆组成。

6.10.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“外观形状”页。



1. 一般属性

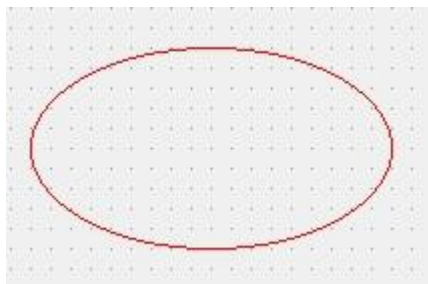
1) 对象名称

每个对象都可以通过设置对象名称（Object Name）来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 边框

2) 线颜色

设置线颜色。



3) 线条宽度

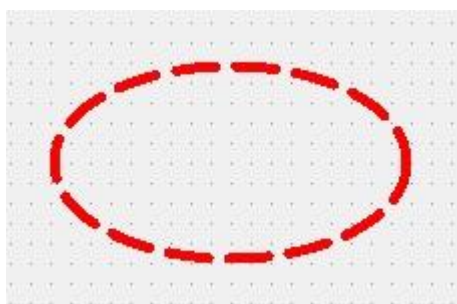
设置线条宽度。



3. [线类型](#)（[跳转到线类型说明](#)）

设置线型。

设置为虚线线型时的椭圆样式如下，



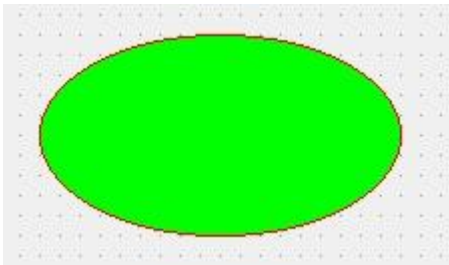
4. 预览

预览外观设置的效果。

5. 背景颜色

4) 背景颜色

设置椭圆组件的填充颜色。



（注：画刷的填充类型不等于“无画刷”时，背景颜色设置生效）

5) [填充类型](#)（[跳转到画刷风格说明](#)）

设置画刷的风格。

6. [几何尺寸](#)（[跳转到几何尺寸说明](#)）

该属性控制组件在画面中的位置与大小。

6.10.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。椭圆组件属性编辑器里的属性设置，功能与属性设置弹窗一致。可参照属性设置弹窗了解每项功能。

6.10.5. 动作编辑器

椭圆组件无事件触发。

6.10.6. 脚本宏

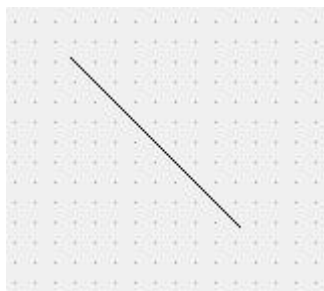
具体可见 macroWizard 文件夹的“Ellipse_Designer_CN.xml”。

6.11. 直线

6.11.1. 功能介绍

绘制一条直线。

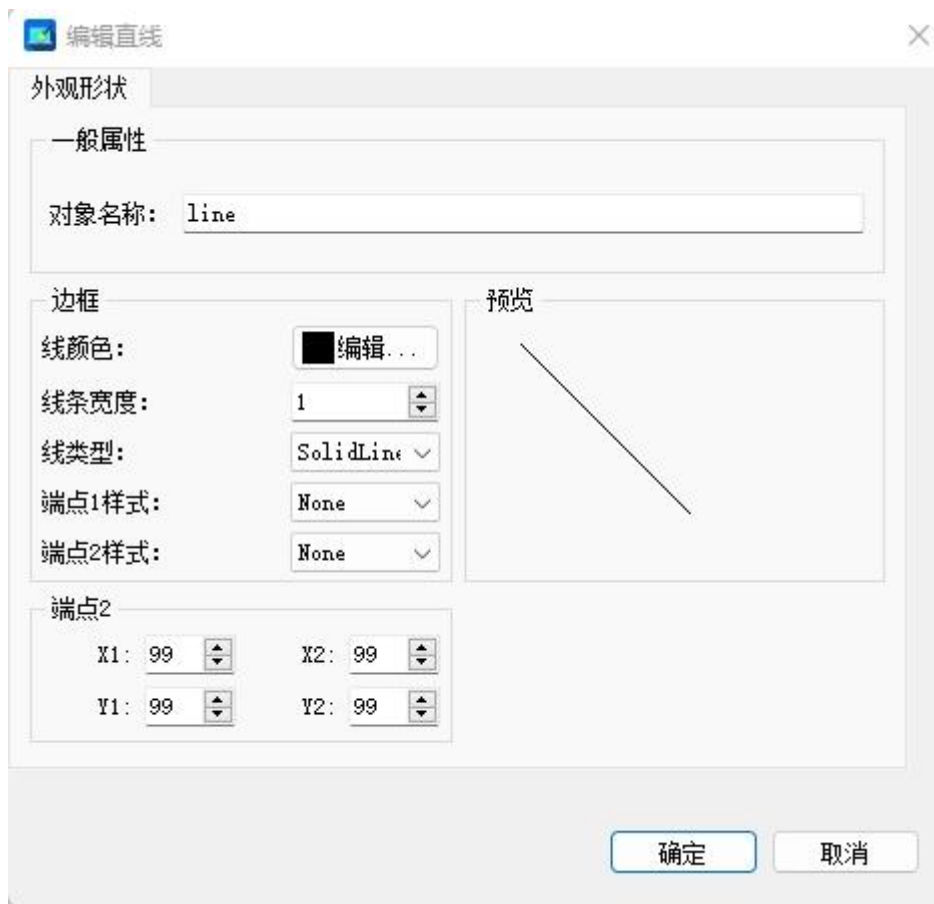
6.11.2. 组件结构



直线组件仅由一条直线组成。

6.11.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“外观形状”页。



1. 一般属性

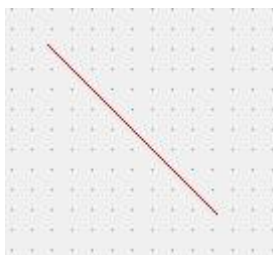
1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 边框

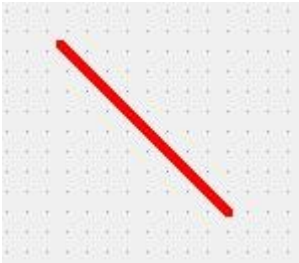
2) 线颜色

设置线颜色。



3) 线条宽度

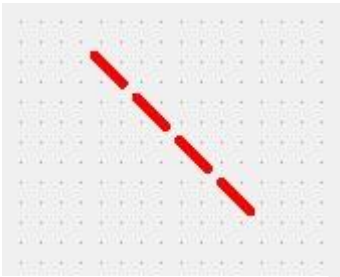
设置线条宽度。



4) [线类型](#) (跳转到线类型说明)

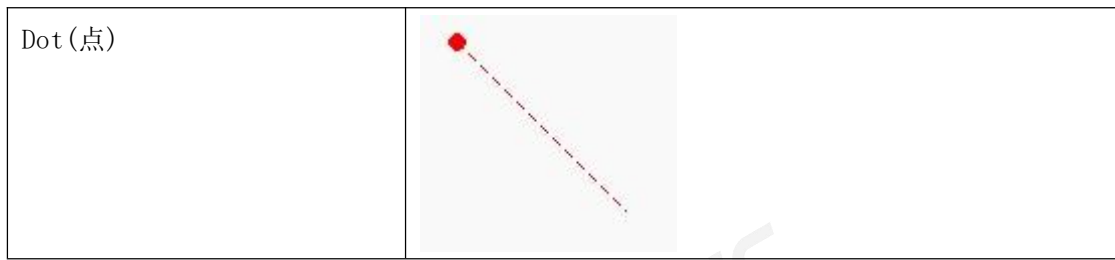
设置线型。

设置为虚线线型时的直线样式如下，



5) 端点样式

选项	样式
None	
Arrow(箭头)	



3. 预览

预览外观设置的效果。

4. 端点

设置直线的端点坐标。

6.11.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。椭圆组件属性编辑器里的属性设置，功能与属性设置弹窗一致。可参照属性设置弹窗了解每项功能。

6.11.5. 动作编辑器

直线组件无事件触发。

6.11.6. 脚本宏

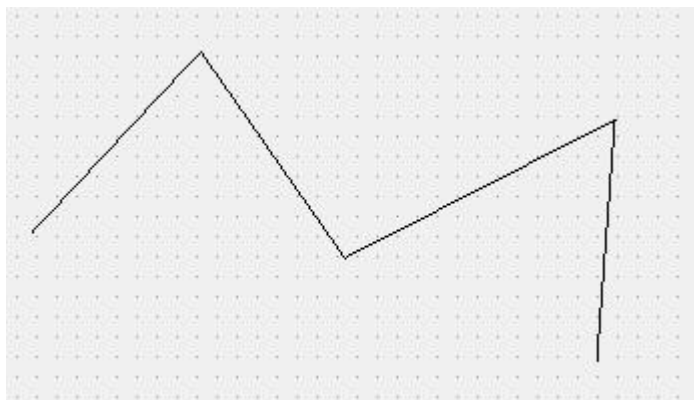
具体可见 macroWizard 文件夹的“Line2_Designer_CN.xml”。

6.12. 多边形

6.12.1. 功能介绍

绘制多边形。

6.12.2. 组件结构

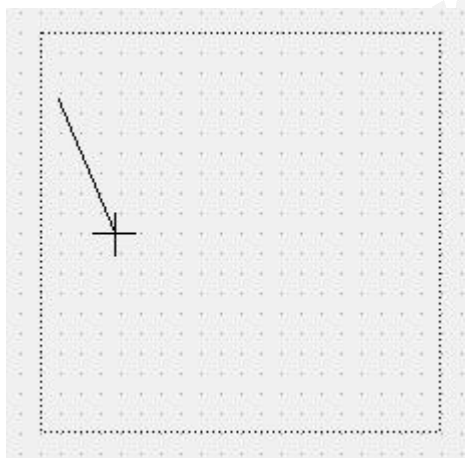


多边形组件仅由多段线组成。

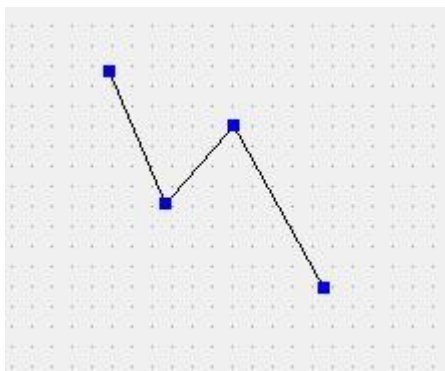
6.12.3. 编辑方式

在 HMI 编辑软件中，使用以下方式编辑多边形。

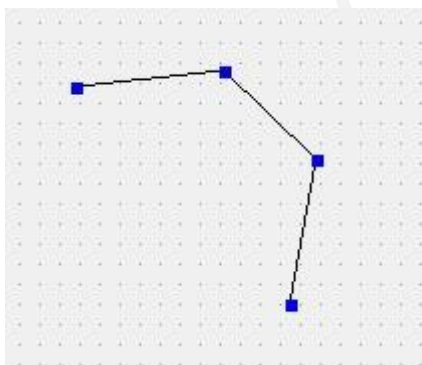
- 1) 在虚线矩形框区域内，使用鼠标左键单击，控制多边形的节点，双击鼠标左键完成多边形的绘制。



- 2) 完成绘制后，虚线矩形框消失。选中多边形时，以蓝色矩形的方式显示多边形的节点。

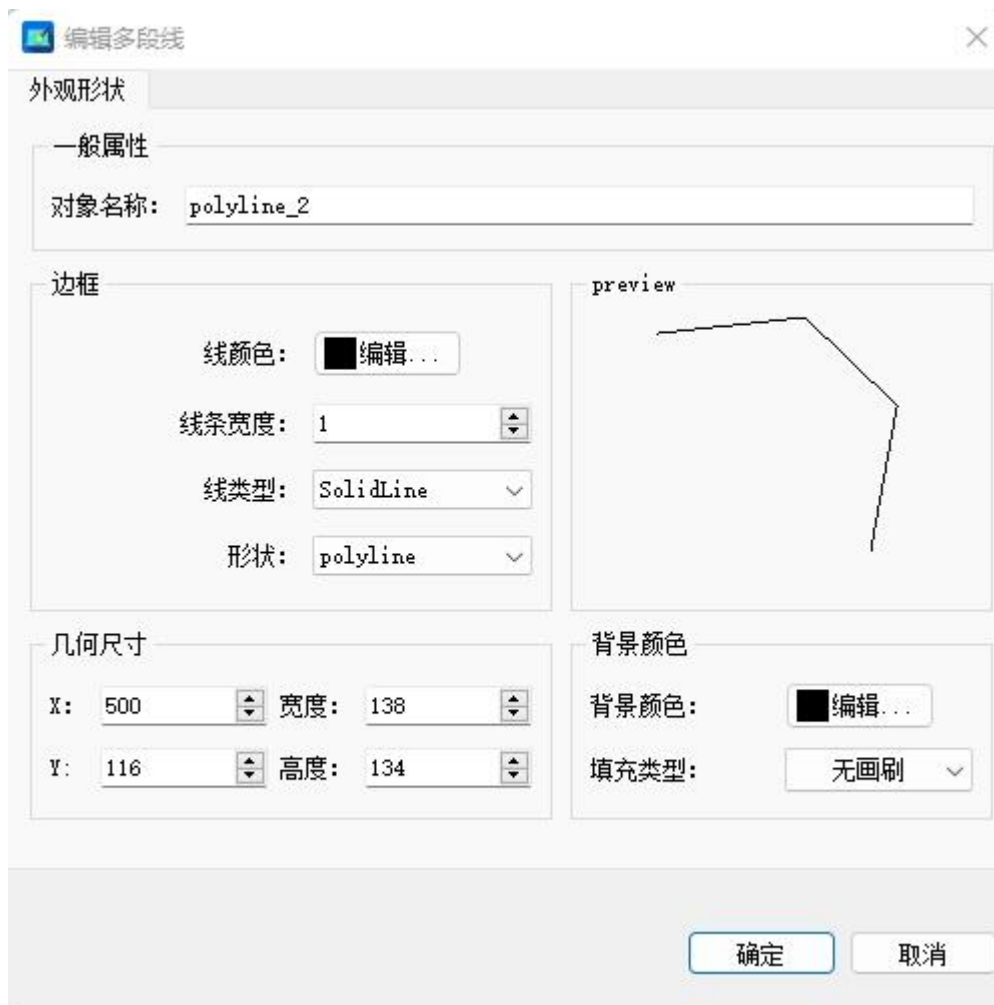


3) 可以通过拖拽多边形节点的方式，调整多边形。



6.12.4. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“外观形状”页。



1. 一般属性

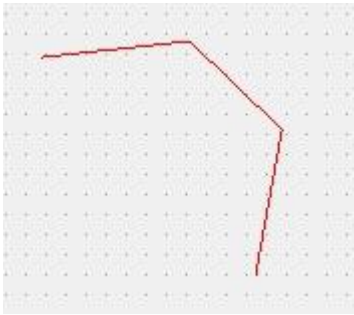
1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 边框

2) 线颜色

设置线颜色。



3) 线条宽度

设置线条宽度。



4) [线类型](#) (跳转到线类型说明)

设置线型。

设置为虚线线型时的多边形样式如下：



5) 形状

选项	样式
----	----

Polyline(多段线)	
Polygon(多边形)	

3. 预览

预览外观设置的效果。

4. [几何尺寸](#)(跳转到几何尺寸说明)

该属性控制组件在画面中的位置与大小。

5. 背景颜色

6) 背景颜色

设置多边形组件的填充颜色。



(注：“形状”选择为“polygon”，且画刷的填充类型不等于“无画刷”时，背景颜色设置生效)

7) [填充类型](#) (跳转到画刷风格说明)

设置画刷的风格。

6.12.5. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。多边形组件属性编辑器里的属性设置，功能与属性设置弹窗一致。可参照属性设置弹窗了解每项功能。

6.12.6. 动作编辑器

多边形组件无事件触发。

6.12.7. 脚本宏

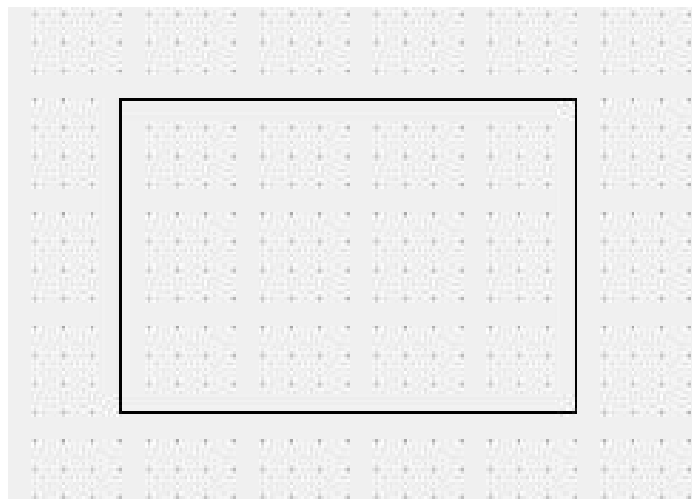
具体可见 macroWizard 文件夹的“Polyline_Designer_CN.xml”。

6.13. 矩形

6.13.1. 功能介绍

绘制一个矩形。

6.13.2. 组件结构



矩形组件仅由矩形框组成。

6.13.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“外观形状”页。



1. 一般属性

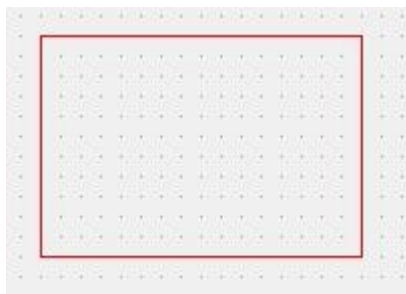
1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 边框

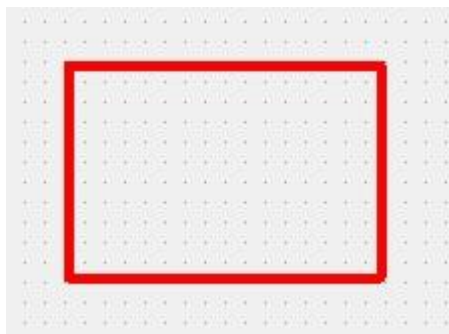
2) 线颜色

设置线颜色。



3) 线条宽度

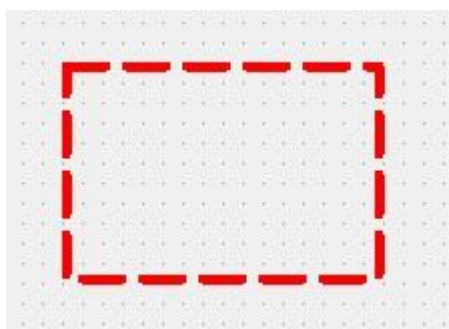
设置线条宽度。



4) 线类型 (跳转到线类型说明)

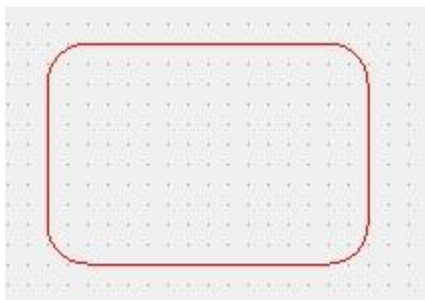
设置线型。

设置为虚线线型时的多边形样式如下，



5) 圆角半径

设置矩形的圆角半径



3. 预览

预览外观设置效果。

4. 几何尺寸

该属性控制组件在画面中的位置与大小。

5. 背景颜色

6) 背景颜色

设置矩形组件的填充颜色。



（注：画刷的填充类型不等于“无画刷”时，背景颜色设置生效）

7) 填充类型（[跳转到画刷风格说明](#)）

设置画刷的风格。

6.13.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。多边形组件属性编辑器里的属性设置，功能与属性设置弹窗一致。可参照属性设置弹窗了解每项功能。

6.13.5. 动作编辑器

矩形组件无事件触发。

6.13.6. 脚本宏

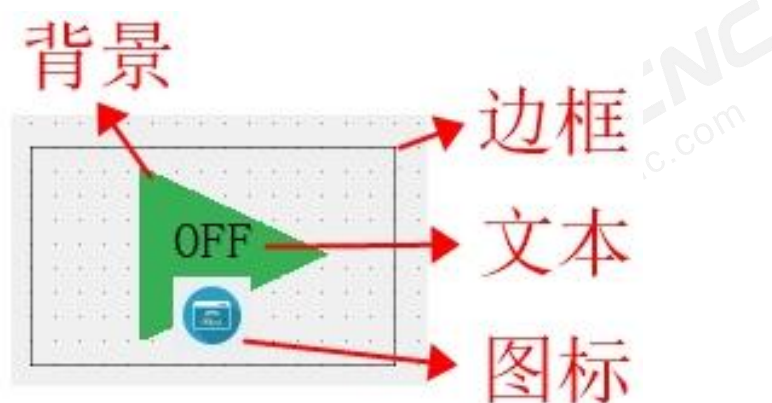
具体可见 macroWizard 文件夹的 “Rectangle_Designer_CN.xml”。

6.14. 位按钮

6.14.1. 功能介绍

- (1) 读写系统的位变数。
- (2) 通过触发“单击”、“鼠标按下”、“鼠标释放”等事件，执行脚本宏。

6.14.2. 组件结构

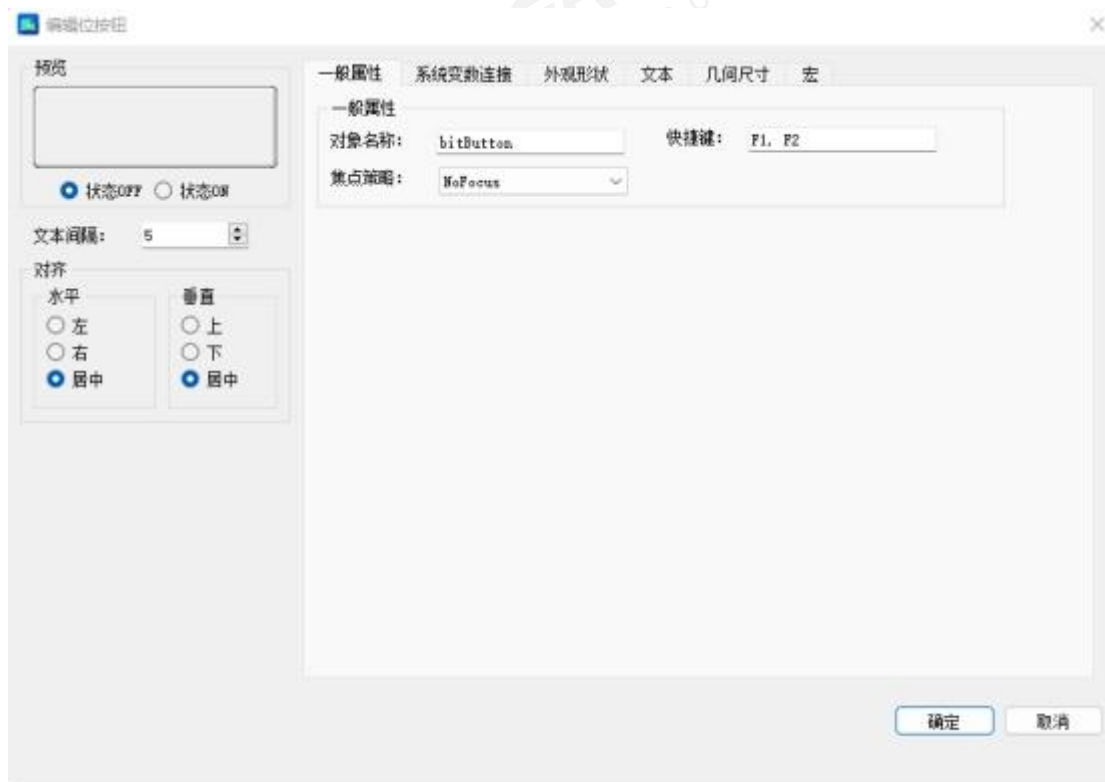


位按钮组件由“边框”、“文本”、“图标”、“背景”组成。

6.14.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“一般属性”、“系统变数连接”、“外观形状”、“文本”、“几何尺寸”、“宏”，共五页。以及“预览”，“文本间隔”，“对齐”三项设置。

【预览，文本间隔，对齐】



1. 预览

状态 OFF：预览按钮处于 OFF 状态时的外观。

状态 ON：预览按钮处于 ON 状态时的外观。

2. 文本间隔

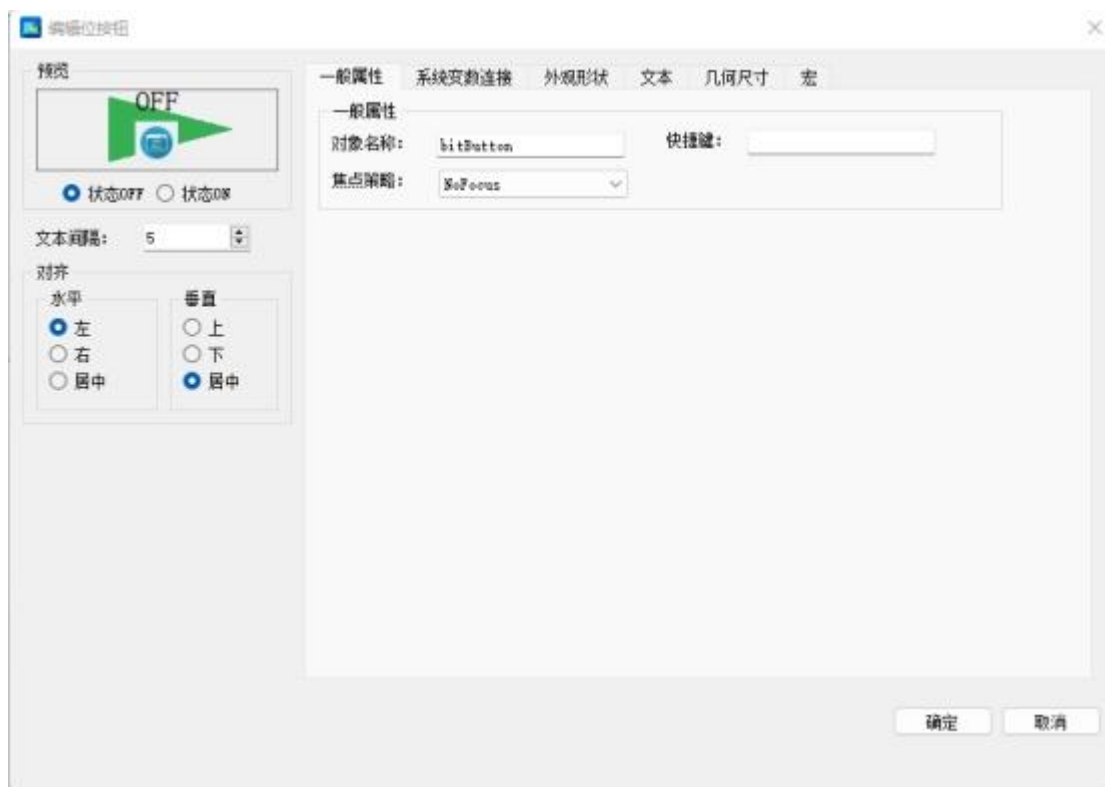
设置按钮图标与文本之间的间隔。

3. 对齐

设置文本对齐策略。详情可见“[对齐](#)”。需要注意的是，设置了图标后，文本对齐策略将失

效，而图标位置策略生效。

【一般属性】



1. 一般属性

1) 对象名称

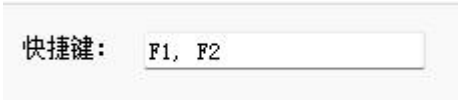
每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2) [焦点策略](#) ([跳转到聚焦策略说明](#))

焦点策略 (Focus Policy) 决定了部件 (Widget) 如何获取焦点和处理焦点事件。

3) 快捷键

快捷键在应用程序中的作用是提供一种快速访问和执行特定功能或操作的方式，而无需通过鼠标进行繁琐的操作。使用快捷键可以大大提高用户的工作效率和操作便捷性。



设置多个快捷键时，则多个快捷键都可触发该按钮。

在快捷键输入框处右键点击，会出现清空快捷键的选项



【系统变数连接】



位变数有两种类型。一种由地址类型+位地址构成。另一种由通道+地址类型+变数地址+位地址构成。具体分类如下：

位变数类型	地址类型
-------	------

地址类型+位地址	I_BIT, O_BIT, C_BIT, S_BIT, A_BIT, Cn_BIT, Tm_BIT 位变数范例: S79, 即地址类型+位地址
通道+地址类型+变数地址+ 位地址	User_BIT, Mcm_BIT, Sys_BIT, Reg_BIT, Bus_BIT, Com_BIT 位变数范例: [1]Usr1.0, 即[通道]地址类型+变数地址. 位 地址

1. 地址类型

设置变数地址的类型。None_BIT, I_BIT, O_BIT, C_BIT, S_BIT, A_BIT, Cn_BIT, Tm_BIT, User_BIT, Mcm_BIT, Sys_BIT, Reg_BIT, Bus_BIT, Com_BIT。注意选择 None_BIT 地址类型时, 表示不关联变数。

2. 变数地址

变数地址的填写需要注意当前选择的地址类型。

地址类型	读取变数地址
I_BIT, O_BIT, C_BIT, S_BIT, A_BIT, Cn_BIT, Tm_BIT	填写位地址 位变数范例: S79
User_BIT, Mcm_BIT, Sys_BIT, Reg_BIT, Bus_BIT, Com_BIT	填写变数地址 位变数范例: [1]Usr1.0, 即[通道]地址类型+变数地址. 位 地址。

3. 位编号

位编号的填写需要注意当前选择的地址类型。

地址类型	位编号
I_BIT, O_BIT, C_BIT, S_BIT, A_BIT, Cn_BIT, Tm_BIT	不需要填写位编号
User_BIT, Mcm_BIT,	填写变数地址的位编号

Sys_BIT, Reg_BIT, Bus_BIT, Com_BIT	位变数范例：[1]Usr1.0，即[通道]地址类型+变数地址.位地址。 此处的“位地址”即是“位编号”。
---------------------------------------	---

1. 通讯 ID

该属性为无效属性。

2. 通道

设置读写变数的通道号。

3. 读取变数地址

该属性可设置位变数关联按钮状态。根据位变数的开关，显示为按下，释放状态。

位变数状态	按钮状态
开（即位变数为 1）	按下 
关（即位变数为 0）	释放 

通过位变数改变按钮状态不会触发“宏”动作。

4. 写入地址

该属性可设置按钮状态关联位变数。根据按钮的“按下”，“释放”状态设置位变数数值。位变数的值与“动作”设置有关，详情可见“动作”。

5. 控制是否可用

该属性设置按钮是否使能。当组件处于禁用状态时，它会自动忽略用户的输入事件，例如点击事件、键盘事件等。同时，禁用的控件通常以视觉上的禁用样式显示，以便与启用的控件进行区分。

位变数	使能状态
开（即位变数为 1）	使能 
关（即位变数为 0）	 注意这里并不是“按下”，而是“未使能”效果。

6. 控制是否可见

该属性设置按钮是否可见。当一个控件设置为可见时，它将显示在用户界面上；而当一个控件设置为不可见时，它将被隐藏起来，用户将无法看到它。

1) 位开时可见

设置按钮可见状态与所关联位变数的关系。

位开时可见	位变数状态	可见状态
勾选	开（即位变数为 1）	可见
	关（即位变数为 0）	不可见
不勾选	开（即位变数为 1）	不可见
	关（即位变数为 0）	可见

7. 动作

该属性设置按钮按下时，关联写入地址的位变数（写入位）如何改变。

动作	位变数
设 1	按钮按下一次，写入位持续处于 on 状态。

设 0	按钮按下一次，写入位持续处于 off 状态。
按下 1	按钮按下时，写入位持续处于 on 状态。按钮释放时，写入位持续处于 off 状态。
按下 0	按钮按下时，写入位持续处于 off 状态。按钮释放时，写入位持续处于 on 状态。
取反	按钮按下时，如果写入位处于 on 状态，则改变写入位状态为 off; 如果写入位处于 off 状态, 则改变写入位状态为 on。

【外观形状】



1. 位开状态

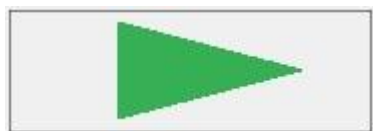
1) 按钮颜色

设置位按钮处于“按下”状态时的背景颜色。



2) 图片

设置位按钮处于“按下”状态时的背景图片。图片的添加方式以及清除方式可见“[图片资源](#)”说明。



“图片”属性会覆盖“按钮颜色”属性。

2. 位关状态

3) 按钮颜色

设置位按钮处于“释放”状态时的背景颜色。



4) 图片

设置位按钮处于“释放”状态时的背景图片。图片的添加方式以及清除方式可见“[图片资源](#)”说明。



“图片”属性会覆盖“按钮颜色”属性。

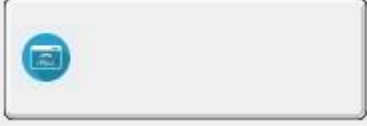
3. 图标

5) 图标

设置位按钮的图标。图片的添加方式以及清除方式可见“[图片资源](#)”说明。



6) 图标位置

图标位置	效果
上	
下	
左	
右	

7) 图标宽度

设置图标的宽度。

图标宽度	效果
16	
32	
48	

8) 图标高度

设置图标的高度。

图标高度	效果
16	
32	
48	

9) 图标间隔

设置图标与按钮边框的间隔。

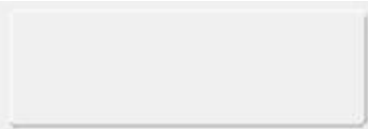
图标间隔	效果
5	
10	
20	

4. 边框

10) 边框有效

设置是否显示边框。

状态	效果
----	----

勾选	
不勾选	

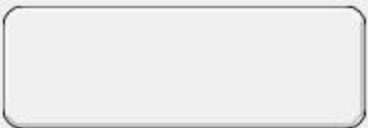
11) 边框颜色

设置边框颜色。

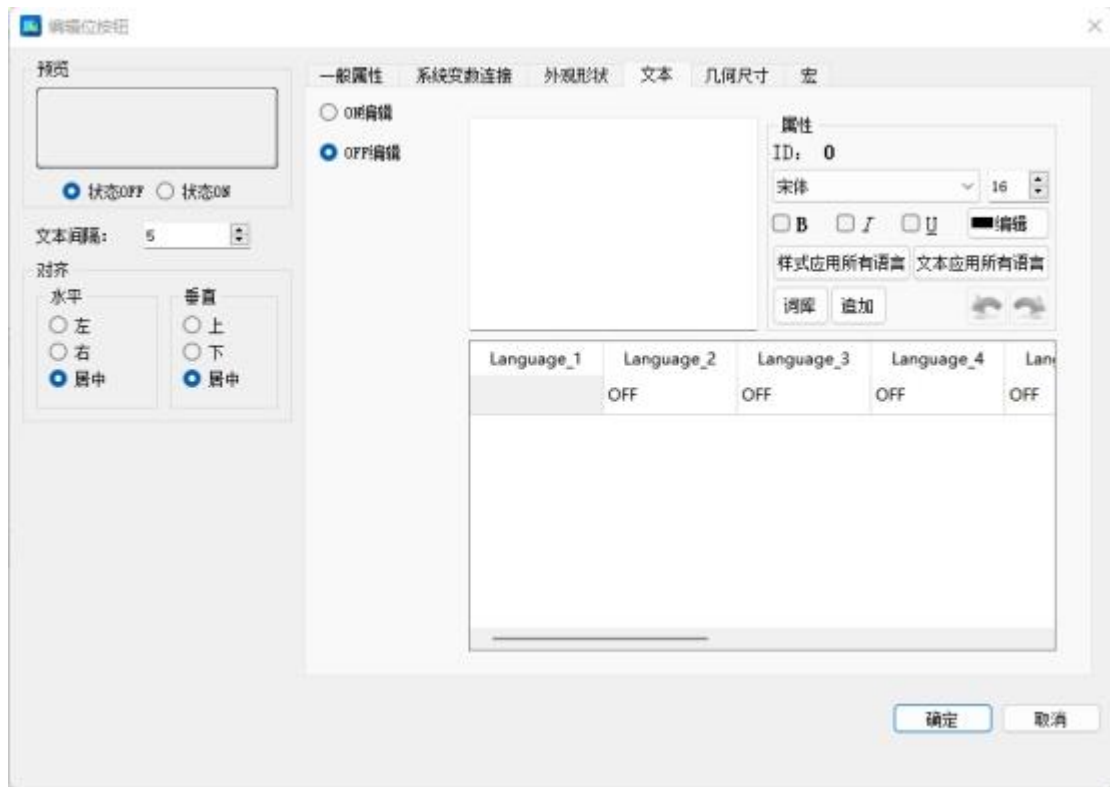
颜色	效果
RGB(255, 0, 0)	
RGB(0, 0, 0)	

12) 圆角半径

设置按钮的圆角半径。

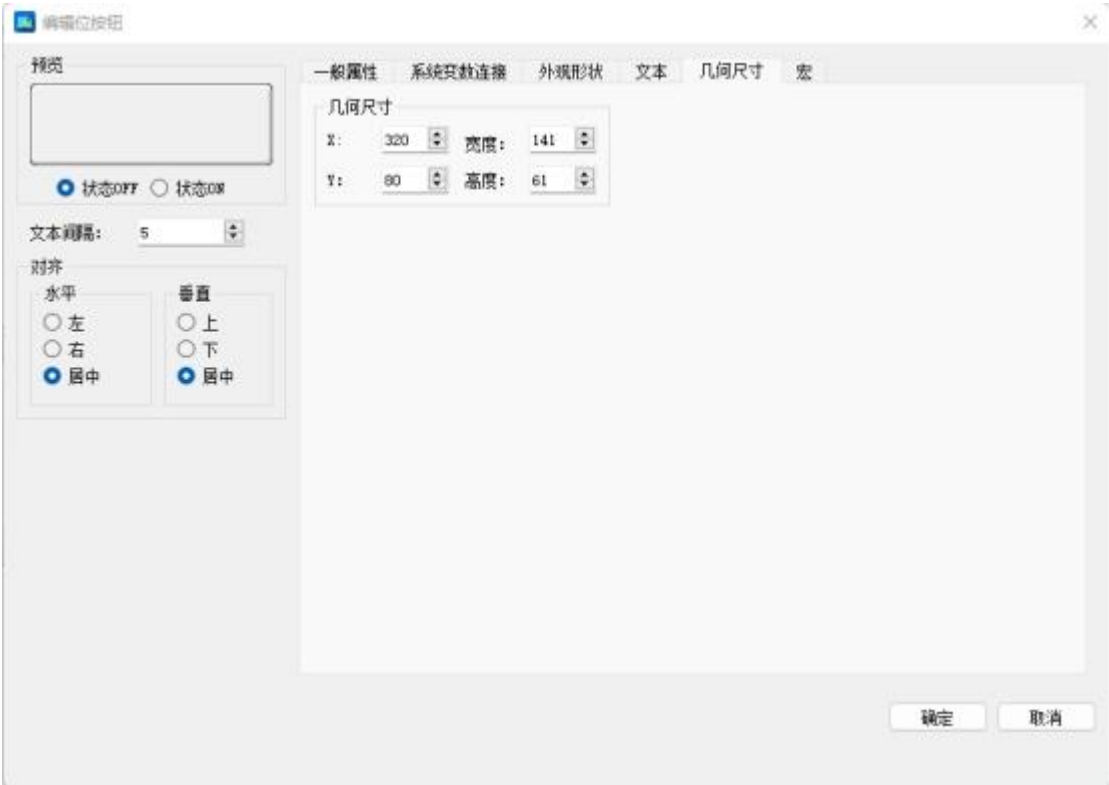
半径	效果
5	
10	
20	

【文本】



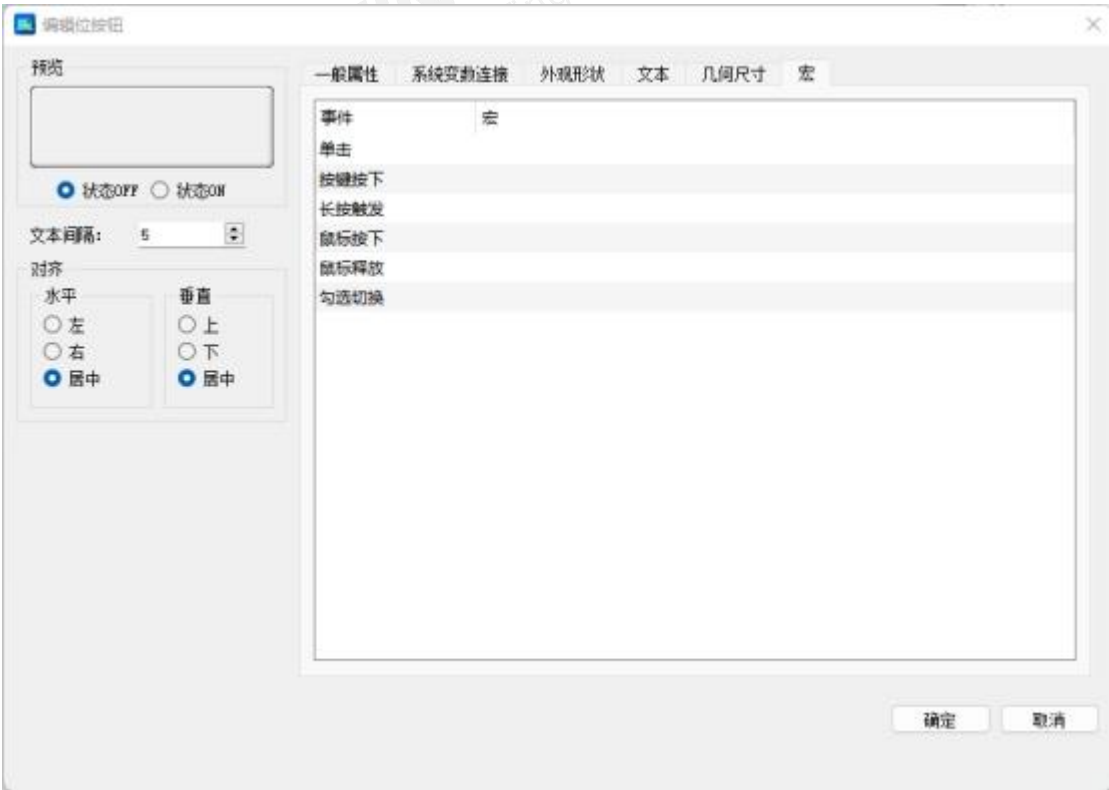
设置按钮的“按下”，“释放”两种状态的文本。且文本设置支持多国语言，详情可见“[多国语言编辑](#)”。

【几何尺寸】



该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

【宏】



事件	触发过程
单击 (MouseClicked)	使用鼠标或快捷键按下按钮，然后释放按钮。
按键按下	拥有焦点的情况下，使用快捷键按下按钮，再按任意按钮触发
长按触发	使用鼠标或快捷键按下按钮，并保持一段时间，配合延时触发功能一同使用。
鼠标按下 (MouseDown)	使用鼠标或快捷键按下按钮
鼠标释放 (MouseRelease)	使用鼠标或快捷键按下按钮，然后释放按钮
勾选切换 (Toggled)	按钮处于可勾选状态，切换按钮状态。

触发多个事件时，触发事件的顺序如下：

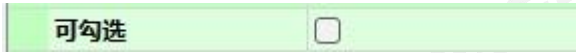
勾选	重复	操作	事件
不可勾选	不自动重复	单击	MouseDown MouseRelease MouseClicked
		长按	MouseDown MouseRelease MouseClicked
	自动重复	单击	MouseDown MouseRelease MouseClicked
		长按	MouseDown MouseRelease MouseClicked ... (循环)
可勾选	不自动重复	单击	MouseDown Toggled

			MouseRelease MouseClicked
		长按	MousePress Toggled MouseRelease MouseClicked
	自动重复	单击	MousePress Toggled MouseRelease MouseClicked
		长按	MousePress Toggled MouseRelease MouseClicked ...（循环）

6.14.4. 属性编辑器

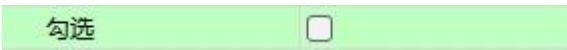
属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。属性编辑器里大部分的属性设置，功能与属性设置弹窗一致。这里只介绍属性设置弹窗里面未介绍的功能。

1. 可勾选



设置组件是否可勾选，当一个组件被设置为可勾选时，用户可以通过点击组件来切换其选中状态。

2. 勾选



此属性设置按钮勾选状态。开启“可勾选”后生效。

3. 自动排他

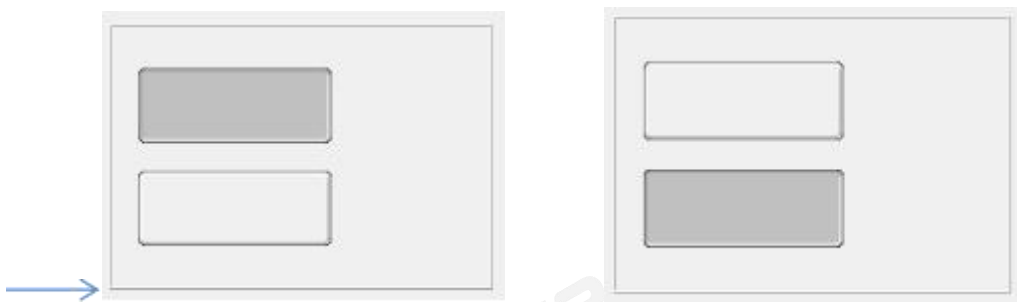
自动排他



如果启用了自动排他性，属于同一父项的可勾选按钮每次只能勾选一个按钮；勾选另一个按钮时，先前的按钮自动取消勾选。开启“可勾选”后生效。

自动排他效果示例：

两个按钮属于同一个父部件，按钮的“可勾选”属性已开启。按下其中一个按钮，然后按下另一个按钮，则先前的按钮自动取消勾选。



4. 自动重复

自动重复



此属性保存按钮在被按下和按住时是否重复发送 pressed、released 和 clicked 信号。延迟和重复间隔由 autoRepeatDelay 和 autoRepeatInterval 定义，以毫秒为单位。

5. 自动重复延时

自动重复延时

300

此属性以毫秒为单位设置自动重复的延时时间。即按下按钮，经过延时时间后，进入自动重复状态。

6. 自动重复间隔

自动重复延时

300

此属性以毫秒为单位设置自动重复的间隔。

7. 延时触发

延时触发	☐
------	--------------------------

此属性设置按钮释放时，是否触发“长按触发”动作。

8. 延时触发时间

延时触发时间	1000
--------	------

此属性以毫秒为单位设置按钮释放时，经过多久的延时时间，触发“长按触发”动作。

9. 操作权限

操作权限	0
------	---

设置组件的操作权限。如果权限不足，按钮组件将无法操作。[\(跳转到权限说明\)](#)

10. 使用

使用	☒
----	-------------------------------------

该属性设置按钮是否使能。当组件处于禁用状态时，它会自动忽略用户的输入事件，例如点击事件、键盘事件等。同时，禁用的控件通常以视觉上的禁用样式显示，以便与启用的控件进行区分。

11. 通道号变数控制

通道号变数控制	☐
---------	--------------------------

开启该选项后，则所有关联变数的通道号实际上为 Com40017 的值。

040017	I	I	W/R	Q	NULL	I	0~2^32	*当前通道号, 用于多通道的画面切换使用*
--------	---	---	-----	---	------	---	--------	-----------------------

12. 操作记录

操作记录	☐
------	--------------------------

组件开启操作记录功能。

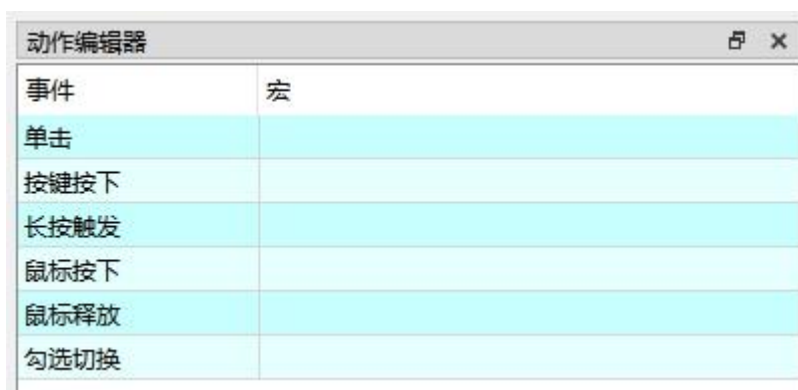
将位按钮的“按下”、“释放”动作记录到操作记录中。（[跳转到操作记录说明](#)）

13. 操作记录名称

> 操作记录名称 Form_bitButton

设置组件在操作记录表格中的显示名称。

6.14.5. 动作编辑器



详情可见“[6.14.3.7 宏](#)”的说明。

6.14.6. 脚本宏

具体可见 macroWizard 文件夹的“BitButton_Designer_CN.xml”。

6.15. 按钮组

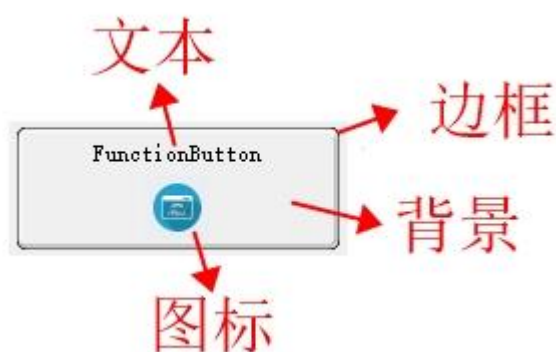
6.16. 复选框

6.17. 功能按钮

6.17.1. 功能介绍

1. 读写变数
2. 通过触发“单击”、“鼠标按下”、“鼠标释放”等事件，执行脚本宏。

6.17.2. 组件结构

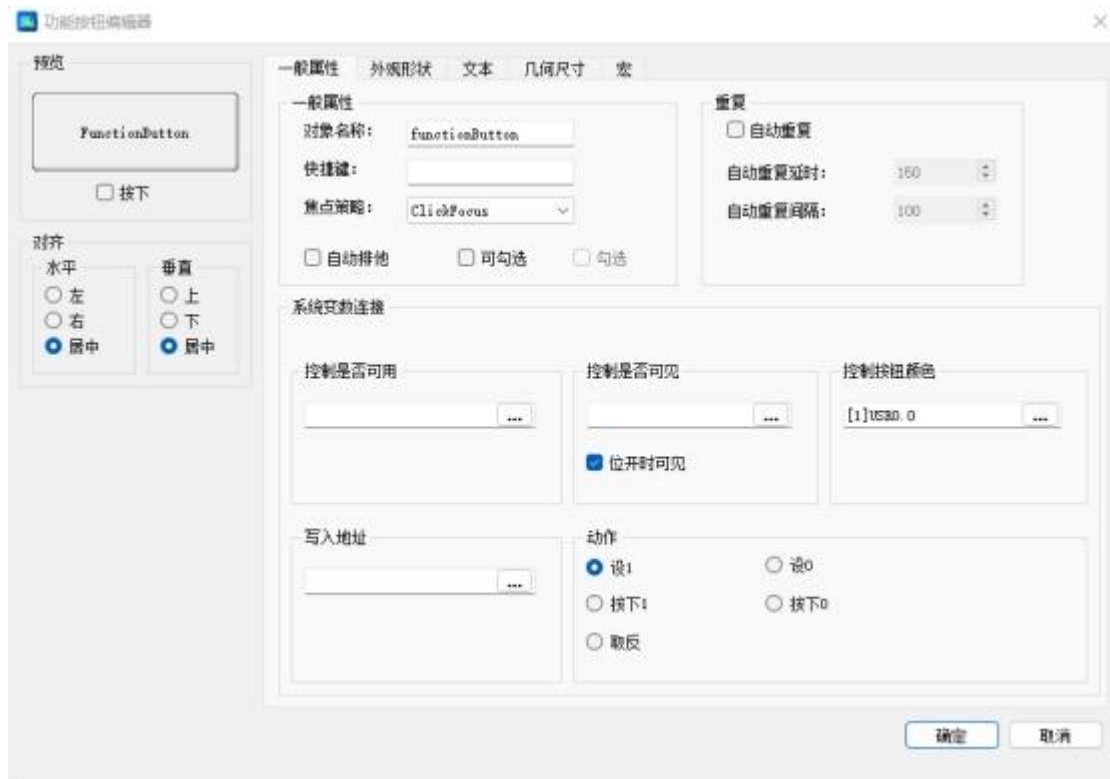


功能按钮组件由“边框”、“文本”、“图标”、“背景”组成。

6.17.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“一般属性”、“外观形状”、“文本”、“几何尺寸”、“宏”，共五页。以及“预览”，“对齐”两项设置。

6.17.3.1. 预览



1. 预览

1) 状态 OFF

预览按钮处于 OFF 状态时的外观。

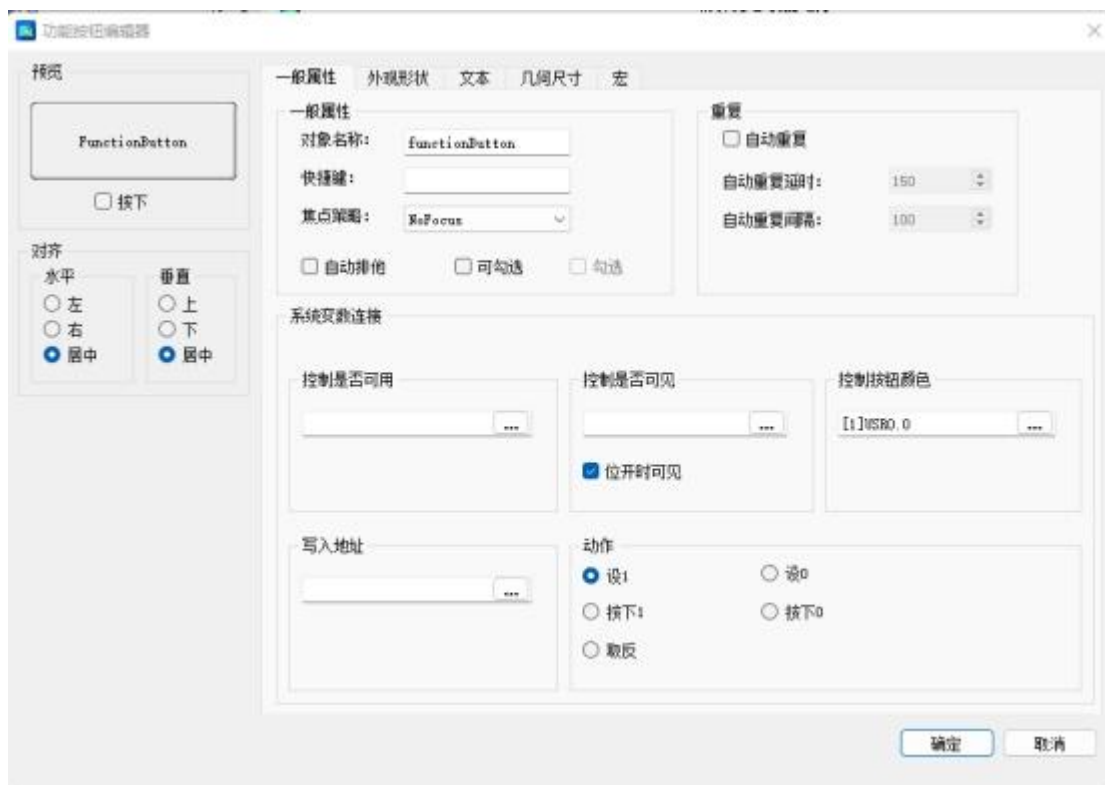
2) 状态 ON

预览按钮处于 ON 状态时的外观。

2. 对齐

设置文本对齐策略。详情可见“[对齐](#)”。需要注意的是，设置了图标后，文本对齐策略将失效，而图标位置策略生效。

6.17.3.2. 一般属性



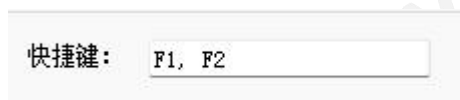
1. 一般属性

1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2) 快捷键

快捷键在应用程序中的作用是提供一种快速访问和执行特定功能或操作的方式，而无需通过鼠标进行繁琐的操作。使用快捷键可以大大提高用户的工作效率和操作便捷性。



设置多个快捷键时，则多个快捷键都可触发该按钮。

在快捷键输入框处右键点击，会出现清空快捷键的选项



3) [焦点策略](#) ([跳转到聚焦策略说明](#))

焦点策略 (Focus Policy) 决定了部件 (Widget) 如何获取焦点和处理焦点事件。

4) 自动排他

如果启用了自动排他性，属于同一父项的可勾选按钮每次只能勾选一个按钮；勾选另一个按钮时，先前的按钮自动取消勾选。开启“可勾选”后生效。

自动排他效果示例：

两个按钮属于同一个父部件，按钮的“可勾选”属性已开启。按下其中一个按钮，然后按下另一个按钮，则先前的按钮自动取消勾选。



5) 可勾选

设置组件是否可勾选，当一个组件被设置为可勾选时，用户可以通过点击组件来切换其选中状态。

6) 勾选

此属性设置按钮勾选状态。开启“可勾选”后生效。

2. 重复

7) 自动重复

此属性保存按钮在被按下和按住时是否重复发送 pressed、released 和 clicked 信号。延迟和重复间隔由“自动重复延时”和“自动重复间隔”定义，以毫秒为单位。

8) 自动重复延时

此属性以毫秒为单位设置自动重复的延时时间。即按下按钮，经过延时时间后，进入自动重复状态。

9) 自动重复间隔

此属性以毫秒为单位设置自动重复的间隔。

3. 系统变数连接

使用“[变数输入弹窗](#)”设置变数。

10) 控制是否可用

该属性设置按钮是否使能。当组件处于禁用状态时，它会自动忽略用户的输入事件，例如点击事件、键盘事件等。同时，禁用的控件通常以视觉上的禁用样式显示，以便与启用的控件进行区分。

位变数	使能状态
开（即位变数为 1）	使能 
关（即位变数为 0）	 注意这里并不是“按下”，而是“未使能”效果。

11) 控制是否可见

该属性设置按钮是否可见。当一个控件设置为可见时，它将显示在用户界面上；而当一个控件设置为不可见时，它将被隐藏起来，用户将无法看到它。

“位开时可见”：设置按钮可见状态与所关联位变数的关系。

位开时可见	位变数状态	可见状态
勾选	开（即位变数为 1）	可见
	关（即位变数为 0）	不可见
不勾选	开（即位变数为 1）	不可见
	关（即位变数为 0）	可见

12) 控制按钮颜色

该属性设置按钮文本颜色、背景颜色关联位变数。当位变数值位为 1 时，使用“位开时颜色”设置背景颜色和前景色。

13) 写入地址

该属性可设置按钮状态关联位变数。根据按钮的“按下”，“释放”状态设置位变数数值。位变数的值与“动作”设置有关，详情可见“动作”。

14) 动作

该属性设置按钮按下时，关联写入地址的位变数（写入位）如何改变。

动作	位变数
设 1	按钮按下一次，写入位持续处于 on 状态。
设 0	按钮按下一次，写入位持续处于 off 状态。
按下 1	按钮按下时，写入位持续处于 on 状态。按钮释放时，写入位持续处于 off 状态。
按下 0	按钮按下时，写入位持续处于 off 状态。按钮释放时，写入位持续处于 on 状态。
取反	按钮按下时，如果写入位处于 on 状态，则改变写入位状态

	为 off；如果写入位处于 off 状态，则改变写入位状态为 on。
--	------------------------------------

6.17.3.3. 外观形状



按钮存在多种状态下的背景颜色设置，相关背景颜色优先级如下：

优先级	颜色属性
1	按下时背景颜色
2	聚焦时背景颜色
3	位开时背景颜色
4	释放时背景颜色

按钮存在两种状态下的文本颜色设置，相关文本颜色优先级如下：

优先级	颜色属性
1	位开时文本颜色

2	文本颜色
---	------

1. 正常

1) 按钮颜色

该属性设置按钮处于“释放”状态下的背景颜色。



2) 位开时颜色

该属性与“控制按钮颜色”位变数相关。当位变数数值为 1 时，则按钮背景颜色更改为“位开时颜色”，否则按钮颜色为“按钮颜色”。

2. 按下

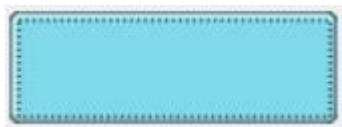
3) 按钮颜色

该属性设置按钮处于“按下”状态下的背景颜色。



3. 聚焦

该属性设置按钮处于“聚焦”状态下的背景颜色。



4. 文本样式

4) 文本间隔

该属性设置图标与文本之间的间隔。

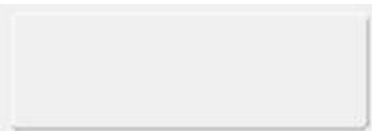
5) 位开时颜色

该属性与“控制按钮颜色”位变数相关。当位变数数值为 1 时，则按钮文本颜色更改为“位开时颜色”。

5. 边框

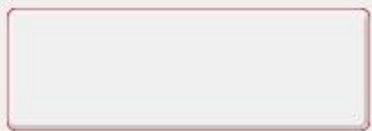
6) 边框有效

设置是否显示边框。

状态	效果
勾选	
不勾选	

7) 边框颜色

设置边框颜色。

颜色	效果
RGB (255, 0, 0)	
RGB (0, 0, 0)	

8) 圆角半径

设置按钮的圆角半径。

半径	效果
----	----

5	
10	
20	

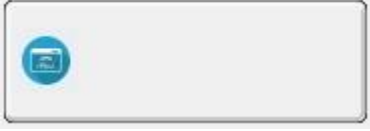
6. 图标

9) 图标

设置功能按钮的图标。图片的添加方式以及清除方式可见“[图片资源](#)”说明。



10) 图标位置

图标位置	效果
上	
下	
左	
右	

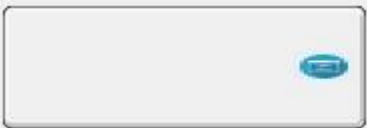
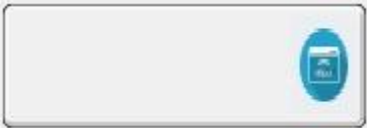
11) 图标宽度

设置图标的宽度。

图标宽度	效果
16	
32	
48	

12) 图标高度

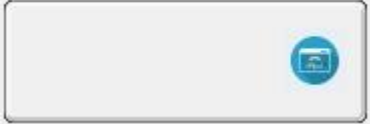
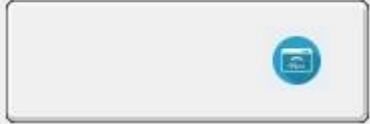
设置图标的高度。

图标高度	效果
16	
32	
48	

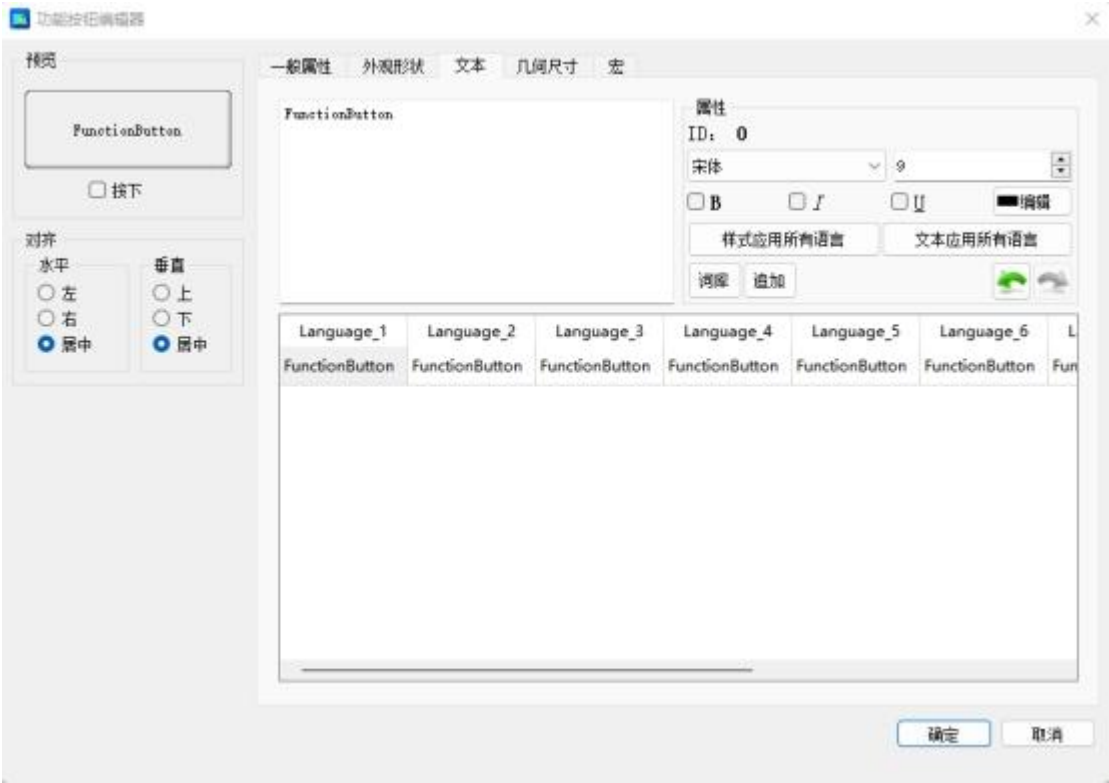
13) 图标间隔

设置图标与按钮边框的间隔。

图标间隔	效果
------	----

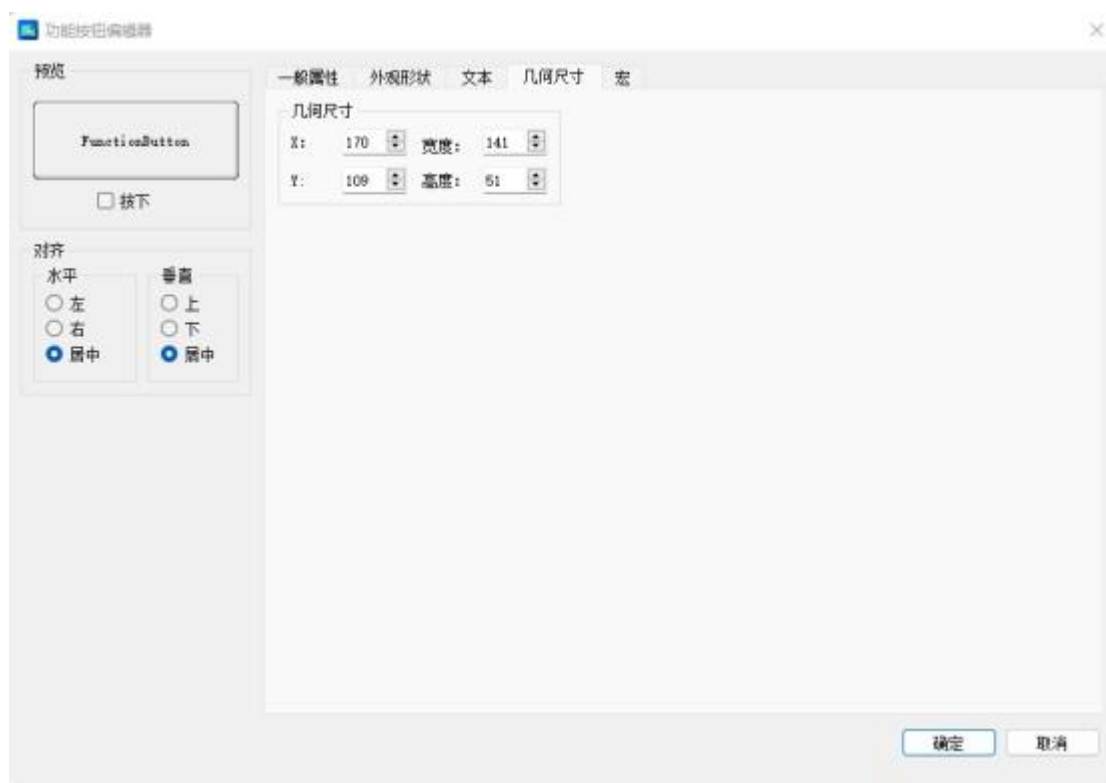
5	
10	
20	

6.17.3.4. 文本



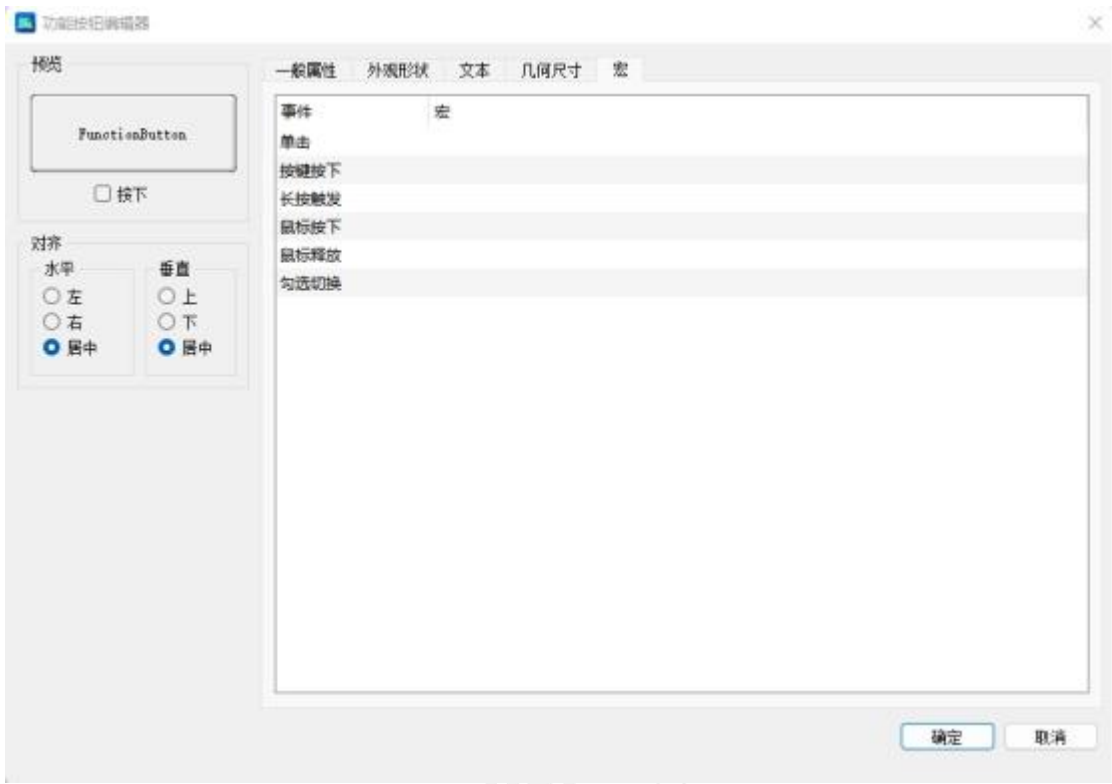
设置按钮的文本。且文本设置支持多国语言，详情可见“[多国语言编辑](#)”。

6.17.3.5. 几何尺寸



该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

6.17.3.6. 宏



事件	触发过程
单击 (MouseClicked)	使用鼠标或快捷键按下按钮，然后释放按钮。
按键按下	使用快捷键按下按钮
长按触发	使用鼠标或快捷键按下按钮，并保持一段时间，配合延时触发功能使用。
鼠标按下 (MousePress)	使用鼠标或快捷键按下按钮
鼠标释放 (MouseRelease)	使用鼠标或快捷键按下按钮，然后释放按钮
勾选切换 (Toggled)	按钮处于可勾选状态，切换按钮状态。

触发多个事件时，触发事件的顺序如下，

勾选	重复	操作	事件
不可勾选	不自动重复	单击	MousePress MouseRelease

			MouseClicked
		长按	MousePress MouseRelease MouseClicked
	自动重复	单击	MousePress MouseRelease MouseClicked
		长按	MousePress MouseRelease MouseClicked ... (循环)
可勾选	不自动重复	单击	MousePress Toggled MouseRelease MouseClicked
		长按	MousePress Toggled MouseRelease MouseClicked
	自动重复	单击	MousePress Toggled MouseRelease MouseClicked
		长按	MousePress Toggled MouseRelease MouseClicked ... (循环)

6.17.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。属性编辑器里大部分的属性设置，功能与属性设置弹窗一致。这里只介绍属性设置弹窗里未介绍的功能。

1. 延时触发

延时触发	☐
------	--------------------------

此属性设置按钮释放时，是否触发“长按触发”动作。

2. 延时触发时间

延时触发时间	1000
--------	------

此属性以毫秒为单位设置按钮释放时，经过多久的延时时间，触发“长按触发”动作。

3. 操作权限

操作权限	0
------	---

设置组件的操作权限。如果权限不足，按钮组件将无法操作。（[跳转到权限说明](#)）

4. 使用

使用	☒
----	-------------------------------------

该属性设置按钮是否使能。当组件处于禁用状态时，它会自动忽略用户的输入事件，例如点击事件、键盘事件等。同时，禁用的控件通常以视觉上的禁用样式显示，以便与启用的控件进行区分。

5. 操作记录

操作记录	☐
------	--------------------------

组件开启操作记录功能。

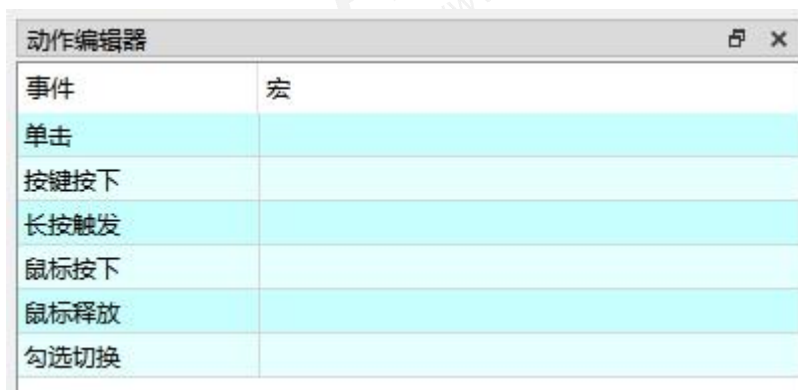
将位按钮的“按下”、“释放”动作记录到操作记录中。（[跳转到操作记录说明](#)）

6. 操作记录名称

> 操作记录名称	Form_functionButton
----------	---------------------

设置组件在操作记录表格中的显示名称。

6.17.5. 动作编辑器



详情可见“[6.17.3.6 宏](#)”的说明。

6.17.6. 脚本宏

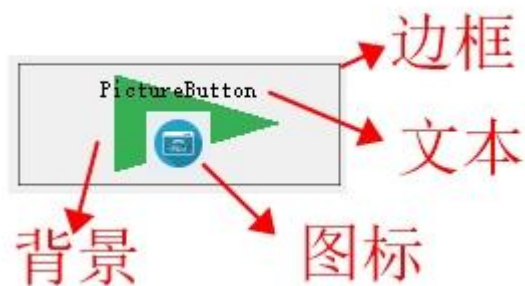
详情可见 macroWizard 文件夹的“FunctionButton_Designer_CN.xml”。

6.18. 图片按钮

6.18.1. 功能介绍

1. 通过触发“单击”、“鼠标按下”、“鼠标释放”等事件，执行脚本宏。
2. 作为软键盘的功能键使用。

6.18.2. 组件结构

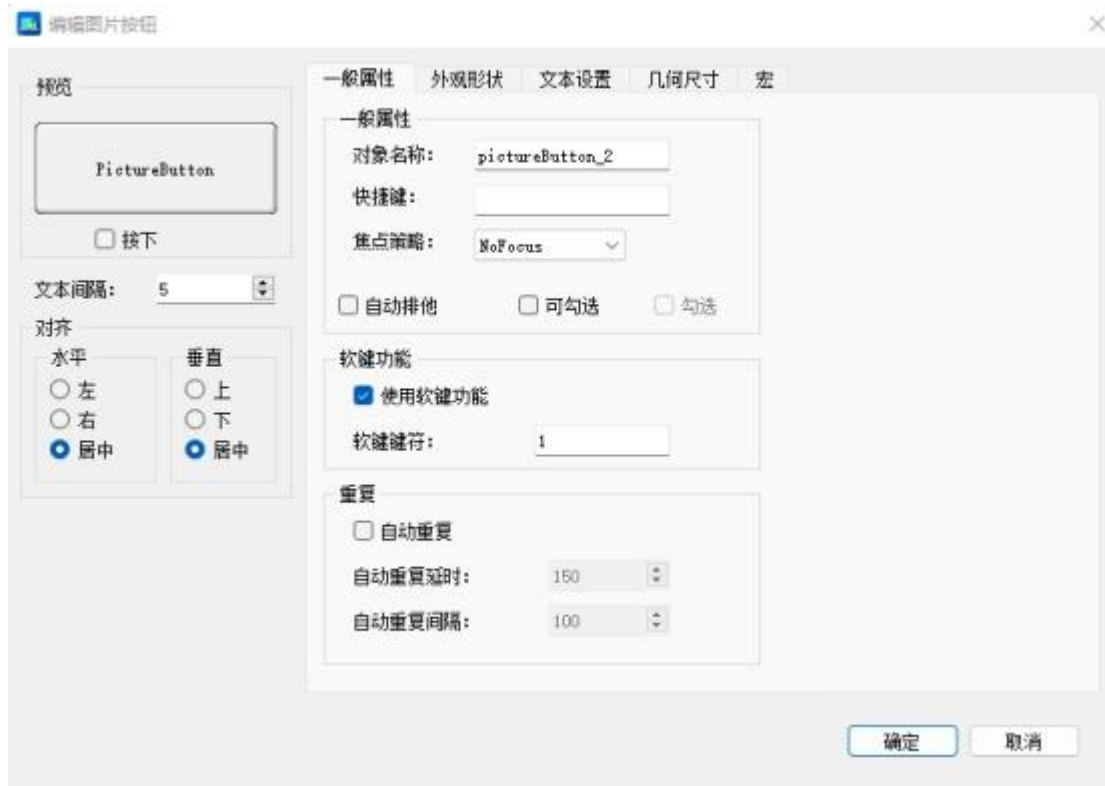


图片按钮组件由“边框”、“文本”、“图标”、“背景”组成。

6.18.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“一般属性”、“外观形状”、“文本设置”、“几何尺寸”、“宏”，共五页。以及“预览”，“文本间隔”，“对齐”三项设置。

6.18.3.1. 预览



1. 预览

1) 状态 OFF

预览按钮处于 OFF 状态时的外观。

2) 状态 ON

预览按钮处于 ON 状态时的外观。

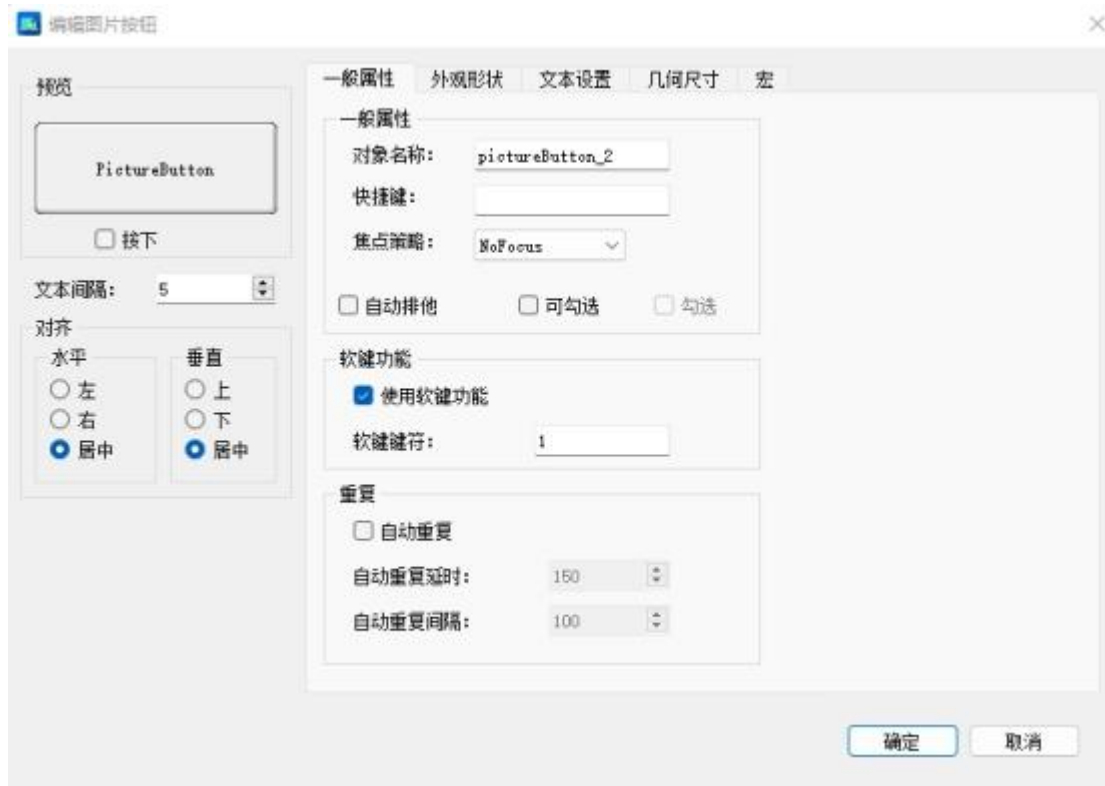
2. 文本间隔

设置按钮图标与文本之间的间隔。

3. 对齐

设置文本对齐策略。详情可见“[对齐](#)”。需要注意的是，设置了图标后，文本对齐策略将失效，而图标位置策略生效。

6.18.3.2. 一般属性



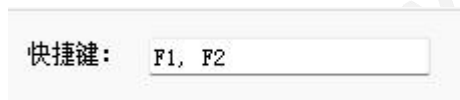
1. 一般属性

1) 对象名称

每个对象都可以通过设置对象名称（Object Name）来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2) 快捷键

快捷键在应用程序中的作用是提供一种快速访问和执行特定功能或操作的方式，而无需通过鼠标进行繁琐的操作。使用快捷键可以大大提高用户的工作效率和操作便捷性。



设置多个快捷键时，则多个快捷键都可触发该按钮。

在快捷键输入框处右键点击，会出现清空快捷键的选项



3) [焦点策略](#) ([跳转到聚焦策略说明](#))

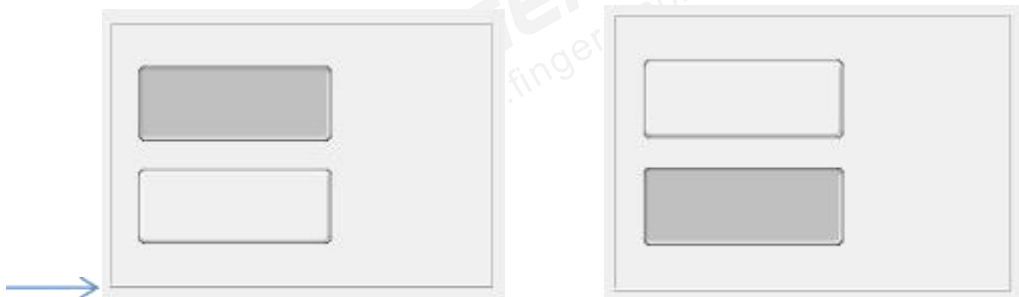
焦点策略 (Focus Policy) 决定了部件 (Widget) 如何获取焦点和处理焦点事件。

4) 自动排他

如果启用了自动排他性，属于同一父项的可勾选按钮每次只能勾选一个按钮；勾选另一个按钮时，先前的按钮自动取消勾选。开启“可勾选”后生效。

自动排他效果示例：

两个按钮属于同一个父部件，按钮的“可勾选”属性已开启。按下其中一个按钮，然后按下另一个按钮，则先前的按钮自动取消勾选。



5) 可勾选

设置组件是否可勾选，当一个组件被设置为可勾选时，用户可以通过点击组件来切换其选中状态。

6) 勾选

此属性设置按钮勾选状态。开启“可勾选”后生效。

2. 软键功能

图片按钮可以作为软键盘的一个功能按键使用。

如下范例所示，开启各图片按钮的“使用软键功能”，并且设置“软件键符”依次为“1”“2”“3”“4”“5”“6”“7”“8”“9”。则运行时点击按钮，会将“软键键符”输入到当前焦点部件（文本框）。



7) 使用软键功能

该属性设置图片按钮是否启用“软键功能”。

8) 软键盘符

该属性设置图片按钮启用“软件功能”时，对应的 Ascii 字符。

3. 重复

9) 自动重复

此属性保存按钮在被按下和按住时是否重复发送 pressed、released 和 clicked 信号。延迟和重复间隔由“自动重复延时”和“自动重复间隔”定义，以毫秒为单位。

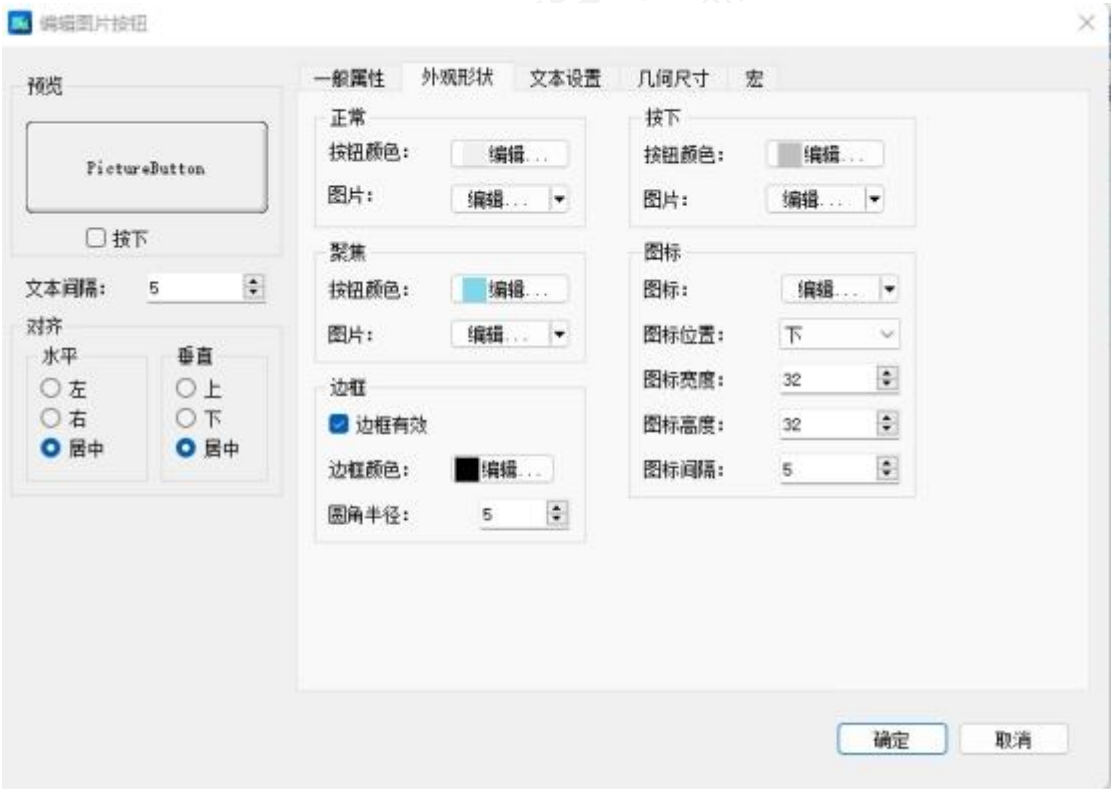
10) 自动重复延时

此属性以毫秒为单位设置自动重复的延时时间。即按下按钮，经过延时时间后，进入自动重复状态。

11) 自动重复间隔

此属性以毫秒为单位设置自动重复的间隔。

6.18.3.3. 外观形状



按钮存在多种状态下的背景颜色设置，相关背景颜色优先级如下，

优先级	颜色属性
1	按下时背景颜色
2	聚焦时背景颜色
3	释放时背景颜色

按钮存在多种状态下的背景图片设置，相关背景图片优先级如下，

优先级	颜色属性
1	按下时背景图片

2	聚焦时背景图片
3	释放时背景图片

1. 正常

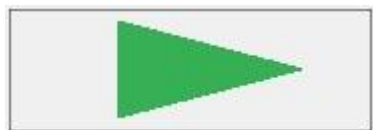
1) 按钮颜色

该属性设置按钮处于“释放”状态下的背景颜色。



2) 图片

设置位按钮处于“释放”状态时的背景图片。图片的添加方式以及清除方式可见“[图片资源](#)”说明。



“图片”属性会覆盖“按钮颜色”属性。

2. 按下

3) 按钮颜色

该属性设置按钮处于“按下”状态下的背景颜色。



4) 图片

该属性设置按钮处于“按下”状态下的背景图片



“图片”属性会覆盖“按钮颜色”属性。

3. 聚焦

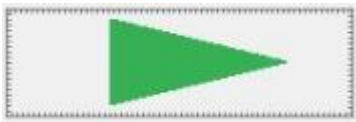
5) 按钮颜色

该属性设置按钮处于“聚焦”状态下的背景颜色。



6) 图片

该属性设置按钮处于“聚焦”状态下的背景图片。



“图片”属性会覆盖“按钮颜色”属性。

4. 图标

7) 图标

设置图片按钮的图标。图片的添加方式以及清除方式可见“[图片资源](#)”说明。



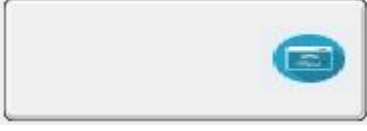
8) 图标位置

图标位置	效果
上	
下	

左	
右	

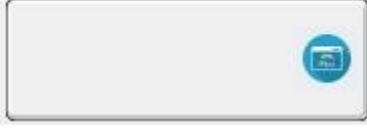
9) 图标宽度

设置图标的宽度。

图标宽度	效果
16	
32	
48	

10) 图标高度

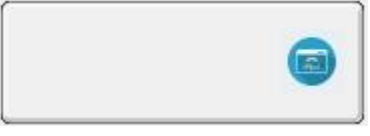
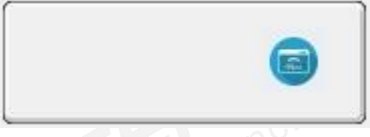
设置图标的高度。

图标高度	效果
16	
32	

48	
----	---

11) 图标间隔

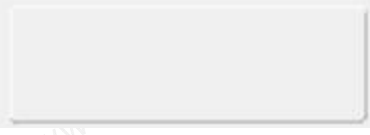
设置图标与按钮边框的间隔。

图标间隔	效果
5	
10	
20	

5. 边框

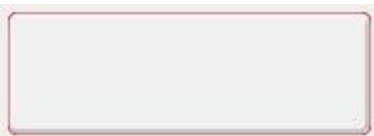
12) 边框有效

设置是否显示边框。

状态	效果
勾选	
不勾选	

13) 边框颜色

设置边框颜色。

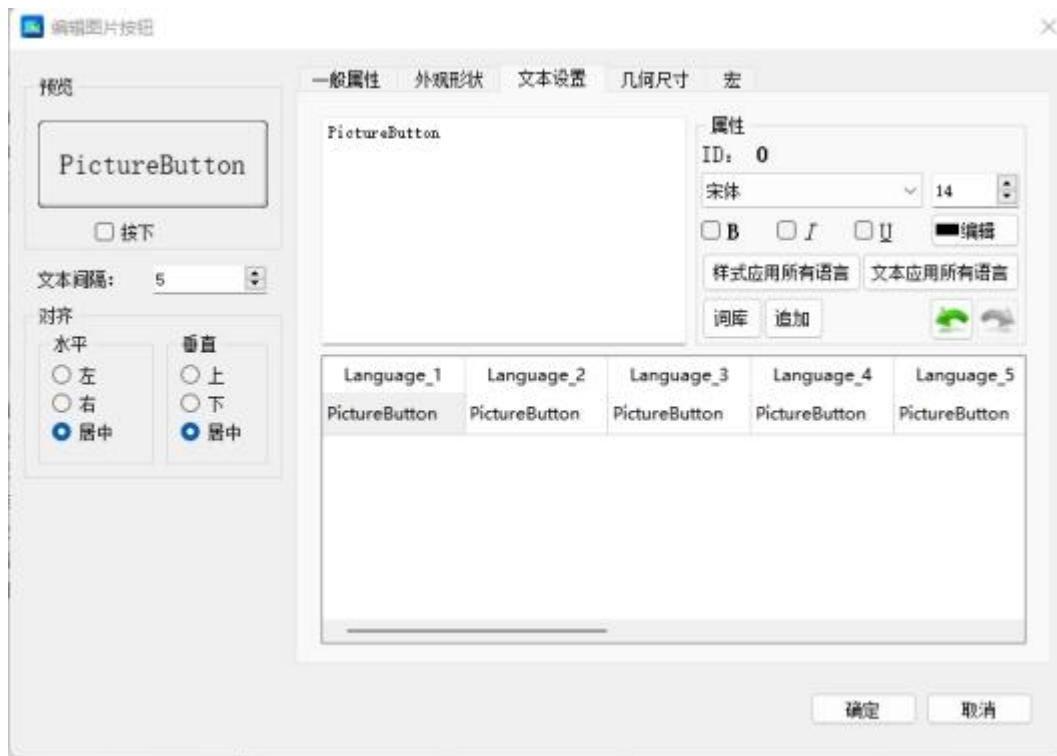
颜色	效果
RGB (255, 0, 0)	
RGB (0, 0, 0)	

14) 圆角半径

设置按钮的圆角半径。

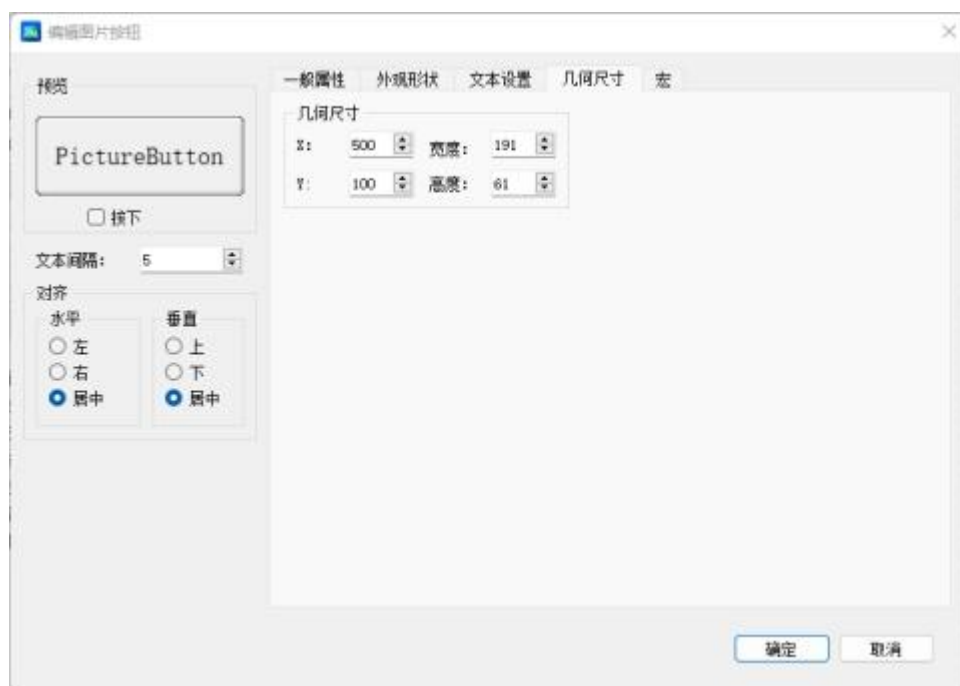
半径	效果
5	
10	
20	

6.18.3.4. 文本设置



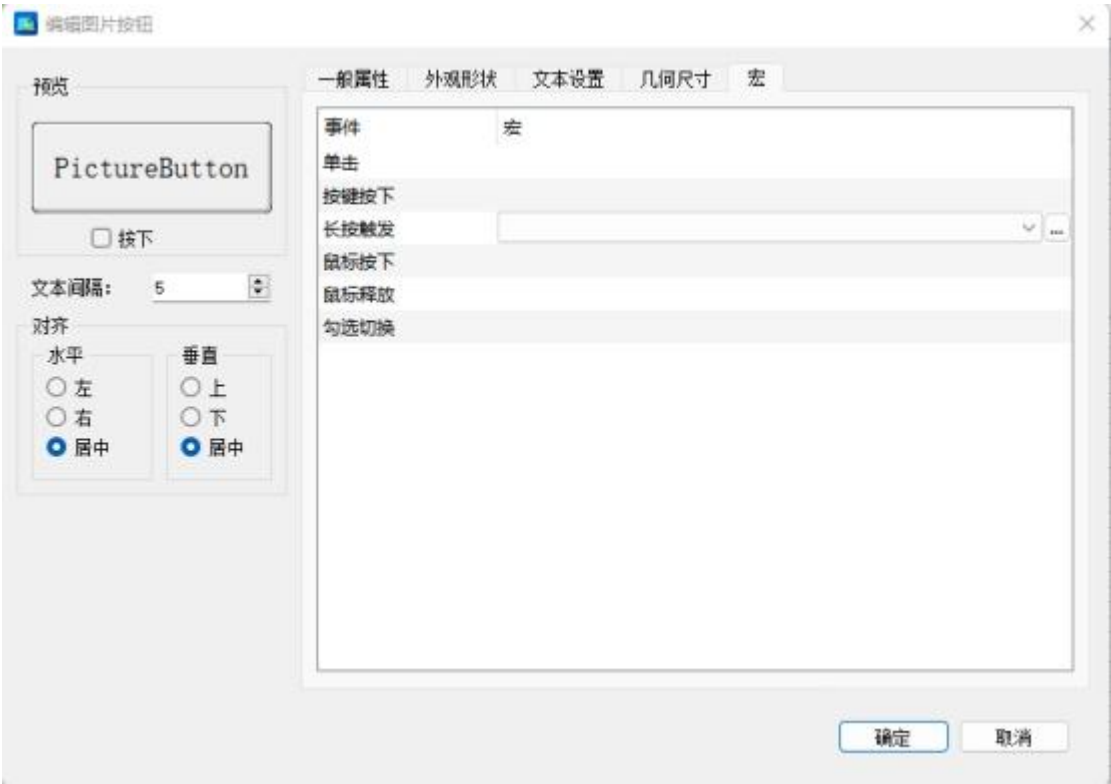
设置按钮的文本。且文本设置支持多国语言，详情可见“[多国语言编辑](#)”。

6.18.3.5. 几何尺寸



该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

6.18.3.6. 宏



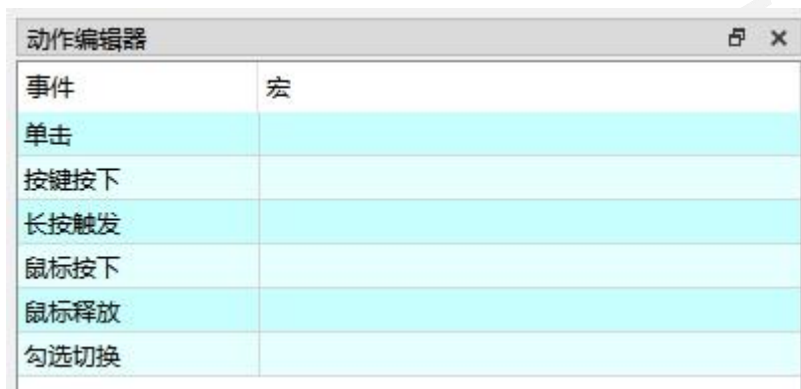
事件	触发过程
单击 (MouseClicked)	使用鼠标或快捷键按下按钮，然后释放按钮。
按键按下	使用快捷键按下按钮
长按触发	使用鼠标或快捷键按下按钮，并保持一段时间，配合延时触发功能一同使用。
鼠标按下 (MousePress)	使用鼠标或快捷键按下按钮
鼠标释放 (MouseRelease)	使用鼠标或快捷键按下按钮，然后释放按钮
勾选切换 (Toggled)	按钮处于可勾选状态，切换按钮状态。

触发多个事件时，触发事件的顺序如下，

勾选	重复	操作	事件
不可勾选	不自动重复	单击	MousePress MouseRelease

			MouseClicked
		长按	MousePress MouseRelease MouseClicked
	自动重复	单击	MousePress MouseRelease MouseClicked
		长按	MousePress MouseRelease MouseClicked ... (循环)
可勾选	不自动重复	单击	MousePress Toggled MouseRelease MouseClicked
		长按	MousePress Toggled MouseRelease MouseClicked
	自动重复	单击	MousePress Toggled MouseRelease MouseClicked
		长按	MousePress Toggled MouseRelease MouseClicked ... (循环)

6.18.4. 动作编辑器



详情可见“[6.18.3.6 宏](#)”的说明。

6.18.5. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。属性编辑器里大部分的属性设置，功能与属性设置弹窗一致。这里只介绍属性设置弹窗里未介绍的功能。

1. 延时触发



此属性设置按钮释放时，是否触发“长按触发”动作。

2. 延时触发时间



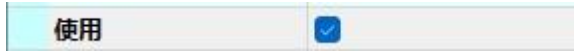
此属性以毫秒为单位设置按钮释放时，经过多久的延时时间，触发“长按触发”动作。

3. 操作权限



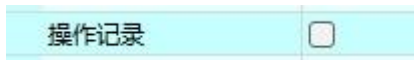
设置组件的操作权限。如果权限不足，按钮组件将无法操作。（[跳转到权限说明](#)）

4. 使用



该属性设置按钮是否使能。当组件处于禁用状态时，它会自动忽略用户的输入事件，例如点击事件、键盘事件等。同时，禁用的控件通常以视觉上的禁用样式显示，以便与启用的控件进行区分。

5. 操作记录



组件开启操作记录功能。

将图片按钮的“按下”、“释放”动作记录到操作记录中。（[跳转到操作记录说明](#)）

6. 操作记录名称



设置组件在操作记录表格中的显示名称。（[跳转到操作记录说明](#)）

6.18.6. 脚本宏

详情可见 macroWizard 文件夹的“PictureBox_Designer_CN.xml”。

6.19. 单选按钮

6.20. CadWidget (CAD)

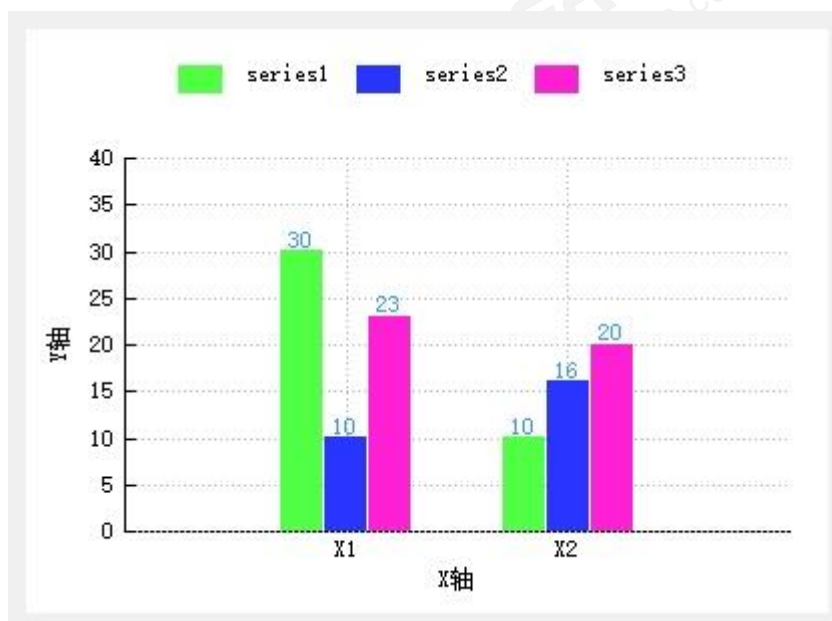
详情参考如下文档：

[二维 CAD/CAM 组件使用指南](#)

6.21. 柱状图

功能介绍

柱状图组件是在HMI上使用的组件，是一种以条形形式来展示数据的统计报告图，由一系列高度（或长度）不等的纵向（或横向）条纹表示数据分布的情况，通常利用于较小的数据集分析。柱状图亦可横向排列，对于用户来说非常直观。



6.21.1. 一般属性

对象名称、焦点策略、几何尺寸查看[公共属性](#)说明

6.21.2. 外观形状

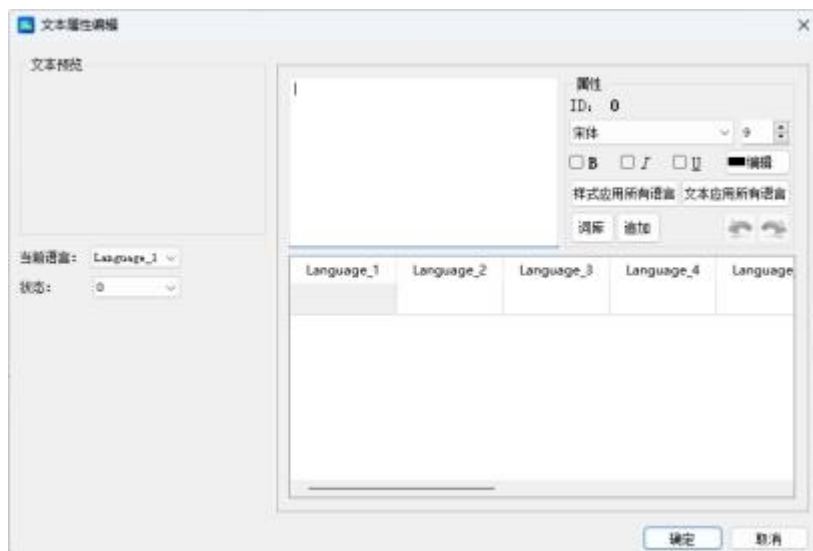
【显示标题】

表示设置组件的标题是否可见。

【标题】



表示设置组件的标题，点击  按钮可进行多语言设置。

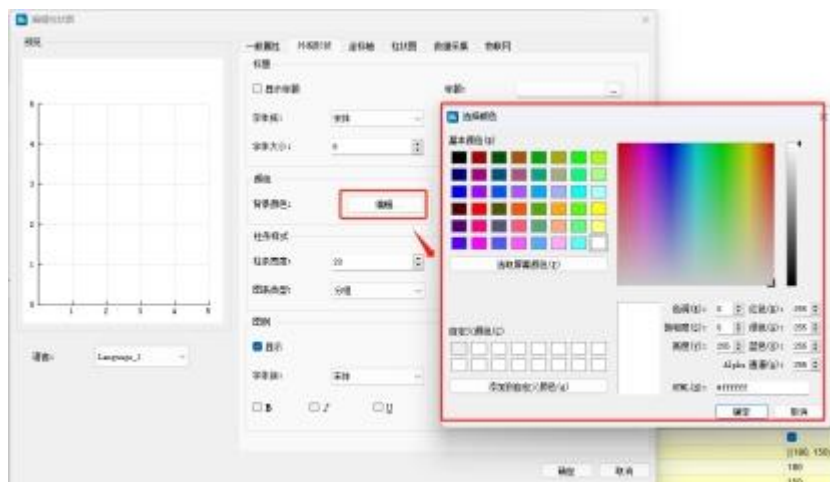


【字体族】

设置组件标题的[字体族](#)。

【颜色】

背景颜色



表示通过点击  按钮进行设置组件的背景颜色。

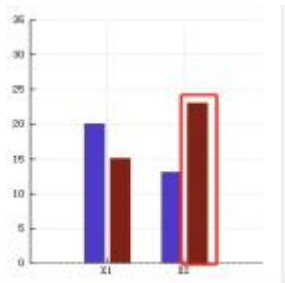
【缩放】

启用鼠标缩放

设置组件是否开启缩放功能。

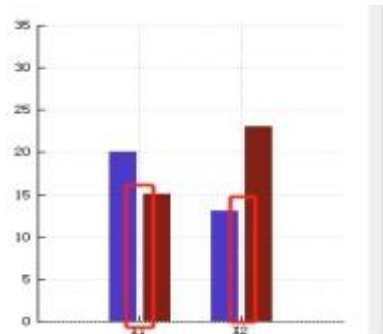
【柱条样式】

柱条宽度



用于设置柱条的宽度。

系列间距



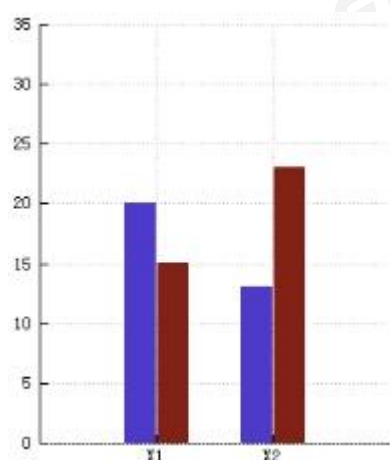
用于设置数据系列之间的显示间距。

图表类型



用于设置组件的图表类型。

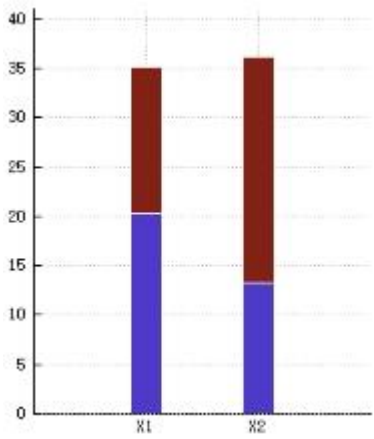
分组



注意：此类型图表要求的数据结构格式如下：

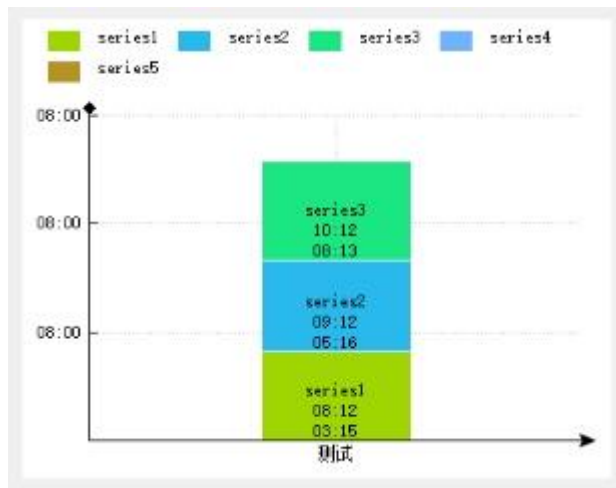
```
[
  {
    "s":1,
    "x":"12/03",
    "y":"20"
  },
  {
    "s":1,
    "x":"12/04",
    "y":"50"
  },
  {
    "s":2,
    "x":"12/03",
    "y":"40"
  }
]
```

普通堆叠



注意：此类型图表要求的数据结构同“分组”的要求。

按时间类型堆叠



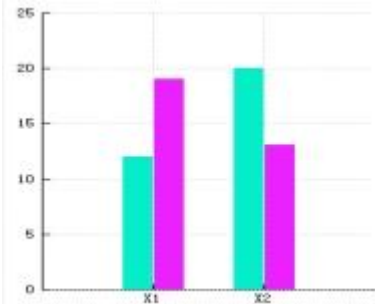
注意：此类型图表要求的数据结构格式如下：

```
[
  {
    "x": "测试",
    "y": {
      "startTime": "2022-09-18 08:12:12",
      "endTime": "2022-09-19 03:15:11"
    },
    "s": "1"
  },
  {
    "x": "测试",
    "y": {
      "startTime": "2022-09-19 09:12:12",
      "endTime": "2022-09-20 05:16:17"
    },
    "s": "2"
  },
]
```

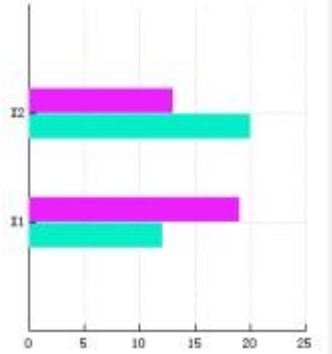
展示方向



■ 垂直



■ 水平

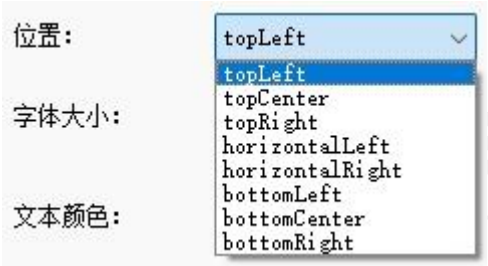


【图例】

显示

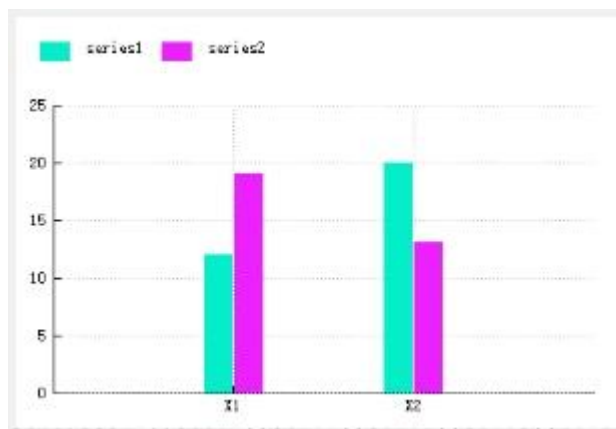
表示设置组件图例是否显示。

位置

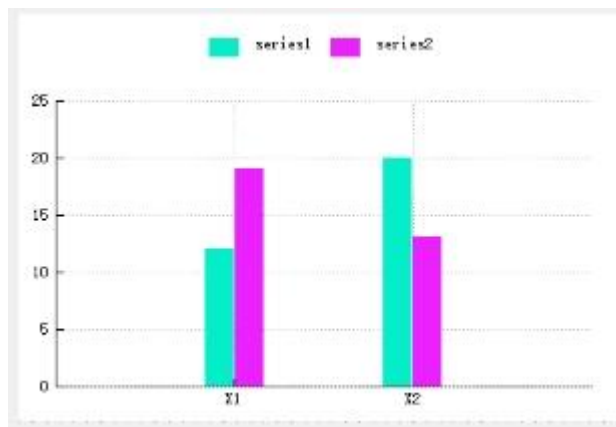


表示设置图例的显示位置。

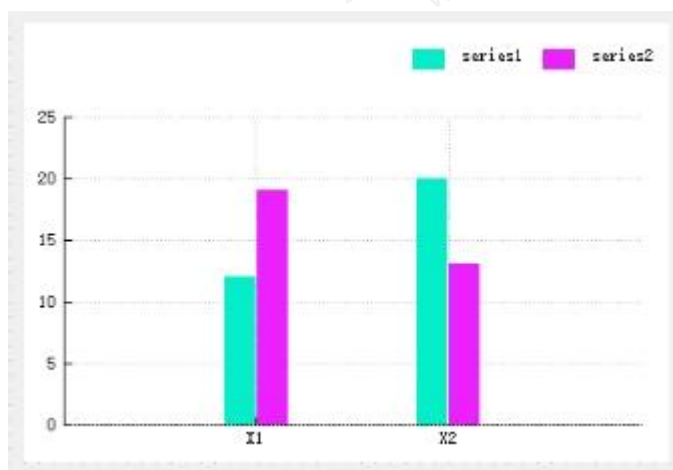
topLeft（顶部居左）



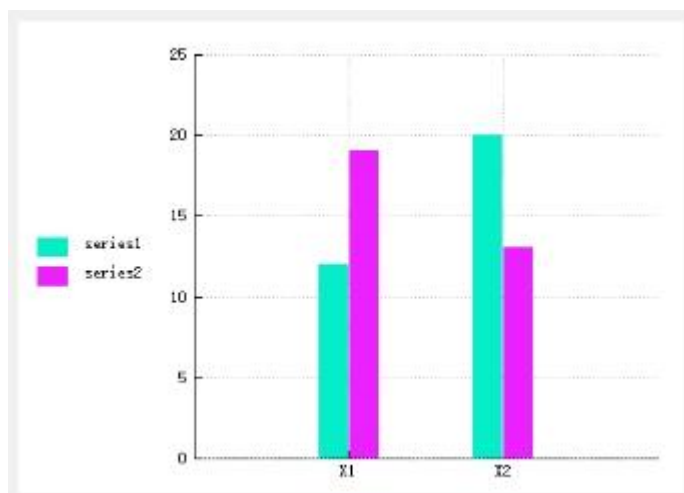
topCenter (顶部居中)



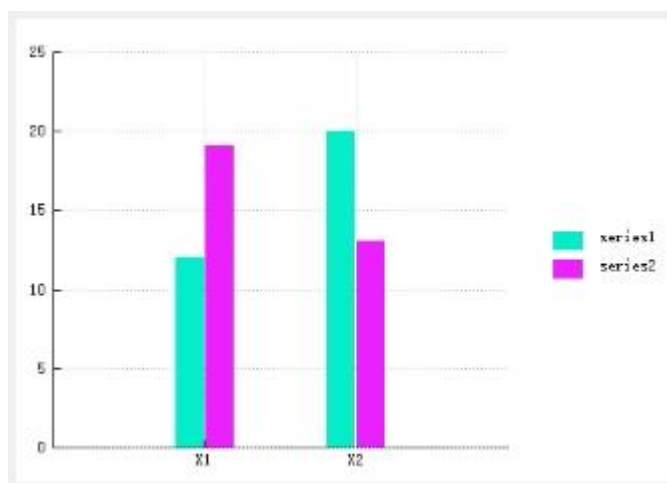
topRight (顶部居右)



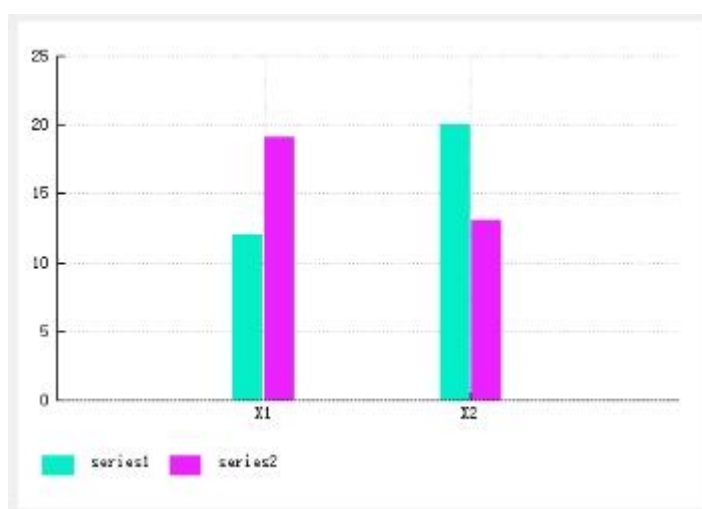
horizontalLeft (水平居左)



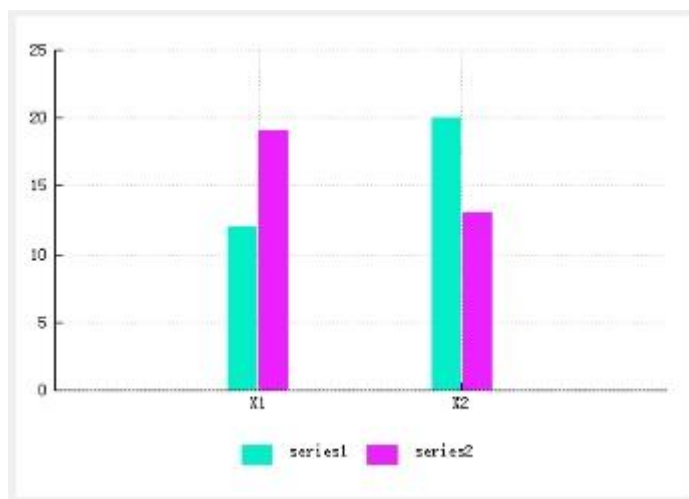
horizontalRight (水平居右)



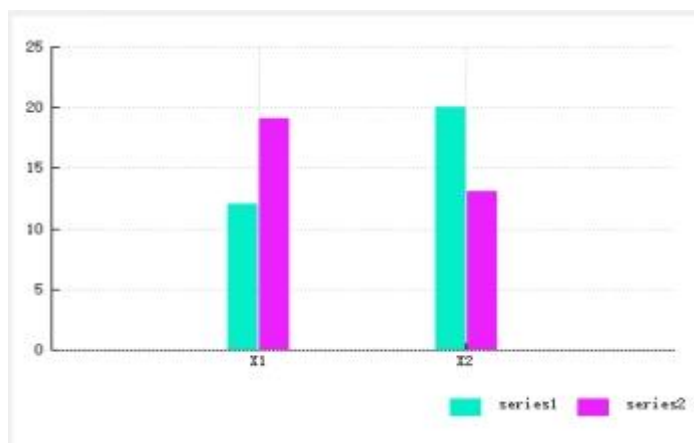
bottomLeft (底部居左)



bottomCenter (底部居中)



bottomRight (底部居右)

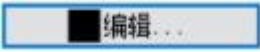


字体族

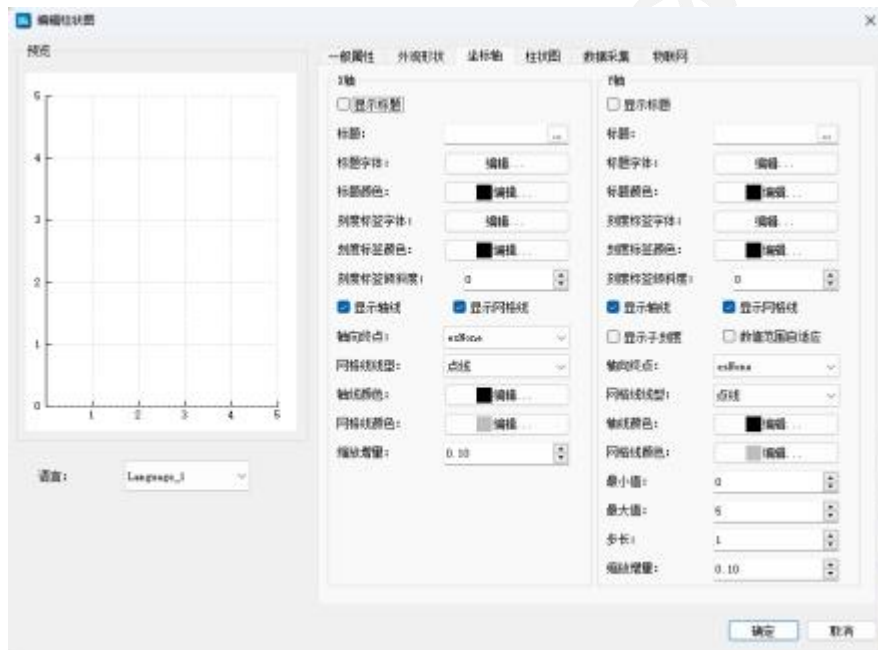
设置图例文本的字体族。

文本颜色



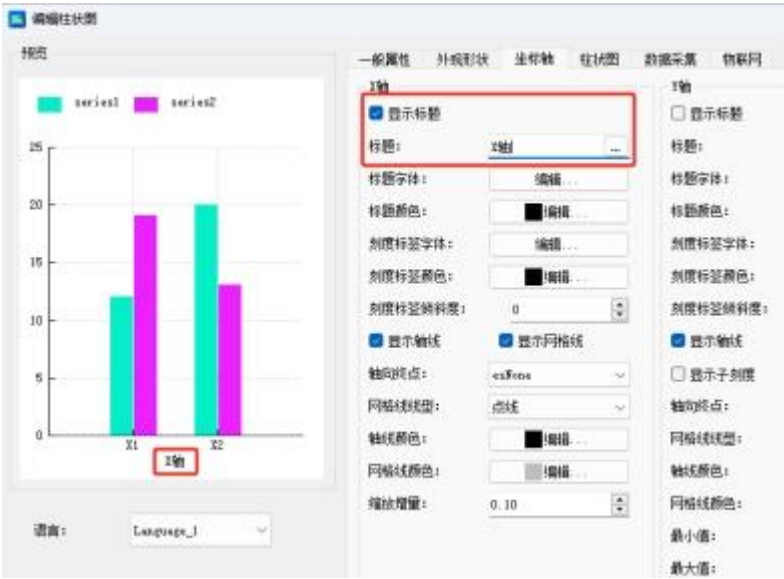
表示通过点击  按钮进行设置组件图例文本的字体颜色。

6.21.3. 坐标轴



【X 轴】

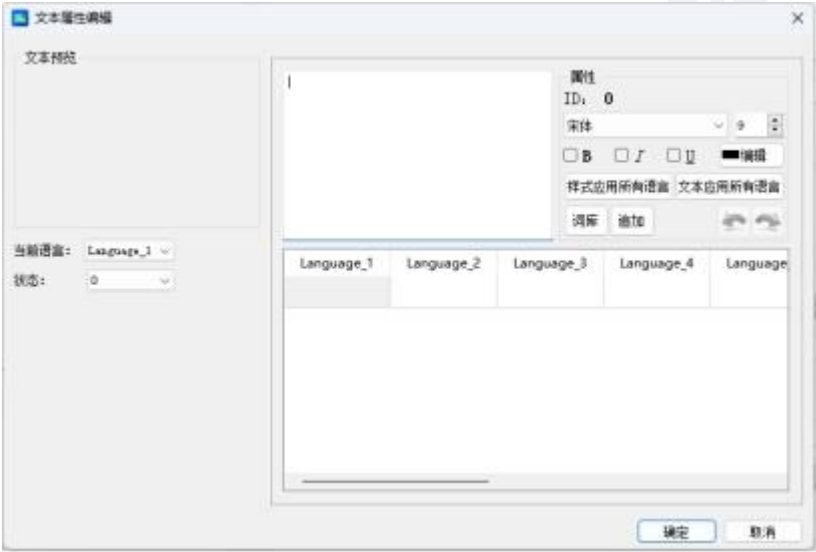
【显示标题】



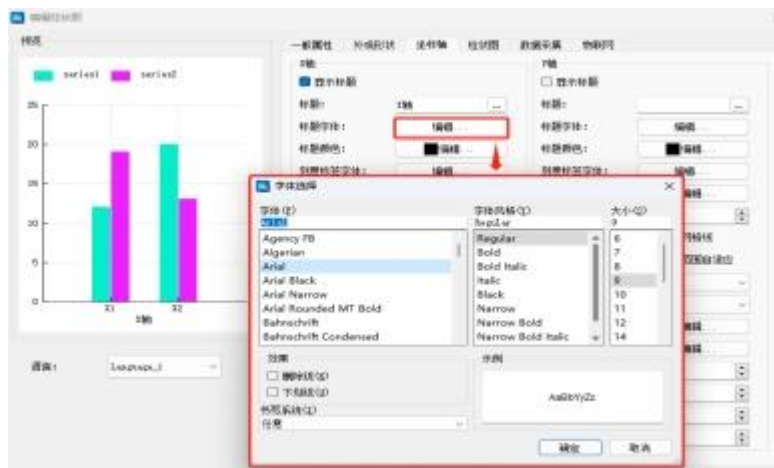
设置是否显示X轴标题。

【标题】

设置X轴的标题。点击  按钮可进行多语言设置。

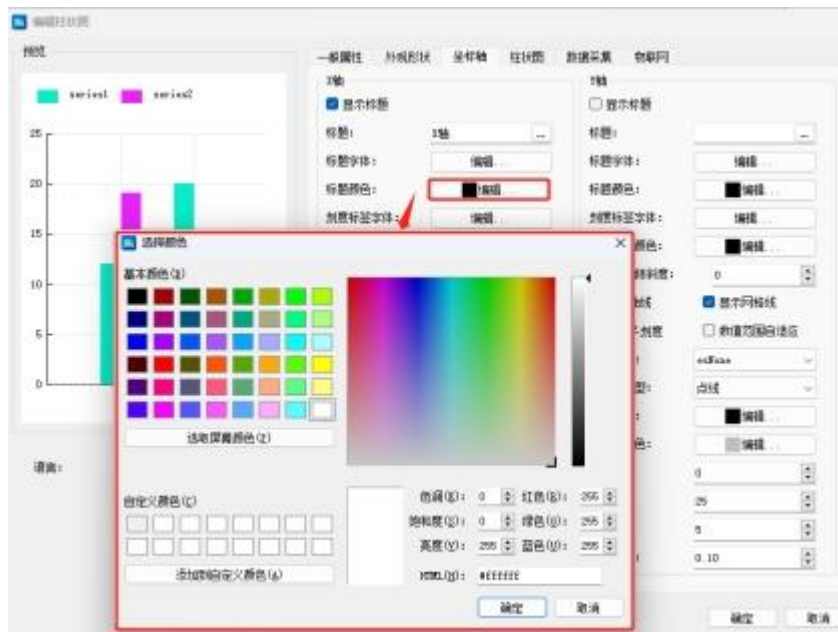


【标题字体】



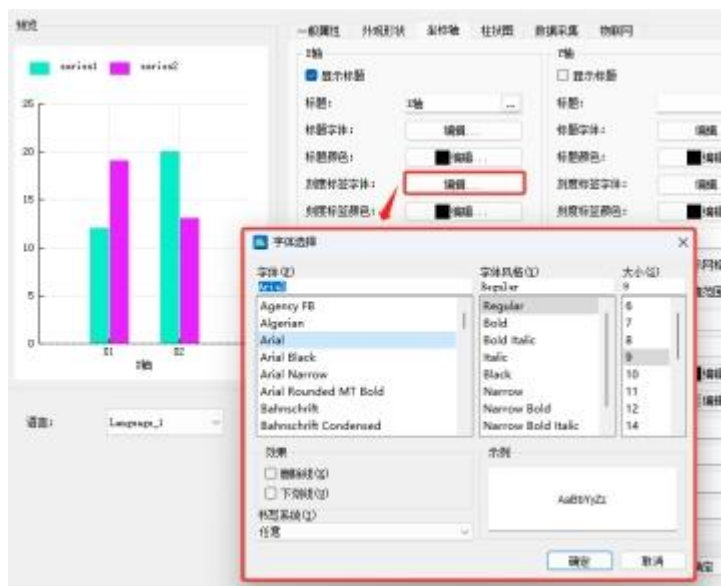
设置X轴标题的字体。

【标题颜色】



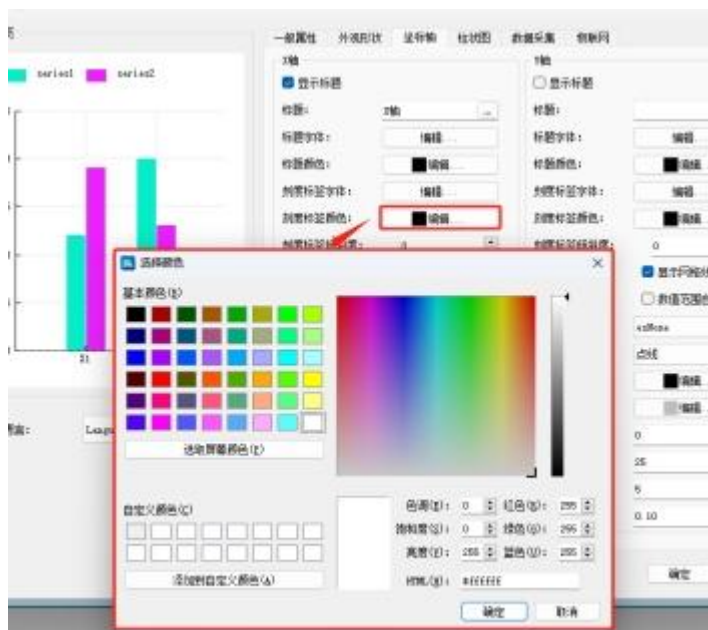
表示通过点击  按钮进行设置组件X轴标题文本的字体颜色。

【刻度标签字体】



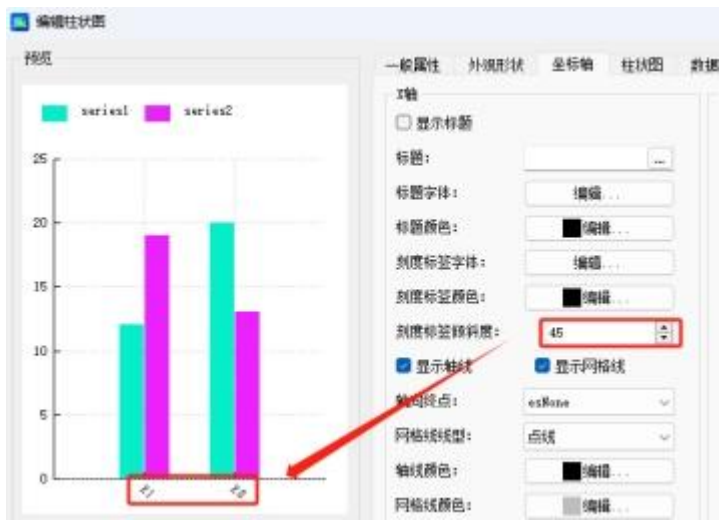
设置X轴刻度标签的字体。

【刻度标签颜色】



表示通过点击  按钮进行设置组件X轴刻度标签的字体颜色。

【刻度标签倾斜度】



表示设置X轴刻度标签的倾斜角度。范围 0-180。

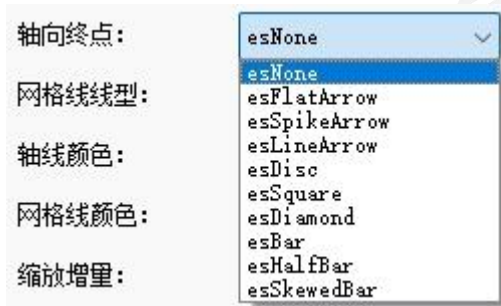
【显示轴线】

设置是否显示X轴轴线。

【显示网格线】

用于设置X轴网格线是否显示。

【轴向终点】



用于设置X轴的轴向终点样式。

esNone

表示没有终点样式。

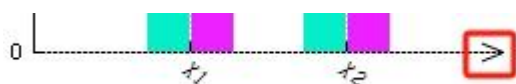
esFlatArrow



esSpikeArrow



esLineArrow



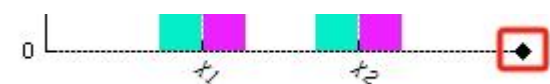
esDisc



esSquare



esDiamond



esBar



esHalfBar



esSkewedBar

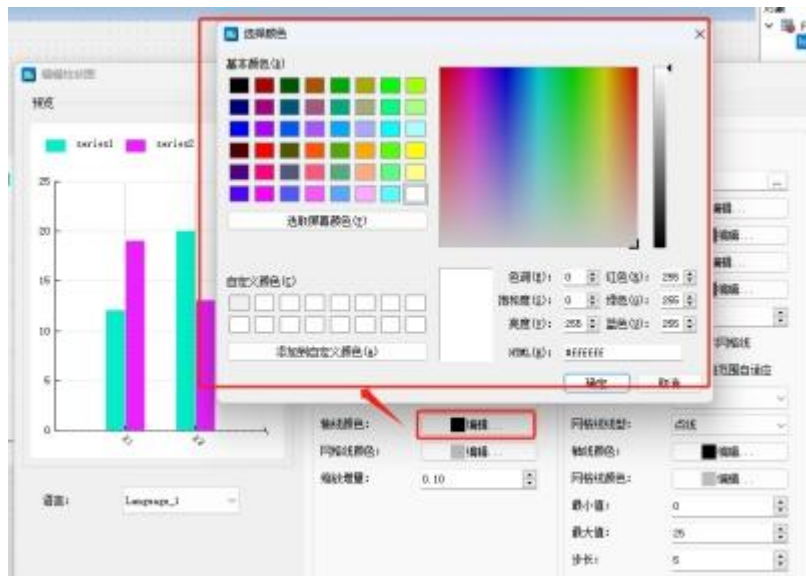


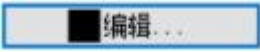
【网格线线型】



表示设置X轴的网格线线型。

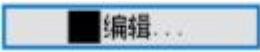
【轴线颜色】



表示通过点击  按钮进行设置组件X轴轴线的颜色。

【网格线颜色】



表示通过点击  按钮进行设置组件X轴网格线的颜色。

【缩放增量】

表示设置X轴在启用“鼠标缩放”功能时的缩放增量。

【Y 轴】

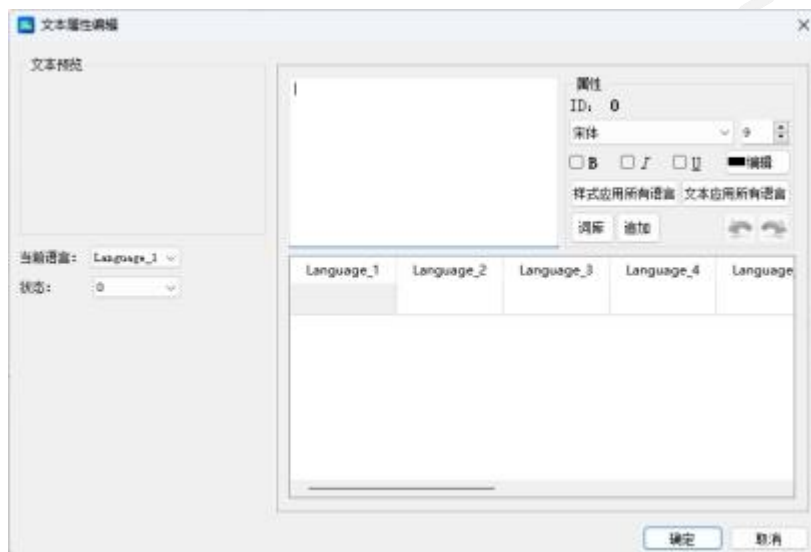
【显示标题】



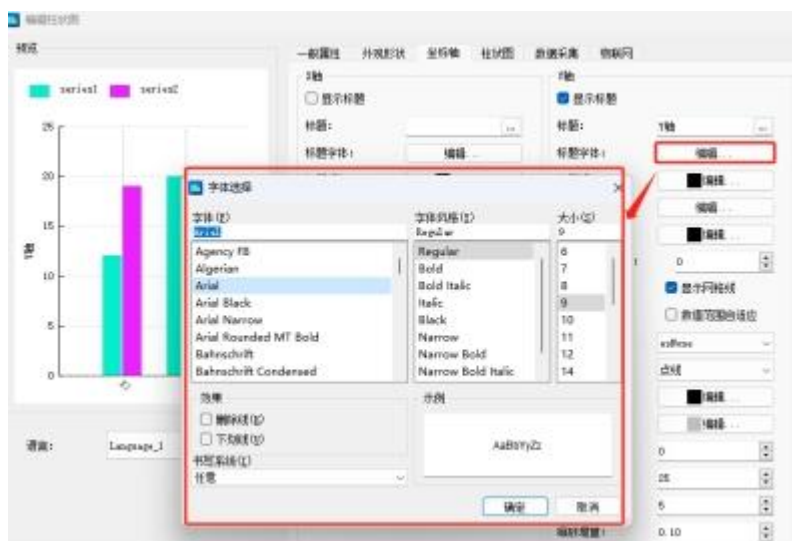
设置是否显示Y轴标题。

【标题】

设置Y轴的标题。点击  按钮可进行多语言设置。

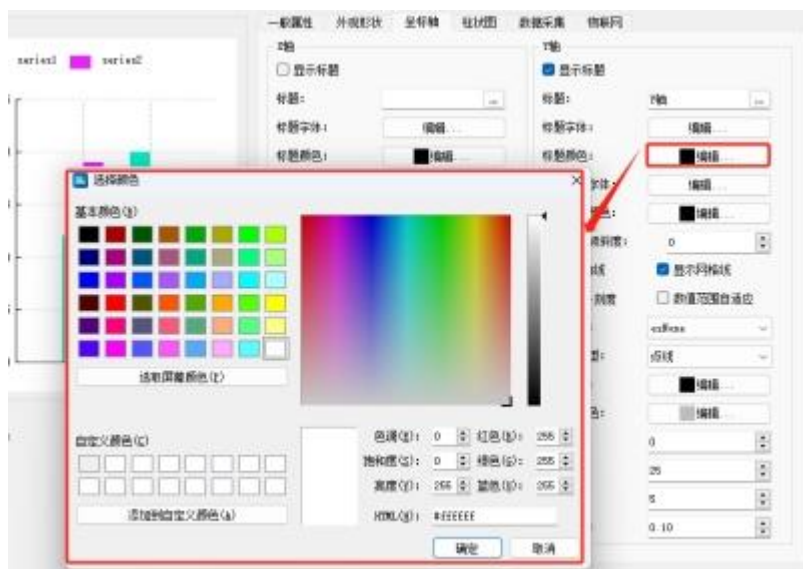


【标题字体】



设置Y轴标题的字体。

【标题颜色】



表示通过点击  按钮进行设置组件Y轴标题文本的字体颜色。

【刻度标签字体】

设置Y轴标题的字体。

【刻度标签颜色】

设置Y轴刻度标签的颜色。

【刻度标签倾斜度】

表示设置Y轴刻度标签的倾斜角度。范围 0-180。

【显示轴线】

设置是否显示Y轴轴线。

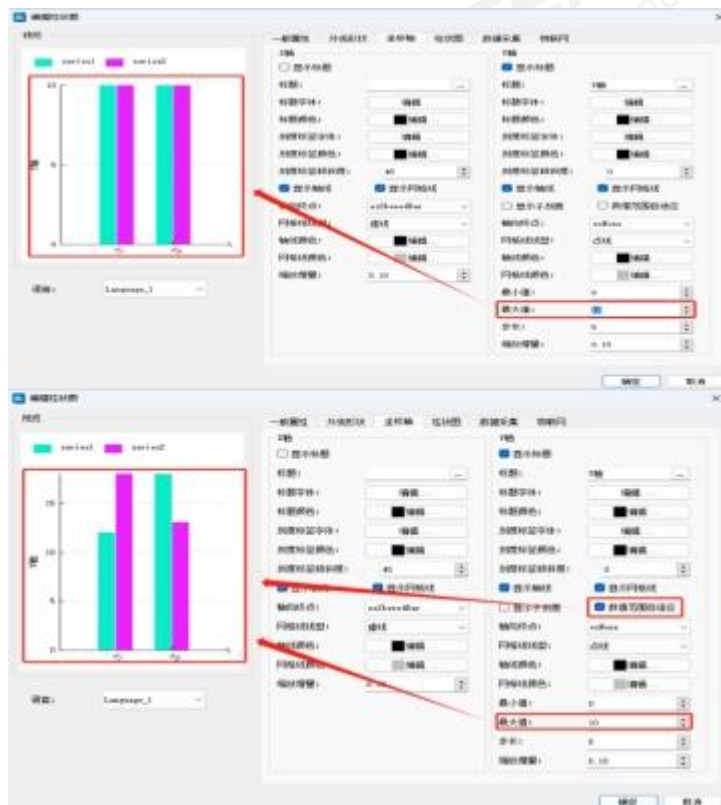
【显示网格线】

设置是否显示Y轴网格线。

【显示子刻度】

设置是否显示Y轴子刻度。

【数值范围自适应】



设置是否自适应数值范围。

【轴向终点】

设置Y轴的轴向终点样式，参考上述“[X轴](#)”>轴向终点”的描述。

【网格线线型】

设置Y轴网格线的线型。

【轴线颜色】

设置Y轴轴线的颜色。

【网格线颜色】

设置Y轴网格线的颜色。

【最小值】

设置Y轴数值的最小值，此设置项显示的前提：上述“[外观形状](#)>>[柱条样式](#)>>图表类型”选择非“按时间类型堆叠”。

【最大值】

设置Y轴数值的最大值，此设置项显示的前提：上述“[外观形状](#)>>[柱条样式](#)>>图表类型”选择非“按时间类型堆叠”。

【标签时间格式】

设置Y轴标签为时间类型时的显示格式，此设置项显示的前提：上述“[外观形状](#)>>[柱条样式](#)>>图表类型”选择“按时间类型堆叠”。

【数据时间格式】

设置组件图表类型为“按时间类型堆叠”时所接收的时间数据的规定格式，此设置项显示的前提：上述“[外观形状](#)>>[柱条样式](#)>>图表类型”选择非“按时间类型堆叠”。

例如在使用宏接口“[setIOTData \(设置柱状图数据\)](#)”设置数据时，若要使下图数据的startTime和endTime字段被组件所接受，那么就要求将此项设置为yyyy-MM-dd HH:mm:ss：

```
{
  {
    "x": "测试",
    "y": { "startTime": "2022-09-18 08:12:12", "endTime": "2022-09-19 03:15:11" },
    "s": "1"
  },
  {
    "x": "测试",
    "y": { "startTime": "2022-09-19 09:12:12", "endTime": "2022-09-20 05:16:17" },
    "s": "2"
  },
  {
    "x": "测试",
    "y": { "startTime": "2022-09-20 10:12:12", "endTime": "2022-09-21 08:13:16" },
    "s": "3"
  }
}
```

【步长】

设置Y轴的刻度步长。

【缩放增量】

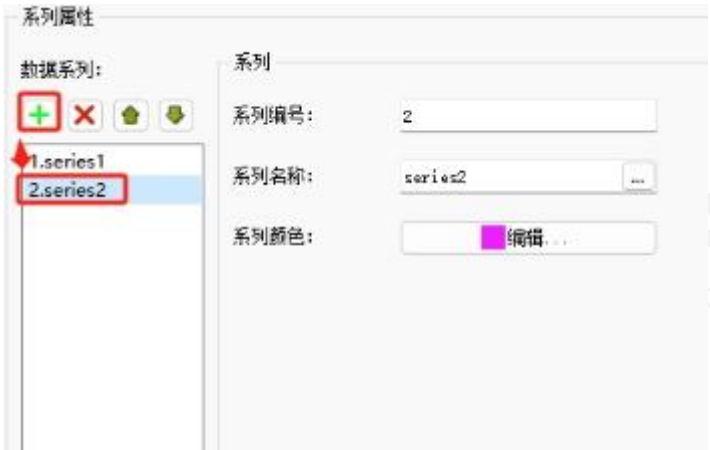
表示设置Y轴在启用“鼠标缩放”功能时的缩放增量。

6.21.4. 柱状图



【数据系列】

新增



点击  按钮将新增一个数据系列。

删除



点击  按钮将删除一个数据系列。

上移

点击  按钮将选中的数据系列进行上移。

下移

点击  按钮将选中的数据系列进行下移。

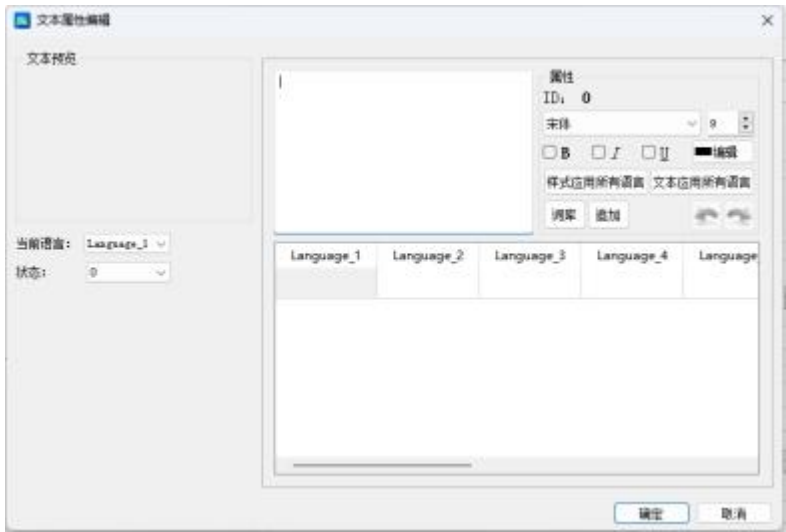
【系列】

系列编号

用于显示数据系列的编号，不可编辑。

系列名称

用于设置数据系列的名称，点击  按钮可进行多语言设置。



系列颜色

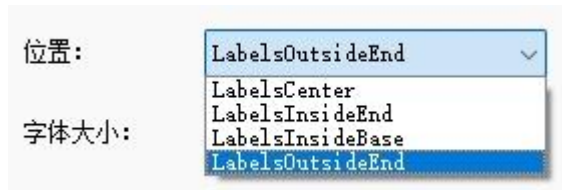
用于设置数据系列的颜色。

【数据标签】

显示

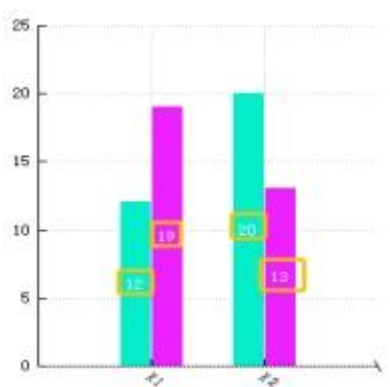
用于设置是否显示柱条的数据标签。

位置

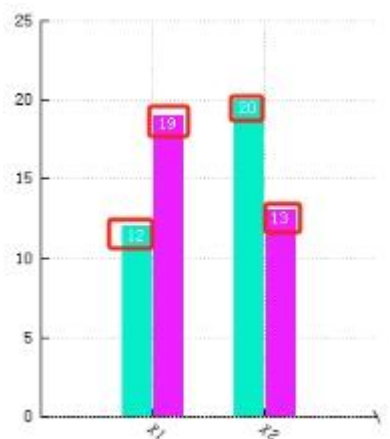


表示设置数据标签相对于柱条的显示位置。

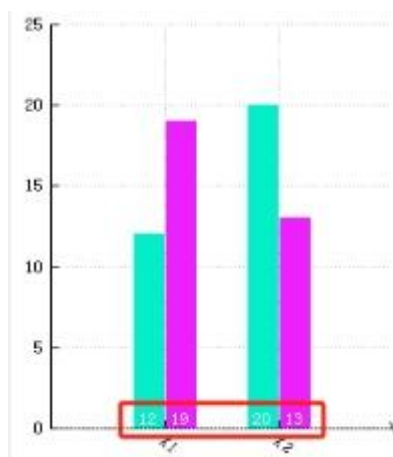
LabelsCenter



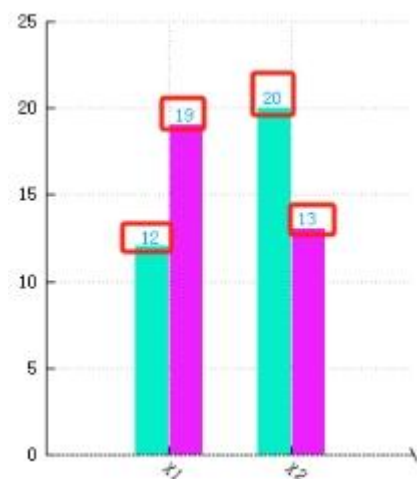
LabelsInsideEnd



LabelsInsideBase

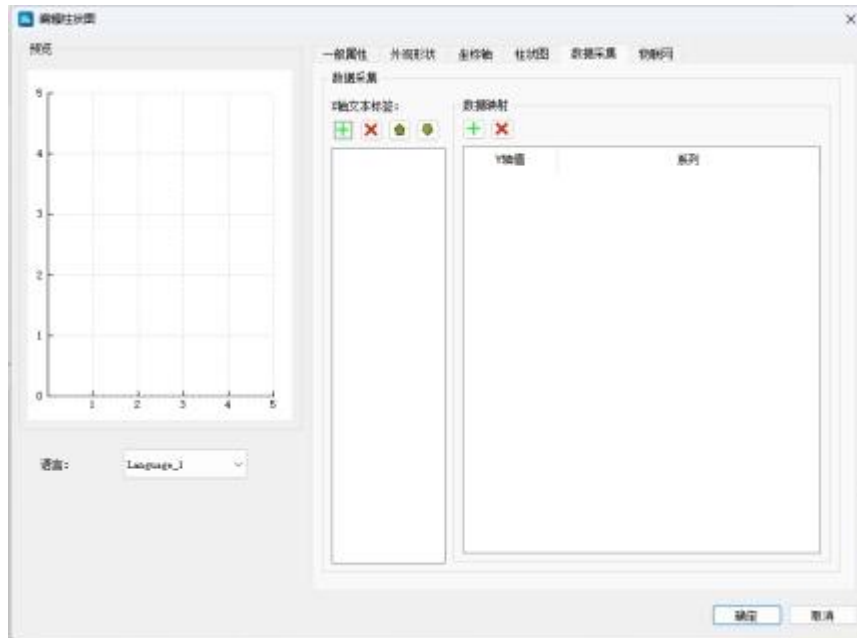


LabelsOutsideEnd

**【字体族】**

用于设置数据标签的字体族。

6.21.5. 数据采集



【X 轴文本标签】

新增



通过点击  按钮将新增一个X轴文本标签。

删除

通过点击  按钮将删除一个X轴文本标签。

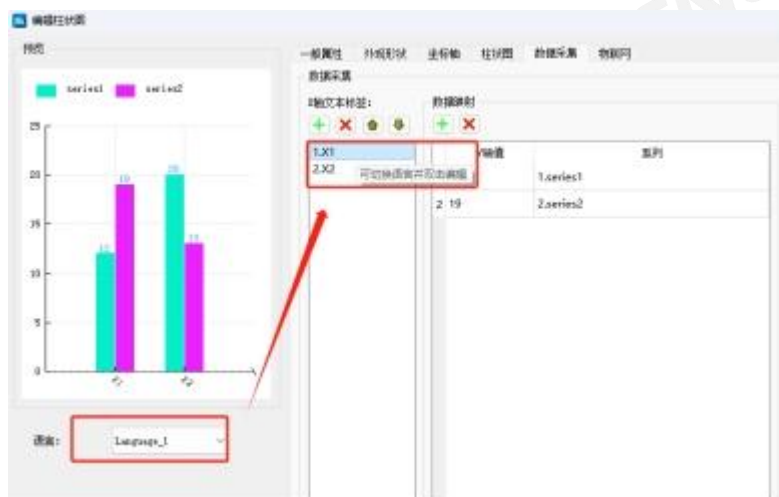
上移

通过点击  按钮将选中的X轴文本标签进行上移。

下移

通过点击  按钮将选中的X轴文本标签进行下移。

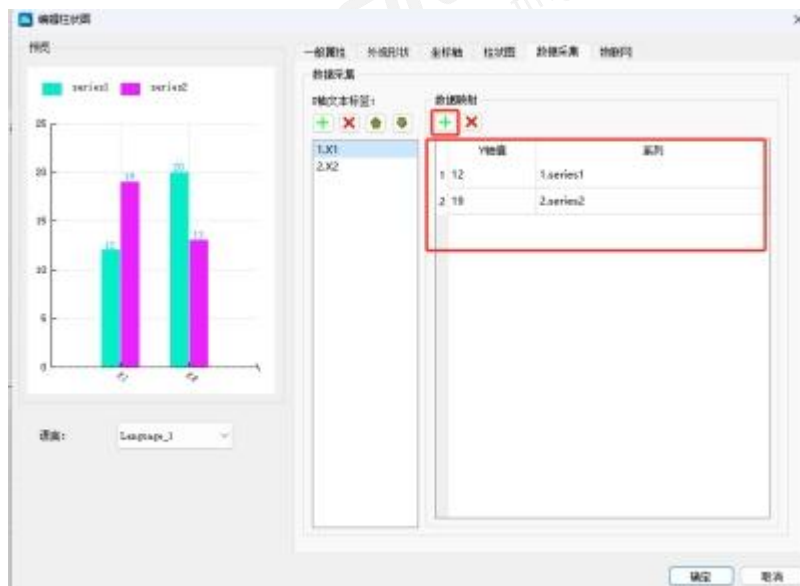
修改X轴文本标签



可先进行切换语言，然后双击“X轴文本标签”进行修改文本。

【数据映射】

新增



通过点击  按钮将新增一个数据映射。

删除

通过点击按钮将删除一个数据映射。

表格数据说明

	Y轴值	系列
1	12	1.series1
2	19	2.series2

Y轴值

	Y轴值	系列
1	12	1.series1
2	19	2.series2

双击单元格，然后可以直接输入数值，作为静态数据。



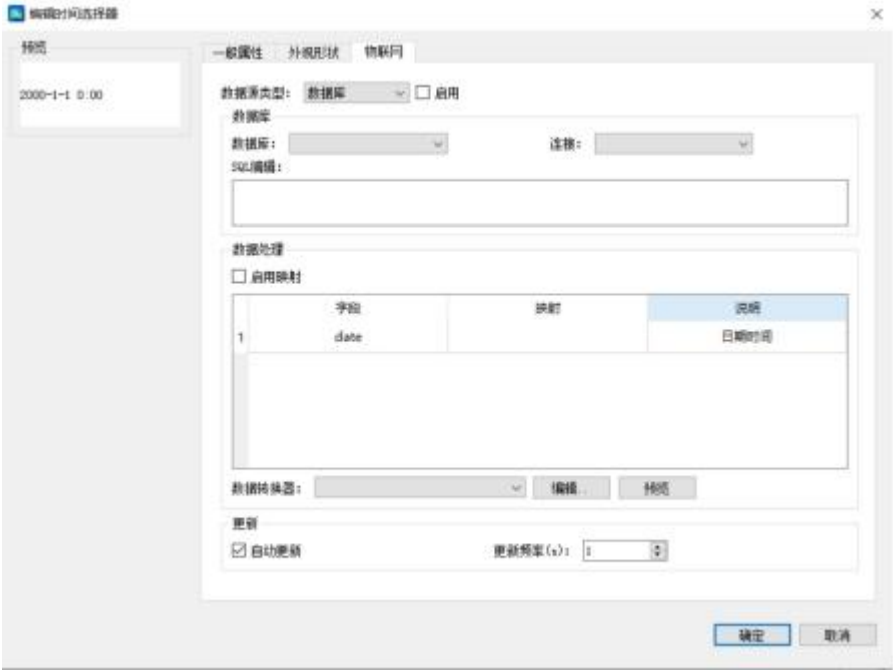
或者点击按钮将打开变数编辑窗口进行关联变数地址。注意：目前不支持变数设置（这里只是预留实现）。

系列



双击单元格，通过下拉列表进行选择数据系列。

6.21.6. 物联网



此栏目主要用于设置与物联网相关的属性。

【数据源类型】

数据源类型



用于设置物联网的数据来源。

启用

表示是否启用物联网功能。

【数据库】

数据库



通过下来列表选择要使用的数据库类型。目前仅支持MySQL数据库。仅限上述“数据源类型”选择“数据库”选项时显示设置项。

连接

通过下来列表选择数据库连接，下拉列表数据项来源于物联网设置中的数据库连接配置。仅限上述“数据源类型”选择“数据库”选项时显示设置项。

SQL编辑

用于编辑SQL查询语句，例如：select col,col2 from table。仅限上述“数据源类型”选择“数据库”选项时显示设置项。

API



请求方式



用于设置API请求的请求方式。仅限上述“数据源类型”选择“API”选项时显示设置项。

URL

用于设置API请求地址，例如：<https://www.baidu.com>。仅限上述“数据源类型”选择“API”选项时显示设置项。

请求数据头

用于设置API请求的数据头，以键（Key）-值（value）形式设置，如：



通过点击  按钮进行添加，点击  按钮进行删除。

仅限上述“数据源类型”选择“API”选项时显示设置项。

【数据处理】

启用映射

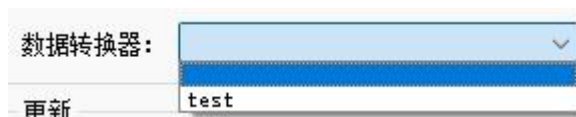
设置是否启用数据字段映射。若未启用映射，则组件的数据要求字段必须为s、x、y（参考下述映射表），否则数据未映射，将不会在组件中设置成功。

映射表

数据处理			
☒ 启用映射			
	字段	映射	说明
1	s		系列
2	x		x轴
3	y		y轴

在启用映射后，通过数据库查询得到的数据字段不是s、x、y情况下，用户可以使用查询到的数据字段映射到组件要求的s、x、y字段即可。

数据转换器



通过下拉列表选择需要使用的数据转换器。下拉列表中的数据项来源于下述“编辑”中所创建的数据转换器。

编辑



如上图，通过点击 **编辑** 按钮进行管理数据转换器。

预览



如上图所示，通过点击 **预览** 按钮打开数据预览窗口。

【更新】

自动更新

表示组件是否启用定时获取物联网数据进行自动刷新。

更新频率

表示组件数据定时刷新的周期，单位为秒（s）。

6.21.7. 组件宏接口

1. enabledZoom (设置缩放功能)

```
/**
 * @描述 设置是否开启缩放功能
 * @参数 enabled 开启状态【true: 开启; false: 关闭】
 * @返回值 无
```

示例: Form.hBarChart.enabledZoom(true)

2. exportContent (导出物联网数据)

```
/**
 * @描述 导出数据 (仅支持.csv 或.xlsx 或.xls)
 * @参数 type 【0: 导出数据; 1: 导出图形; 2: 导出数据和图形】
 * @参数 path 保存地址 (全路径, 包含文件名称和后缀)
 * @返回值 无
```

示例: Form.hBarChart.exportContent(0, "C:\\\\Users\\123.csv")

注意: 此功能仅限在PCBased平台生效。

3. setAxisZoom (设置轴缩放)

```
/**
 * @描述 设置轴缩放
 * @参数 direction 缩放方向【0: 下移; 1: 上移; 2: 左移; 3: 右移; 4: 收缩; 5: 放大; 6: 适中】
 * @返回值 无
```

示例: Form.hBarChart.setAxisZoom(0)

4. setAxisZoomIncrement (设置轴缩放增量)

```
/**
 * @描述 设置轴的缩放增量
 * @参数 axis 轴【0: X轴; 1: Y轴】
 * @参数 increment 缩放增量
 * @返回值 无
```

示例: Form.hBarChart.setAxisZoomIncrement(0,10.00)

5. setBarChartBackground (设置柱状图背景颜色)

```
/**
 * @描述 设置柱状图背景颜色
 * @参数 color 颜色名
 * @返回值 无
```

示例: Form.hBarChart.setBarChartBackground("#b7ffd4")

6. setBarChartLegend (设置柱状图图例)

```
/**
 * @描述 设置柱状图图例
 * @参数 position 图例位置【0: 顶部居左; 1: 顶部居中;
 2: 顶部居右; 3: 水平居左; 4: 水平居右; 5: 底部居左; 6:
 底部居中; 7: 底部居右】
 * @参数 family 字体族
 * @参数 fontSize 字体大小
 * @参数 bold 是否粗体
 * @参数 italic 是否倾斜
 * @参数 underline 是否下划线
 * @参数 color 字体颜色
 * @返回值 无
```

示例:

Form.hBarChart.setBarChartLegend(1,"Microsoft YaHei",true,true,true,"#000000")

7. setBarChartSeries (设置柱状图数据系列)

```
/**
 * @描述 设置柱状图数据系列
 * @参数 series 数据系列 json 串
 *
 * 【 [{"id":"1","name":"series1","seriesColor":
 * :"\#ffffff"},...]】
 * -- --
```

示例：

```
Form.hBarChart.setBarChartSeries("[{"id":"1","name":"series1","seriesColor":
"\#f737ca"}, {"id":"2","name":"series2","seriesColor": "\#b7ffd
4"}]")
```

8. setBarChartTitle (设置柱状图标题)

```
/**
 * @描述 设置柱状图标题 (调用该方法默认标题可见)
 * @参数 title 标题
 * @参数 family 字体族
 * @参数 fontSize 字体大小
 * @参数 bold 是否粗体
 * @参数 italic 是否倾斜
 * @参数 underline 是否下划线
 * @参数 color 字体颜色
 * @返回值 无
 */
```

示 例 ： Form.hBarChart.setBarChartTitle(" 标 题 1","Microsoft YaHei",true,true,true,"#000000")

9. setBarChartXAxisColor (设置柱状图 X 轴线颜色)

```
/**  
* @描述 设置 X 轴线颜色  
* @参数 color 颜色  
* @返回值 无
```

示例：Form.hBarChart.setBarChartXAxisColor("#000000")

10. setBarChartXGridColor (设置柱状图 X 轴网格线颜色)

```
/**  
* @描述 设置 X 轴网格线颜色  
* @参数 color 颜色  
* @返回值 无
```

示例：Form.hBarChart.setBarChartXGridColor("#000000")

11. setBarChartXGridLineType (设置柱状图 X 轴网格线线型)

```
/**  
* @描述 设置 X 轴网格线线型  
* @参数 type 线型【0：虚线；1：实线；2：点线】  
* @返回值 无
```

示例：Form.hBarChart.setBarChartXGridLineType(1)

12. setBarChartXTitle (设置柱状图 X 轴标题)

```

/**
 * @描述 设置柱状图 X 标题（调用该方法默认标题可见）
 * @参数 title 标题
 * @参数 family 字体族
 * @参数 fontSize 字体大小
 * @参数 bold 是否粗体
 * @参数 italic 是否倾斜
 * @参数 underline 是否下划线
 * @参数 color 字体颜色
 * @返回值 无
 */

```

示 例：Form.hBarChart.setBarChartXTitle(" 标 题 1", "Microsoft YaHei", true, true, true, "#000000")

13. setBarChartXUpperEnding（设置柱状图 X 轴轴向终点样式）

```

/**
 * @brief 设置 X 轴轴向终点样式
 * @param style 样式【0: esNone; 1: esFlatArrow; 2: esSpikeArrow; 3: esLineArrow; 4: esDisc; 5: esSquare; 6: esDiamond; 7: esBar; 8: esHalfBar; 9: esSkewedBar】
 * @返回值 无

```

示例：Form.hBarChart.setBarChartXUpperEnding(1)

14. setBarChartYAxisColor（设置柱状图 Y 轴线颜色）

```

/**
 * @描述 设置 Y 轴线颜色
 * @参数 color 颜色
 * @返回值 无

```

示例：Form.hBarChart.setBarChartYAxisColor("#000000")

15. setBarChartYAxisValRangeAutoAdjust (设置柱状图 Y 轴数值范围自适应)

```
/**
 * @描述 设置柱状图 Y 轴数值范围自适应数据
 * @参数 enabled 是否开启自适应
 * @返回值 无
 */
```

示例:

```
Form.hBarChart.setBarChartYAxisValRangeAutoAdjust(true)
```

16. setBarChartYGridColor (设置柱状图 Y 轴网格线颜色)

```
/**
 * @描述 设置 Y 轴网格线颜色
 * @参数 color 颜色
 * @返回值 无
 */
```

示例: `Form.hBarChart.setBarChartYGridColor("#000000")`

17. setBarChartYGridLineType (设置柱状图 Y 轴网格线线型)

```
/**
 * @描述 设置 Y 轴网格线线型
 * @参数 type 线型【0: 虚线; 1: 实线; 2: 点线】
 * @返回值 无
 */
```

示例: `Form.hBarChart.setBarChartYGridLineType(1)`

18. setBarChartYRange (设置柱状图 Y 轴数值范围)

```
/**
 * @描述 设置柱状图 Y 轴数值范围
 * @参数 mininum 最小值
 * @参数 maxinum 最大值
 * @返回值 无
 */
```

示例：Form.hBarChart.setBarChartYRange(0, 20)

19. setBarChartYStep（设置柱状图 Y 轴刻度步长）

```
/**
 * @描述 设置柱状图 Y 轴刻度步长
 * @参数 step 步长
 * @返回值 无
```

示例：Form.hBarChart.setBarChartYStep(2)

20. setBarChartYTitle（设置柱状图 Y 轴标题）

```
/**
 * @描述 设置柱状图 Y 标题（调用该方法默认标题可见）
 * @参数 title 标题
 * @参数 family 字体族
 * @参数 fontSize 字体大小
 * @参数 bold 是否粗体
 * @参数 italic 是否倾斜
 * @参数 underline 是否下划线
 * @参数 color 字体颜色
 * @返回值 无
 */
```

示 例：Form.hBarChart.setBarChartYTitle(" 标 题 1", "Microsoft YaHei", true, true, true, "#000000")

21. setBarChartYUpperEnding（设置柱状图 Y 轴轴向终点样式）

```
/**
 * @描述 设置Y轴轴向终点样式
 * @参数 style 样式【0: esNone; 1: esFlatArrow; 2:
 esSpikeArrow; 3: esLineArrow; 4: esDisc; 5: esSquare;
 6: esDiamond; 7: esBar; 8: esHalfBar; 9: esSkewedBar】
 * @返回值 无
 */
```

示例: Form.hBarChart.setBarChartYUpperEnding(1)

22. setBarsStyle (设置柱状图柱条样式)

```
/**
 * @描述 设置柱状图柱条样式
 * @参数 direction 展示方向【0: 垂直; 1: 水平】
 * @参数 barWidth 柱条宽度
 * @参数 spacing 系列间距
 * @参数 xLabelsRotation X轴文本标签倾斜度
 * @返回值 无
 */
```

示例: Form.hBarChart.setBarsStyle(0, 30, 5, 0)

23. setIOTData (设置柱状图数据)

```
/**
 * @描述 设置时间类型堆叠图数据
 * @参数 data json 字符串数据【[{"x": "2022/09/18
08:12:12", "y": "2022/09/19
03:15:11", "s": "1"},...]】
 * @返回值 无
 */
```

示例: Form.hBarChart.setIOTData("[{"x": "2022/09/18
08:12:12", "y": "2022/09/19
03:15:11", "s": "1"},...]") 或
hBarChart.setIOTData("[{"x": "label1", "y": "12", "s": "1"},...]")

注意: 此处的参数data的数据结构需根据上述“[柱条样式](#)>>1图表类型”中的描述进行规范,

否则数据无效。

24. updateIOTData（获取物联网数据并刷新）

```
/**
 * @描述 手获取物联网数据刷新
 * @返回值 无
 */
```

示例：Form.hBarChart.updateIOTData()

6.21.8. 组件宏属性

属性	类型	描述	示例
backgroundColor	QColor	背景颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.backgroundColor=H Property.setQColor(colorMap)</pre>
barChartWidth	Int	柱条宽度	Form.hBarChart.barChartWidth=10
dataLabelsColor	QColor	柱状图数据标签文本颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.dataLabelsColor=H Property.setQColor(colorMap)</pre>
dataLabelsFont	QFont	柱状图数据标签文本字体	<pre>var fontMap={}; fontMap["weight"]=12; fontMap["pointSize"]=12; fontMap["family"]="宋体"; fontMap["italic"]=true;</pre>

			<pre>fontMap["underline"]=true; Form.hBarChart.dataLabelsFont=HProperty.setQFont(fontMap)</pre>
dataLabelsVisible	bool	柱状图数据标签可见性	<pre>Form.hBarChart.dataLabelsVisible=true</pre>
displayDirection	int	柱状图图表展示方向, 0 表示垂直, 1 表示水平	<pre>Form.hBarChart.displayDirection=0</pre>
legendFont	QFont	图例文本字体	<pre>var fontMap={}; fontMap["weight"]=12; fontMap["pointSize"]=12; fontMap["family"]="宋体"; fontMap["italic"]=true; fontMap["underline"]=true; Form.hBarChart.legendFont=HProperty.setQFont(fontMap)</pre>
legendPosition	int	图例显示位置, 0 表示 topLeft, 1 表示 topCenter, 2 表示 topRight,	<pre>Form.hBarChart.legendPosition=1</pre>

		3 表示 horizontalLeft, 4 表示 horizontalRight, 5 表示 bottomLeft, 6 表示 bottomCenter, 7 表示 bottomRight	
legendTextColor	QColor	图例文本 颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.legendTextColor=H Property.setQColor(colorMap)</pre>
legendVisible	bool	图例可见 性	Form.hBarChart.legendVisible=true
seriesSpacing	int	柱状图系 列间距	Form.hBarChart.seriesSpacing=2
title	QString	柱状图标 题	Form.hBarChart.title=" 标题"
titleColor	QColor	柱状图标 题文本颜 色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0;</pre>

			<pre>colorMap["blue"]=0; Form.hBarChart.titleColor=HProperty.setQColor(colorMap)</pre>
titleFont	QFont	柱状图标题文本字体	<pre>var fontMap={}; fontMap["weight"]=12; fontMap["pointSize"]=12; fontMap["family"]="宋体"; fontMap["italic"]=true; fontMap["underline"]=true; Form.hBarChart.titleFont=HProperty.setQFont(fontMap)</pre>
titleVisible	bool	柱状图标题可见性	<pre>Form.hBarChart.titleVisible=true</pre>
xAxisColor	QColor	X轴轴线颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.xAxisColor=HProperty.setQColor(colorMap)</pre>
xAxisVisible	bool	X轴轴线可见性	<pre>Form.hBarChart.xAxisVisible=true</pre>
xGridColor	QColor	X轴网格线颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.xGridColor=HProperty.setQColor(colorMap)</pre>
xGridLineType	int	X轴网格线	<pre>Form.hBarChart.xGridLineType=1</pre>

		线型，0 表示虚线，1 表示实线，2 表示点线	
xGridVisible	bool	X轴网格线可见性	Form.hBarChart.xGridVisible=true
xLabelRotation	int	X轴文本标签倾斜度	Form.hBarChart.xLabelRotation=10
xTitle	QString	X轴标题	Form.hBarChart.xTitle=" X"
xTitleColor	QColor	X轴标题颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.xTitleColor=HProperty.setQColor(colorMap)</pre>
xTitleFont	QFont	X轴标题字体	<pre>var fontMap={}; fontMap["weight"]=12; fontMap["pointSize"]=12; fontMap["family"]="宋体"; fontMap["italic"]=true; fontMap["underline"]=true; Form.hBarChart.xTitleFont=HProperty.setQFont(fontMap)</pre>
xTitleVisible	bool	X轴标题可见性	Form.hBarChart.xTitleVisible=true
xUpperEnding	int	X轴轴向终点样式，0 表示	Form.hBarChart.xUpperEnding=1

		esNone, 1 表示 esFlatArrow, 2 表示 esSpikeArrow, 3 表示 esLineArrow, 4 表示 esDisc, 5 表示 esSquare, 6 表示 esDiamond, 7 表示 esBar, 8 表示 esHalfBar, 9 表示 esSkewedBar	
yAxisColor	QColor	Y轴轴线颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.yAxisColor=HProperty.setQColor(colorMap)</pre>
yAxisMaximum	int	Y轴数值范围最大值	<pre>Form.hBarChart.yAxisMaximum=20</pre>

yAxisMinimum	int	Y轴数值范围最小值	Form.hBarChart.yAxisMinimum=0
yAxisStep	int	Y轴刻度步长	Form.hBarChart.yAxisStep=5
yAxisValRangeAutoAdjust	bool	Y轴数值范围是否自适应	Form.hBarChart.yAxisValRangeAutoAdjust=true
yAxisVisible	bool	Y轴轴线可见性	Form.hBarChart.yAxisVisible=true
yGridColor	QColor	Y轴网格线颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.yGridColor=HProperty.setQColor(colorMap)</pre>
yGridLineType	int	Y轴网格线线型，0表示虚线，1表示实线，2表示点线	Form.hBarChart.yGridLineType=1
yGridVisible	bool	Y轴网格线可见性	Form.hBarChart.yGridVisible=true
yTitle	QString	Y轴标题	Form.hBarChart.yTitle=" Y"
yTitleColor	QColor	Y轴标题文本颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.hBarChart.yTitleColor=HProperty</pre>

			erty.setQColor(colorMap)
yTitleFont	QFont	Y轴标题字体	<pre> var fontMap={}; fontMap["weight"]=12; fontMap["pointSize"]=12; fontMap["family"]="宋体"; fontMap["italic"]=true; fontMap["underline"]=true; Form.hBarChart.yTitleFont=HProperty.setQFont(fontMap) </pre>
yTitleVisible	bool	Y轴标题可见性	Form.hBarChart.yTitleVisible=true
yUpperEnding	int	Y轴轴向终点样式，0 表示 esNone，1 表示 esFlatArrow，2 表示 esSpikeArrow，3 表示 esLineArrow，4 表示 esDisc，5 表示 esSquare，6 表示 esDiamond，7 表示	Form.hBarChart.yUpperEnding=1

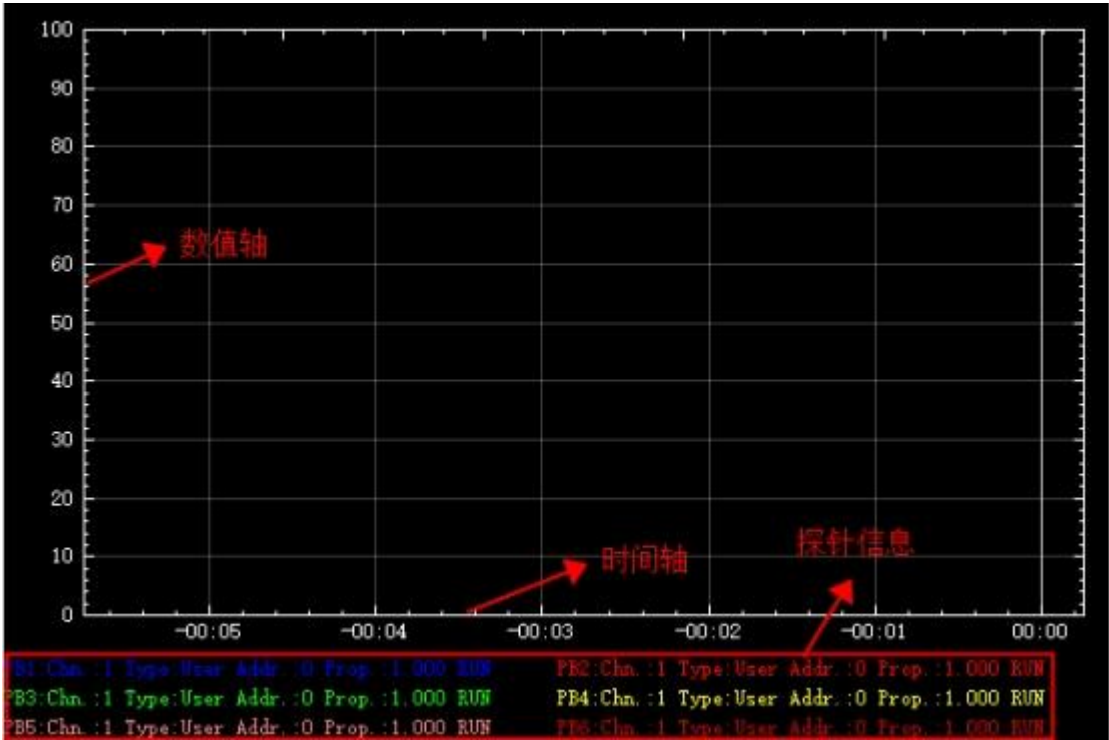
		esBar , 8 表 示 esHalfBar , 9 表 示 esSkewedB ar	
zoomEnabled	bool	柱 状 图 是 否 启 用 缩 放	Form.hBarChart.zoomEnabled=true

6.22. 变量示波器

6.22.1. 功能介绍

变数示波器对变数进行连续采样,将数字转为可视化的波形图来帮助工程师和技术人员分析问题。

6.22.2. 组件结构



1. 数值轴

数值轴用于显示变数数值对应位置。

2. 时间轴

时间轴是示波器上用于表示变数在时间上的变化的水平轴。

3. 探针信息

显示探针信息，目前支持最多六个探针。

PB1: Chn:1 Type: User Addr: 0 Prop: 1.000 RUN

探针序号 通道 地址类型 地址 缩放比例 运行状态

6.22.3. 属性编辑器

1. 刷新时间

设置绘制的刷新周期，单位为 ms。该值设置越小，则刷新速度越快，但是会导致 CPU 资源消耗增加。

2. 时间范围

设置时间轴的时间长度，单位为 s。

3. 时间格式

- 1) HHMMSS
- 2) MMSS
- 3) SS

4. 时间轴显示

设置是否显示时间轴

5. 竖直标尺最小值

设置数值轴的最小值。当波形不在视口范围内时，可调整数值轴的最小、最大值将波形完全显示。

6. 竖直标尺最大值

设置数值轴的最大值。当波形不在视口范围内时，可调整数值轴的最小、最大值将波形完全显示。

7. 竖直标尺可见

设置是否显示竖直轴。

8. 背景颜色

设置背景颜色。

9. 零线颜色

设置数值轴零刻度线颜色。

10. 网格线颜色

设置网格线的颜色。

11. 网格显示

设置网格是否显示。

12. PBx 显示

设置探针 x 是否显示。

13. PBx 通道

设置探针 x 使用的通道。

14. PBx 数据类型

设置探针 x 的数据类型。

15. PBx 地址

设置探针 x 的地址。

16. PBx 线宽

设置探针 x 的波形线宽。

17. PBx 描绘颜色

设置探针 x 的颜色。

18. 设备

设置使用远程可视化时，组件连接的设备。

6.22.4. 参数设置弹窗

在运行阶段，可以使用宏 paraDataSetDialog 调出参数设置弹窗。参数设置弹窗共有“显示”“数值设置”“文件设置”三页。

【显示】



1. 刷新周期

设置绘制的刷新周期，单位为 ms。该值设置越小，则刷新速度越快，但是会导致 CPU 资源消耗增加。

2. 水平轴

1) 水平轴样式

无效属性。

2) 显示时长

设置时间轴的时间长度，单位为 s。

3) 样式

- a. HHMMSS
- b. MMSS
- c. SS

4) 可见

设置水平轴是否可见。

3. 垂直轴

1) 最小值

设置数值轴的最小值。

2) 最大值

设置数值轴的最大值。

3) 可见

设置垂直轴是否可见。

4. 零线

设置垂直轴的零线颜色。

5. 背景

设置背景颜色。

6. 栅格

设置栅格颜色。

7. 探针适中 (ZoomAuto)

设置进行适中的轴，将该轴的波形完全显示在视口内，调用 setPBZoomAuto 后生效。

8. 小数位数

设置垂直轴的小数位数。

【数值设置】

参数设定

显示

数值设置

文件设置

Device:

Device0

	地址	颜色	线宽	比例	启用
PB1	[i]USR1		1	1.000	
PB2	[i]USR0		1	1.000	
PB3	[i]USR0		1	1.000	
PB4	[i]USR0		1	1.000	
PB5	[i]USR0		1	1.000	
PB6	[i]USR0		1	1.000	
H-Axis	[i]USR0				

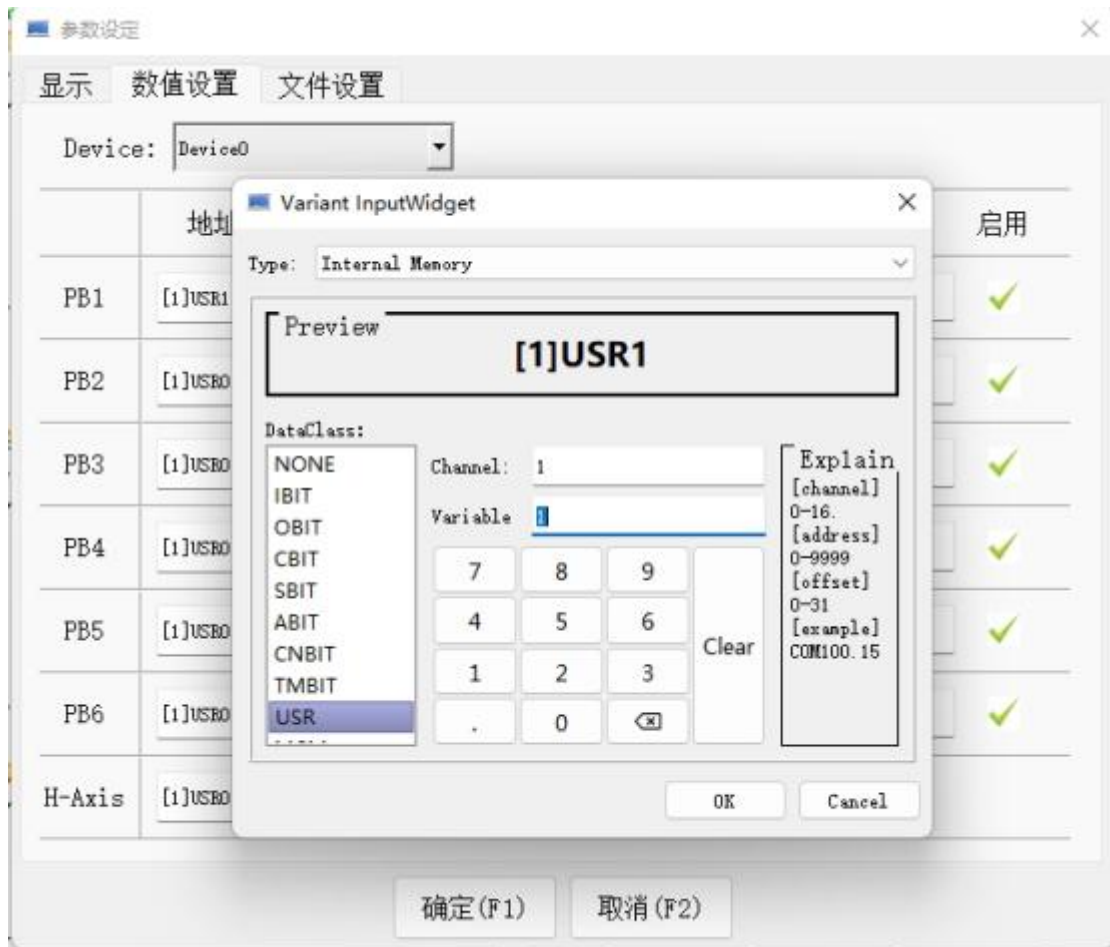
确定 (F1)

取消 (F2)

1. Device (设备)

设置使用远程可视化时，组件连接的设备。

2. 地址



点击地址栏时，使用变数输入弹窗编辑变数地址，详情可见“[变数输入弹窗](#)”。

3. 颜色

设置探针对应波形的颜色。

4. 线宽

设置探针对应波形的线宽。

5. 比例

设置探针对应波形的缩放比例。

6. 启用

设置探针是否启用，启用探针后方可绘制波形。

7. H-Axis

无效参数。

【文件设置】

规划中的功能，将波形数据保存到文件，且可以从文件中载入波形数据。目前未实现。

6.22.5. 脚本宏

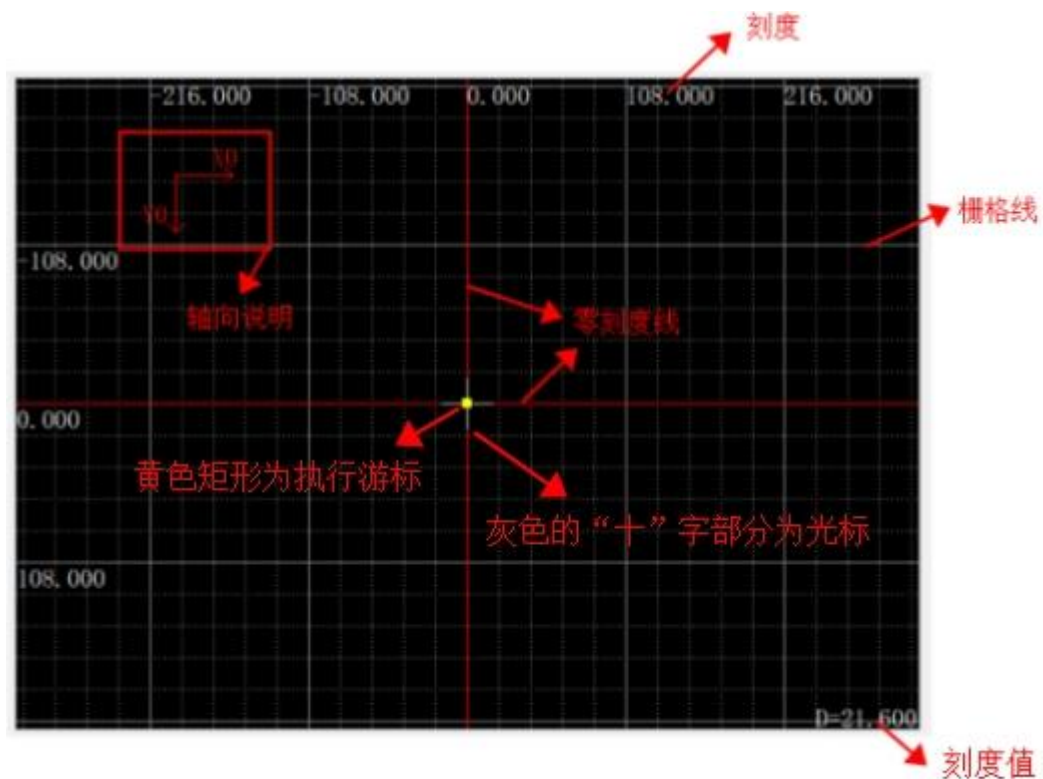
详情可见“HCurvsGraphics_A4_Designer_CN.xml”。

6.23. 绘图

6.23.1. 功能介绍

描图组件具有静态描图与动态描图两个主要功能，静态描图用来预览程序轨迹，动态描图绘制实际运行时的程序轨迹。

6.23.2. 组件结构



1. 轴向说明

轴向通常指的是机床的坐标轴方向，指示当前设置的轴向。

2. 零刻度线

零刻度线是指标尺或刻度尺上的参考线，通常用于确定测量起点或基准位置。它通常表示为一条明显的线或标记，用于确定起始位置。

3. 刻度

刻度是用于标记图形位置的等分线。在绘图中，刻度按照一定的比例进行划分，以便于确定位置。

4. 栅格线

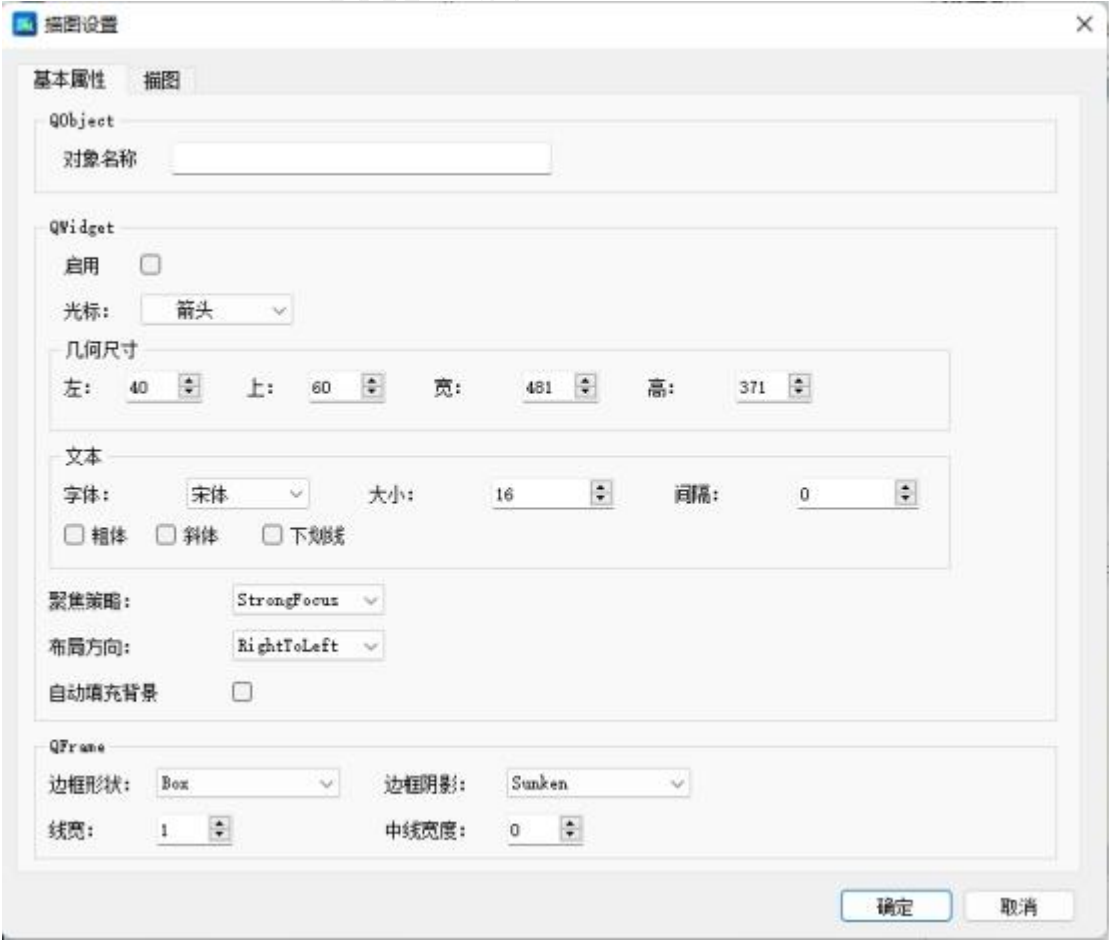
栅格线是一种用于绘图中的参考线，以等间隔的平行线形式出现。

5. 刻度值

栅格线间的实际步长。

6.23.3. 属性设置弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“基本属性”、“描图”两页。



【一般属性】

1. 对象名称

每个对象都可以通过设置对象名称（Object Name）来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 几何尺寸

该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

3. 文本

设置组件的文本字体。各项参数说明可见“[字体](#)”。

4. 边框

设置组件的边框。具体参数说明可见“[框架](#)”。

【描图】



1. 数据类型

设置绘图的数据来源。可设置为以下选项：

- 1) 程序坐标（加减速前）
- 2) 程序坐标（加减速后）
- 3) 机械坐标
- 4) 相对坐标
- 5) 回授坐标

2. 通道

指定绘图的数据来源的通道。

3. 轴向

- 1) 水平轴

指定绘图的水平轴对应机台的实际轴向。

- 2) 垂直轴

指定绘图的垂直轴对应机台的实际轴向。

- 3) 竖直轴

指定绘图的竖直轴对应机台的实际轴向。

- 4) 镜像轴

指定绘图的镜像轴，指定镜像轴后，绘制时会进行镜像的绘制。



1. 绘图

1) 动态描图

绘制程序实际运行时的轨迹，可设置是否启用动态描图，以及动态描图轨迹的颜色。

2) 静态描图

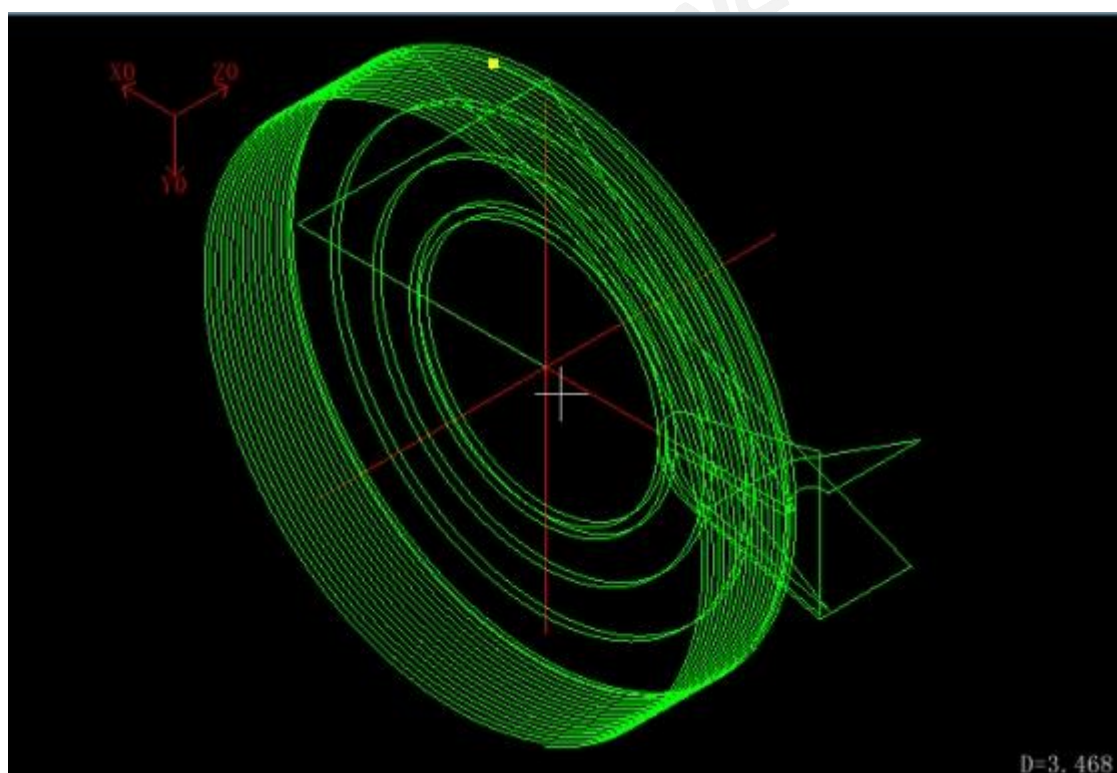
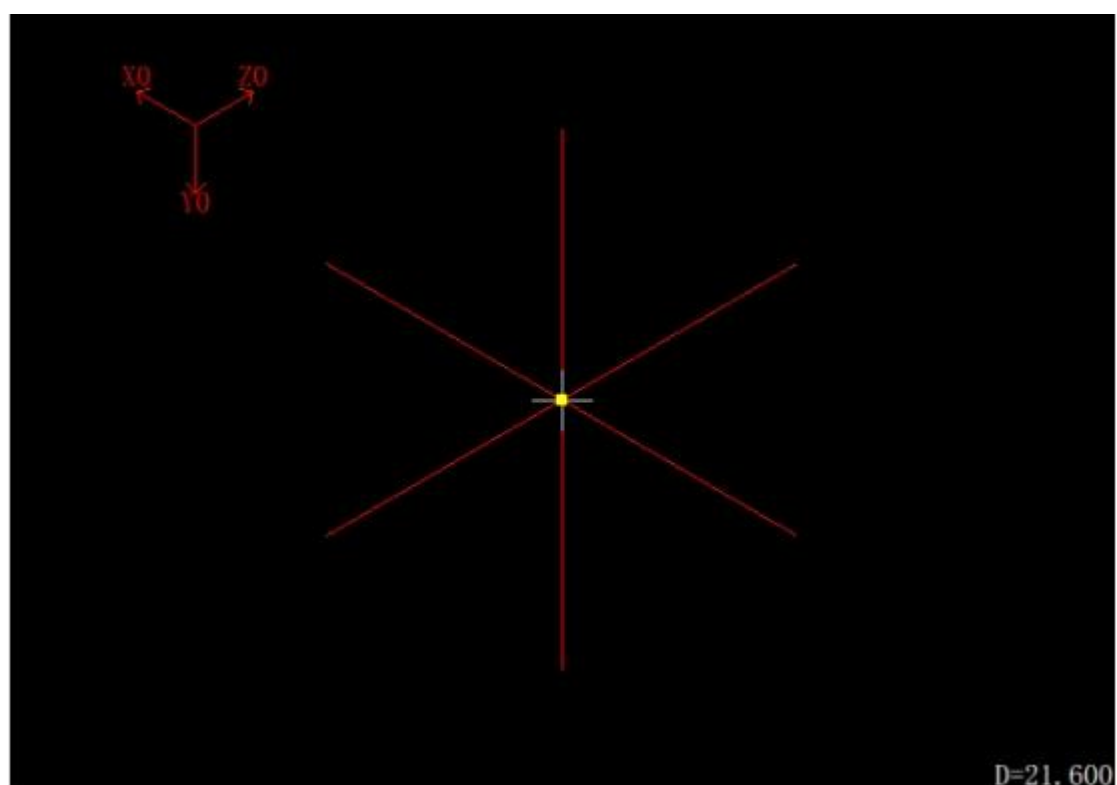
预览程序运行轨迹。静态描图可设置 G0 代码、其它 G 代码的颜色，用以区别快速定位和加工轨迹。

2. 视口

3) 等轴测显示

等轴测是一种用于绘制立体物体的图形表示的方法，它能够在二维平面上展示物体的三维形状。在等轴测中，物体的三个坐标轴（通常为 x、y、z 轴）在图纸上以等角度相交，从而使得物体在图纸上的表现更加直观和真实。

等轴测下的视图（注意需要设置水平轴、垂直轴、竖直轴之后，等轴测才可生效）：



4) 背景

设置视口背景色。

5) 零线

设置零线颜色。

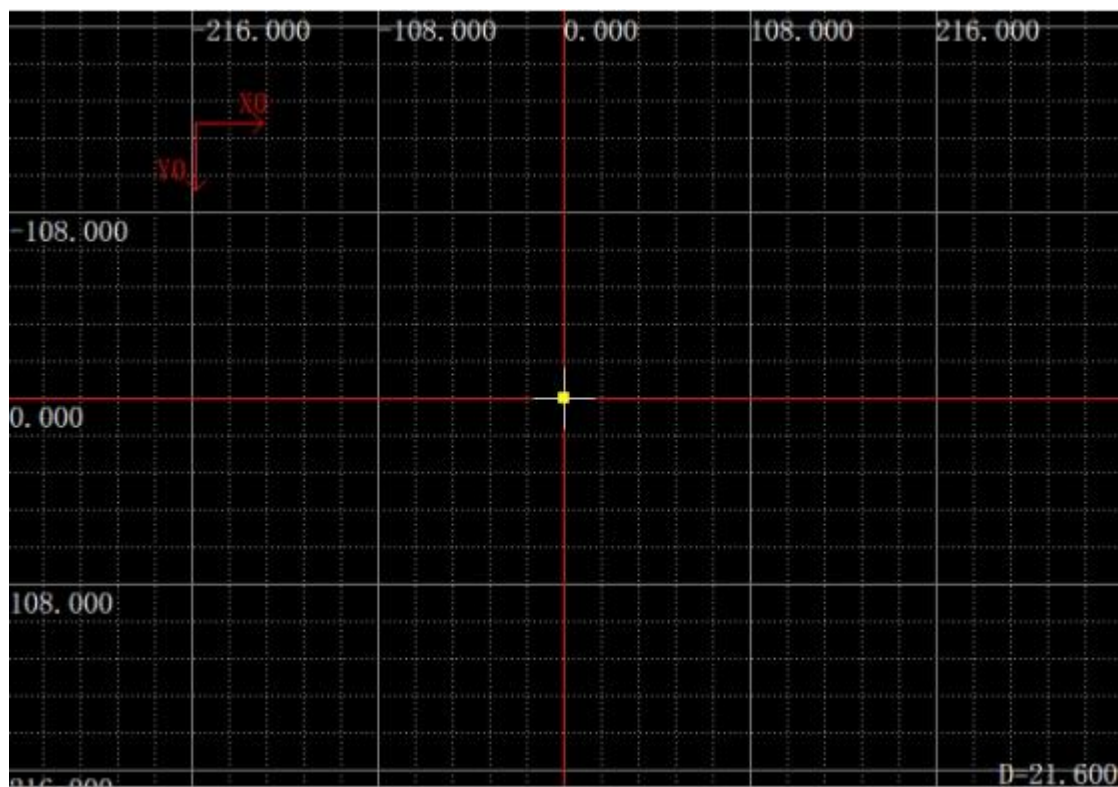
6) 文本

设置文本颜色。

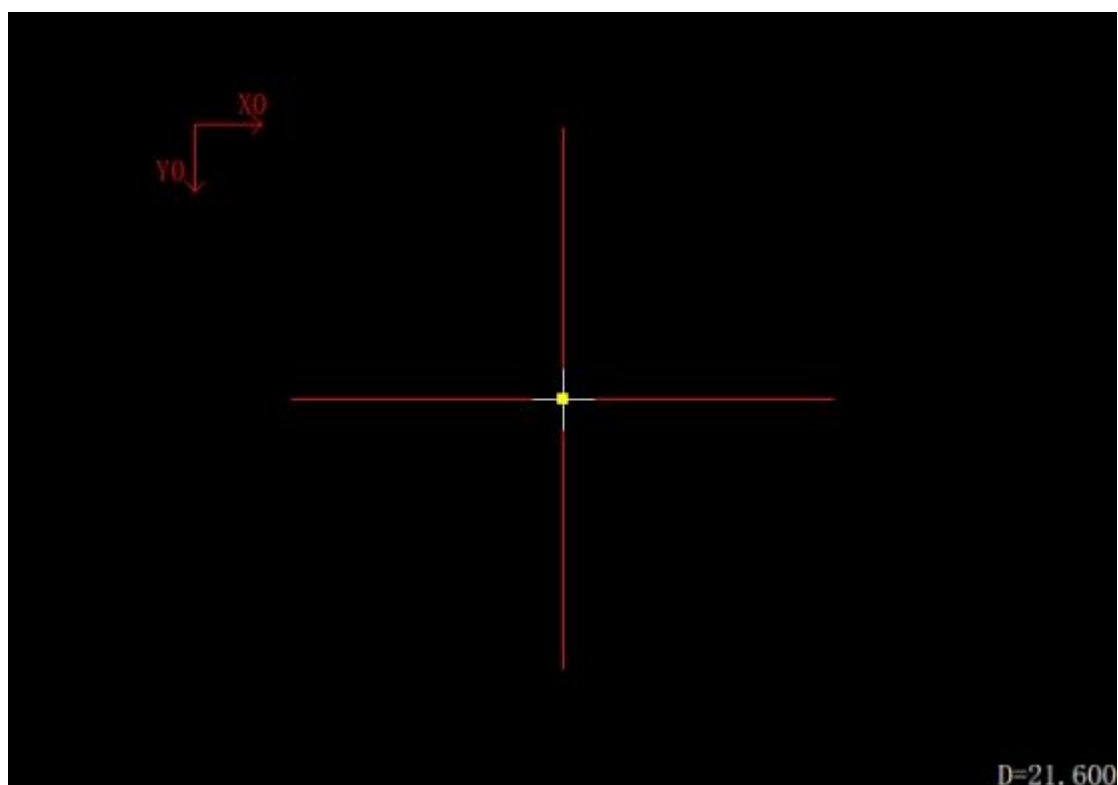
3. 栅格

7) 显示

显示栅格的样式（注意，设置竖直轴后，栅格会自动隐藏）



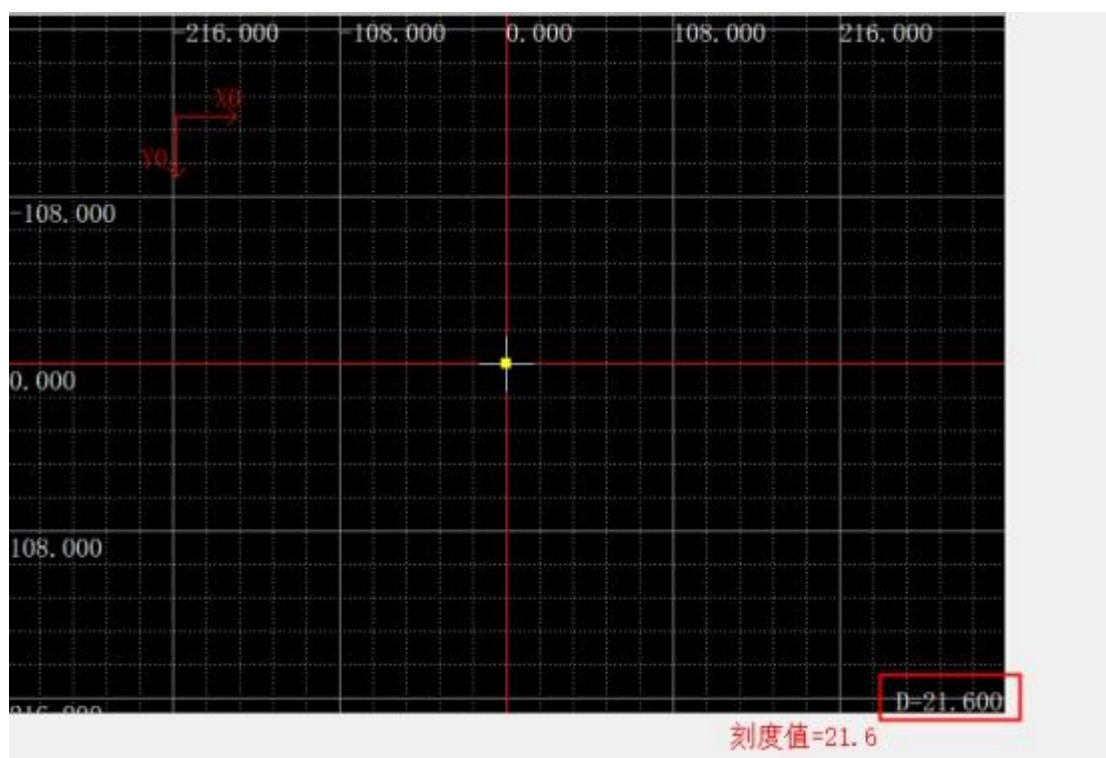
不显示栅格时样式：



8) 步长

设置栅格的步长。实际上步长会根据缩放比例进行缩放，实际栅格步长= 设置值/缩放比例。

因此实际的栅格步长由“刻度值”表示。



9) 颜色

设置栅格线的颜色。

4. 执行游标



动态描图时，执行游标指示当前绘制的位置。

10) 显示

设置执行游标是否显示。

11) 大小

设置执行游标的大小。

12) 颜色

设置执行游标的颜色。

5. 静态游标

静态描图时，静态游标指示当前绘制的位置。

13) 显示

设置静态游标是否显示。

14) 大小

设置静态游标的大小。

15) 颜色

设置静态游标的颜色。

6. 光标

16) 大小

设置光标的大小。

17) 颜色

设置光标的颜色。

6.23.4. 属性编辑器

属性编辑器提供了便捷的属性设置方式，不需要双击打开属性设置弹窗。属性编辑器里大部分的属性设置，功能与属性设置弹窗一致。这里只介绍属性设置弹窗里未介绍的功能。

1. 刷新周期

设置绘制的刷新周期。该值设置越小，则刷新速度越快，但是会导致 CPU 资源消耗增加。

2. 游标移动量

使用“上”“下”“左”“右”的宏移动光标时，每次移动的距离（像素）。

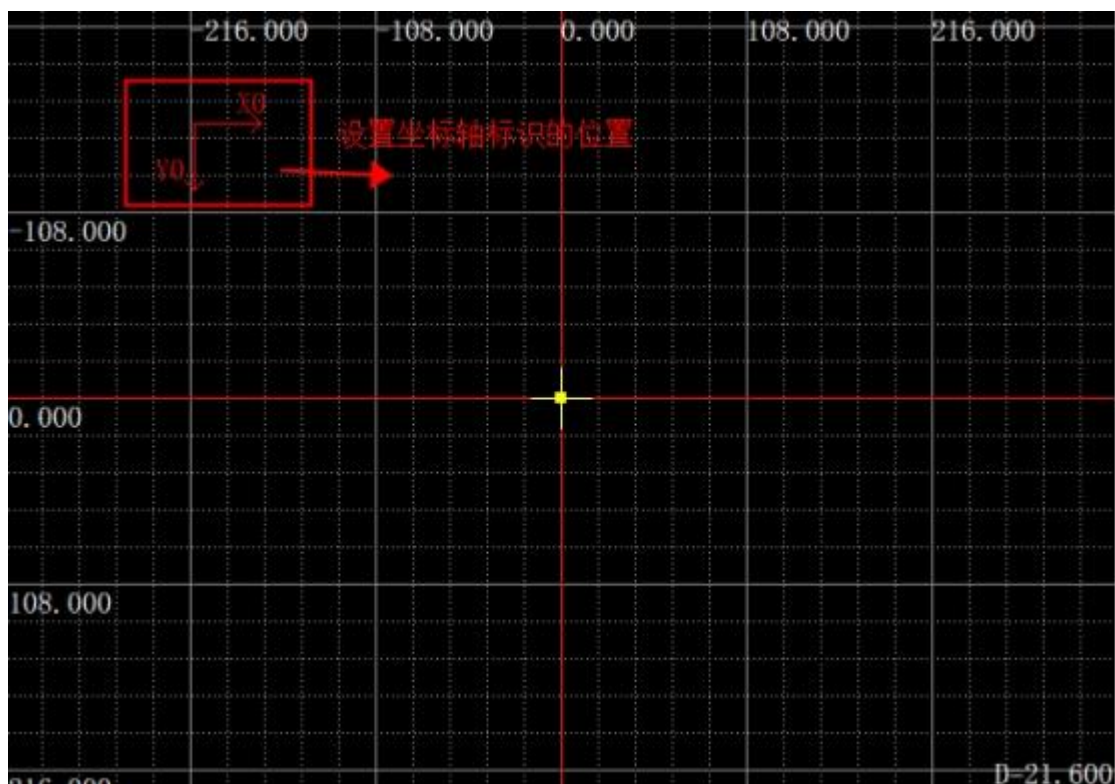
3. 中心移动量

使用“上”“下”“左”“右”的宏移动视口时，每次移动的距离（像素）。

4. 缩放增量值

使用“放大”“缩小”的宏缩放视口时，每次缩放的比例。

(1) 零点坐标系位置



5. 执行游标形状

执行游标可以设置为 Rect (矩形), 和 Circle (圆形)

6. 游标颜色

即光标颜色。

7. Dxf 颜色

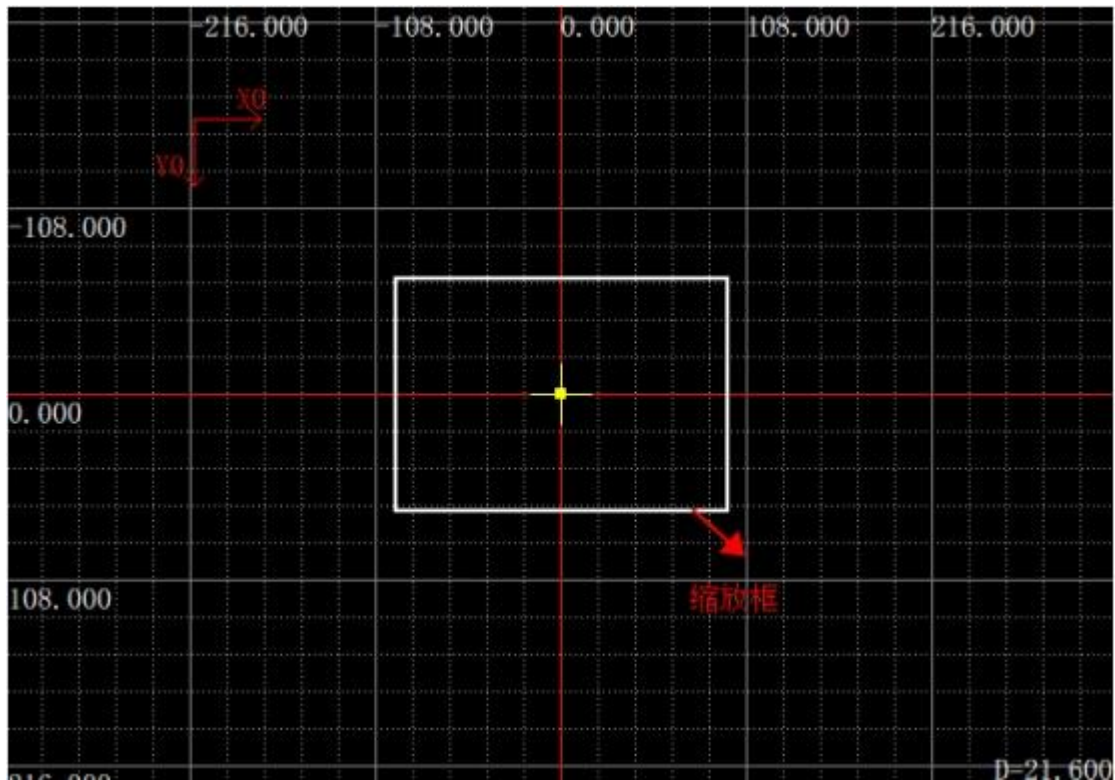
无效属性。

8. 绘图颜色

无效属性。

9. 缩放框颜色

缩放框的作用是对框选部分进行缩放，可设置缩放框的颜色。



10. 缩放框线宽

设置缩放框的线宽。

11. 缩放框移动增量

使用“上”“下”“左”“右”移动缩放框时，每次移动距离=视口宽度*缩放框移动增量

12. 缩放框缩放增量

使用“放大”“缩小”对缩放框进行缩放时，每次缩放的比例。

13. 缩放框线型

缩放框的线型可设置为 SolidLine (实线)、DashLine (虚线)

14. 设备

设置使用远程可视化时，组件连接的设备。

6.23.5. 脚本宏

详情可见“HGraphics_Designer_CN.xml”。

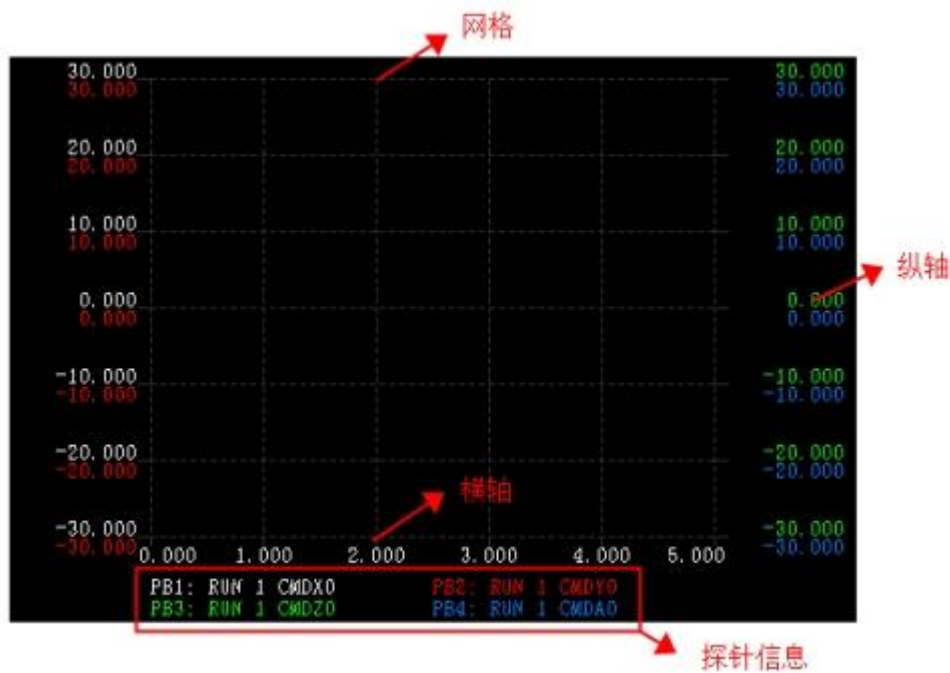
6.24. 折线图

6.25. 示波器

6.25.1. 功能介绍

示波器对程序执行时的轴向数据进行连续采样，将数字转为可视化的波形图来帮助工程师和技术人员分析问题。

6.25.2. 组件结构



1. 纵轴

纵轴为数值轴，以不同颜色显示各个数据对应的数值。

2. 横轴

横轴为时间轴，用于表示数据在时间上的变化。

3. 网格

网格是一种用于绘图中的参考线，以等间隔的平行线形式出现。

4. 探针信息

显示探针信息，目前支持最多四个探针。

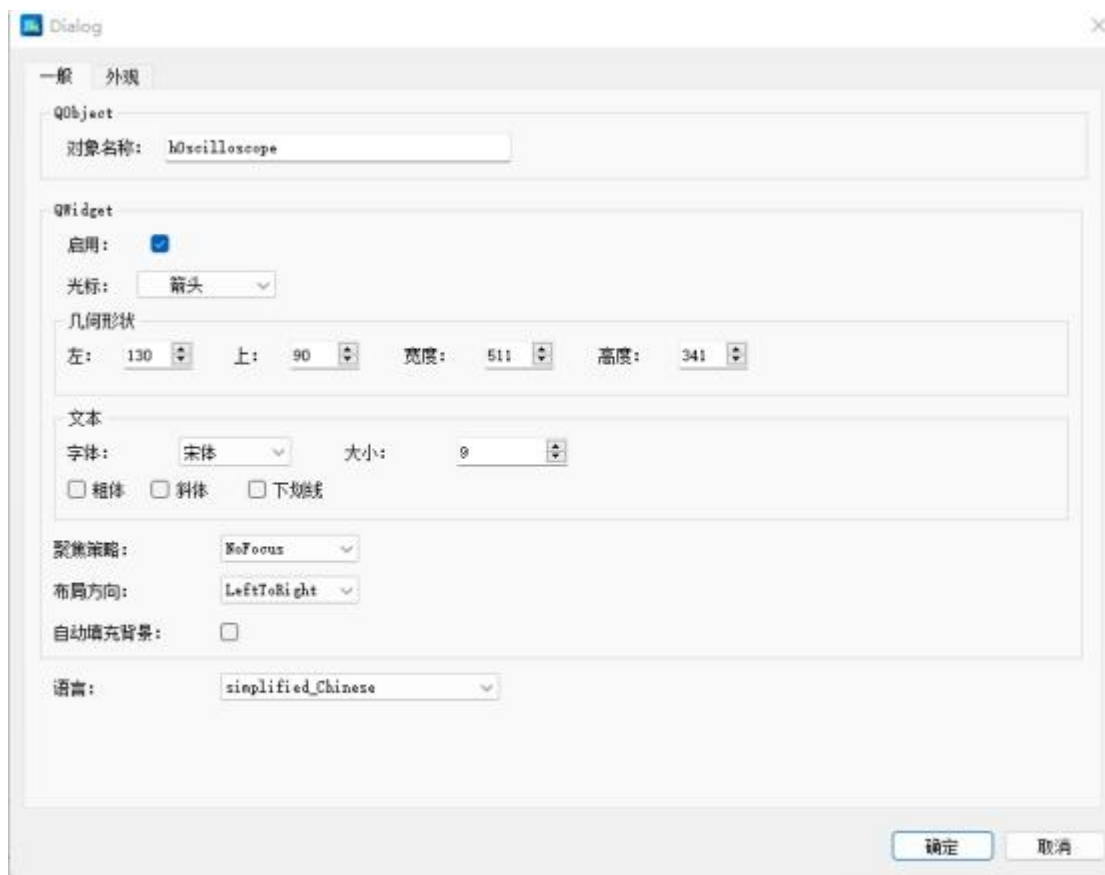
PB1: RUN 1 CMD X0

探针序号 运行状态 通道 类型 轴

6.25.3. 属性编辑弹窗

在 HMI 软件编辑界面，使用鼠标左键双击组件，即可弹出属性设置界面。属性设置弹窗包含“一般”、“外观”两页。

【一般】



1. 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在代码中引用和操作对象。

2. 几何尺寸

该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

3. 文本

设置组件的文本字体。各项参数说明可见“[字体](#)”。

【外观】

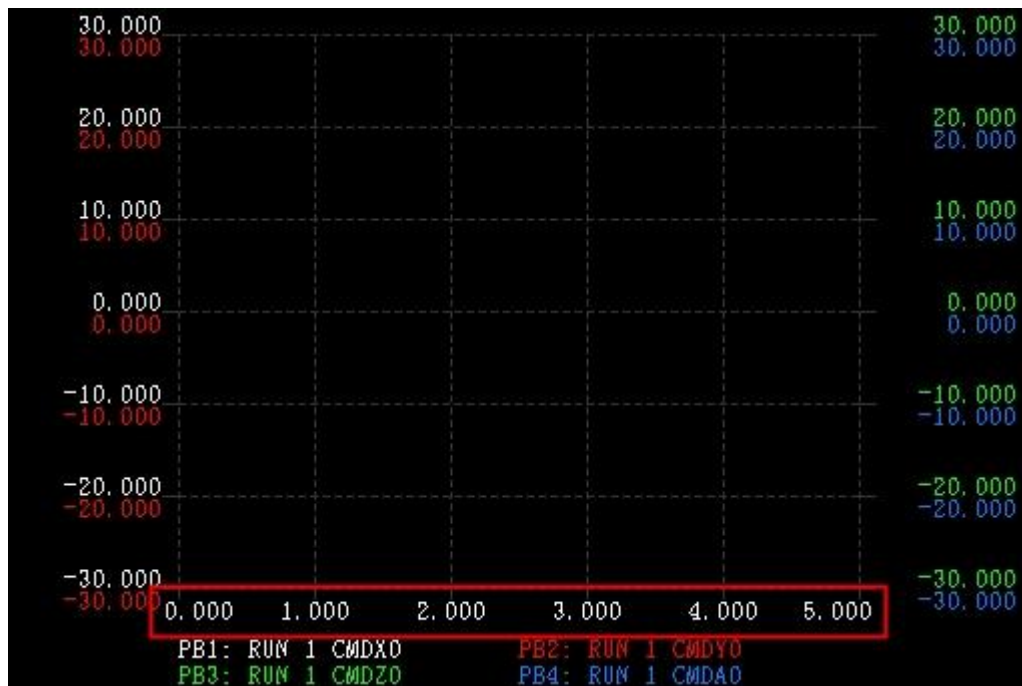


4. 刷新周期

设置绘制的刷新周期，单位为 ms。该值设置越小，则刷新速度越快，但是会导致 CPU 资源消耗增加。

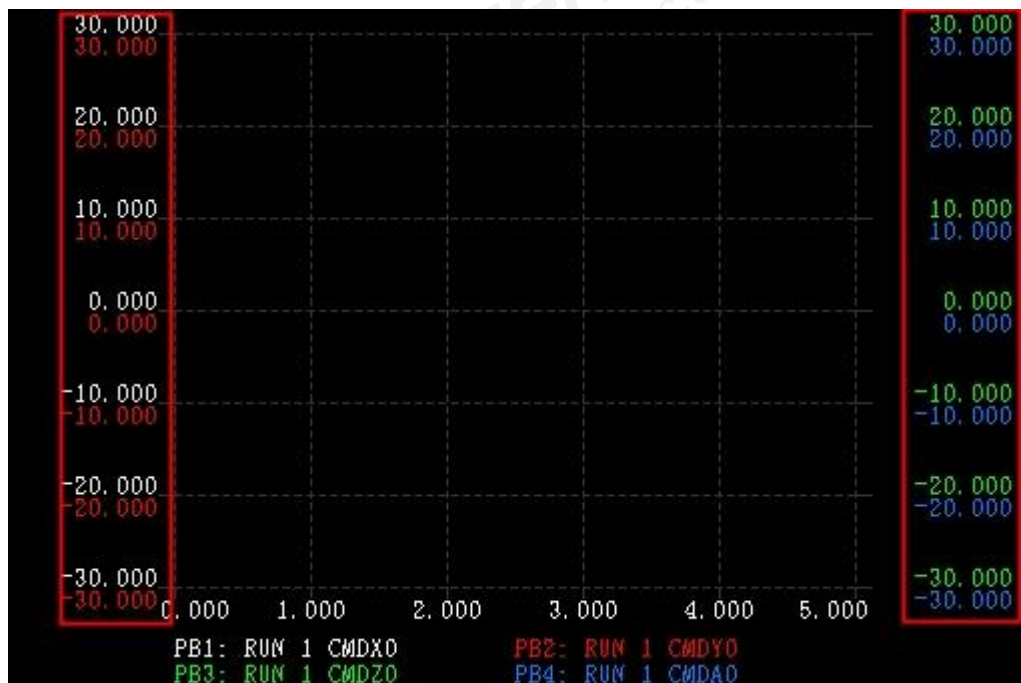
5. 水平文本显示

设置水平轴文本是否显示。



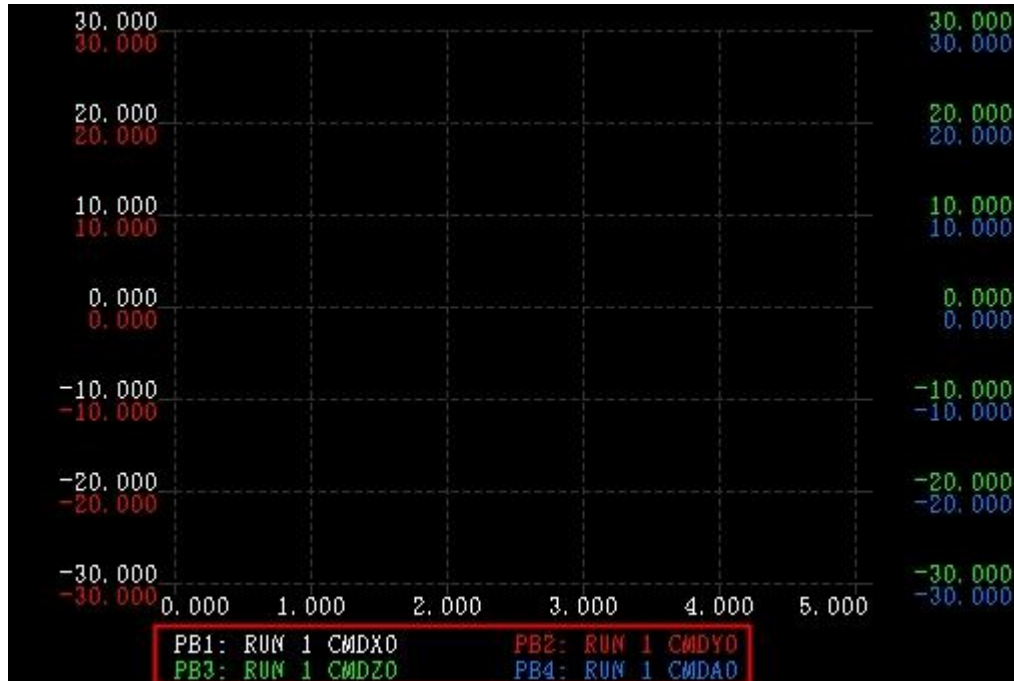
6. 垂直文本显示

设置垂直轴文本是否显示。



7. 探针标识显示

设置探针标识是否显示。



8. 网格是否显示

设置网格是否显示。

9. 网格颜色

设置网格颜色。

10. 水平网格数量

设置水平网格的数量。

11. 水平缩放增量

设置水平缩放增量，在使用“放大”“缩小”宏方法对水平轴进行缩放时， $\text{缩放比例} = \text{缩放比例} * \text{水平缩放增量}$ 。

12. 水平步长

设置水平网格的步长。

13. 水平缩放比例。

设置水平缩放比例。

14. 垂直网格数量

设置垂直网格数量。

15. 垂直缩放增量

设置垂直缩放增量，在使用“放大”“缩小”宏方法对垂直轴进行缩放时， $\text{缩放比例} = \text{缩放比例} * \text{垂直轴缩放增量}$ 。

16. 垂直步长

设置垂直步长。

17. 垂直移动增量

设置垂直移动增量，在使用“上”“下”宏方法对垂直轴进行移动时，每次移动的距离（像素）。

18. PBx 显示

设置探针数据是否显示，勾选后方可绘制波形。

19. PBx 取反

设置探针数据是否取反。

20. PBx 线性

设置探针数据是否以直线连接。不勾选该选项时，探针数据将以散点图的形式进行绘制。

21. PBx 比例

设置探针数据的缩放比例。

22. PBx 零线

设置探针数据的零线位置。

23. PBx 颜色

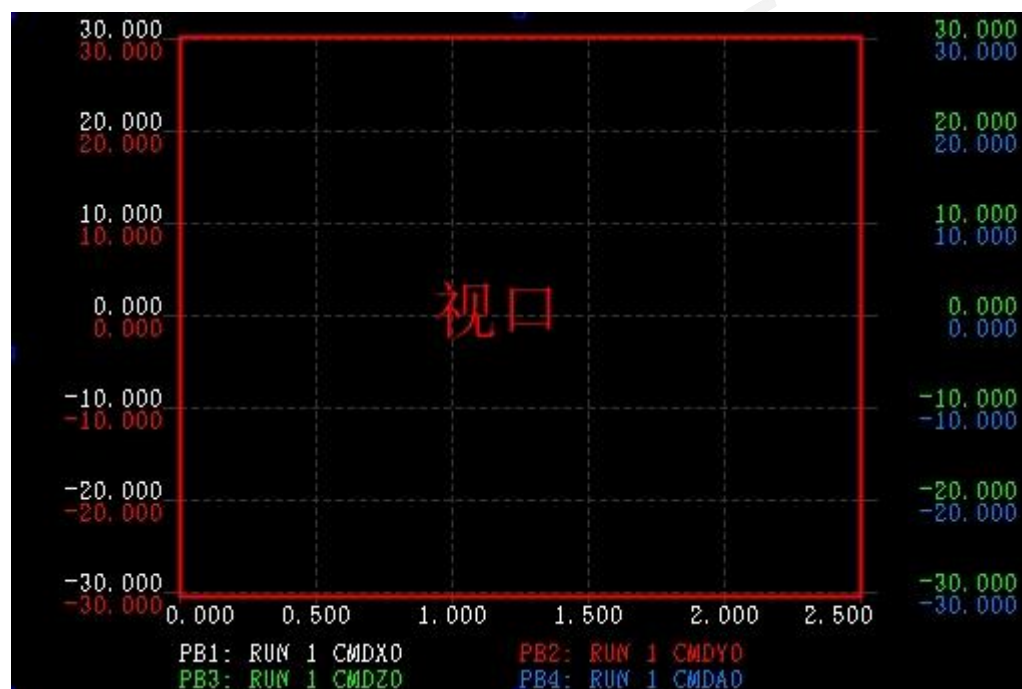
设置探针波形的颜色。

24. 背景颜色

设置背景颜色。

25. 视口颜色

设置视口的颜色。



26. 水平文本颜色

设置水平文本颜色。

27. 字体

设置所有文本的字体，各项参数详情可见“[字体](#)”。

6.25.4. 脚本宏

详情可见“H0scilloscope_Designer_CN.xml”。

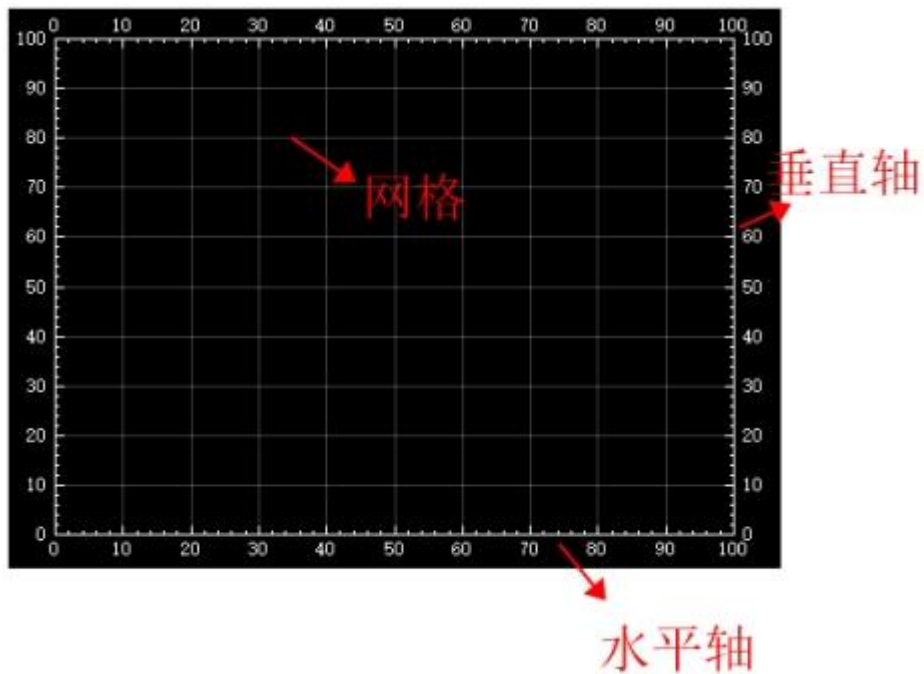
6.26. 饼状图

6.27. 自定义绘图

6.27.1. 功能介绍

自定义绘图组件供用户绘制自定义的图形，例如直线、圆弧、矩形。

6.27.2. 组件结构



1. 水平轴

在坐标平面中，横轴通常与纵轴（y 轴）垂直相交，共同构成了一个二维坐标系，用来描述平面上的点的位置。

2. 垂直轴

在坐标平面中，纵轴通常与横轴（x 轴）垂直相交，共同构成了一个二维坐标系，用来描述平面上的点的位置。

3. 网格

网格是一种用于绘图中的参考线，以等间隔的平行线形式出现。

6.27.3. 属性编辑器

1. 水平步长

水平步长是指在一个坐标系或图表中，横轴上相邻两个刻度之间的距离。

2. 垂直步长

垂直步长是指在一个坐标系或图表中，纵轴上相邻两个刻度之间的距离。

3. 水平标尺最小值

水平标尺的最小值是指水平标尺上显示的最小刻度值。

4. 水平标尺最大值

水平标尺的最大值是指水平标尺上显示的最大刻度值。

5. 竖直标尺最小值

竖直标尺的最小值是指竖直标尺上显示的最小刻度值。

6. 竖直标尺最大值

竖直标尺的最大值是指竖直标尺上显示的最大刻度值。

7. 水平标尺可见

设置水平标尺是否可见。

8. 垂直标尺可见

设置垂直标尺是否可见。

9. 网格显示

设置网格是否可见。

10. 背景颜色

设置背景颜色。

11. 零线颜色

设置横轴、纵轴零刻度线的颜色。

12. 网格线颜色

设置网格线的颜色。

13. 动态描图颜色

无效属性。

14. 等比例

坐标零点居中，按照步长绘制刻度。

15. 水平反向

水平轴的方向取反。

16. 垂直反向

垂直轴的方向取反。

6.27.4. 脚本宏

详情可见“HUserGraphics_Designer_CN.xml”。

6.28. 计算器

6.29. 权限管理

6.29.1. 组件外观



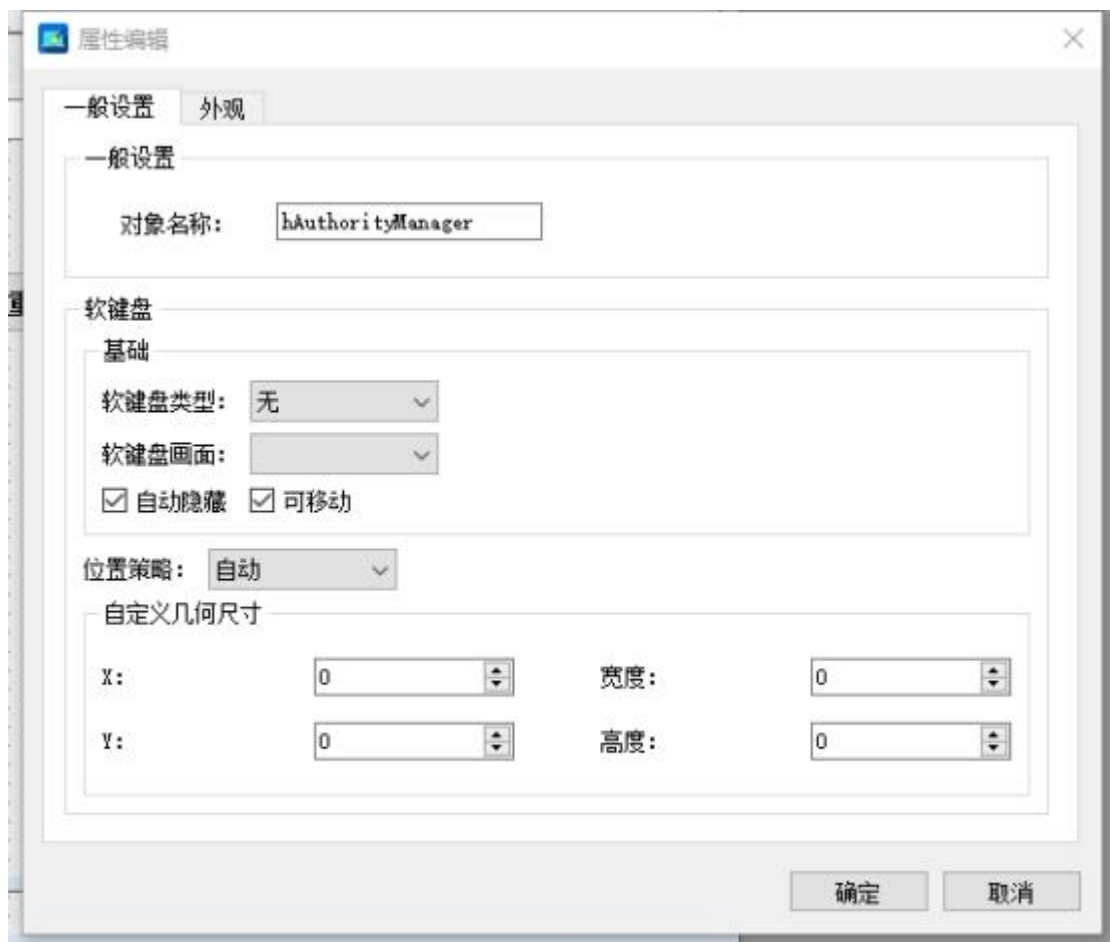
组件包括等级选择下拉框、密码输入栏、记住密码勾选框、登录按钮、取消按钮和重置密码按钮组成。

6.29.2. 简介

权限管理插件提供登录 HMI 权限、重置权限密码的功能。

6.29.3. 属性编辑弹窗

属性编辑弹窗包含“一般设置”和“外观”两页。



1. 一般设置

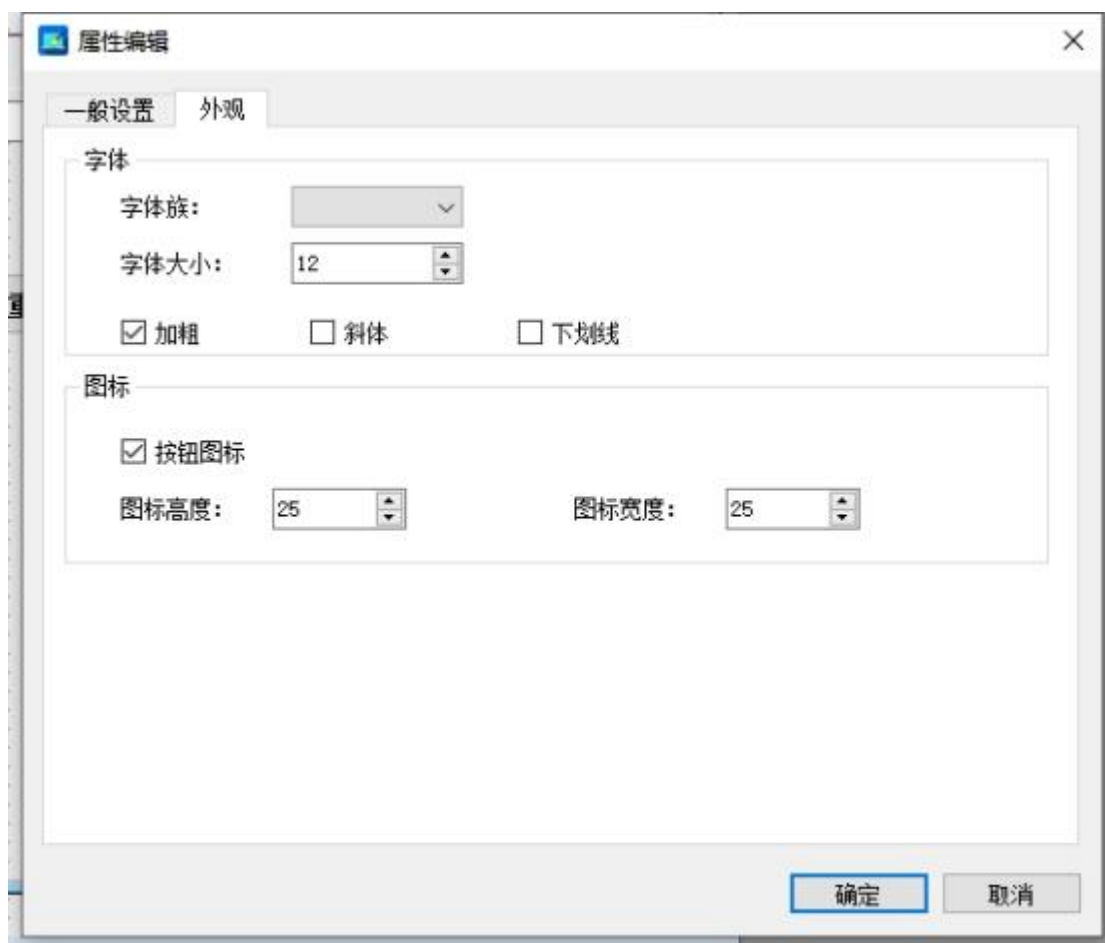
1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在脚本代码中引用和操作对象。

2. 软键盘

使用触控的方式输入时，可以使用软键盘。(跳转到软键盘说明)，默认是使用密码输入键盘。

6.29.4. 外观



1. 字体

插件整体字体设置。

2. 图标

插件整体图标设置，图标包括登录按钮、取消按钮和重置密码按钮上的三个图标。

6.29.5. 属性编辑器

属性编辑器默认位置可视化开发软件的右侧，所有属性均在属性编辑弹窗上有体现。

HAuthorityManager	
按钮图标	☒
图标大小	25 x 25
控件字体	A [Simsun, 12]
软键盘	1, 1, 无, , 自动, 0, 0, 0, 0

6.29.6. 系统多国语言

插件上存在的固有文字，包括文件信息显示区域、插件提示文本等文字内容的翻译都可以在“工具”菜单栏中“组件固有文字翻译”中设置。



固有文字翻译编辑窗口功能请浏览相应的说明文档。

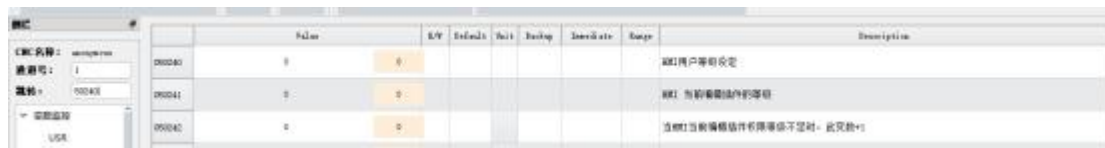
6.29.7. 功能详述

6.29.7.1. 权限

- 1. 输入类型的插件（数值插件、代码编辑器、表格等等）能够设置操作权限，插件默认权限为 0，即不设权限。
- 2. 当控制器的人机权限数值大于插件权限数值时，权限不足。（操作插件所需权限为 1，

控制器权限为 2 时权限不足无法操作插件。)

3. 权限相关的变量如下图所示:



Name	Value	Edit	Execute	Delete	Backup	Initialize	Range	Description
COM50240	0	0	0	0	0	0	0	HMI 用户等级设定
COM50241	0	0	0	0	0	0	0	HMI 当前编辑插件的等级
COM50242	0	0	0	0	0	0	0	当前编辑插件权限不足时, 此变量+1

COM50240: HMI 用户等级设定, 即当前控制器的人机权限。

COM50241: HMI 当前编辑插件的等级, 当前焦点部件所需权限。(当前焦点部件为数值插件、数值插件设置用户权限为 2, 那么 COM50241 变量值为 2)

COM50242: HMI 当前编辑插件权限不足时变数值递增。(控制器的人机权限为 2, 目标插件所需权限为 1, 当编辑目标插件时 COM50242 会+1, 意味需要登录更高权限等级) 应用工程师会关联这个变量, 弹出权限登录的窗口。

6.29.8. 权限设置

HMI 人机权限可以通过 finger hmi designer 软件设置。在“设置”-->“工程设置”-->“权限设置”页面。



1. “重置权限等级及密码”，勾选后，在更新画面工程时，会使用当前画面工程的权限设置信息覆盖控制器下的权限信息。如果不勾选，更新画面工程后将保持之前控制器下的权限设置。
2. “开机恢复默认等级”，勾选后，每次重启控制器都会自动将权限等级调整至目标等级，否则维持关机前的权限等级。
3. 能够通过“添加”和“移除”按钮调整权限数量，权限名称支持设置多国语言。

6.29.9. 登录权限



权限管理

等级: Lv. 1

密码:

☐ 记住密码

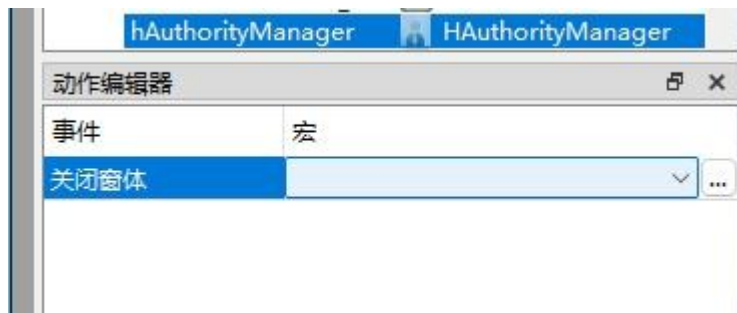
登录 (F1) 取消 (F2) 重置密码 (F3)

选择所需登录的权限等级，输入密码，登录到对应的权限等级。

6.29.10. 重置密码

选择需要重置密码的权限等级，需要先输入原有权限密码，之后两次输入重置的新密码，即可重置权限密码。

6.29.11. 关闭窗口体



动作编辑器里有个关闭窗口体事件，在登录成功、取消登录、重置密码成功时会发送一个 closewindows 信号，触发关闭窗口体事件，执行对应的宏程序，但实际不会关闭当前插件所在的窗体画面，目前 HMI 没有关闭主窗体画面的功能。

6.30. 万年历

6.31. 无线管理

6.31.1. 组件外观



插件以列表的形式，罗列出所有扫描出来的 wifi 名称。

6.31.2. 简介

无线管理插件主要用于连接 wifi。插件仅能在控制器下使用，需要搭配 wifi 模块。

6.31.3. 属性编辑弹窗

插件无属性编辑弹窗

6.31.4. 属性编辑器

WifiManager	
> 字体	A [SimSun, 10]
> 字体颜色	■ [0, 0, 0] (255)
> 背景颜色	■ [255, 255, 255] (255)
> 软键盘	1, 1, 无, , 自动, 0, 0, 0, 0

1. 字体，统一设置插件整体字体。
2. 字体颜色，统一设置插件整体字体颜色。
3. 背景颜色，插件背景颜色。
4. 软键盘，插件软键盘设置。

6.31.5. 功能详述

无线管理插件提供扫描 wifi、连接和断开 wifi 等功能。详细可见插件的接口说明。

6.31.6. 插件函数表

函数原型	描述	参数	返回值
connectWifi()	连接所选 WiFi，触发后将判断如果是记住密码的 WiFi 就直接连接，否则弹出密码输入窗口	无	无
cursorDown()	光标下移	无	无
cursorUp()	光标上移	无	无
disconnect()	断开当前已连接的 WiFi 无线网卡处于 Inactive（不活动的状态）	无	无
getWifiInformation()	获取已连接 WiFi 的信息	无	Array: 存储 WiFi 信息的字符串列表 Inf[0] = Ssid (WiFi 名) Inf[1] = Status (状态) Inf[2] = IP Inf[3] = Encryption (加密) Inf[4] = Authentication (认证) Inf[5] = signal (信号强弱) Inf[6] = bssid
scan()	扫描 WiFi，把能连接的	无	无

	WiFi 显示到主窗体。		
--	--------------	--	--

6.32. 代码显示器

6.32.1. 组件外观



组件由行号显示区域和文本显示区域组成。

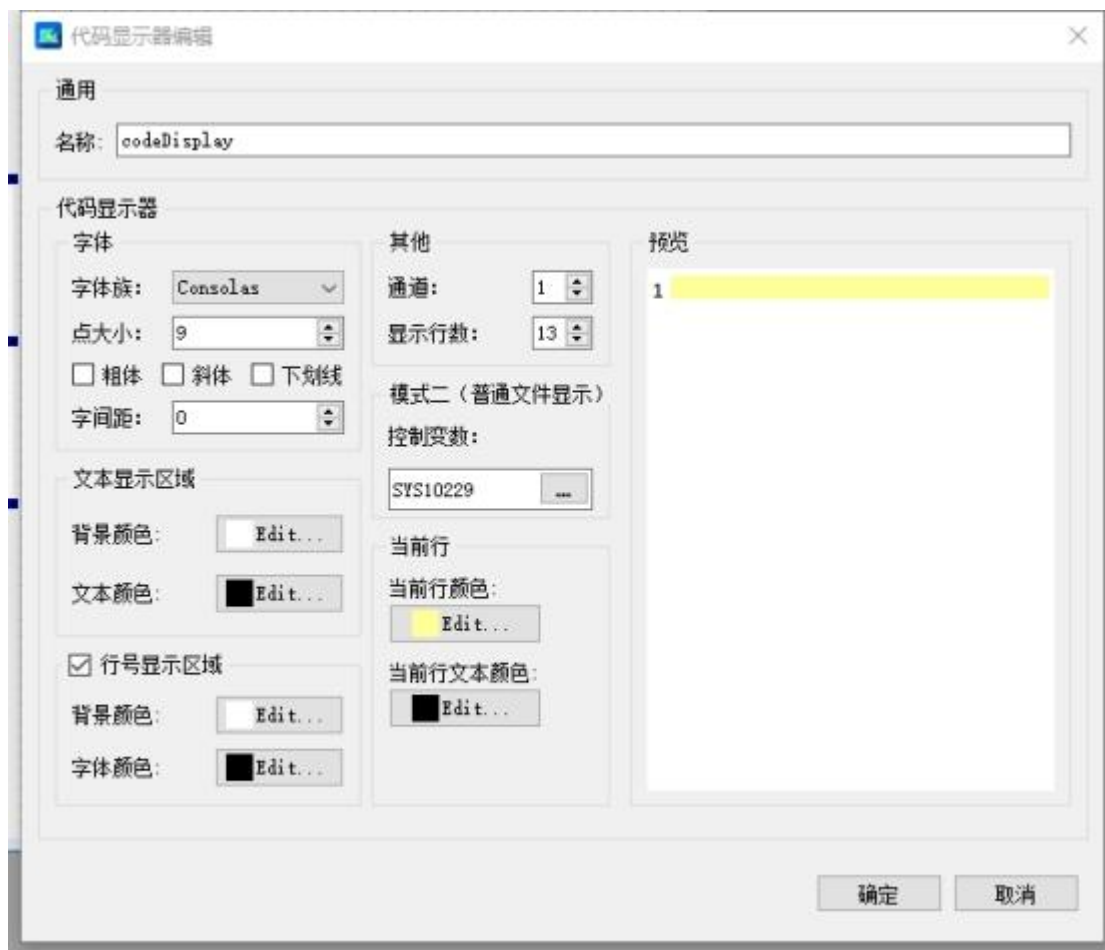
6.32.2. 简介

代码显示器插件用于显示当成控制器加载的程式内容。

代码显示器的 3 个工作模式：

- 最常用的是默认的工作模式，会自动加载设定通道所执行的程式文件，根据 SYS10228（当前执行程式的位置）和 SYS10229（当前执行程式的行号）刷新插件视口显示。
- 模式 1，显示模拟的程式文件，会自动加载当前模拟的程式文件，根据 SYS10297（当前执行程式的位置）和 SYS10298（当前执行程式的行号）刷新插件视口显示。
- 当通过函数接口开启模式二时，需要配合接口打开目标文件，以及设置模式二的控制变数刷新插件视口。

6.32.3. 属性编辑弹窗



1. 通用

1) 名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在脚本代码中引用和操作对象。

2. 代码显示器

2) 字体

设置插件整体字体属性，包括文本显示区域和行号显示区域。

3) 文本显示区域

- a. 背景颜色：设置文本显示区域的背景颜色。
- b. 文本颜色：设置文本显示区域的文本颜色。

- 4) 行号显示区域
 - a. 可以控制行号显示区域是否显示。
 - b. 背景颜色：设置行号显示区域的背景颜色。
 - c. 字体颜色：设置行号显示区域的文本颜色。
- 5) 其他
 - a. 通道：控制代码显示器显示哪个通道下加载的程式内容。
 - b. 行号：设置代码显示器最大视口显示行号。
- 6) 模式二
 - a. 控制变数：当代码显示器开启模式二时，当前行由关联的控制变数决定。
- 7) 当前行
 - a. 当前行颜色：当前行的高亮颜色。
 - b. 当前行文本颜色：当前行的文本颜色。

6.32.4. 属性编辑器

属性编辑器默认位置可视化开发软件的右侧，所有属性均在属性编辑弹窗上有体现。



6.32.5. 属性接口

属性	描述	用法
visible	读写插件可见属性	<code>var visible = Form.codeDisplay.visible</code>
enable	读写插件使能属性	<code>Var enable = Form.codeDisplay.enable</code>
geomerty	读写插件的几何尺寸属性, 需要借助 HProperty 读写具体值。	<code>var geomerty = Form.codeDisplay.geomerty</code>
font	读写插件的字体属性, 需要借助 HProperty 读写具体值。	<code>var font = Form.codeDisplay.font</code>
backgroundColor	读写代码显示器的背景颜色, 需要借助 HProperty 读写具体值。	<code>var color = Form.codeDisplay.backgroundColor</code>
channelNo	读写代码显示器的通道号	<code>Form.codedisplay.channelNo=1</code>
ChnNoVarEnable	设置代码显示器的通道号变数控制是否有效	<code>Form.codedisplay.ChnNoVarEnable = true</code>
currentLineColor	读写代码显示器的当前行颜色	<code>var color = Form.codeDisplay.currentLineColor</code>
currentLineTextColor	读写代码显示器	<code>var color = Form.codeDisplay.</code>

	的当前行文本颜色	currentLineTextColor
lineCount	读写代码显示器的行数	var count= Form.codeDisplay. lineCount
lineNumBackgroundColor	读写代码显示器的行号栏背景颜色	var color =Form.codedisplay. lineNumBackgroundColor
lineNumberAreaVisible	读写行号区域是否可见	Form.codedisplay.lineNumberAreaVisible = true
lineNumColor	读写代码显示器的行号字体颜色	var color =Form.codedisplay.lineNumColor
txtColor	读写代码显示器的字体颜色	var color =Form.codedisplay.txtColor

6.32.6. 接口函数

函数原型	描述	参数	返回值
void openFile(filepath)	设置要打开的程式文件	(string)filePath: 字符串型文件路径。	无
void setDataMode(mode)	设置要显示的程式类型	int)mode[0-1], int 型模式类型, 有效值为 1 或 0。 =0 显示实际执行程序 =1 显示模拟执行程序 =2 用户自定义	无

6.33. 代码编辑器

组件外观



组件由文件信息显示区域、行号显示区域、编辑器主体与滑动块组成，其中文件信息显示区域显示内容包括文件名、通道（通道由文件所在路径决定/usr/sys0001/program为通道1）、光标所在行和光标所在列。插件可以最多提供3个编辑器。

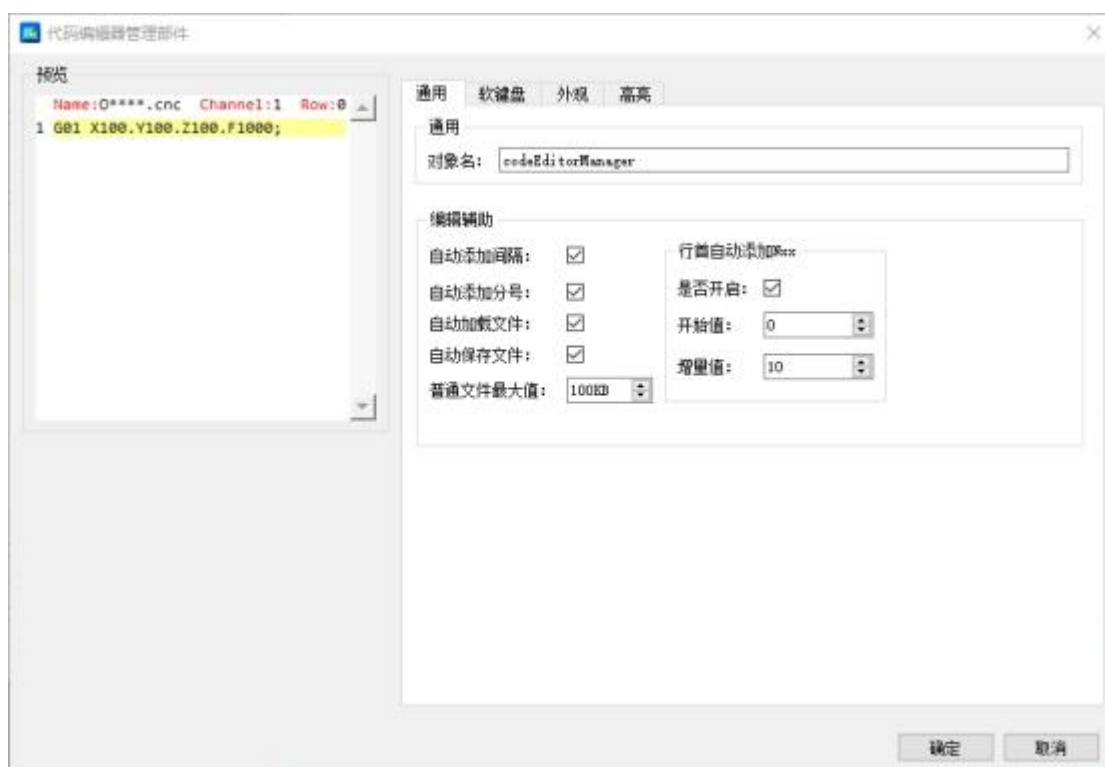
6.33.1. 简介

代码编辑器用于读写文件，文件格式不限，多个用于编辑程式文件（G代码）。

6.33.2. 属性编辑弹窗

属性编辑弹窗包含“通用”、“软键盘”、“外观”和“高亮”四页。

【通用】



1. 通用

1) 对象名称

每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在脚本代码中引用和操作对象。

2. 辅助编辑

1) 自动添加间隔

在编辑 G 代码时，输入两个 G 代码间会自动添加空格。输入 X100 后输入 Y200，显示结果“X100 Y200”。

2) 自动添加分号

在编辑 G 代码时，输入回车换行会自动在行末添加分号 ‘;’。

3) 自动加载文本

开启功能后，画面工程运行过程中，画面切换至代码编辑器画面，插件会自动加载所记录的程式文件。

4) 自动保存文件

开启功能后，画面工程运行过程中，当画面从编辑器页面切换至其他画面时，插件会自动保

存当前文件。

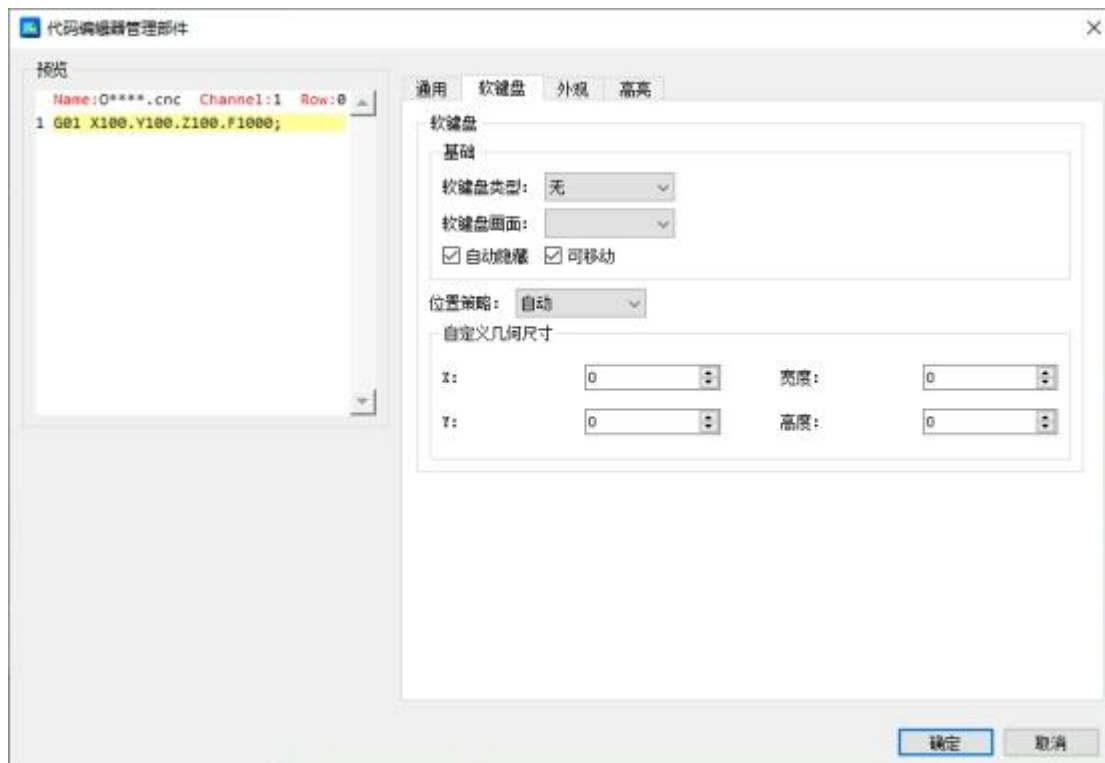
5) 普通文件最大值

插件将程式文件分为“大文件”和“小文件”以便更好得进行读写处理，文件大小大于“普通文件最大值”为大文件，否则为小文件。

6) 行首自动添加 Nxx（自动编号）

相关属性有“自动编号”、“自动编号开始值”、“自动编号增量值”，开启属性后，编辑器换行时，会自动插入 N 码，N 码的值由开始值和增量值决定（开始值+换行数*增量值）。功能可通过属性或函数控制。

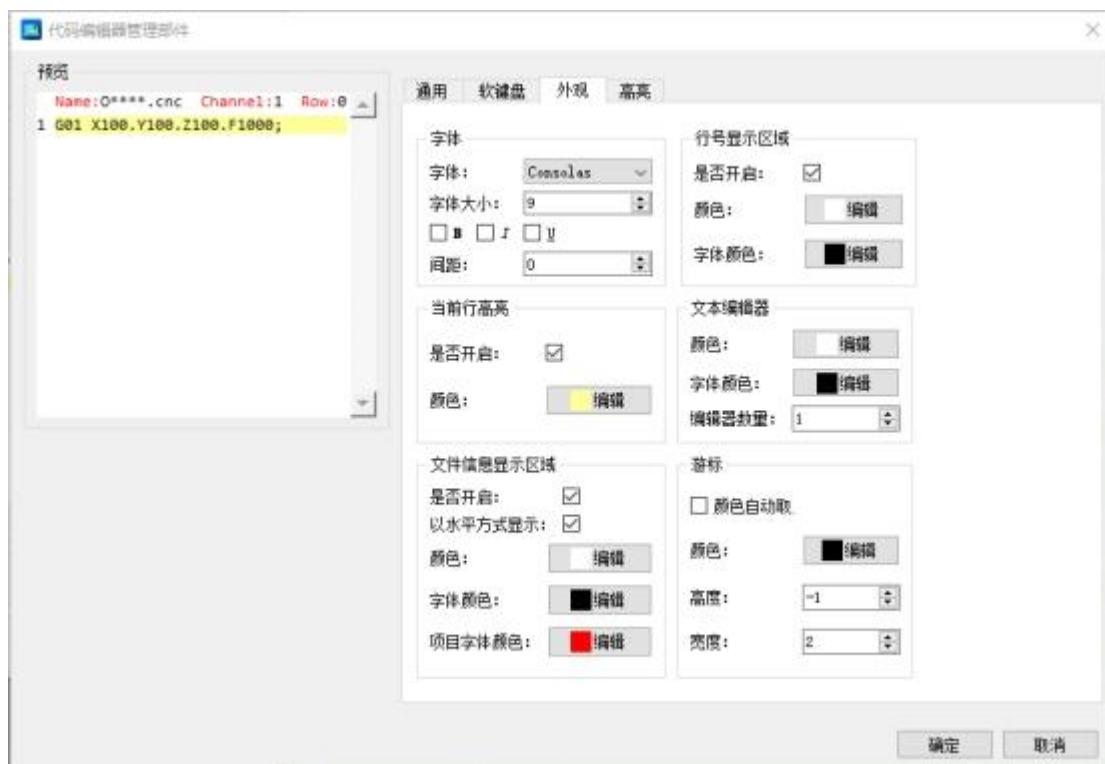
【软键盘】



(1) 软键盘

使用触控的方式输入时，可以使用软键盘。（[跳转到软键盘说明](#)）

【外观】



1. 字体

统一设置插件字体，包括文件信息显示区域、行号显示区域和编辑器主体的字体。

2. 高亮当前行

1) 是否开启

控制属性使能，开启属性后，插件后以“当前行颜色”高亮游标所在行的文本块。

2) 颜色（当前行颜色）

高亮当前行颜色。

3. 文件信息显示区域

1) 是否开启

控制文件信息显示区域是否可见。

2) 以水平方式显示

使能属性，文件信息条目以水平方式显示，否则文件信息条目以垂直方式显示。

3) 颜色

文件信息显示区域背景颜色。

4) 字体颜色

文件信息显示区域字体颜色。

5) 项目字体颜色

文件信息显示区域项目字体颜色，项目包括“名称”、“通道”、“行”和“列”。

4. 行号显示区域

1) 是否开启

控制行号显示区域是否可见。

2) 颜色

行号显示区域的背景颜色。

3) 字体颜色

行号显示区域的字体颜色。

5. 文本编辑器

1) 颜色

编辑器主体背景颜色。

2) 字体颜色

编辑器主体背景颜色。

3) 编辑器数量

插件编辑器数量，插件最多可提供三个编辑器。

6. 游标

1) 颜色自动取反

开启属性后，游标颜色会自动根据所在行的文本块背景颜色进行取反，且游标颜色不可设置。

2) 颜色

设置游标颜色，开始“颜色自动取反”后，属性不可设置。

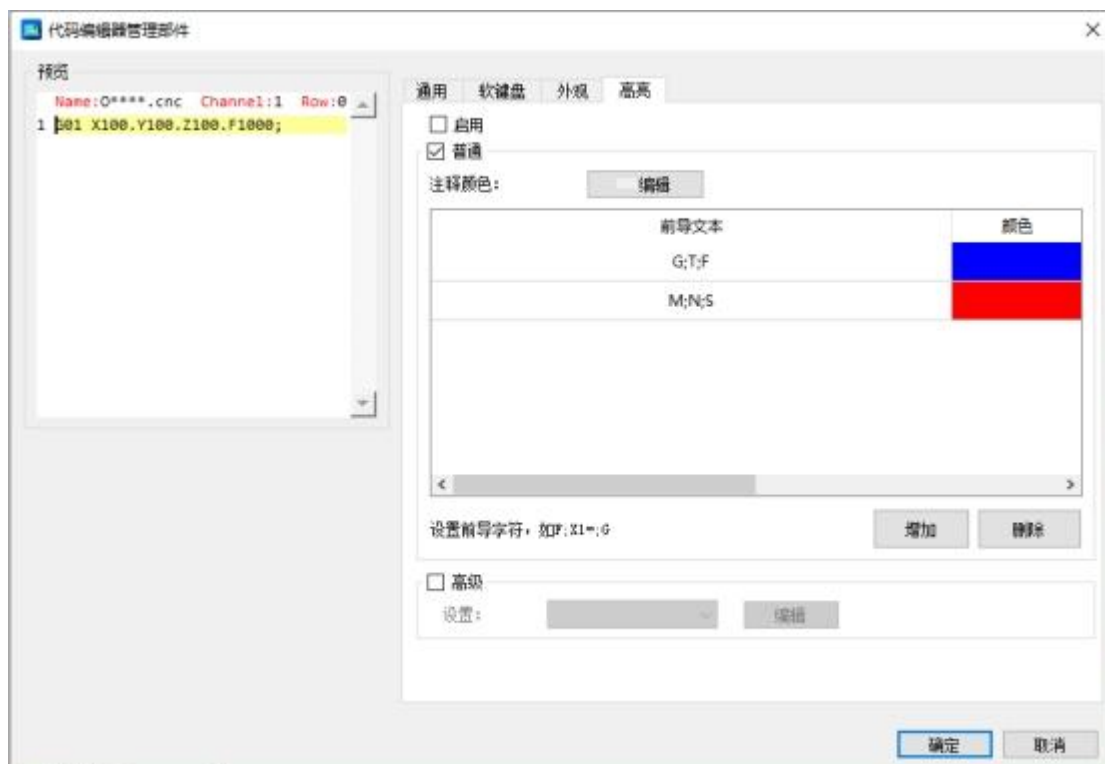
3) 高度

可设置游标高度，默认值为-1 即根据字体大小自动调整。

4) 宽度

可设置游标的宽度，默认值为 2.

【高亮】



1. 启用

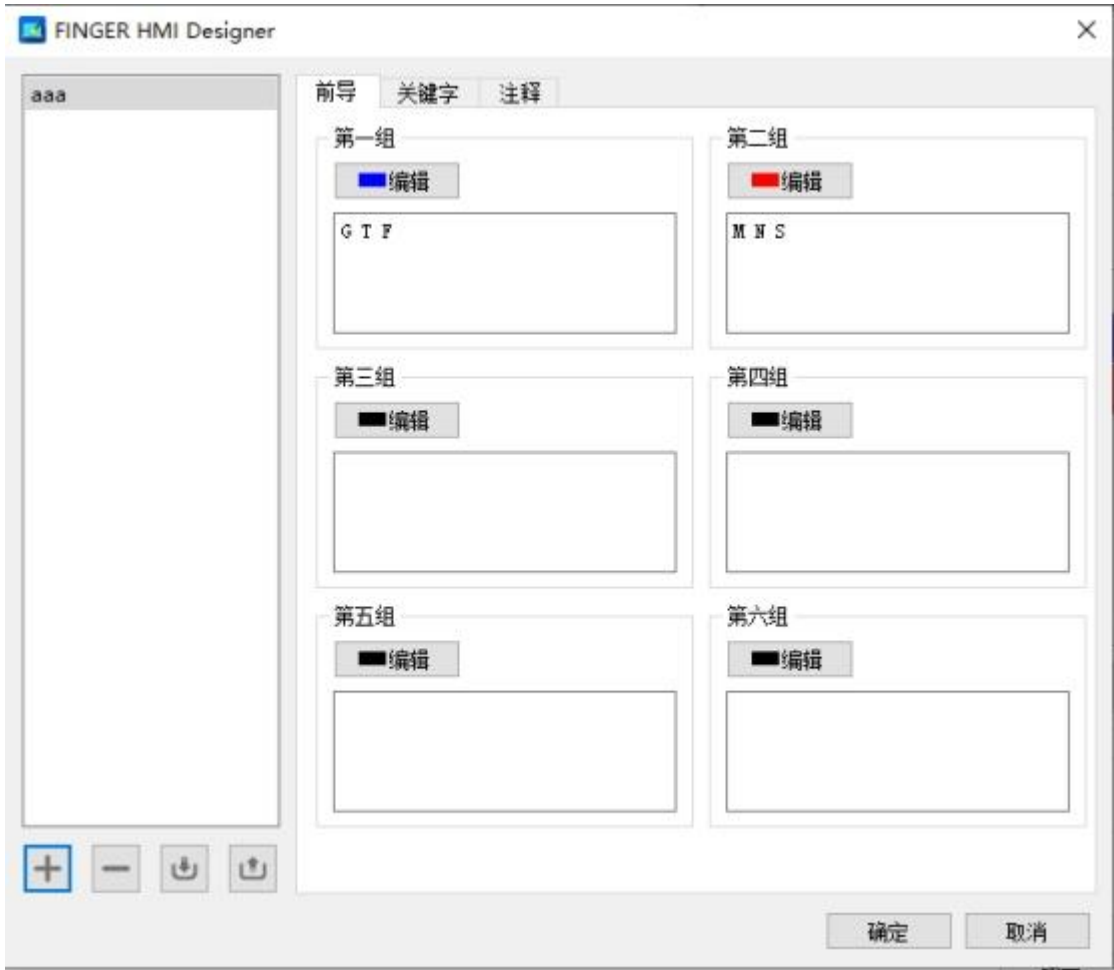
用于控制高亮的功能是否启用，高亮功能分为两个，普通高亮功能与高级高亮功能，两个不能同时启用，使用普通高亮功能时，高级高亮功能处于不可用的状态，反之亦然。

1) 普通

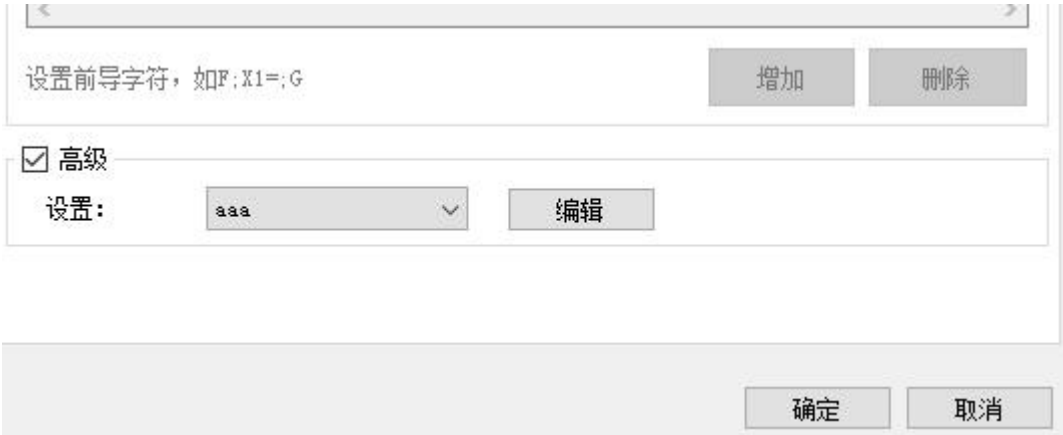
普通高亮功能允许在属性编辑弹窗中编辑设置，包括注释颜色与前导文本颜色（G 代码），不同的 G 码使用分号“;”间隔开。可以增加或删除高亮的颜色。

2) 高级

多用于复杂和特定的文本高亮要求，勾选高级复选框后，高级功能使能，通过“编辑”唤出弹窗编辑高级高亮编辑窗口，如下图所示：



允许新建、删除、导出、导入高亮配置，高亮配置中包括前导、关键字和注释三个选项，完成编辑后，在代码编辑器的属性编辑弹窗“高亮”页面中选用，如下图所示：



6.33.3. 属性编辑器

属性编辑器默认位置可视化开发软件的右侧，大部分属性在“属性编辑弹窗”中都有体现，在这不一一赘叙，下文仅针对编辑弹窗中不包含的属性进行介绍。

1. 嵌入键盘

光标高度	-1
游标宽度	2
嵌入软键盘	☒
操作权限	0
滚动条宽或高	20
只读控制位	COM50200.0

启用软键盘的情况下，用于控制软键盘是否嵌入插件中，嵌入的软件盘无法移动，并且与插件等宽，且高度为插件高度的一半。

2. 操作权限

游标宽度	2
嵌入软键盘	☒
操作权限	0
滚动条宽或高	20

默认值为 0，即不开启权限限制，设置组件的操作权限。如果权限不足，插件将无法编辑。

([跳转到权限说明](#))

3. 滑动块宽或高

光标高度	-1
游标宽度	2
嵌入软键盘	☒
操作权限	0
滚动条宽或高	20
只读控制位	COM50200.0
只读	☐

用于控制插件的滑动块宽度或高度。

4. 只读

滚动条宽或高	20
只读控制位	COM50200.0
只读	☐
软键盘	1, 1, 无, , 自动, 0, 0, 0, 0

涉及到的属性有“只读”和“只读控制位”，只读属性开启后，编辑器处于不可编辑的状态，可以通过设置只读控制位与系统地址进行关联，通过系统地址控制插件的只读属性。

5. 输入模式（是否使用广数输入模式）

只读	☐
软键盘	1, 1, 无, 自动, 0, 0, 0, 0
输入模式	0
操作记录	☐
操作记录名称	Form_codeEditorManager
文本间隔	0.000000

输入模式为 1 时，插件处于“广数输入模式”，非 1 时为普通输入模式。当插件处于“广数输入模式”时，无法通过直接输入或者软键盘输入，需要通过“输入区块”结合 Insert 与 Delete 按键，对编辑器内的文本进行新增、删除、修改等操作。

6. 操作记录

输入模式	1
操作记录	☐
操作记录名称	Form_codeEditorManager
文本间隔	0.000000

- 1) 设置是否开启操作记录
- 2) 设置操作记录上插件的名称，默认值为画面名_插件对象名。

操作记录具体功能浏览相应说明文档。

6.33.4. 系统多国语言

插件上存在的固有文字，包括文件信息显示区域、插件提示文本等文字内容的翻译都可以在“工具”菜单栏中“组件固有文字翻译”中设置。



固有文字翻译编辑窗口功能请浏览相应的说明文档。

6.33.5. 功能详述

插件功能不仅限于上述属性中提交到的，下文将逐一对代码编辑器插件的功能进行详述。

【跳转】

代码编辑器提供多个接口，两种方式实现跳转功能。第一种方式通过接口函数 openJumpDialog 唤出跳转窗口，如下图所示：



通过跳转窗口设置目标行，并且触发跳转。第二种方式是通过接口函数 `jumpTo` 设置要跳转的目标行并触发跳转。

【查找与替换】

插件提供多个接口，两种方式实现查找、替换和全部替换的功能。第一种方式是通过接口函数 `openFindDialog` 唤出查找替换的窗口，如下图所示：



设置查找替换的内容，并且设置搜索条件，包括反向查找、大小写敏感和全字匹配，触发查找、替换和全部替换实现功能。第二种方式是通过接口函数设置需要查找 `setFindString` 或替换 `setReplaceString` 的内容后，触发查找替换 `editorReplace` 功能。

【内容载入 MDI】

插件提供接口函数“`loadToMdi`”实现将编辑器内容写入控制器 MDI 的功能。

【程式再启动】

插件通过接口函数“`setRestartFormHere`”实现控制器加载当前程式，并且跳转到光标所在的行执行程式。

【高亮修改行】

插件提供接口“`setHighlightChangedLine`”启用高亮修改行。功能启用后，“高亮当前行功能”将自动关闭。当文本内容被修改后会以“当前行颜色”高亮被修改行的文本块。当文件被保存后，高亮显示消失。注意功能只在“小文件”情况下可用。

【代码帮助】

插件提供接口启用 G 代码帮助，调用接口打开代码帮助窗口后，编辑器会根据当前游标所在的 G 代码，在代码帮助窗口中显示相应的内容，默认有车床和铣床两种代码帮助类型，系统帮助文档的路径保存在 “/system/config/help/”，需要先确认默认路径下有帮助文档才能正常使用帮助功能。

【程序再启动】

接口 setRestartFromHere，加载编辑器当前文件，并且从当前行开始执行。

6.33.6. 插件属性

属性	描述	
autoAddN	读写代码编辑器自动添加 N 码属性值	codeEditorManager.autoAddN=true
autoLoad	读写代码编辑器自动加载属性值	codeEditorManager.autoLoad=true
autoSave	读写代码编辑器自动保存属性	codeEditorManager.autoSave=true
autoSemicolon	读写代码编辑器自动添加分号属性	codeEditorManager.autoSemicolon=true
autoSpace	读写代码编辑器控件自动添加属性	codeEditorManager.autoSpace=true
backgroundColor	读写代码编辑器背景颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.backgroundColor</pre>

		=HProperty.setQColor(colorMap)
currentLineColor	读写代码编辑器当前行颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.currentLineColor =HProperty.setQColor(colorMap)</pre>
cursorColor	读写代码编辑器光标颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.cursorColor =HProperty.setQColor(colorMap)</pre>
cursorHeight	读写代码编辑器光标高度	codeEditorManager.cursorHeight=10
cursorWidth	读写代码编辑器光标宽度	codeEditorManager.cursorWidth=10
editorNum	获取代码编辑器控件的编辑器数量，属性只读	num = codeEditorManager.editorNum
fileInfBackgroundColor	读写代码编辑器文件信息的背景颜色	<pre>var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.fileInfBackground Color =HProperty.setQColor(colorMap)</pre>

fileInfHFormat	读写代码编辑器是否水平显示文件信息	codeEditorManager.fileInfHFormat=true
fileInfItemColor	读写代码编辑器文件信息项目字段文本颜色	<pre> var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.fileInfItemColor =HProperty.setQColor(colorMap) </pre>
fileInfTxtColor	读写代码编辑器文件信息的文本颜色	<pre> var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.fileInfTxtColor =HProperty.setQColor(colorMap) </pre>
fileTxtColor	读写代码编辑器文本颜色	<pre> var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.fileTxtColor =HProperty.setQColor(colorMap) </pre>
highlightCurrentLine	读写代码编辑器是否高亮当前行	<pre> var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.highlightCurrentLine </pre>

		=HProperty.setQColor(colorMap)
inputMode	读写代码编辑器输入模式	codeEditorManager.inputMode=1
insetSoftKeypad	读写代码编辑器是否嵌入软件盘	codeEditorManager.insetSoftKeypad=true
lineNum	读写代码编辑器是否显示行号	codeEditorManager.lineNum=true
lineNumBackground Color	读写代码编辑器行号栏背景颜色	var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.lineNumBackground Color =HProperty.setQColor(colorMap)
lineNumColor	读写代码编辑器行号栏文本颜色	var colorMap={}; colorMap["red"]=255; colorMap["green"]=0; colorMap["blue"]=0; Form.codeEditorManager.lineNumColor =HProperty.setQColor(colorMap)
normalFileMaxSi ze	读写代码编辑器普通文件大小限制	codeEditorManager.normalFileMaxSize=102400
openFileInf	读写代码编辑器是否显示文件信息	codeEditorManager.openFileInf=true
readOnly	读写代码编辑器是否只读	codeEditorManager.readOnly=false
scrollBarExtent	读写代码编辑器滑动块大小	codeEditorManager.scrollBarExtent=25

softKeypad	读写代码编辑器是否使用软键盘	codeEditorManager.softKeypad=true
softKeypadAutoHide	读写代码编辑器是否自动隐藏	codeEditorManager.softKeypadAutoHide=true

6.33.7. 插件函数表

函数原型	描述	参数	返回值
activeDeleteOneLine()	删除当前活动编辑器中光标所在行。	无	无
activeEditorID()	获取当前激活的编辑器的 ID。	无	● 当前激活的编辑器的 ID (0-2)
activeInsertText(text)	为当前活动编辑器插入文本	● (string)text 文本内容	无
activeJumpTo(line)	跳转到当前活动的编辑器的第 N 行	● (int)line 目标行	无
activeOpenFile(fileName, type, device)	在当前活动的编辑器内打开文件。	● (string)fileName: 文件路径, 字符型。 ● (int)type[0-1]: 文件类型=0 普通 CNC 文件;=1 含有文件信息的 CNC 文件; 此参数也可省略。 ● (string)device: 设备名, 默	● (bool) 是否打开成功

		认参数"Device0"	
activeSaveFile(fileName, device)	保存当前活动的编辑器内打开文件。	● (string)fileName: 文件路径, 字符型。 ● (string)device: 设备名, 默认参数"Device0"	(bool) 是否保存成功
chageFindDlgFocus()	在查找替换对话框内切换输入焦点	无	无
changeJumpDlgFocus()	切换跳转对话框中的输入框中的输入焦点, 由于此对话框仅有一个输入框, 所以宏执行的效果为输入框获取输入焦点。	无	无
clearInputWidget()	清空输入框, 仅能在输入模式 1 下使用。	无	无
clickFindBtn()	点击查找替换对话框中的 Find 按钮	无	无
clickJumpDlgCancelBtn()	点击跳转对话框上的 Cancel 按钮, 关闭对话框	无	无

	框并取消跳转。		
clickJumpDlgOKBtn()	点击跳转对话框上的OK按钮，当前激活的对话框实现跳转功能。	无	无
clickReplaceAllBtn()	点击查找替换对话框中的ReplaceAll按钮。	无	无
clickReplaceBtn()	点击查找替换对话框中的Replace按钮	无	无
closeAssistantDialog()	关闭帮助窗口。	无	无
closeFindDialog()	关闭查找替换对话框	无	无
confirmModify()	在背景编辑功能下，调用此函数确认修改并保存后，下一次执行才会采用新修改的内容。	无	无
deleteLine(line, ID)	删除指定编辑器的行内容	● (int)line: 指定行 ● (int)ID[0-2]: 编辑器标识	空
deleteOneLine(ID)	删除指定的编	● (int)ID[0-2]	无

	辑器中光标所在行。	]: 编辑器标识	
editorCopy()	复制选中的文本	无	● (string) 返回复制的文本内容
editorCut()	剪切选中的文本	无	无
editorFind(findBackward, caseSensitively, wholeWords)	查找下一个。	● (bool) findBackward: 是否反向查找=false, 正向查找。=true, 反向查找。 ● (bool) caseSensitively: 是否区分大小写=false, 不区分大小写。=true, 区分大小。 ● (bool) wholeWords: 是否全字匹配=false, 非全字匹配。=true, 全字匹配。	● (bool) 是否发现目标字符串
editorIsHide(ID)	获取指定编辑器的显示/隐藏状态	● (int) ID[0-2]: 编辑器标识	● (bool) 是否隐藏
editorMoveCursor(mode)	在当前的活动的编辑器中移动光标位置	● (int) mode, 移动模式, 具体意义如下: =0, 移动	无

		至行尾处;=1, 移动至行首处;=2, 移动至文档开始处;=3, 移动至文档结尾处;=其他, 移动至行尾处;	
editorPaste()	粘贴选中的文本	无	无
editorRedo()	重新操作	无	无
editorReplace(findBackward, caseSensitively, wholeWords)	替换下一个。	● (bool)findBackward: 是否反向查找=false, 正向查找。=true, 反向查找。 ● (bool)caseSensitively: 是否区分大小写=false, 不区分大小写。=true, 区分大小写。 ● (bool)wholeWords: 是否全字匹配=false, 非全字匹配。=true, 全字匹配。	● (bool) 是否发现目标字符串
editorUndo()	撤销操作	无	无
getActiveIsAddN()	获取当前激活	无	● (bool) 是否开

	的编辑器是否开启换行自动添加 N 功能。		启换行时自动添加 N
getActiveNoffset()	获取当前激活的编辑器的自动添加 N 功能中数值的增量	无	● (int) 自动添加 N 功能中数值的增量。
getActiveNstart()	获取当前激活的编辑器的自动添加 N 功能中 N 的起始值	无	● (int) 自动添加 N 功能中 N 的起始值
getActiveOpenFileName()	获取当前活动的编辑器内打开文件的路径。	无	● (string) 文件路径
getCurrentLineNo()	获取编辑器当前行号	无	● (int) 编辑器当前行号
getCurrentLineText()	获取编辑器当前行的内容。	空	● (string) 编辑器当前行的文本内容
getFindString()	获取查找替换对话框中查找输入框中的字符串	无	● (string) 查找替换对话框中查找输入框中的字符串。
getIsAddN(ID)	获取指定的编辑器是否开启换行自动添加 N 功能。	● (int) ID[0-2]: 编辑器标识	● (bool) 是否开启换行时自动添加 N
getNoffset(ID)	获取指定编辑	● (int) ID[0-2]	● (int) 自动添加

	器的自动添加 N 功能中数值的增量	]：编辑器标识。	N 功能中数值的增量。
getNstart(ID)	获取当前激活的编辑器的自动添加 N 功能中 N 的起始值	● (int) ID[0-2]：编辑器标识	● (int) 自动添加 N 功能中 N 的起始值
getOpendFileName(ID)	获取当指定的编辑器内打开文件的路径	● (int) ID[0-2]：编辑器标识	● (string) 文件路径
getOperationalAuthority()	获取插件限制权限	空	● (int) authority: 插件限制权限
getReplaceString()	获取查找替换对话框中替换输入框中的字符串。	无	● (string) 查找替换对话框中替换输入框中的字符串。
getTxt()	获得指定编号编辑器的文本；	● (int) ID: 编辑器编号，范围 0~2；	● (string) 获得的文本内容。
inputCode()	将输入的代码输入到编辑器中，仅能在输入模式 1 下使用。	空	空
insertCode()	将输入的代码插入到编辑器中，仅能在输入模式 1 下使用。	空	空
insertLine(line, content)	向指定编辑器	● (int) line:	无

ID)	插入一行文本	指定行 ● (string)content:待插入文本 ● (int)ID[0-2]: 编辑器标识	
insertText(ID, text)	为指定编辑器插入文本。	● (int)ID[0-2]:编辑器标识 ● (string)text:插入的文本内容	无
isTextEdited(ID)	获取指定编号编辑器是否有被编辑的状态。	● (int)ID: 编辑器编号, 范围0~2;	● (bool) 内容是否有修改过。
jumpTo(ID, n)	跳转到指定编辑器的第 N 行。	● (int)ID[0-2]: 编辑器标识 ● (int)n: 目标行, 整数型	无
openAssistantDialog()	打开帮助窗口。	无	无
openFile(ID, fileName, type, device)	在指定的编辑器内打开文件。	● (int)ID[0-2]: 编辑器标识 ● (string)fileName: 文件路径, 字符型。 ● (int)type[0-1]: 文件类型=0 普通 CNC 文件=1 含有文件信息的	● (bool) 文件是否打开成功

		CNC 文件；此参数也可省略。 ● (string)device: 设备名，默认参数"Device0"	
openFindDialog()	打开查找替换对话框	无	无
openJumpDialog()	打开跳转对话框。	无	无
overwriteLine(line , content , ID)	复写指定编辑器的行内容	● (int)line: 指定行 ● (string)content: 待插入文本 ● (int)ID[0-2]: 编辑器标识	无
pageDown()	下翻页。	无	无
pageUp()	上翻页	无	无
readLine(line , ID)	获取指定编辑器的 line 行文本	● (int)line: 指定行 ● (int)ID[0-2]: 编辑器标识	● (string) 行内容
saveFile(ID, fileName)	保存指定的编辑器内已打开文件。	● (int)ID[0-2]: 编辑器标识 ● (string)fileName: 文件路径，字符型。	● (bool) 文件是否保存成功

		● (string) device: 设备名, 默认参数"Device0"	
setActiveIsAddN(open)	开启或关闭换行当前激活的编辑器的自动添加 N 功能	● (bool) open: 是否开启自动添加 N 功能。	无
setActiveNoffset(value)	设置当前激活的编辑器的自动添加 N 功能中数值的增量	● (int) value: 自动添加 N 功能中数值的增量, 整数型	无
setActiveNstart(value)	设置当前激活的编辑器的自动添加 N 功能中 N 的起始值	● (int) value: 自动添加 N 功能中 N 的起始值	无
setDragMoveEnable(enable)	是否允许文本拖拽。	● (bool) enable: 是否能够拖拽文本	无
setEditorFocus(ID)	将焦点转移到指定的编辑器	● (int) ID[0-2]: 编辑器标识	无
setEditorManagerFocus()	将焦点转移到编辑器管理器上, 由管理器决定哪个编辑器应获取焦点	无	无
setEditorSelectionMode(open)	开启或关闭当前活动的编辑器选择模式	● (bool) open, 是否开启选择模式	无

setFindString(str)	设置查找替换对话框中查找输入框中的字符串。	● (string) 查找替换对话框中替换输入框中的字符串。	无
setHighlightChangedLine(enable)	高亮文本内容被修改的行	● (bool) enable: 是否高亮文本内容被修改的行	无
setIsAddN(ID, open)	开启或关闭换行指定编辑器的自动添加N功能	● (int) ID[0-2]: 编辑器标识 ● (bool) open: 是否开启自动添加N功能 true: 换行时自动添加N false: 换行时不自动添加N	无
setLineSelectMode(enable)	是否开启行选择模式	(bool) enable: 是否开启行选择模式	无
setNoffset(ID, value)	设置指定编辑器的自动添加N功能中数值的增量	● (int) ID[0-2]: 编辑器标识 ● (int) value: 自动添加N功能中数值的增量	无
setNstart(ID, value)	设置指定的编辑器的自动添加N功能中N的起始值	● (int) ID[0-2]: 编辑器标识 ● (int) value: 自动添加N功能中N的起始值	无

setOperationalAuthority(authority)	设置插件限制权限	(int) authority: 插件限制权限	无
setReplaceString()	设置查找替换对话框中替换输入框中的字符串。	(string) 查找替换对话框中替换输入框中的字符串。	无
setTxt(ID, text)	设置指定编号编辑器的文本	(int) ID[0-2]: 编辑器编号 (string) text: 字符型, 设置的文本内容;	无
showHideEditor(ID, show)	显示或隐藏指定的编辑器	(int) ID[0-2]: 编辑器标识 (bool) show: 是否显示, 此参数省略时, 默认为 true	无
toggleCaseSensitively()	切换查找替换对话框中的 FindCaseSensitively 选择状态, 当其选中时开启大小写敏感。	无	无
toggleEditorFocus()	当存在多个编辑器时, 用于切换当前活动的	无	无

	编辑器。		
toggleSearchDirect()	切换查找替换对话框中的FindBackward选择状态，当其选中时开启反向查找。	无	无
toggleWholeWords()	切换查找替换对话框中的FindWholeWords选择状态，当其选中时开启全词匹配。	无	无
setRestartFromHere()	从当前行再启动执行程序文件	无	无

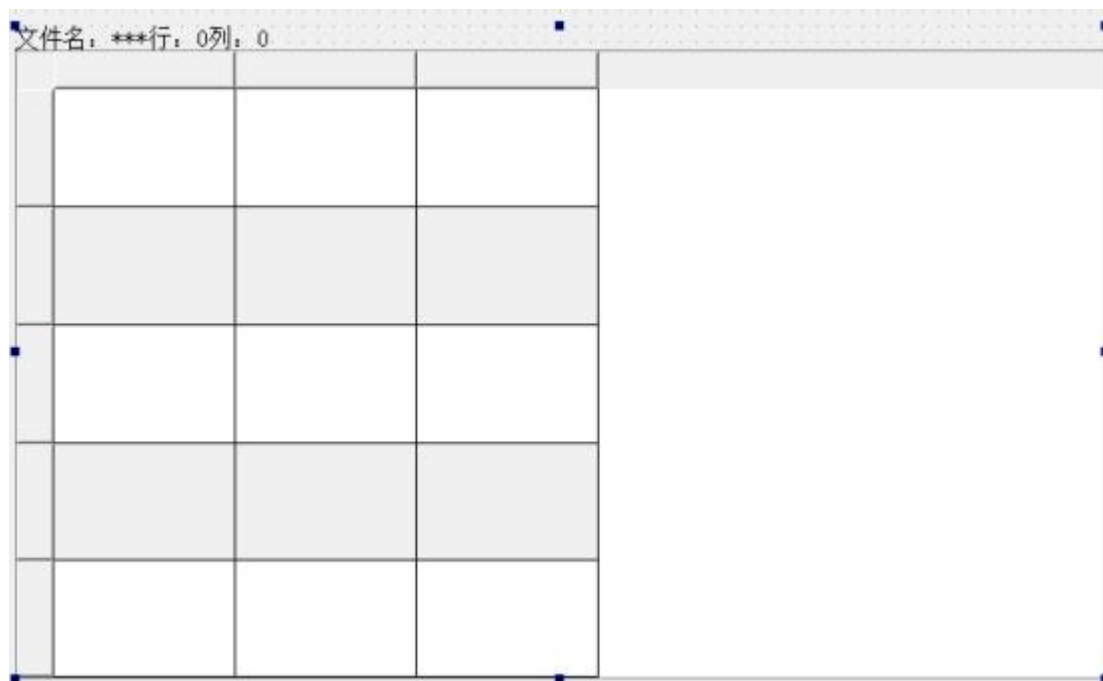
6.34. 文本框

6.35. 多文本框

6.36. 下拉框

6.37. 表格集

6.37.1. 组件外观



组件包括，文件名、行和列显示，以及用于显示加载程式文件内容的表格。

6.37.2. 简介

表格集目前共有 5 个模式，分别是模式 1 刀补模式；模式 2 填充格模式；模式 3 文件读写模式；模式 4 自定义模式；模式 5 物联网模式。

模式 2，填充格模式能够加载并且显示程式文件内容。

6.37.3. 属性编辑弹窗

属性编辑弹窗包括基础、表格、样式、特殊.1、特殊.2。表格集插件中各个模式的公有属性在属性弹窗的基础、表格和样式页面中，模式特有的属性放置在特殊页面中。



1. 基础

1) 对象名：每个对象都可以通过设置对象名称（Object Name）来标识和访问。对象名称是一个字符串，用于在脚本代码中引用和操作对象。

2) 操作权限：设置插件的操作权限。

2. 几何尺寸

设置插件的位置和大小的属性。

3. 软键盘

使用触控方式输入时，可使用软键盘，具体看软键盘说明，表格集模式二中默认软键盘是十进制软键盘。

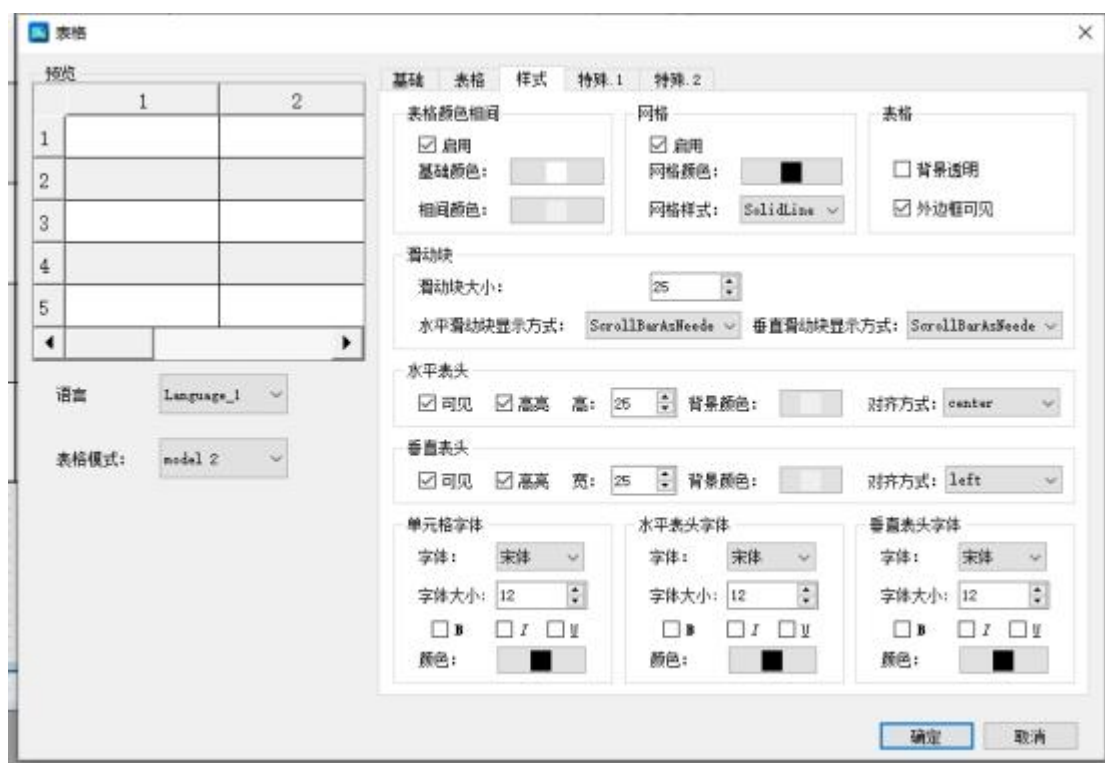


1. 表格

- 1) 不可见位：控制器表格集插件是否可见，可关联系统位变量。
- 2) 视口列数：设置表格视口列数，视口列数等于表格实际列数。
- 3) 视口行数：设置表格视口行数，视口行数不等于表格实际行数。

2. 属性

- 1) 最后一列铺满：设置表格最后一列的列宽自动调整，铺满表格。
- 2) 均分列宽：根据当前表格宽度、垂直表头宽度，滑动块宽度和列数，表格列均分列宽。
- 3) 列宽：单独设置列宽度。
- 4) 只读：控制列单元格是否只读。
- 5) 水平对齐：单元格水平对齐方式（左对齐、居中、右对齐）
- 6) 垂直对齐：单元格垂直对齐方式（上对齐、居中、下对齐）
- 7) 不可见位：控制表格该列是否可见，可关联系统位变量。



1. 表格颜色相间

- 1) 启用：启用表格颜色相间功能
- 2) 基础颜色：奇数行颜色
- 3) 相间颜色：偶数行颜色

2. 网格

- 1) 启用：启用表格网格
- 2) 网格颜色：网格颜色属性
- 3) 网格样式：网格样式属性，实现、虚线等样式选择。

3. 表格

- 1) 背景透明：设置表格背景透明属性
- 2) 外边框可见：设置表格外边框可见属性

4. 滑动块

- 1) 滑动块大小：控制水平滑动块的高度和垂直滑动块的宽度。
- 2) 水平、垂直滑动块显示方式：可选中总是显示、总是不显示和需要是显示。

5. 水平表头

- 1) 可见
- 2) 高亮：控制是否高亮显示当前列的水平表头文本
- 3) 高：控制水平表头高度
- 4) 背景颜色
- 5) 对齐方式：水平表头文本的对齐方式

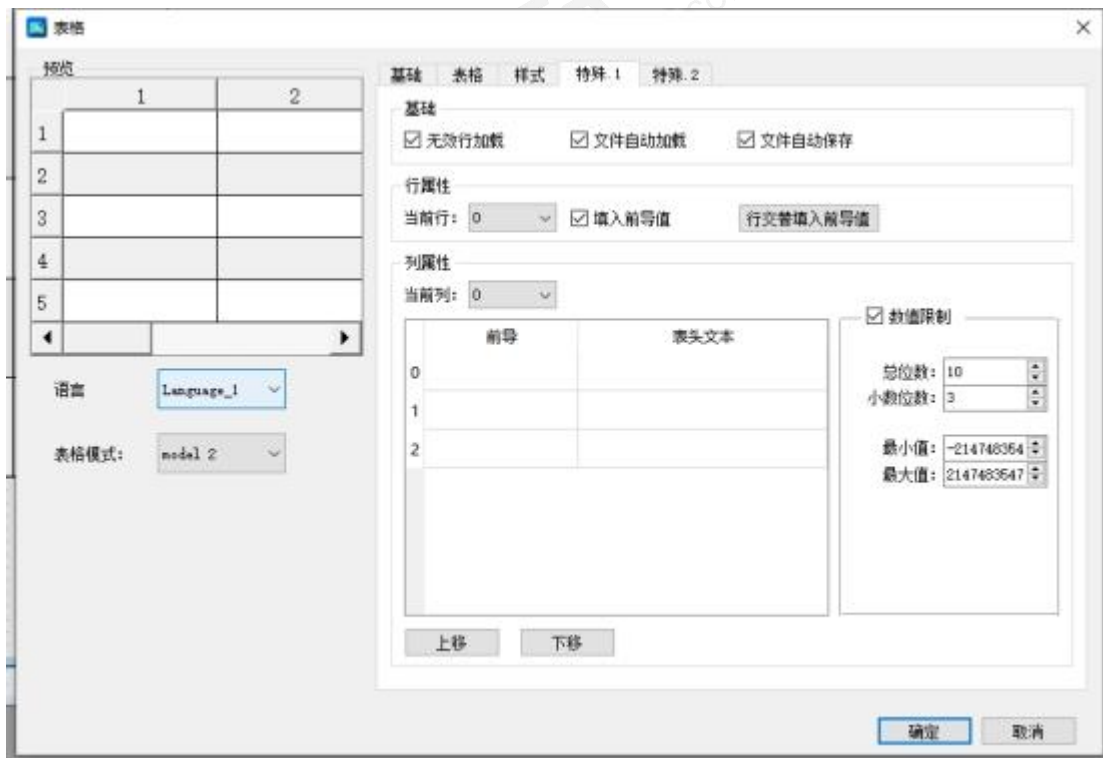
6. 垂直表头

- 1) 可见
- 2) 高亮：控制是否高亮显示当前行的垂直表头
- 3) 宽：垂直表头的宽度
- 4) 背景颜色
- 5) 对齐方式：垂直表头文本的对齐方式。

7. 单元格字体：统一设置表格单元格字体的样式和颜色。

8. 水平表头字体：统一设置表格水平表头字体的样式和颜色

9. 垂直表头字体：统一设置表格垂直表头字体的样式和颜色。



1. 基础

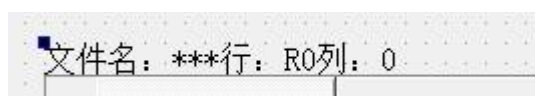
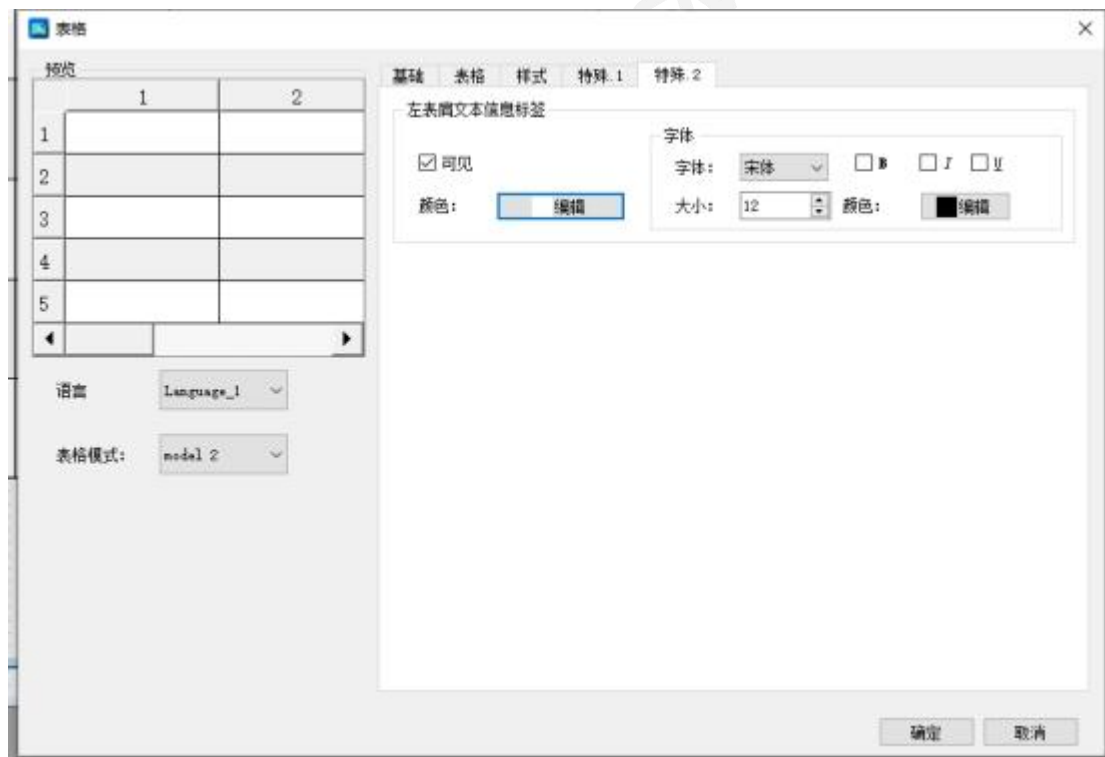
- 1) 无效行加载：表格是否加载显示程式文件中的空白行。
- 2) 文件自动加载：切换画面时，自动加载上一次加载的程式文件。
- 3) 文件自动保存：离开画面时，自动保存正在编辑的程式文件。

2. 行属性

- 1) 当前行：切换当前行，能够修改对应“填入前导值”属性。
- 2) 填入前导值：控制“当前行”的表格视口行，是否显示前导对应的值。
- 3) 行交替填入前导值：自动设置视口行间隔显示前导值。

3. 列属性

- 1) 当前列，显示正在编辑的列。
- 2) 前导：设置列的前导值（即 G 码），插件根据各列设置的前导值，读取程式文件中的内容并且显示到表格中。
- 3) 表头文本：设置列表头文本，可通过鼠标右键单击编辑对应的多国语言文本。
- 4) 数值限制：每一列都有对应的数值限制，控制对应列的显示和输入。



- 5) 左表肩文本标签

- 1) 可见：设置标签是否可见。
- 2) 颜色：设置标签背景颜色。
- 3) 字体：设置标签字体样式和前景颜色。

6.37.4. 属性编辑器



选择行为：设置选择方式，选项包括单元格、整行和整列。仅是选择方式，通过表格接口获取当前单元格仍只能获取到焦点所在的单元格。

6.37.5. 系统多国语言

插件没有系统多国语言

6.37.6. 功能详述

插件读取程式文件，以表格呈现方式，展示程式文件各行中对应 G 码的值。

6.37.7. 属性接口

属性	描述	用法
visible	读写表格 可见	hTableWidget.visible=false
enable	读写表格 是否启用	hTableWidget.enable=false
geometry	读写表格 几何尺寸， 需要借助	var geometry= hTableWidget.geometry

	HProperty 读写具体值。	
alternatingRowColors	读写是否交替显示行颜色	hTableWidget.alternatingRowColors=true
authority	读写表格集的权限	hTableWidget.authority=1
columnCount	获取表格集的列数，属性只读	var column = hTableWidget.columnCount
gridColor	读写表格集网格颜色，需要借助 HProperty 读写具体值。	var color =hTableWidget.gridColor
gridStyle	读写表格集网格样式，值对应属性编辑弹窗下拉框的 index	hTableWidget.gridStyle=0
gridVisible	读写表格集网格是否可见	hTableWidget.gridVisible=true

horizontalHeader BackgroundColor	读写表格 集水平表 头背景颜 色,需要借 助 HProperty 读写具体 值。	var color = hTableWidget. horizontalHeaderBackgroundColor
horizontalHeaderFont	读写表格 集水平表 头字体,需 要借 助 HProperty 读写具体 值。	var font = hTableWidget .horizontalHeaderFont
horizontal HeaderFontColor	读写表格 集水平表 头文本颜 色,需要借 助 HProperty 读写具体 值。	var color = hTableWidget. horizontalHeaderFontColor
horizontalHeaderHeight	读写表格 集水平表 头高度	hTableWidget.horizontalHeaderHeight =35
horizontalHeader HighlightSections	读写表格 集水平表	hTableWidget. horizontalHeaderHighlightSections=true

	头是否高亮	
horizontalHeaderVisible	读写表格集水平表头是否可见	hTableWidget.horizontalHeaderVisible=true
horizontalScrollBarPolicy	读写表格集水平滑块策略, 值对应属性编辑弹窗下拉框index	hTableWidget.horizontalScrollBarPolicy=0
rowCount	获取表格集的行数, 属性只读	var row = hTableWidget.rowCount
scrollBarExtent	读写表格集滑块尺寸	hTableWidget.scrollBarExtent=25
softKeypadGeometry	读写表格集的软键盘大小, 需要借助HProperty读写具体值。	var geometry= hTableWidget.softKeypadGeometry
softKeypadPosition	读写表格集的软键	var pos= hTableWidget.softKeypadPosition

	盘位置, 需要借助 HProperty 读写具体值。	
softKeypadType	读写表格集的软键盘类型, 值对应属性编辑弹窗下拉框的值	hTableWidget.softKeypadType=1
stretchLastSection	读写表格集是否最后一列铺满表格	hTableWidget.stretchLastSection=true
tableWidget DoubleRowColors	读写表格集偶数行的颜色	var color = hTableWidget. tableWidgetDoubleRowColors
tableWidgetFont	读写表格集字体, 需要借助 HProperty 读写具体值。	var font = hTableWidget.tableWidgetFont
tableWidgetFontColor	读写表格集文本颜色, 需要借	var color = hTableWidget .tableWidgetFontColor

	助 HProperty 读写具体 值。	
tableWidget SingleRowColors	读写表格 集单数行 颜色, 需要 借 助 HProperty 读写具体 值。	var color = hTableWidget. tableWidgetSingleRowColors
verticalHeader BackgroundColor	读写表格 集垂直表 头背景颜 色, 需要借 助 HProperty 读写具体 值。	var color = hTableWidget. verticalHeaderBackgroundColor
verticalHeaderFont	读写表格 集垂直表 头字体, 需 要 借 助 HProperty 读写具体 值。	var font = hTableWidget.verticalHeaderFont
verticalHeaderFontColor	读写表格 集垂直表	var color = hTableWidget. verticalHeaderFontColor

	头文本颜色	
verticalHeaderHighlightSections	读写表格 集垂直表 头是否高 亮	hTableWidget. verticalHeaderHighlightSections=true
verticalHeaderVisible	读写表格 集垂直表 头是否可 见	hTableWidget.verticalHeaderVisible=true
verticalHeaderWidth	读写表格 集垂直表 头宽度	hTableWidget.verticalHeaderWidth=35
verticalScrollBarPolicy	读写表格 集垂直滑 块策略	hTableWidget.verticalScrollBarPolicy=0

6.37.8. 宏函数

原型	描述	参数	返回值
cellBackground Color(row,col)	获取单元格的背景颜色。	(var)row, 行。 (var)col, 列。	(string) 十六进制的颜色值
cellChangedColumn()	获得内容发生改变的单元格所在列。	无。	(int) 内容发生改变的单元格所在列。
cellChangedRow()	获得内容发生改变的单元	无。	(int) 内容发生改变的单元格所在

	元 格 所 在 行。		行。
cellData(row, col)	获取某单元 格的值。	(var)row, 行。 (var)col, 列。	(var) data, 单元 格的值
cellEnable(row, col)	获取单元格 是否启用。	(var)row, 行。 (var)col, 列。	(bool) ok, 单元 格是否启用
cellForeground Color(row, col)	获取单元格的 前 景 颜 色。	(int)row, 行。 (int)col, 列。	(string)color , 颜色名称或十六 进制的颜色值
cellsRange()	获取所有选 中 的 单 元 格。	无	(Array) 单元格对 象数值, 对象成员 包括 row, col 两个 成员。
clearCellData(row, col)	清除某单元 格的值。	(var)row, 行。 (var)col, 列。	无
clearContent()	清空表格内 容。	无	无
clearRowData(row)	清空某行的 值。	(var)row, 行。	无
copyCellData(row, col)	复制某单元 格的值	(var)row, 行。 (var)col, 列。	无
copyRowData()	复 制 某 行 的值。	(var)row, 行。	无
cornerText()	获取表格转 角文本, 仅 模 式 四 可 用。	无。	(var) cornerText
currentColumn()	获取表格当	无	(var)当前列

	前列		
currentRow()	获取表格当前行。	无	(var)当前行
cursorDown()	下一项	无	无
cursorLeft()	左一项	无	无
cursorRight()	右一项	无	无
cursorUp()	上一项	无	无
dataBeforeChange	获取表格单元格改变前的值	无	(var) data
exportContent (type, path)	导出物联网显示内容到 excel 文件 (仅在模式五有效)	(int)type, 导出参数 0-导出数据 1-导出图形 2-导出数据和图形 (string)path, 导出的文件路径	无。
find(type, content, col, wholeWords, caseSensitive)	查找表格内的数据(目前模式四可以用)	(int)type, 查找类型 0-文本 1-变数 2-位 (string)content, 查找的内容, ("Target"/"USR100"/"COM75962.10") (int)col, 指定搜索的列, 默认值是-1 搜索所有列 (bool)wholeWords, 全	(Map)包含单元格的行列的数据结构

		词匹配搜索，默认值是 true (bool)caseSensitive, 区分大小写，默认值 true	
freezeFirstColumn(freeze)	设置是否冻结表格首列	(bool)freeze , 是否冻结	无。
freezeFirstRow(freeze)	设置是否冻结表格首行	(bool)freeze , 是否冻结	无。
getColumnWidth(column)	获取列宽	(int) column	(int) width
horizontalHeaderText (index)	获取某列水平表头文本	(int) index, 指定某列水平表头。	(string) 字符串类型的水平表头文本。
insertRow(row)	在指定位置插入一行。	(int) row, 行	无。
isCellEmpty(row, col)	判断某单元格的值是否为空	(var) row, 行。 (var) col, 列。	(bool)值, 单元格是否为空
isColumnHidden(col)	获取某列是否隐藏	(int) col , 列。	(bool)hide, bool值。
maximum(row, col)	获取单元格的 最大值限制	(int) row 行。 (int) col 列。	(var) maximum
minimum(row, col)	获取单元格的 最小值限制	(int) row 行 (int) col 列	(var) minimum
openFile(path)	打开指定文件路径, 并	(string) path, 字符串类型的文件路径	(bool)类型 , 是否打开成功

	载入		
pageDown()	下一页	无。	无。
pageUp()	上一页	无。	无。
pasteCellData(row,col)	粘贴某单元格的值，复制粘贴的函数配合使用。	(int) row 行 (int) col 列	无
pasteRowData(row)	粘贴某行的值，复制粘贴的函数配合使用。	(var)row, 行。	无
redo()	恢复撤销前的表格操作（目前模式二可以用）	无。	无。
removeRow(row)	删除指定行	(int)row, 行。	无。
saveAs(path)	表格数据另存到文件。	(string)path, 字符串类型的文件路径。	无
saveFile()	表格数据保存到文件。	无	无
scrollBarValue()	获取滑动块当前值	无	(int) value 滑动块值
setCellBackgroundColor(row,col,color)	修改单元格的背景颜色。	(int) row 行 (int) col 列 (string)color	无
setCellData(row,col,data)	修改某单元格的值。	(var)row, 行。 (var)col, 列。	无

		(var)data, 写入的值	
setCellEnable(row, col, ok)	设置是否启用单元格	(var)row, 行。 (var)col, 列。 (var)ok, 是否启用。	无。
setCellFont(row, column, fontName, fontSize, bold, italic, underline)	修改单元格字体。	(int)row, 指定行。 (int)column, 指定列。 (string)fontName, 字体名称。 (int)fontSize, 字体大小, 默认值 12。 (int)bold, 加粗, 默认值 true。 (int)italic, 斜体, 默认值 true。 (int)underline, 下划线, 默认值 true。	无
setCellForegroundColor(row, col, color)	修改单元格的前景颜色。	(int)row, 行。 (int)col, 列。 (string)color	无
setColumnHidden(col, true)	设置某列是否隐藏	(int)col, 列。 (bool)hide, bool 值。	无。
setColumnReadOnly(col, true)	设置某列是否只读。	(int)col, 列。 (bool>true, bool 值。	无。
setColumnWidth(column, width)	设置列宽	(int) column 指定列 (int) width 列宽	无。
setCurrentCell(row, col)	设置表格当前的单元格。	(int)row, 行。 (int)col, 列。	无。

setCurrentColumn(col)	设置表格当前列	(int)col, 列。	无。
setCurrentRow(row)	设置表格当前行	(int)row, 行。	无。
setHorizontalHeaderCellFont(index, fontName, FontSize)	修改某列水平表头字体	(int)index, 指定某列水平表头。 (string)fontName, 字体名称。 (int)FontSize, 字体大小。	无。
setHorizontalHeaderText(index, text)	设置某列水平表头文本。	(int)index, 指定修改的水平表头。 (string)text, 字符串类型的表头文本。	无。
setMultiSelectionEnable(enable)	设置使能表格是否能够多选	(bool)enable, 表格单元格多选使能。	无
setSelectionBehavior(value)	设置表格的选择行为	(int)value: 0: 选择单元格 1: 选择行 2: 选择列	无
setVerticalHeaderText(index, text)	设置某行垂直表头文本	(int)index, 指定修改的垂直表头。 (string)text, 字符串类型的表头文本。	无
titleText()	获取表格标题文本	无	(var) title
totalRowCount()	获取表格总	无	(int) 表格总行

	行数		数。
Undo()	撤销表格操作（目前模式二可以用）	无。	无。

6.38. 树状图

6.38.1. 组件外观



组件包括表头和节点内容。

6.38.2. 简介

插件以树状形式呈现数据。

6.38.3. 属性编辑弹窗

属性编辑弹窗包含“基础”、“样式”和“树”三页。

【基础】



1. 基础

- 1) 每个对象都可以通过设置对象名称 (Object Name) 来标识和访问。对象名称是一个字符串，用于在脚本代码中引用和操作对象。对象名称
- 2) 操作权限

设置树状图插件的操作权限。

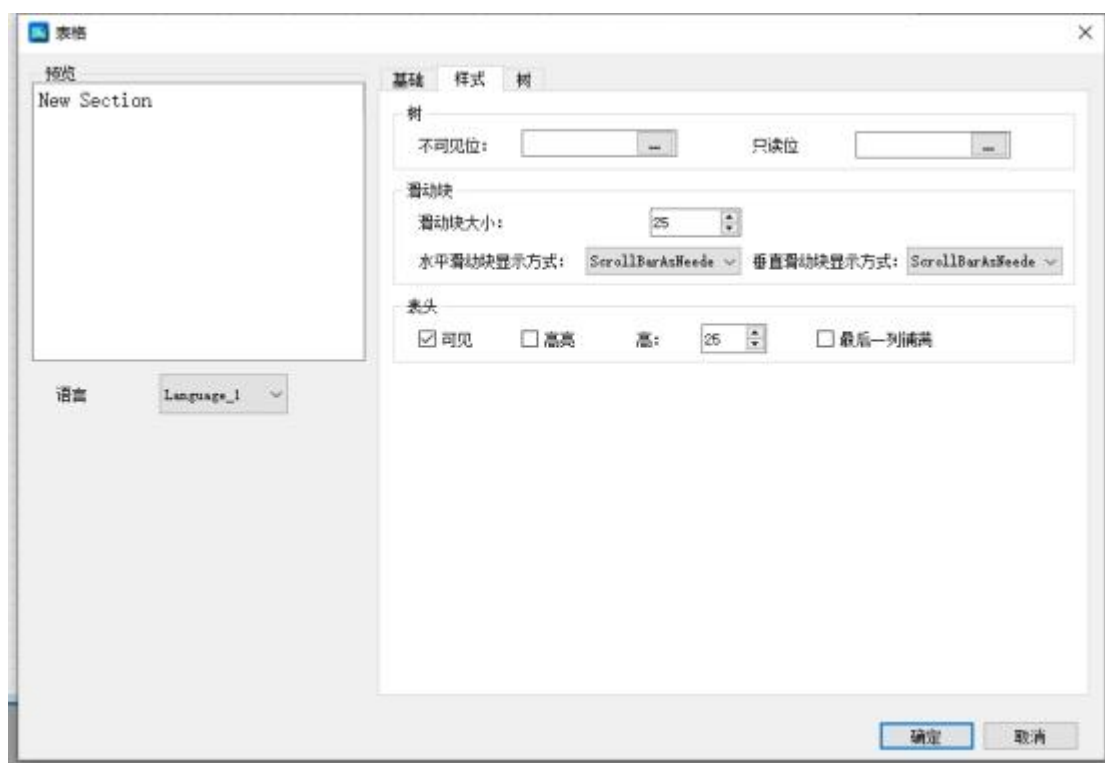
2. 几何尺寸

设置和保存插件的位置和大小的属性。

3. 软键盘

使用触控的方式输入时，可以使用软键盘。[\(跳转到软键盘说明\)](#)，默认是使用字符输入键盘。

【外观】



1. 树

1) 不可见位

不可见位关联系统地址，当不可见位为 true 时，树状图被隐藏不可见。

2) 只读位

只读位关联系统地址，当只读位为 true 时，树状图不可通过软键盘或输入区块编辑。

2. 滑动块

1) 滑动块大小

设置垂直和水平滑动块的尺寸。

2) 水平滑动块显示方式

设置水平滑动块的显示方式三种策略（滑动块在需要时显示、总是显示滑动块、总是不显示滑动块）

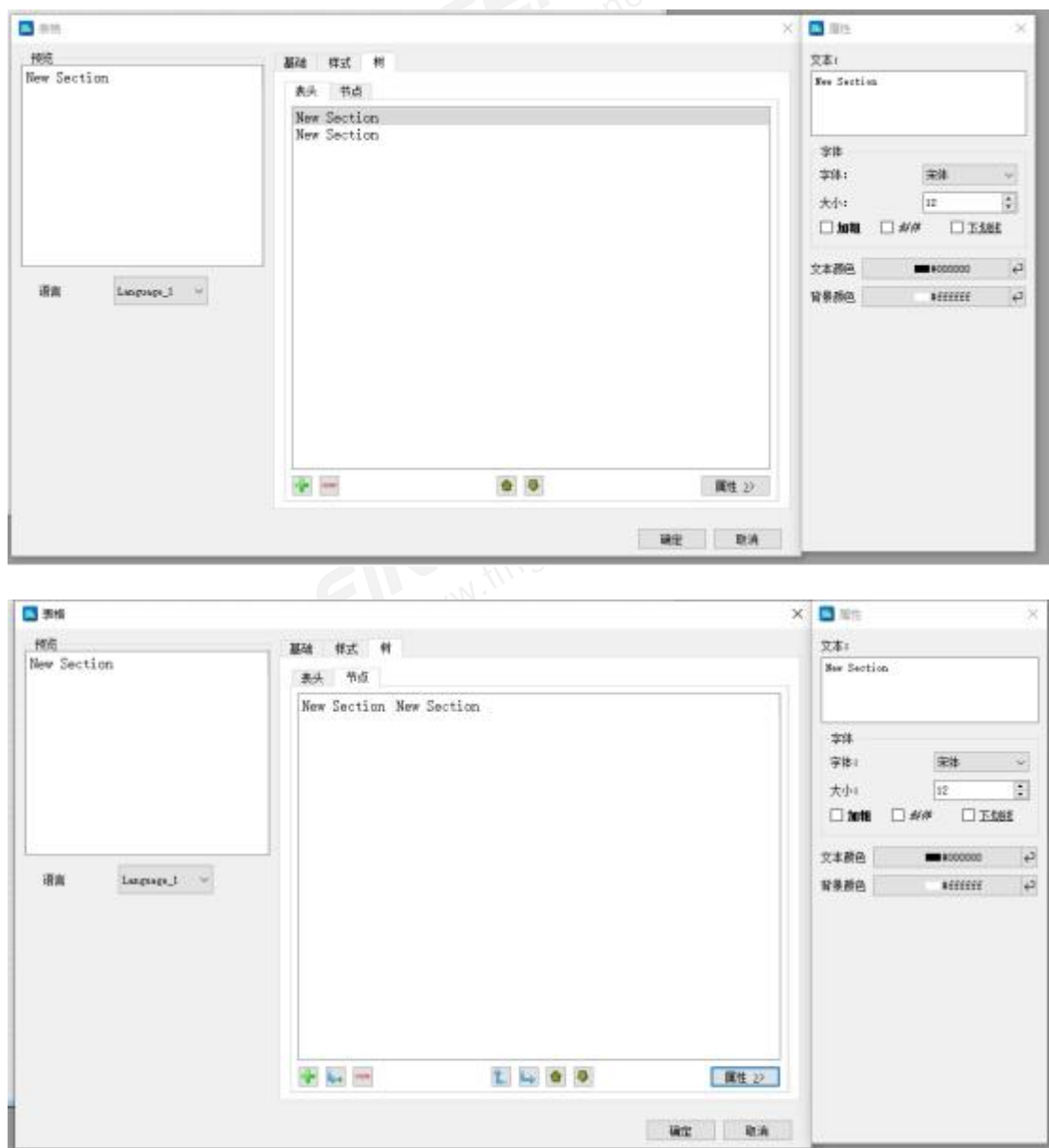
3) 垂直滑动块显示方式

设置垂直滑动块的显示方式三种策略（滑动块在需要时显示、总是显示滑动块、总是不显示滑动块）

3. 表头

- 1) 可见，水平表头是否可见。
- 2) 高亮，水平表头是否高亮。
- 3) 高，水平表头的高度。
- 4) 最后一列铺满，水平表头最后一列是否铺满视口。

【树】



树状图显示内容设置页面。

1. 表头

以列表的方式，设置水平表头数据。对应下方功能按钮，增加、删除、上移和下移。表头都有对应的属性，属性包括显示文本、字体、文本颜色和背景颜色。

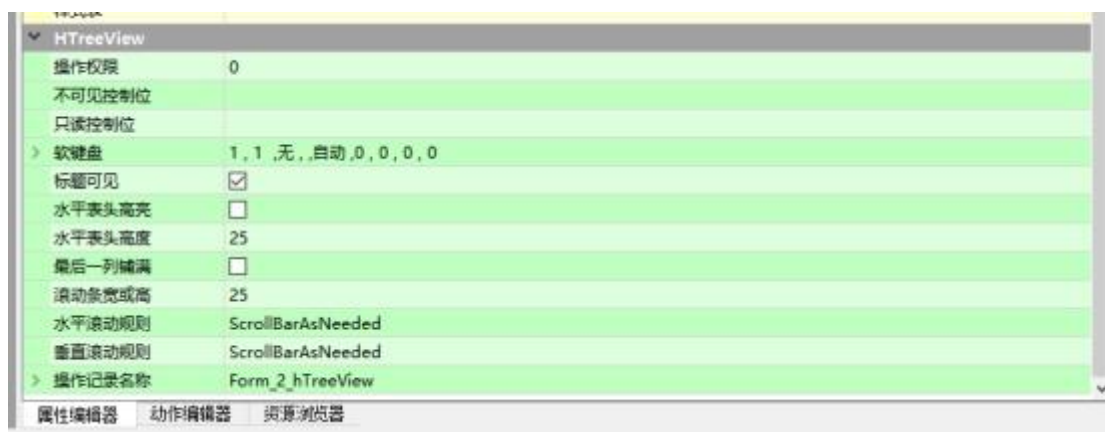
2. 节点

节点页面设置树状图的节点显示内容，对应功能按钮增加、增加子节点、删除、节点上移一层级、节点下移一层级、节点上移和节点下移。每个节点有对应属性，属性包括显示文本、字体、文本颜色和背景颜色。

3. 语言

树状图支持多国语言，切换预览窗口下的语言，可以编辑对应预览语言的多国语言属性。

6.38.4. 属性编辑器



所有内容在属性编辑弹窗上有说明，这里不再赘述。

6.38.5. 系统多国语言

插件没有系统多国语言。

6.38.6. 功能详述

插件以树状形式呈现数据内容，提供读写树状图节点内容的相关接口，包括节点文本、字体和颜色等等。

6.38.7. 属性

属性	描述	用法
visible	读写插件可见属性	var visible = Form.hTreeView.visible
enable	读写插件使能属性	Var enable = Form.hTreeView.enable
geomerty	读写插件的几何尺寸属性, 需要借助HProperty 读写具体值。	var geomerty = Form.hTreeView.geomerty
columnCount	属性只读, 读取树状图列数	var columnCount = Form.hTreeView.columnCount
headerHeight	读写树状图表头高度	var height=Form.hTreeView.headerHeight
headerHighlightSections	读写插件的表头是否高亮	var value = hTreeView.headerHighlightSections
headerVisible	读写插件表头是否显示	var value = hTreeView.headerVisible
operateRecordName	读写插件操作记录名称	var value = hTreeView.operateRecordName
operationalAuthority	读写插件的权限	var value = hTreeView.operationalAuthority
scrollBarExtent	读写滑动块的尺寸	hTreeView.scrollBarExtent=50
stretchLastSection	读写插件是否最后一列铺满	hTreeView.stretchLastSection=true

6.38.8. 函数接口

函数原型	描述	参数	返回值
clearSection(parent)	清空	(Index)parent 如果	无

	parent 节点下 的所有 子节点	不设置 parent 则清 空树状图，Index 类 型变量可通过函数 index/currentIndex 获取	
collapsesSection(index)	折 叠 目 标节点	(Index)index 目标 节点，可通过函数 index/currentIndex 获取	无
Collapse()	折 叠 树 状 图 的 所 有 节 点	无。	无。
currentIndex()	获 取 树 状 图 当 前节点	无	(Index) 当 前节点
expand()	展 开 树 状 图 所 有节点	无	无
expandSection	展 开 树 状 图 index 节 点	(Index)index 目标 节点，可通过函数 index/currentIndex 获取	无
headerText()	获 取 第 column 列 的 表 头文本	(int)column	(string) 表头文本
Index(row, column, parent)	获 取	(int)row, 行。	(Index) 目

	parent 节 点 下 row 行 column 列 的 Index	(int)column, 列。 (Index)parent, 目标 节点的 parent 节点, 如果不设置则为根节 点	标节点
indexOfParent (parent)	返 回 基 于 父 节 点 的 行 位置。	(Index)目标节点	(int) 目标 节点基于 父节点的 行定位
insertColumn(column)	为 树 状 图 在 column 插 入 一 列	(int) column 在第 column 列插入一列	无
insertRow(row, parent)	在 parent 节 点 下 插 入 第 row 行	(int)row 在 row 行 插入 (Index)parent 在 parent 下插入, 如果 不设置 parent 则在 根节点下插入, Index 类型变量可通过函数 index/currentIndex 获取	无
parentSection(index)	获 取 index 节 点 的 父 节点	(Index)目标节点	(Index) 目 标节点的 父节点

removeColumn(column)	删除树状图第column列	(int)column , 删除列	无
removeRow?2(row , parent)	删除parent节点下的第row行	(int)row 删除第row行 (Index)parent 在parent下插入, 如果不设置parent则在根节点下删除, Index类型变量可通过函数index/currentIndex获取	无
sectionText(index)	获取目标节点的文本	(Index)index 目标节点, 可通过函数index/currentIndex获取	(string) 节点文本
setHeaderText(column, text)	设置第column列的表头文本	(int)column, 列。 (string) 文本	无
setSectionBackgroundColor(index, color)	设置节点的背景颜色	(Index)index 目标节点, 可通过函数index/currentIndex获取 (var)color 颜色, 可通过 HProperty 获取颜色类型	无

setSectionFont(index, font)	设置目标节点的字体	(Index)index 目标节点，可通过函数 index/currentIndex 获取 (var)font 字体，可通过 HProperty 获取字体类型	无
setSectionText(index, text)	设置目标节点的文本	(Index)index 目标节点，可通过函数 index/currentIndex 获取 (string)text 写入文本	无
setSectionTextColor(index, color)	设置节点文本颜色	(Index)index 目标节点，可通过函数 index/currentIndex 获取 (var)color 颜色，可通过 HProperty 获取颜色类型	无

6.38.9. 信号

原型	描述	
sectionDataChanged(Index)	节点数据发生改变	信号关联脚本函数 函数中通过 arguments[0] 访问第一个参数 Index，获取数据发生变化的节点 index

sectionClicked(index)	节点被点击	信号关联脚本函数 函数中通过 arguments[0] 访问第一个参数 Index，获取被点击的节点 index
currentSectionChanged(index, index)	当前节点改变	信号关联脚本函数 函数中通过 arguments[0] 访问第一个参数 Index，获取当前节点，arguments[1] 访问第一个参数 Index，获取改变前的节点。

6.39. 列表框

6.40. 日期时钟

6.41. 动态图片

6.42. 动态文本

6.43. 坐标显示器

6.44. 历史报警显示器

6.45. 当前执行显示器

6.46. 操作记录

6.46.1. 功能介绍

操作记录是记录计算机系统或应用程序的使用历史,包括用户在系统中进行的操作和相应的结果。当出现系统故障或异常时,通过查看操作记录可以确定问题原因,进而采取相应的措施加以解决。操作记录可以帮助企业了解用户行为和需求,分析用户偏好和行为模式,从而优化产品设计和运营策略。通过操作记录组件,可以回溯操作历史。

目前表格集、代码编辑器、文件选择器、文件传输、数值、整数输入、小数输入、按钮、下拉框,拥有用户操作记录。

操作记录提供多种查找方式,

- 按记录者查找。
- 按操作内容查找。
- 按详细查找。
- 支持指定查找时间范围。



可以添加自定义的操作记录。

//添加自定义记录

//operate:操作

//detail:详细

bool addOperateRecord(operate, detail);

宏向导中添加接口，供用户在宏程序中添加自定义的操作记录。



6.46.2. 组件结构



组件由水平表头、垂直表头、表格内容、水平滚动条、垂直滚动条组成。

6.46.3. 属性编辑器

操作记录组件未提供属性设置弹窗，可通过属性编辑器设置相关属性。

1. 滚动条宽或高

滚动条宽或高	18
--------	----

2. 软键盘

软键盘	1, 1, 无, , 自动, 0, 0, 0, 0
自动隐藏	☒
可移动	☒
软键盘类型	无
软键盘画面	
位置策略	自动
X	0
Y	0
宽度	0
高度	0

使用触控或鼠标的方式输入时，可以使用软键盘。（[跳转到软键盘说明](#)）

3. 设备

设备	Device0
----	---------

设置使用远程可视化时，需要连接的设备。

6.46.4. 脚本宏

具体可见 macroWizard 文件夹的 “H0perationRecord_A20_Designer_CN.xml”

6.47. 进度条

6.48. HVisionView（视觉插件）

6.49. 标签

6.50. 指示灯

6.51. 框架

6.52. 网格分割器

6.53. 选项组

6.54. 切换卡

6.55. HCam2D

6.56. 五面钻

6.57. CAM 插件

6.58. 凸轮磨边

6.59. SpringEditorPlugins（3D 弹簧）

注意：功能开启相关的系统变数

要使用 3D 弹簧功能需要确保控制器的下述系统参数为开启状态

COM【40782】=1，该参数控制 3D 弹簧是否开启

COM【73831】=64，即 C73831.6=1，此参数控制 3D 弹簧是否使能

注意：设置 73831=64 之后需要点击 Trans 上的【重置】按钮再重启才能保存此变量，如果重启后该变量的值依然为 64 说明开启成功

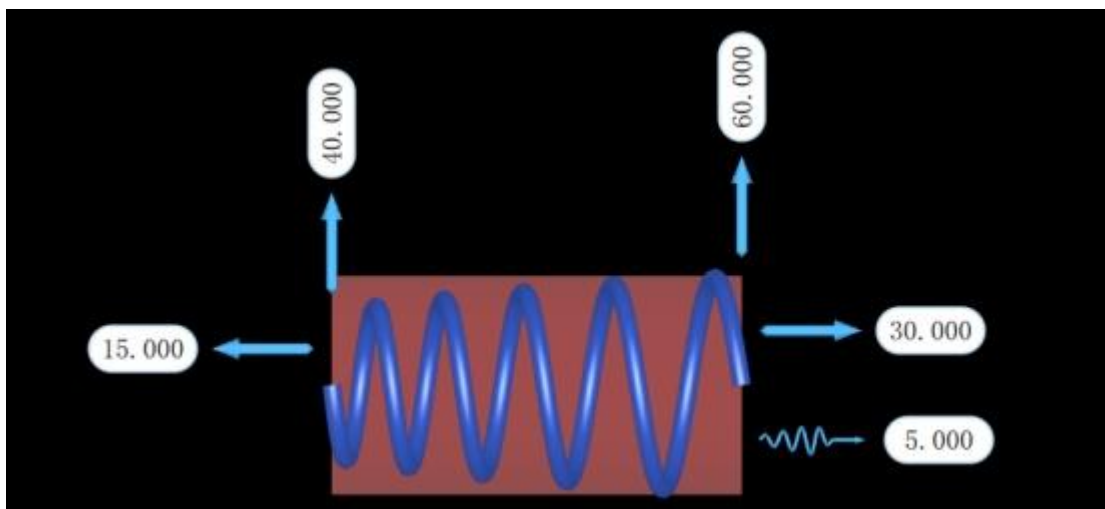
	Value	R/W	Default	Unit	Backup	Immediate	Range	
040780	1	1	R	0	NULL	N	I	0~1 六面钻功能是否开启
040781	1	1						*****
040782	1	1						*****



参数	定义	范围	单位	默认值	生效方式
#C73831.6	3D 弹簧机功能是否开启	0~1	位型	0	重启
	=0，未开启 =1，开启，请致电业务开启该功能才可正常使用， B0x_V1_2023-09-06-09-16.axBin 该版本及之后版本支持				

6.59.1. 弹簧属性

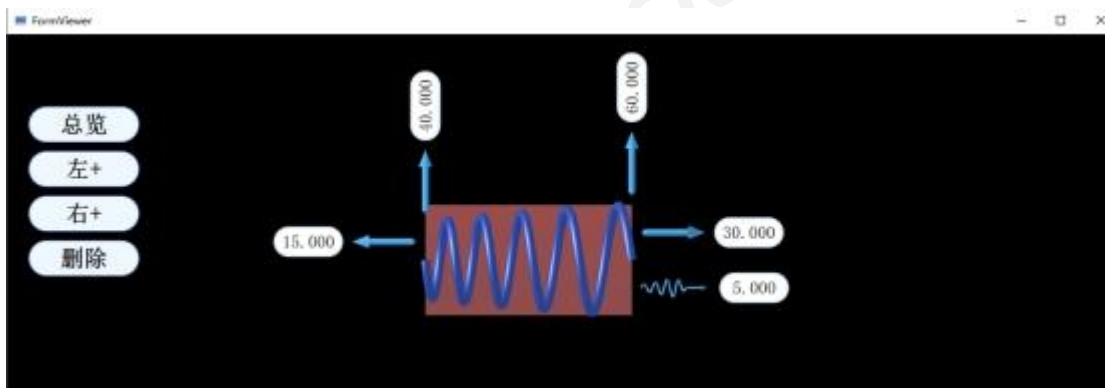
- 1. 左外径
- 2. 右外径
- 3. 左螺距
- 4. 右螺距
- 5. 圈数



如图示为一个弹簧段，左外径为 40，右外径为 60，左螺距为 15，右螺距为 30，圈数为 5。

6.59.2. 交互操作

1. 3D 界面展示



在 3D 界面内只提供方便常用简易的操作，更多的接口功能请在 HMI 接口中使用。

2. 基本操作

- 1) 弹簧属性值修改: 选中需要编辑的弹簧，在弹簧段周围会出现 5 个箭头以及属性值显示，
- 2) 拖动箭头实现快速的属性修改，或点选属性值手动输入
- 3) 删除弹簧: 选中弹簧后，点选界面左边的删除按钮，即可删除该弹簧。注意：如果只剩一个弹簧，则不能删除。
- 4) 增加弹簧: 选中弹簧后，点选界面左边的（左+）按钮，即可在该弹簧的左边增加一段弹簧。（右+）按钮同理。

- 5) 总览：点选总览按钮后，整段弹簧会水平显示在视图中央。
- 6) 视图操作：
 - a. 按住鼠标左键并移动，可实现视图的旋转
 - b. 按住鼠标中键并移动，可实现视图的平移
 - c. 滚动鼠标滚轮，可实现视图的放大缩小

6.59.3. HMI 接口

函数名	描述
setSpringData(QVariant dataList, double wire_diameter)	设置弹簧数据，包括弹簧段数据二维数组和线径
setSingleSpringData(int index, QVariant data)	设置下标为 index 的弹簧的数据
springData()	返回弹簧数据
setBackgroundColor(int r, int g, int b)	设置背景颜色
setSpringColor(int index, int r, int g, int b)	设置下标为 index 的弹簧颜色
setDefaultTurns(int turns)	设置在 3D 界面编辑中新增弹簧时的默认圈数
setRings(int rings)	设置弹簧圆弧精度
setSlices(int slices)	设置弹簧截面精度
setEditEnabled(bool enabled)	设置 3D 编辑使能
setWireDiameter(double diameter)	设置线径
wireDiameter()	返回线径
springCount()	返回弹簧段数
setSelected(index)	选中下标为 index 的弹簧
deleteSpring(index)	删除下标为 index 的弹簧
addSpring(int index, QVariant data)	在下标 index 处添加弹簧
showAll()	总览视图

heightAt(int index)	返回下标为 index 的弹簧高度
totalHeight()	返回弹簧总高度
lengthAt(int index)	返回下标为 index 的弹簧线长
totalLength()	返回弹簧总线长
panAboutViewCenter(double angle)	按角度（度）调整相机绕视图中心的平移
tiltAboutViewCenter(double angle)	按角度（度）调整相机围绕视图中心的倾斜
rollAboutViewCenter(double angle)	按角度（度）调整摄影机绕视图中心的滚动
undo()	撤销。只通过接口进行弹簧参数的增删改操作后才生效
redo()	重做。只通过接口进行弹簧参数的增删改操作后才生效

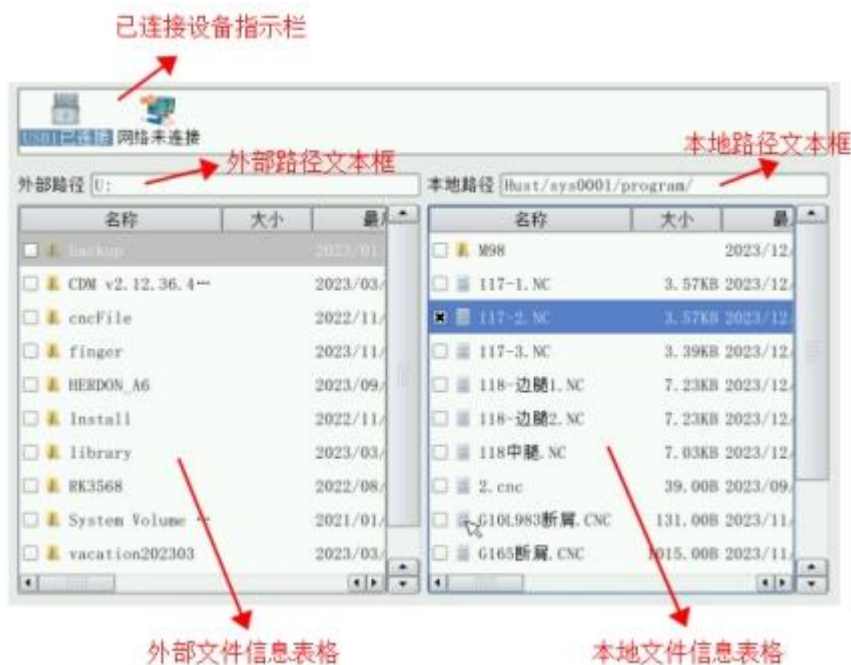
信号	描述
editFinished()	3D 编辑完成后发送信号
indexChanged(int)	选中弹簧时发送信号返回弹簧下标

6.60. 程序传输

6.60.1. 功能介绍

文件传输的作用是将一个文件从设备传输到另一个设备，常用的有从控制器本地文件系统传输文件到 U 盘，或从 U 盘将文件传输到控制器。此外，文件传输也提供了删除、新增、重命名、复制、粘贴等常用的文件操作，具体可见“HFilecopy_Designer_CN.xml”。

6.60.2. 组件结构



1. 已连接设备指示栏

显示当前已连接的设备，目前支持 U 盘设备，暂不支持网络设备。

2. 外部路径文本框

显示外部设备的当前目录路径。

3. 本地路径文本框

显示本地设备的当前目录路径。

4. 外部文件信息表格

显示外部设备的文件信息。

5. 本地文件信息表格

显示本地设备的文件信息。

6.60.3. 属性编辑器

文件传输组件未提供属性设置弹窗，可通过属性编辑器设置相关属性。

1. 几何尺寸

几何尺寸	[(60, 70), 621 x 401]
X	60
Y	70
宽度	621
高度	401

该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

2. 字体

字体	A [宋体, 9]
字体族	宋体
点大小	9
粗体	☐
斜体	☐
下划线	☐
删除线	☐
字距调整	☒
反锯齿	首选默认

此属性保存当前为组件设置的字体，各项参数说明可见“[字体](#)”。

3. 软键盘

软键盘	1, 1, 无, , 自动, 0, 0, 0, 0
自动隐藏	☒
可移动	☒
软键盘类型	无
软键盘画面	
位置策略	自动
X	0
Y	0
宽度	0
高度	0

设置组件使用的软键盘，具体可见“[软键盘](#)”。

4. 默认通道路径

默认通道路径	1
--------	---

该属性设置组件使用“限制打开路径”中对应 ID 号的限制路径。

注：“限制打开路径”是指只能打开该路径和该路径下的子目录，避免用户访问到重要的系统目录，进行了误操作。

5. 限制打开路径

限制打开路径	1=/home/root/hust/usr/sys000
--------	------------------------------

设置限制打开路径。



注：“限制打开路径”是指只能打开该路径和该路径下的子目录，避免用户访问到重要的系统目录，进行了误操作。

6. 游标颜色

> 游标颜色	[0, 120, 215] (255)
--------	---------------------

设置表格中游标的颜色。



7. 聚焦时文本颜色

> 聚焦时文本颜色 [0, 0, 0] (255)

设置游标所在行的文本颜色。

8. 选中时背景颜色

> 选中时背景颜色 [255, 255, 191] (255)

设置文件被选中时的背景颜色。

优先级： 游标颜色 > 选中时背景颜色 > 交替色 > 背景颜色

9. 文本颜色

> 文本颜色 [85, 255, 0] (255)

设置表格中的文本颜色。

优先级： 聚焦时文本颜色 > 文本颜色

10. 背景颜色

> 背景颜色

[255, 255, 255] (255)

设置组件的背景颜色。



11. 第一种交替色

> 第一种交替色

[85, 255, 127] (255)

设置表格的第一种交替色。



优先级： 游标颜色 > 选中时背景颜色 > 交替色 > 背景颜色

12. 第二种交替色

> 第二种交替色 [255, 255, 255] (255)

设置表格的第二种交替色。

13. 水平表头文本颜色

> 水平表头文本颜色 [0, 0, 0] (255)

设置表格的水平表头文本颜色。

14. 水平表头背景颜色

> 水平表头背景颜色 [255, 255, 255] (255)

设置表格的水平表头背景颜色。

15. 设备

设备	Device0
----	---------

设置远程可视化时，该组件显示哪个设备的文件。

16. 操作记录

操作记录	☐
------	--------------------------

设置是否开启该组件的操作记录功能。具体可见“[操作记录](#)”。

17. 操作记录名称

> 操作记录名称	Form_hFilecopy
----------	----------------

设置该组件在操作记录中的命名。具体可见“[操作记录](#)”。

6.60.4. 动作编辑器

动作编辑器	
事件	宏
双击	

双击表格中的某行，会触发该事件。

6.60.5. 脚本宏

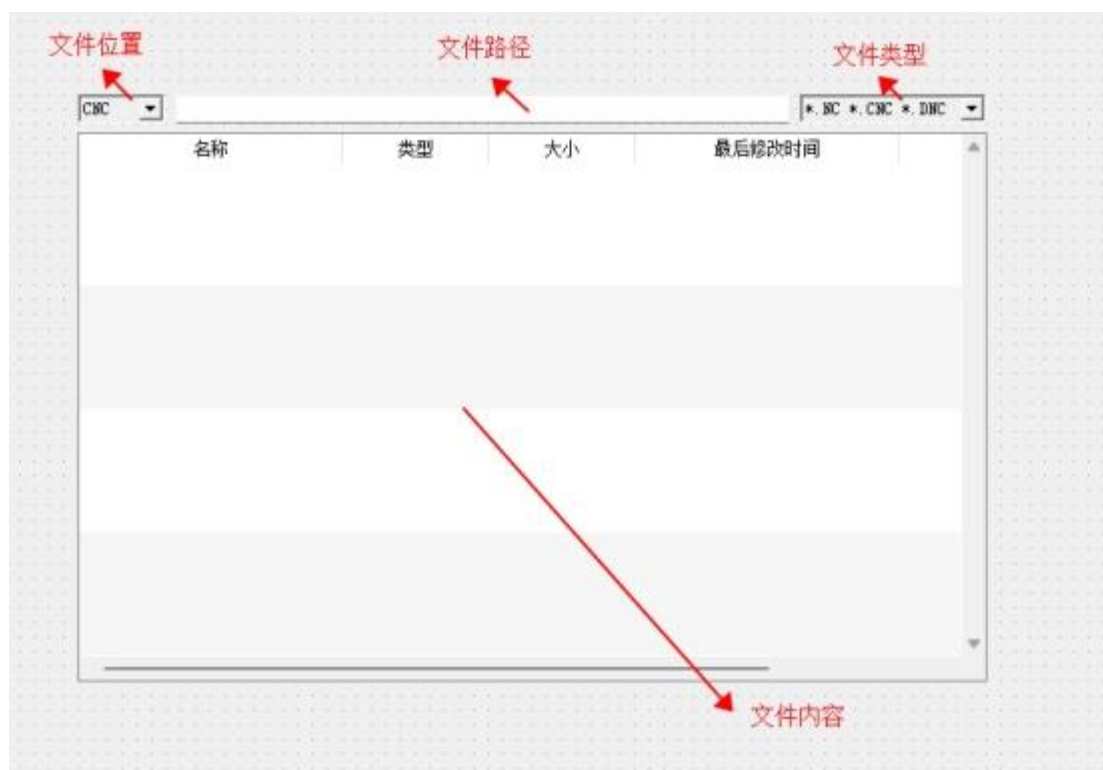
详情可见“HFilecopy_Designer_CN.xml”。

6.61. 文件选择器

6.61.1. 功能介绍

文件选择器用于从文件系统中选择文件。它允许用户浏览文件系统，选择一个或多个文件。此外，文件选择器也提供了删除、新增、重命名、复制、粘贴等常用的文件操作，具体可见“HFiledialog_Designer_CN.xml”。

6.61.2. 组件结构



1. 文件位置

在控制器中运行时，文件来源于本地文件、U 盘（插入 U 盘后）、远程 PC（将 PC 文件挂载到控制器），可以通过“文件位置”下拉框切换。

CNC

UIDisk

../sys0001/program

*. NC *. CNC *. DNC

名称	类型	大小	最后修改时间	信息
M98	Folder		18:28:20 2023/12/14	
117-1	nc	3.567KB	18:28:20 2023/12/14	(...
117-2	nc	3.567KB	18:28:20 2023/12/14	(...
117-3	nc	3.392KB	18:28:20 2023/12/14	(...
118-边腿1	nc	7.228KB	18:28:20 2023/12/14	(...
118-边腿2	nc	7.227KB	18:28:20 2023/12/14	(...
118中腿	nc	7.033KB	18:28:20 2023/12/14	(...
2	cnc	39B	11:11:37 2023/09/25	[%...
G10L983断屑	cnc	131B	12:33:30 2023/11/24	G1...
G165断屑	cnc	1015B	12:33:30 2023/11/24	#M...
G32车牙	cnc	255B	12:33:30 2023/11/24	G1...
G76车牙	cnc	119B	12:33:30 2023/11/24	G1...

2. 文件路径

显示当前文件夹路径。

3. 文件类型

表格中只显示指定文件类型的文件。

4. 文件表格

显示文件信息。

6.61.3. 属性编辑器

文件选择器组件未提供属性设置弹窗，可通过属性编辑器设置相关属性。

1. 几何尺寸

√ 几何尺寸	[(60, 70), 621 x 401]
X	60
Y	70
宽度	621
高度	401

该属性控制组件在父窗口中的位置与大小，各项参数说明可见“[几何尺寸](#)”。

2. 字体

√ 字体	A [宋体, 9]
字体族	宋体
点大小	9
粗体	☐
斜体	☐
下划线	☐
删除线	☐
字距调整	☒
反锯齿	首选默认

此属性保存当前为组件设置的字体，各项参数说明可见“[字体](#)”。

3. 文件打开类型

文件打开类型	*.NC *.CNC *.DNC
--------	------------------

设置文件选择器中可见的文件类型，不同类型之间使用空格分隔。

4. “选择”列是否可见

selectColumnVisible	☐
---------------------	--------------------------

设置表格中“选择”列是否可见。

5. 文件名显示

文件名显示	☒
-------	-------------------------------------

设置表格中“名称”列是否可见。

6. 文件类型显示

文件类型显示	☒
--------	-------------------------------------

设置表格中“类型”列是否可见。

7. 文件大小显示

文件大小显示	☒
--------	-------------------------------------

设置表格中“大小”列是否可见。

8. 最后修改日期显示

最后修改日期显示	☒
----------	-------------------------------------

设置表格中“最后修改时间”列是否可见

9. 文件信息显示

文件信息显示	☒
--------	-------------------------------------

设置表格中“信息”列是否可见。

10. “选择”列宽

selectColumnWidth	30
-------------------	----

设置表格中“选择”列的宽度。

11. 文件名列宽

文件名列宽	180
-------	-----

设置表格中“名称”列的宽度。

12. 文件类型列宽

文件类型列宽	100
--------	-----

设置表格中“类型”列的宽度。

13. 文件大小列宽

文件大小列宽	100
--------	-----

设置表格中“大小”列的宽度。

14. 最后修改日期列宽

最后修改日期列宽	180
----------	-----

设置表格中“最后修改时间”列的宽度。

15. 文件信息列宽

文件信息列宽	200
--------	-----

设置表格中“信息”列宽。

16. 文件位置可见

fileLocationVisible	☒
---------------------	-------------------------------------

设置“文件位置”下拉框是否可见。

17. 文件路径显示

文件路径显示	☒
--------	-------------------------------------

设置“文件路径”文本框是否可见。

18. 文件类型筛选显示

文件类型筛选器显示	☒
-----------	-------------------------------------

设置“文件类型”下拉框是否可见。

19. 最后修改日期格式

最后修改日期格式	HMSYMD
----------	--------

设置表格中“最后修改日期”的显示格式，

YMDHMS：年月日时分秒

HMSYMD：时分秒年月日

20. 软键盘

√ 软键盘	1, 1, 无, , 自动, 0, 0, 0, 0
自动隐藏	☒
可移动	☒
软键盘类型	无
软键盘画面	
位置策略	自动
X	0
Y	0
宽度	0
高度	0

设置组件使用的软键盘，具体可见“[软键盘](#)”。

21. 默认通道路径

默认通道路径	1
--------	---

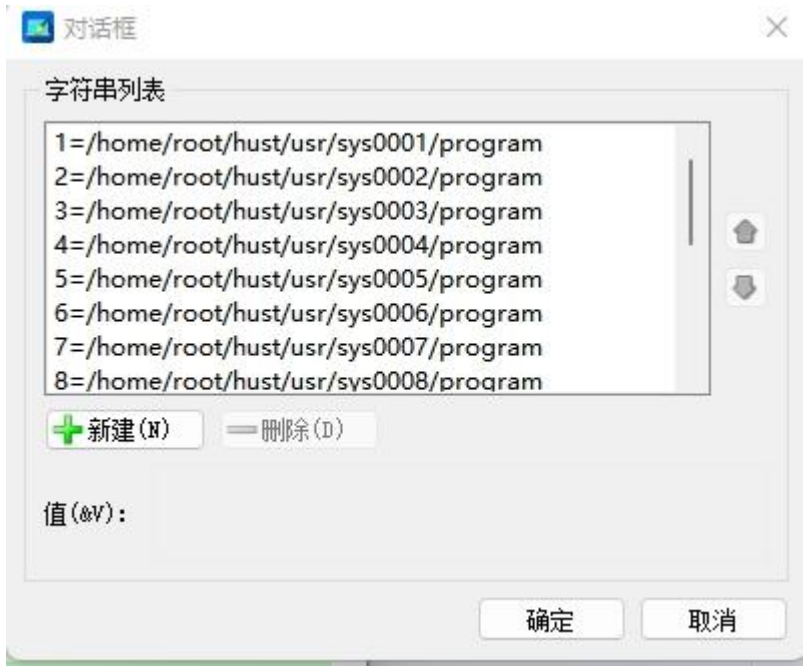
该属性设置组件使用“限制打开路径”中对应 ID 号的限制路径。

注：“限制打开路径”是指只能打开该路径和该路径下的子目录，避免用户访问到重要的系统目录，进行了误操作。

22. 限制打开路径

限制打开路径	1=/home/root/hust/usr/sys000
--------	------------------------------

设置限制打开路径。

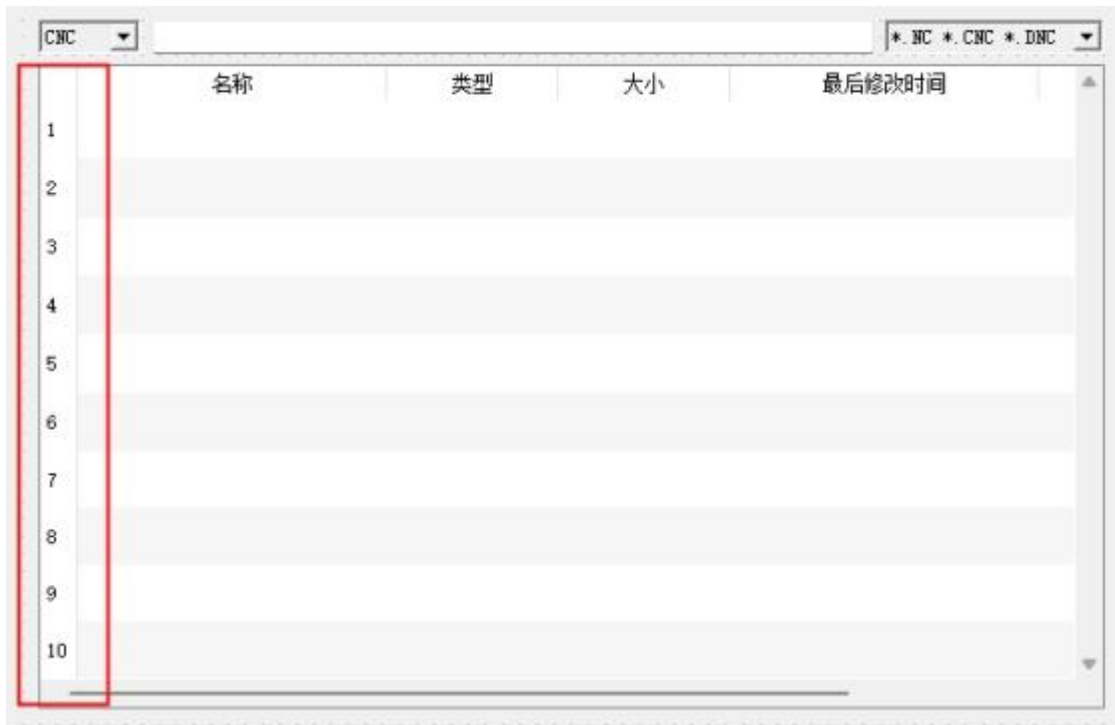


注：“限制打开路径”是指只能打开该路径和该路径下的子目录，避免用户访问到重要的系统目录，进行了误操作。

23. 文件序号显示



设置表格中显示文件序号。



24. 水平游标显示

水平游标显示	☒
--------	-------------------------------------

设置表格是否显示水平滚动条。

25. 垂直游标显示

垂直游标显示	☒
--------	-------------------------------------

设置表格是否显示垂直滚动条。

26. 水平滚动条大小

水平滚动条大小	22
---------	----

设置水平滚动条的高度。

27. 垂直滚动条大小

垂直滚动条大小	22
---------	----

设置垂直滚动条宽度。

28. 文件大小列模式

文件大小列模式	Size
---------	------

设置表格中“大小”列显示的内容。

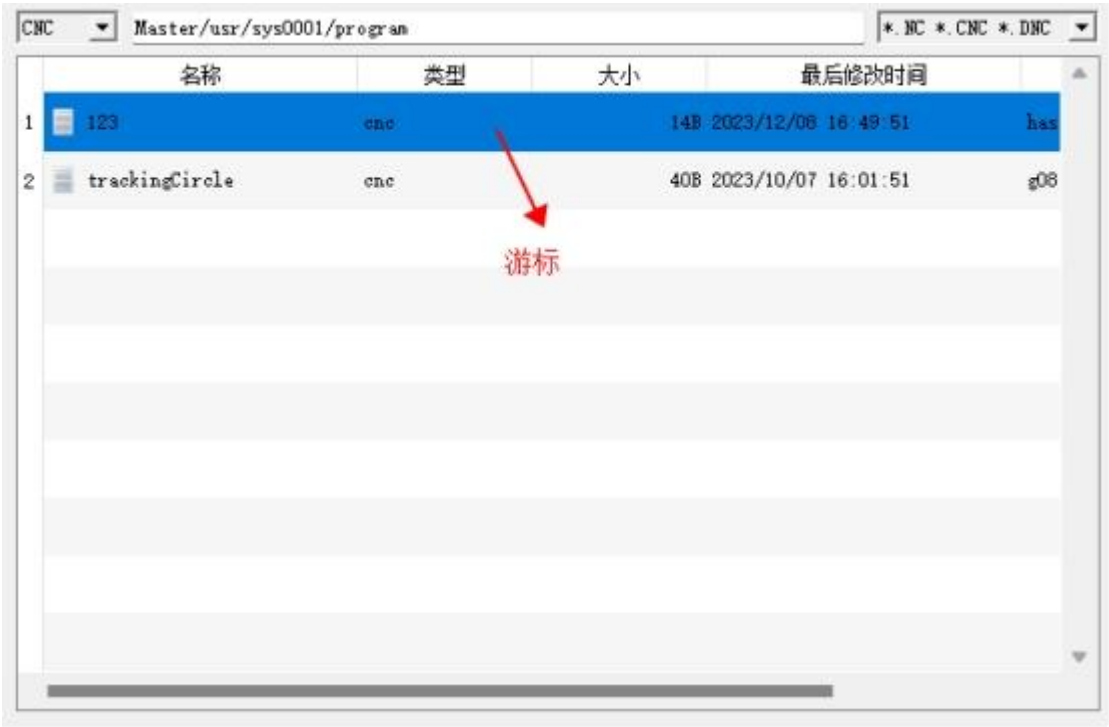
Size：显示文件大小。

LineCount：显示文件行数。

29. 游标颜色

> 游标颜色	☒ [0, 120, 215] (255)
--------	---

设置表格中游标的颜色。



30. 聚焦时文本颜色。

> 聚焦时文本颜色 [0, 0, 0] (255)

设置光标所在行的文本颜色。

31. 选中时背景颜色

> 选中时背景颜色 [255, 255, 191] (255)

设置文件被选中时的背景颜色。

优先级： 光标颜色 > 选中时背景颜色 > 交替色

32. 文本颜色

> 文本颜色 [85, 255, 0] (255)

设置表格中的文本颜色。

优先级： 聚焦时文本颜色 > 文本颜色

33. 第一种交替色

> 第一种交替色 [255, 255, 255] (255)

设置表格的第一种交替色。



34. 第二种交替色

> 第二种交替色 [246, 246, 246] (255)

设置表格的第二种交替色。

35. 水平表头文本颜色

> 水平表头文本颜色 [0, 0, 0] (255)

设置表格的水平表头文本颜色。

36. 水平表头背景颜色

> 水平表头背景颜色 [255, 255, 255] (255)

设置表格的水平表头背景颜色。

37. 垂直表头文本颜色

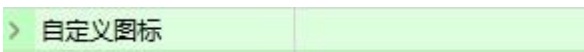


设置垂直表头的文本颜色。

38. 垂直表头背景颜色



39. 自定义图标



设置文件使用的图标，搭配 setUserIconEnable 接口使用。

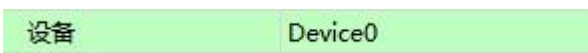
```
//设置使用自定义图标的文件
```

```
//filePath: 文件路径
```

```
//enable: true-使用自定义图标 false-使用默认图标
```

```
void setUserIconEnable(const QString &filePath, bool enable);
```

40. 设备



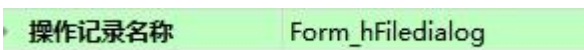
设置远程可视化时，该组件显示哪个设备的文件。

41. 操作记录



设置是否开启该组件的操作记录功能。具体可见“[操作记录](#)”。

42. 操作记录名称



设置该组件在操作记录中的命名。具体可见“[操作记录](#)”。

6.61.4. 动作编辑器

动作编辑器	
事件	宏
currentItemChange	
双击	

1. 当前选中项改变

表格中的当前选中项改变时，会触发该事件。

2. 双击

双击表格中的某行，会触发该事件。

6.61.5. 脚本宏

详情可见“HFiledialog_Designer_CN.xml”。

6.62. 参数监控表格

6.63. 自定义程式跳转

6.64. 伺服参数

6.65. 键盘容器

6.66. 键盘按键

6.67. 键盘显示

7. 基本宏函数

7.1. 宏编辑器

7.2. System 函数群

System 函数群中列举出了系统基本函数，可以对系统进行一些基本操作。

7.2.1. getData（获取变量的值）

getData 函数用于获取系统变量的值，通过其返回值来获取相应系统变量的当前值。

用法

System.getData(channel, dataClass, address)

参数说明

- channel（通道号）：数值型，取值区间为[1, 16]
- dataClass（系统变数类型）：数值型，取值区间为[0, 5]

以下为 dataClass 各取值对应的意义：

=0 User;

=1 Mcm;

=2 Sys;

=3 Reg;

=4 Bus;

=5 Com;

➤ address（变数地址）：数值型。

➤

范例

例 1: 获取通道 1 中#U100 值:

定义一个变量来存放#U100 的值

```
var USR100=System.getData(1,0,100)
```

例 2: 获取通道 2 中#R200 值:

定义一个变量来存放#U100 的值

```
var REG200=System.getData(2,3,200)
```

7.2.2. setData（设定变量的值）

setData 宏用于设定系统变量，无返回值。

用法

```
System.setData(channel, dataClass, address, value)
```

参数说明

➤ channel（通道号）：数值型，取值区间为[1, 16]

➤ dataClass（系统变数类型）：数值型，取值区间为[0, 5]

以下为 dataClass 各取值对应的意义:

=0 User;

=1 Mcm;

=2 Sys;

=3 Reg;

=4 Bus;

=5 Com;

➤ address（变数地址）：数值型。

➤ value（要写入的值）：浮点型。

范例

例 1：将通道 1 中#U100 设定为 10: System.setData(1, 0, 100, 10)

例 2：将通道 2 中#R100 设定为 50: System.setData(2, 3, 100, 50)

7.2.3. getBit（获取变量位的值）

获取系统变量某一位的状态，返回值为 bool 型，true 代表该位状态为 ON，false 代表该位状态为 OFF

用法

System.getBit(channel, bitType, address, bitIndex)

参数说明

➤ channel（通道号）：数值型, 取值区间为[1, 16]

➤ bitType（数据位类型）：数值型, 取值区间为[0, 12]

表 4.1.3.1-1 为 bitType 各取值对应的意义：

=0 I_BIT;

=1 O_BIT;

=2 C_BIT;

=3 S_BIT;

=4 A_BIT;

=5 Cn_BIT;

=6 Tm_BIT;

=7 User_BIT;

=8 Mcm_BIT;

=9 Sys_BIT;

=10 Reg_BIT;

=11 Bus_BIT;

=12 Com_BIT;

➤ address (设置数据地址): 数值型

➤ bitIndex (位索引): 数值型, 取值区间[0, 31], 如果数据位类型是 User、Mcm、Sys、Reg, 才需设置, 表示要设定的系统变数位编号。

范例

例 1: 获取第 1 通道#ABIT1 状态

```
var ABit1= System.getBit(1, 4, 1)
```

例 2: 获取第一通道#U100 第 2 位的状态

```
var ABit1= System.getBit(1, 7, 100, 2)
```

7.2.4. setBit (设置通道数据位为 0 或者 1)

用于设定指定系统变数某一位的状态, 无返回值

用法

```
System.setBit(channel, bitType, address, bitIndex, bit)
```

参数介绍

➤ bit (需要设定的位的值): 数值型, 取值区间[0, 1], 为 0 时表示位 OFF, 为 1 时表示位 ON

➤ channel, bitType, address, bitIndex 等参数详细介绍请参见第 4.1.3.1 节。

范例

例 1: 将第 1 通道#ABIT1 设为 ON

```
System.setBit(1, 4, 1, 1)
```

例 2: 将第 1 通道#U100 第 2 位设为 OFF

```
System.setBit(1, 7, 100, 2, 0)
```

7.2.5. saveData (保存数据)

保存数据, 无返回值, 用于保存自定义数据

用法

```
System.saveData(ident,value)
```

参数说明

- ident（数据标识）：字符型
- value（需要保存的数据）：可以是数值、字符串、变量、组件属性

范例

例 1：保存一个标识为“count”值为 100 的数据

```
System.saveData("count",100)
```

例 2：保存画面“Form”中名为“checkBox1”的控件的“checked”属性

```
System.saveData("checked",Form.checkBox1.checked)
```

7.2.6. loadData（读取数据）

用于读取使用 saveData 宏保存过的数据，返回值即为读取出来的数据。

用法

```
System.loadData(ident)
```

参数说明

- ident（数据标识）：字符型

范例

例 1：读取标识为“count”的数据

```
Var count = System.loadData("count")
```

7.2.7. colorFromString（将字符转换成颜色值）

将字符转换成颜色值，返回值为转换后的颜色值。

用法

```
System.colorFromString(name)
```

参数说明

- name（颜色值）：字符型，可以为如下格式：

#RGB（每个 R,G,B 是一个 16 进制符）

#RRGGBB

#RRRGGBBB

#RRRRGGGBBBB

标准颜色名请参见如下网站：

<http://www.w3.org/TR/SVG/types.html#ColorKeywords>

范例

例 1：将字符串“#00ff00”转换为颜色值

```
System.colorFromString("#00ff00")
```

7.2.8. loadExeProgramFile（载入执行程序式）

载入执行程序式，用于加载控制器中的 CNC 程式，无返回值。

用法

```
System.loadExeProgramFile(channel, filename)
```

参数说明

- channel（通道号）：数值型，取值区间[1, 16]
- filename（文件路径及名称）：字符串，填写该参数时需要加双引号

范例

例 1：加载第一通道中名为“T1.CNC”的程式

```
System.loadExeProgramFile(channel, "/usr/sys0001/program/T1.CNC")
```

7.2.9. saveVar（存储指定变数）

存储指定变数，当指定的通道号，或数据类型正确时，该宏返回 true，否则返回 false。

用法

```
System.saveVar(channel, dataType, filename)
```

参数说明

- channel（通道号）：数值型，取值区间[1, 16]
- dataType（数据类型）：整数型，具体意义如下：
 - =0 指定通道的用户变数固定路径：/usr/sys00xx/var/userCurrent.var;
 - =1 指定通道的 MCM 固定路径：/usr/sys00xx/var/mcmCurrent.var;

=2 指定通道的系统变数（前 10000 个） 固定路径：/usr/sys00xx/var/sysCurrent.var;

=3 指定通道的寄存器固定路径：/usr/sys00xx/var/regCurrent.var;

=4 公共的 Bus 固定路径：/usr/sys0000/var/comCurrentBus.var;

=5 公共的用户变数固定路径：/usr/sys0000/var/comCurrentUser.var;

=6 公共的 MCM 固定路径：/usr/sys0000/var/comCurrentMcm.var;

=7 公共的系统变数（前 30000 个）固定路径：/usr/sys0000/var/comCurrentSys.var;

=8 公共的寄存器固定路径：

/usr/sys0000/var/comCurrentReg.var;

注意：

1. sys00xx 中的 xx 表示通道号，例如通道 1 为 sys0001，通道 12 为 sys0012。

2. 当数据类型为 4~8 时，通道号指定为 0 或不小于最大通道的任意值都可以

filename（储存的文件路径及名称）：字符串，此参数填为"" 或不填时，表示存储到固定的路径

范例

例 1：将通道 1 的用户变数储存在固定路径

System.saveVar(1,0)

例 2：将通道 1 的 MCM 变数储存在/usr/sys0001/val.var

System.saveVar(1,1," /usr/sys0001/val.var")

7.2.10. loadVar（加载指定通道的变数）

加载指定通道的变数，当指定的通道号，或数据类型正确时，该宏返回 true，否则返回 false

用法

System.loadVar(channel,datatype, fileName)

参数

参数介绍请参见第 4.1.9.1 节。

范例

例 1：加载通道 1 的用户变数

System.loadVar(1,0)

例 2：加载通道 1 中储存在/usr/sys0001/val.var 路径下的 MCM 变数

```
System.loadVar(1,1," /usr/sys0001/val.var")
```

7.2.11. initVar（恢复变数为出厂值）

将系统变数恢复为出厂值，当指定的通道号，或数据类型正确时，该宏返回 true，否则返回 false

用法

```
System.initVar(channel,dataType)
```

参数

- channel（通道号）：数值型，取值区间[1,16]
- dataType（数据类型）：具体介绍请参见第 4.1.9.1 节

范例

例 1：恢复通道 1 的用户变数为出厂值

```
System.initVar(1,0)
```

7.2.12. getProgramDir（获取指定通道的程式存储路径）

获得指定通道的程式存储路径，该宏执行成功后返回指定通道的程式存储路径，字符型。

用法

```
programDir = System.getProgramDir(channel)
```

参数

- channel（通道号）：字符型，取值区间为[1,16]

范例

例 1：获取通道 1 中的程式存储路径

```
var programDir
```

```
programDir = System.getProgramDir(1)
```

执行后通道 1 程式存储路径/usr/sys0001/program 被加载到变量 programDir 中

7.2.13. getExePrgFileMain（获取指定通道当前正在执行主程式文件名）

获得指定通道当前正在执行主程式文件名（包含路径），该宏返回值即为上述文件名，字符型。

用法

```
System.getExePrgFileMain(channel)
```

参数

➤ channel（通道号）：数值型，取值区间为[1,16]

范例

例 1：获取 1 通道当前正在执行主程式文件名（包含路径）

```
var cncName = System.getExePrgFileMain(1)
```

7.2.14. getExePrgFileSub（获取指定通道当前正在执行副程式文件名）

该宏用于获得指定通道当前正在执行副程式文件名（包含路径），返回值即为上述文件名。

用法

```
System.getExePrgFileSub(channel)
```

参数

➤ channel（通道号）：数值型，取值区间为[1,16]

范例

例 1：获取 1 通道当前正在执行副程式文件名（包含路径）

```
var cncName = System.getExePrgFileSub(1)
```

7.2.15. getMDItext（获得 MDI 内容）

用于获取 MDI 内容，返回值为指定的 MDI 内容，字符型。

用法

System.getMDIText(channel)

参数说明

- channel (通道号): 数值型, 取值区间为[1, 16]

范例

例 1: 获取第 1 通道 MDI 内容, 并将其显示在画面 “Form” 的代码显示器中

```
Form.codeDisplay.setText(System.getMDIText(1))
```

7.2.16. exeMDI (执行 MDI)

用于执行 MDI, 无返回值

用法

System.exeMDI(channel)

参数

- channel (通道号): 数值型, 取值区间为[1, 16]

范例

例 1: 通道 1 执行 MDI

```
System.exeMDI(1)
```

7.2.17. setMDIText (设置 MDI 内容)

用于设定 MDI 内容, 无返回值

用法

System.setMDIText(channel, string)

参数

- channel (通道号): 整数型, 取值区间为[1, 16]
- string (文本内容): 字符型

范例

例 1: 将通道 1 中的文本设定为 “hello, world”

```
System.setMDIText(1, "hello, world")
```

7.2.18. sleep（延时函数）

延时函数，延时指定时间后再执行下面宏程序，无返回值

用法

System.sleep(time)

参数

➤ time（延时时间）：数值型，单位 ms

范例

例 1：保存数据 A 后 1 秒再保存数据 B

```
System.saveData("A", 100)
```

```
System.sleep(1000)
```

```
System.saveData("B", 50)
```

7.2.19. execute（运行外部程序）

用于运行外部程序，返回值为异常值或程序退出码，为-2 时代表程序不能运行，为-1 时代表程序崩溃，为其他值时代表程序退出码。

用法

System.execute(program)

参数

➤ program（程序路径）：字符型

范例

例 1：运行路径/home 下的 example 程序

```
System.execute("/home/example")
```

7.2.20. resetCNC（重置控制器指定通道）

重置控制器指定通道。此函数类同于 C1，切勿使用定时器频繁调用，无返回值

用法

System.resetCNC(channel)

参数

➤ channel (通道号): 十进制整数, 按参数的位指定控制器通道号, BIT00=1 通道一、BIT01=1 通道二、……BIT07=1 通道八

范例

例 1: 重置控制器通道一、通道二、通道三, 参数需要设置 BIT0、BIT1、BIT2 为 1, 即参数为二进制数 0111, 转为 10 进制为 7

```
System. resetCNC(7)
```

例 2: 重置控制器通道二, 需要设置参数 BIT1=1, 其余位为 1, 即参数为二进制数 0010, 转为十进制数 2

```
System. resetCNC(2)
```

7.2.21. getDatas (获得连续多个变数值)

获得连续多个变数值, 该宏返回一个存放连续多个变数值的数组, 用户需要定义一个数组用于接收该宏的返回值。

用法

```
System. getDatas(channel, dataClass, address, count)
```

参数

- channel (通道号): 整数型, 取值区间为[1, 16]
- dataClass (首变数地址的数据类型): 数值型, 具体意义如下:

=0 User;

=1 Mcm;

=2 Sys;

=3 Reg;

=4 Bus;

=5 Com;

- address (首变数地址的数据地址): 数值型

- count (连续变数的个数): 数值型

范例

例 1：获取#REG100~#REG104 的值

//定义数组，数组大小为 5

```
var array=new Array(5)
```

//初始化数组中的值

```
for(var i=0;i<5;i++)
```

```
array[i]=0
```

//读取连续多个变数，并赋值到数组当中

```
array=System.getDatas(3,0,100,5)
```

//使用 array 数组

```
for(var j=0;j<5;j++)
```

```
total +=array[j]
```

7.2.22. setDatas（设置连续 多个变数值）

用于设置连续多个变数值。注：数组大小与指定的个数不一致时，只取小的生效，无返回值

用法

```
System.setDatas(channel,dataClass,address,count,arrays)
```

参数

- 参数 channel, dataClass, address, count 说明请参见第 4.1.21.1 节
- arrays: 设置变数值的数组

范例

例 1：将#REG100~#REG104 值设定为数组 array 中的值

//定义数组，数组大小为 5

```
var array=new Array(5)
```

//初始化数组中的值

```
for(var i=0;i<5;i++)
```

```
array[i]=123.456
```

// 设置连续多个变数，变数值分别为数组中的数值。

```
System.setDatas(3,0,100,5,array)
```

7.2.23. getBits（获得连续多个 bit 位的值）

用于获得连续多个 bit 位的值，该宏返回连续多个 bit 位的值，bit 位值为布尔型 (true/false)

用法

System.getBits(channel, bitType, address, bitIndex, count)

参数

➤ channel（通道号）：数值型，取值区间为[1,15]

➤ bitType（首位的类型）：数值型，具体意义如下：

=0 I_BIT;

=1 O_BIT;

=2 C_BIT;

=3 S_BIT;

=4 A_BIT;

=5 Cn_BIT;

=6 Tm_BIT;

=7 User_BIT;

=8 Mcm_BIT;

=9 Sys_BIT;

=10 Reg_BIT;

=11 Bus_BIT;

=12 Com_BIT;

➤ address（首位地址）：数值型

➤ bitIndex（起始位编号）：如果数据位类型是 User_Bit、Mcm_Bit、Sys_Bit、Reg_Bit、Com_BIT，需设置首 bit 表示数据的某一位，范围[0,31]，数值型

➤ count（连续位的个数）：数值型

范例

例 1：获取#USR100 中的 BIT0~BIT4 的值

//定义数组，数组大小为 5

```
var array=new Array(5)

//初始化数组中的值

for(var i=0;i<5;i++)

array[i]=0

//读取连续多个 bit 位的值，并赋值到数组当中

array=System.getBits(1, 7, 100,0,5)

//使用 array 数组

if(array[0]==1)

total +=1
```

7.2.24. setBits（设置连续多个 bit 位的值）

用于设置连续多个 bit 位的值，注：数组大小与指定的个数不一致时，只取小的生效，无返回值。

用法

System.setBits(channel, bitType, address, bitIndex, count, arrays)

参数

- 参数 channel, bitType, address, bitIndex, count 说明请参见第 4.1.23.1 节。
- arrays（数组）：设置 bit 位状态，bit 位状态只为 0 或 1

范例

例 1：设定#USR100 中的 BIT0~BIT4 的值为数组 array 中的值

```
//定义数组，数组大小为 5

var array=new Array(5)

//初始化数组中的值

for(var i=0;i<5;i++)

array[i]=1

//设置连续多个 bit 位的值，bit 位的状态分别为数组中的状态。

array=System.setBits(1, 7, 100, 0, 5, array)
```

7.2.25. getUsbDevicePath（获得 U 盘目录）

用于获取 U 盘目录，返回字符串类型的值

用法

System.getUsbDevicePath()

参数：无

范例

例 1：获取 U 盘的档案目录

```
var dir = System.getUsbDevicePath()
```

7.2.26. getPrePrgFileSub（获取预解子程序文件名）

用于获取预解子程序文件名

用法

System.getPrePrgFileSub()

参数：channel（通道）

范例

例 1：预解通道 1 子程序的文件名

```
var dir = System.getPrePrgFileSub(1)
```

7.2.27. getPrePrgFileMain（获取预解主程序文件名）

用于获取预解主程序文件名

用法

System.getPrePrgFileMain()

参数：channel（通道）

范例

例 1：预解通道 1 主程序的文件名

```
var dir = System.getPrePrgFileMain(1)
```

7.2.28. setCurScreenSize（设置当前屏幕大小）

用于设置当前屏幕大小

用法

System.setCurScreenSize()

参数:

(int)width: 宽度

(int)height: 高度

范例

例 1: 设置当前屏幕大小为 800*600;

System.setCurScreenSize(800, 600)

7.2.29. hPrint（在终端打印信息）

用于在终端打印信息

用法

System.hPrint()

参数: (string)string: 需要打印的字符串。

范例

例 1: 预解主程序的文件名。

System.hPrint("someText")

7.2.30. keyToString（将按键值转化为按键文本）

用于将按键值转换为按键文本

用法

System.keyToString()

参数:

(int)value: 按键值。

按键值参考" <https://doc.qt.io/qt-5/qt.html#Key-enum>"

范例

```
例 1: var keyStr = System.keyToString(0x42)
print(keyStr)//字母 B
```

7.2.31. stringToKey（将按键文本转化为按键值）

用于将按键文本转化为按键值

用法

```
System.stringToKey()
```

参数:

(string)value: 按键文本

范例

```
例 1: var keyStr = System.stringToKey("B")
print(keyStr)//0x42
```

7.2.32. convertCompressCnc（将文件转化为 CNC 格式）

用于将文件转化为 CNC 格式

用法

```
System.convertCompressCnc()
```

参数:

(string)cncFileName: 文件名, 包含全路径

范例

```
例 1: var result = System.convertCompressCnc("/usr/sys0001/program/program.CNC")
print(result)//结果
```

7.2.33. simulateHandle（模拟触摸屏操作）

用于模拟触摸屏操作

用法

System.simulateHandle()

参数:

(int)x:坐标 x。

(int)y:坐标 y。

(double)doubleClick:是否双击

=true 双击

=false 单击

范例

例 1: System.simulateHandle(400,300,true)

7.2.34. setInputMethod(设置切换中英文输入,迷你软键盘)

用于设置切换中英文输入,迷你软键盘

用法

System.setInputMethod()

参数:

(int)method:中英文

=0 切换至英文输入

=1 切换至简体中文输入

范例

例 1: System.setInputMethod(1)

7.3. HMI 函数群

7.4. Math 函数群

7.5. String 函数群

7.6. Number 函数群

7.7. HInputDialog 函数群

7.8. HMessageBox 函数群

7.9. Hfile 函数群

7.10. HFileEdit 函数群

7.11. HFileInfo 函数群

7.12. HFileVar 函数群

7.13. HAuthorityManagement 函数群

7.14. HErrorDialog 函数群

7.15. HProperty 函数群

7.16. HSharedFile 函数群

7.17. HCamera 函数群

7.18. HVisionModule 函数群

8. 其他

广州亿达科技有限公司

咨询热线: 020-39389901 维修专线: 18127931302

传真号码: 020-39389903 邮政编码: 511495

电子邮箱: finger@finger-cnc.com

公司官网: www.finger-cnc.com

公司地址: 广东省广州市番禺区钟村街诚鼎街 8 号 201



亿达官网



亿达微信公众号