



iX Developer

用户指南
简体中文

iX Developer 用户指南

序言

iX Developer 软件用于配置 iX Panel 和在 PC 环境下运行的控制应用程序，包括用于 IPC（工业 PC）的应用程序。

通过 iX Developer，您可以轻松创建出合理、灵活且高效的操作面板应用程序，从而为操作员和其他系统适时提供所需信息。

本手册将通过一个示例项目向您介绍 iX Developer 项目的分步设计过程。

有关详细信息，请参阅 iX Developer 帮助文件。

订货号: MACN832G

版权所有 © 2015-09 Beijer Electronics AB. 保留所有权利

本文档中的信息可随时更改，恕不另行通知，并按付印时所能提供的内容呈献给您。Beijer Electronics AB 保留随时更改任何信息的权利，但不会同时更新此出版物。Beijer Electronics AB 对于本文档中出现的任何错误均不承担任何责任。

本文档中的所有示例仅用于加深对功能的了解以及提高操作软件的能力。如果将这些示例用于实际应用，Beijer Electronics AB 对由此产生的任何后果不承担任何责任。

鉴于本软件的适用范围广泛，为确保在特定应用中能够正确使用本软件，用户自身必须掌握充足的知识。应用程序和设备的负责人必须自行确保每项应用在配置和安全性方面均符合所有相关要求、标准和法规。对于在本软件的安装或使用过程中产生的任何损坏，Beijer Electronics AB 不承担任何责任。Beijer Electronics AB 禁止任何修改、变动或转换本软件的行为。

目录

1	简介	6
1.1	控制器	6
1.1.1	变量	6
1.2	手册结构	6
2	安装和启动	7
2.1	系统要求	7
2.1.1	iX Developer	7
2.1.2	iX Runtime	7
2.1.3	某些元件的特殊要求	9
2.1.4	启动 iX Developer	9
2.1.5	帮助	9
3	新建项目	10
3.1	创建新项目	10
3.2	界面	11
3.2.1	项目浏览器	11
3.2.2	功能区组和控件	12
3.2.3	快速访问工具栏	12
3.2.4	迷你工具栏和上下文菜单	12
4	控制器变量	13
4.1	添加变量	13
4.2	保存项目	14
5	编辑元件	15
5.1	添加元件	15
5.1.1	仪表	15
5.1.2	滑动器	15
5.1.3	对齐	16
5.1.4	调整大小	16
5.1.5	更改外观	16
5.2	运行测试项目	16
6	画面浏览和跳转	17
6.1	画面浏览	17
6.1.1	画面管理员	17
6.1.2	添加画面	17
6.1.3	画面跳转	18
6.2	背景和前景画面	18
6.2.1	添加画面	18
6.2.2	背景和前景画面	18
6.3	画面浏览运行测试	20
7	趋势图查看器	21
7.1	添加实时趋势图查看器	21
7.1.1	曲线	21
7.2	运行实时趋势图查看器测试	22
7.3	趋势图查看器历史记录	22
7.3.1	操作	22
7.4	运行历史趋势图查看器测试	23
8	警报管理	24
8.1	警报指示器	24
8.2	警报服务器	25

8.2.1	警报组	25
8.2.2	警报项目	25
8.3	警报显示器	27
8.4	运行测试警报	27
9	配方	28
9.1	创建配方变量	28
9.1.1	添加画面	28
9.1.2	调整浏览按钮	29
9.2	新建元件	30
9.2.1	显示信息	30
9.3	配方项	30
9.4	保存配方	31
9.5	载入配方	31
9.6	配方数据	31
9.7	运行测试配方	32
10	动态	33
10.1	创建元件	33
10.2	调整元件大小	34
10.3	为元件设定颜色	35
10.4	运行测试动态	35
11	脚本	36
11.1	添加元件	36
11.2	脚本标签页	36
11.3	运行脚本	37
12	内部变量	38
12.1	添加内部变量	38
12.1.1	面积变量	38
12.1.2	计算变量	38
12.2	创建模拟数值	39
12.2.1	变量操作	39
12.3	内部变量运行测试	40
13	元件浏览器	41
13.1	添加图形元素	41
13.2	使用元件浏览器	41
14	文本库	43
14.1	配置文本	43
14.2	文本库运行测试	44
15	用户权限	45
15.1	用户权限设定	45
15.1.1	用户权限组	45
15.1.2	用户	45
15.2	访问被拒绝时的登录行为	46
15.3	创建注销按钮	46
15.4	元件用户权限	47
15.4.1	管理员	47
15.4.2	用户	47
15.5	运行测试用户权限	47
16	功能键	48
16.1	定义功能键操作	48
16.1.1	显示画面	48

16.1.2	用户权限	49
16.1.3	设置控制器变量值	49
16.1.4	配方	49
16.1.5	设定时区、地区和夏令时	49
16.2	定义功能键脚本	51
16.2.1	面积计算	51
16.2.2	确认所有警报	51
16.3	功能键运行测试	52
17	对象使用浏览	53
17.1	使用元件使用浏览工具	53

1 简介

iX Developer 软件用于配置 iX Panels 和在 PC 环境下运行的控制应用程序，包括用于 Beijer Electronics 的 IPC（工业 PC）的应用程序。

iX Developer 包含应用程序中所需的所有基本功能。这些功能经过严格测试并以用户需求和用户偏好为导向开发而成。

iX Developer 中的预定义元件可用于创建完整的过程图像，从而得以在总体上概括复杂的应用程序。您可以对预定义的元件进行自定义，也可以自行创建元件。

本产品还可为众多控制器和自动化设备提供通信驱动。

1.1 控制器

操作面板可以与多种类型的自动化设备连接，例如 PLC、伺服机构和驱动器。需要注意的是，本手册中的 *控制器* 一词用作对被连接设备的统称。

1.1.1 变量

控制器中的数据值被称为 *变量*。变量也可以是系统变量或内部变量。变量具有符号名称，且可以是不同的数据类型。

与变量连接的对象可以更改控制器中的值，且变量值可通过多种方式借助更改对象的外观来加以反映。屏幕中的对象在连接到变量之前始终为静态对象。

1.2 手册结构

本 USERS GUIDE 以一个示例项目为依托，使您在使用 iX Developer 时轻松上手。如果认真遵循示例中的各项说明，便会得到一个可进一步开发或用于激发灵感的实用项目。示例针对的是 PC，但对于所有受支持的操作面板，所有功能的工作原理非常相似。

帮助文件中提供了有关 iX Developer 的详细信息，使用本软件时，按 **F1** 即可显示此帮助信息。

有关本 USERS GUIDE 中所述各项说明的详细信息，请参见开头部分。随着示例讲解的不断深入，您会逐渐熟悉 iX Developer 的使用，因此某些重复性的任务说明将会删减或省略。

2 安装和启动

iX Developer 研发 PC 可安装在用于项目开发、设计和编译的开发 PC 上。然后便可在操作面板中运行项目，以观察和操控某个控制器（或者一组控制器）。

2.1 系统要求

2.1.1 iX Developer

参数	推荐配置	
RAM	2 GB	
处理器	2 GHz 或更高	
操作系统	Microsoft Windows 10	
	Microsoft Windows 8.1	
	Microsoft Windows 7 SP1	
显卡	呈现层 2:	
	DirectX 版本	必须高于或等于 9.0。
	视频 RAM	必须大于或等于 120MB。
	像素着色器	版本级别必须高于或等于 2.0。
	顶点着色器	版本级别必须高于或等于 2.0。
	多纹理单元	单元数必须大于或等于 4。

更新

软件、驱动程序和协议可以通过在原有版本之后进行更新。因此，建议您在创建项目前使用 iX Developer 中内置的更新功能进行更新。因此，建议您在创建项目前使用 iX Developer 中内置的更新功能进行更新。

2.1.2 iX Runtime

参数	推荐配置	
RAM	1 GB	
处理器	1.3 GHz 或更高	
操作系统	Microsoft Windows 10	
	Microsoft Windows 8.1	
	Microsoft Windows 7 SP1	

参数	推荐配置	
显卡	呈现层2:	
	DirectX 版本	必须高于或等于 9.0。
	视频 RAM	必须大于或等于 120MB。
	像素着色器	版本级别必须高于或等于 2.0。
	顶点着色器	版本级别必须高于或等于 2.0。
	多纹理单元	单元数必须大于或等于 4。

2.1.3 某些元件的特殊要求

对于 iX Developer 项目中要用到的某些元件，需要使用特定的软件版本。在开发研发 PC 上进行项目模拟也可能因这些元件而受到限制。

元件	最低要求	在PC上运行	在操作面板上运行
媒体播放器	Windows Media Player 10	支持	不支持
PDF 显示器	Acrobat Reader 9	支持	不支持
网络浏览器	Internet Explorer 7	支持	不支持

2.1.4 启动 iX Developer

安装后会在 Windows “开始” 菜单中显示 iX Developer 的配置图标。点击**开始/所有程序/iX Developer2.20/iX Developer2.20**以启动配置工具。

2.1.5 帮助

在 iX Developer 运行的情况下，按 **F1** 可获得帮助主题。

3 新建项目

目标：

- 创建新项目。
- 熟悉各工具窗口和桌面区域布局。

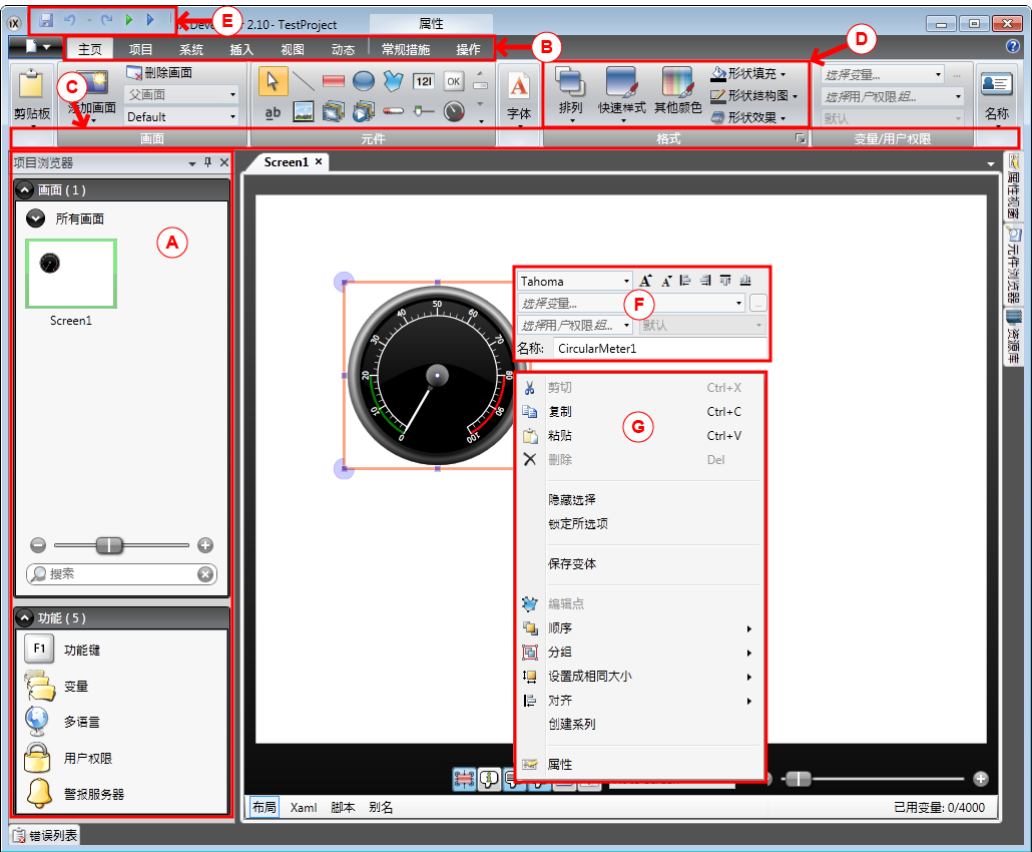
3.1 创建新项目

1. 启动 iX Developer，然后选择**创建新项目**。
这时会弹出一个引导您完成整个项目创建过程的向导。
 2. 选择分辨率为 **1024 × 768** 的 **PC** 作为应用程序的目标。单击**下一步**。
 3. 在控制器列表中选择**DEMO**。单击**下一步**。
“DEMO”控制器包含常规变量（数据容器）和计数器，可直接在开发研发 PC 上设计和模拟运行工程，并进行调试，而无需连接到实际的控制器。
 4. 为项目指定一个名称。对于本教程，将采用 **DEMO_TEST**。检查建议的位置是否合适。如果不合适，请单击**浏览**选择其他位置。
 5. 单击**完成**。
- 项目文件可以存放在计算机环境中您拥有写入权限的任意位置。
项目可由快捷方式直接打开。

3.2 界面

界面用于显示项目组件（例如各个控制器和功能）的画面和配置页面。一次仅显示一个画面或组件。打开多个画面或组件时，会在桌面区域的上部显示一行标签页。单击某一标签可激活其内容以便进行编辑。

如果打开的标签页过多而无法全部显示，可以使用界面上部的下拉菜单在标签页之间来回切换。



图中标号	桌面区域组件	对应章节
A	项目浏览器	项目浏览器
B	功能区标签页	功能区组和控件
C	控件组	
D	控件	
E	快速访问工具栏	快速访问工具栏
F	迷你工具栏	迷你工具栏和上下文菜单
G	上下文菜单	

3.2.1 项目浏览器

新项目打开后，会在界面中显示一个空白画面。项目浏览器位于界面左侧。

3.2.2 功能区组和控制件

功能区标签页位于工具窗口的上部。每个功能区标签页都包含一个或多个控件组。每组中均包含一系列相关控件。您可以使用这些控件来设计画面以及对项目中的元件和控制件进行设置。

如果不习惯使用带有功能区标签页的软件，请花费一些时间来熟悉功能区这个概念。

3.2.3 快速访问工具栏

快速访问工具栏位于桌面区域的顶部且始终可见。随着 iX Developer 的启动，工具栏中将显示**保存**、**撤消**、**恢复**、**运行**和**模拟**命令。

3.2.4 迷你工具栏和上下文菜单

右键单击 iX Developer 中的元件时，会显示迷你工具栏和上下文菜单。迷你工具栏中包含特定于 iX Developer 的命令，例如将元件连接到控制器变量的命令。上下文菜单中包含常规的 Microsoft 应用程序命令，例如**复制**、**粘贴**等。

4 控制器变量

目标：

- 为项目定义变量列表。
- 保存项目。

4.1 添加变量

与变量连接的对象可以更改控制器中的值，且变量值可通过多种方式借助更改对象的外观来加以反映。屏幕中的对象在连接到变量之前始终为静态对象。

1. 单击**项目浏览器**中的**变量**。
这样便会在桌面上打开变量配置页面。默认情况下，变量列表中将包含一个元素。
2. 单击第一行中的第一个字段（**名称**）。
光标位于文本**变量1**边上。
3. 按键盘上的 **[TAB]** 键。
这时将转而选中下一个字段（**变量下的数据类型**）。您不需要更改数据类型。

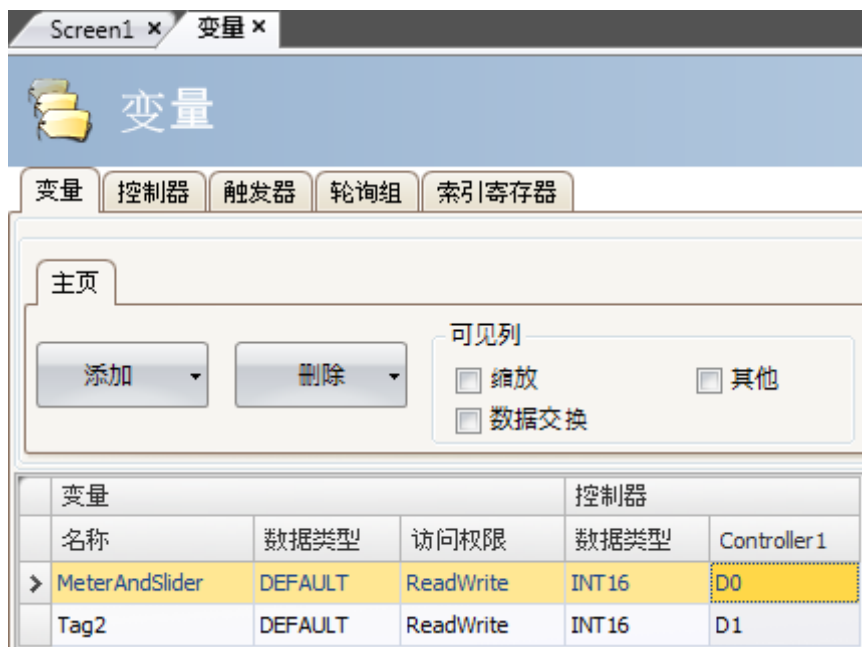
注意：

在某些情况下，变量的数据类型可用作显示格式，例如在使用比例缩放时显示正确的工程单位。默认根据专为**控制器**下的**数据类型**所做的选择。

4. 再次按键盘上的 **[TAB]** 键。
这时会选中下一个字段（**访问权限**）。您此时不需要更改访问权限。
5. 再次按键盘上的 **[TAB]** 键。
这时将转而选中下一个字段（**控制器下的控制器数据类型**）。您此时不需要更改数据类型。
6. 再次按键盘上的 **[TAB]** 键。
这时会选中下一个字段（**控制器1**）。
7. 在**控制器1**字段中输入 **D0**。
控制器列中的条目与所选控制器中的变量保持一致。在“**DEMO**”控制器中存在可通过对应变量的地址访问的预定义变量，例如 **D0** 表示一个整型变量字段。
8. 继续按 **[TAB]** 键将前两行填充完整。为第二个控制器变量输入 **D1**。
某些字段会自动填充，必要时所填的值还会自动递增。数据类型会根据**控制器**列中输入的内容自动更改。

9. 将Tag1 重命名为MeterAndSlider。

注意：
变量的**名称**是变量的识别符，可为以字母（a-z、A-Z）开头的任意字母数字字符串。



在下一节中将使用 D0 变量控制和观察画面中的控制器变量值。

4.2 保存项目

- 单击快速访问工具栏中的保存符号。
项目将保存在创建项目时选择的保存位置。

5 编辑元件

目标：

- 插入滑动器和仪表。
- 了解如何配置元件和对齐元件。
- 测试项目；通过画面中的元件控制和观察控制器变量值。

5.1 添加元件

5.1.1 仪表

1. 单击桌面区域中的 **Screen1** 标签页，并确保选中功能区中的 **主** 功能区标签页。
2. 从 **元件组** 中选择一个仪表。将其置于画面左上方某处。
3. 拖动角控点将此仪表调整到合适大小，使仪表指针和刻度清晰可见。
4. 右击仪表并展开 **选择变量...** 下拉列表。
5. 选择 **MeterAndSlider**，然后点击 **OK**。

5.1.2 滑动器

1. 从 **元件组** 中选择一个滑动器。有时可能需要单击右下角的小箭头来展开 **元件组** 才能选择滑动器。
2. 将滑动器置于画面中的仪表下方即可。
3. 右击滑动器并展开 **选择变量...** 下拉列表。
4. 选择 **MeterAndSlider**，然后点击 **OK**。



5.1.3 对齐

元件经过拖动可与其他元件对齐。

1. 上下缓慢拖动滑动器。
请注意，滑动器会在仪表下方不远处对齐到相应位置。
2. 左右缓慢拖动滑动器。
请注意，滑动器可对齐到相应位置并在滑动器与仪表对齐时显示对齐线。
3. 将滑动器置于仪表下方不远处，并保证滑动器的左边沿与仪表的左边沿对齐。

5.1.4 调整大小

1. 同时选择这两个元件（指向画面中的空白区域，然后斜向拖动指针框选这两个元件）。
多选（组合）时会产生一个主元件。主元件带有橙色框，其他元件则带有蓝色框。对选中的组合执行格式化命令时，主元件将作为模板。
如果仪表不是主元件，则可：
2. 单击该仪表，将组合中的主元件更改为仪表。
接下来请调整组合中元件的宽度：
3. 单击位于主页功能区标签页的**格式组**中的排列控制，并选择**相同宽度**。

5.1.5 更改外观

1. 选择**Screen1**上的滑动器。
2. 单击**格式组**中的**快速样式**控件，然后选择一种新的颜色样式。
3. 单击**格式组**右下角的小箭头可进行边框、阴影/填充效果等其他设置。
4. 选择**Screen1**上的仪表。
5. 选择**常规**功能区标签页，然后找到**样式组**。尝试各种不同的预定义样式，然后选出最符合您个人喜好的样式。

5.2 运行测试项目

项目的编译和运行几乎可以随时进行。这样您便能够不断地反复测试设计结果。

1. 单击**快速访问工具栏**中的**运行**图标。
接下来项目便开始验证。如果未发现任何错误，将在开发环境中编译和执行项目。
2. 前后拖动滑动器控制块。
由于两个元件均连接到同一变量，因此仪表指针会随着滑动器控件值的改变而转动。
3. 关闭**运行**窗口。

6 画面浏览和跳转

一个 iX Developer 项目由带有元件（通常连接到控制器变量）的各个画面组成。所有画面的基本功能相同。通过为画面赋予特定属性可以限定画面在项目中的行为：

- **启动画面：**启动画面是运行时显示的第一个画面。在默认情况下，**Screen1** 将作为启动画面，但也可以通过右键单击某画面然后选择设置为启动画面，将任意画面**指定为启动画面**。
- **背景和前景画面：**项目中的其他画面可以使用带有别名的画面之外的任何画面作为背景或前景画面。有关别名的详细信息，请参阅 iX Developer 帮助文件。
- **画面模板：**另存为画面模板的画面不仅可以用在当前的 iX Developer 项目中，也可以用在今后所有的项目中。

画面跳转可通过操作按钮等元件来实现。使用**画面管理员**添加画面并创建画面之间的链接时，按钮将自动添加到链接所在画面的左上角。

目标：

- 添加新画面并通过按钮设置画面跳转。

6.1 画面浏览

6.1.1 画面管理员

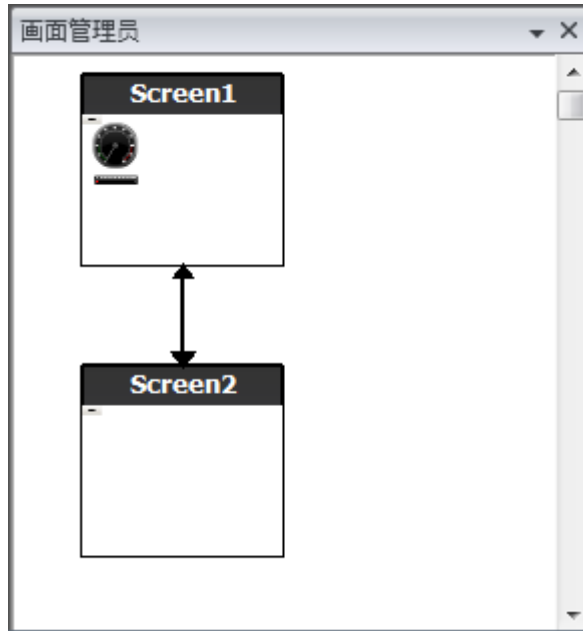
- 单击功能区中的**视图**功能区标签页。单击**画面管理员**。这时会在桌面中打开**画面管理员**。

6.1.2 添加画面

- 指向**Screen1**。单击**Screen1**，然后拖动出一个从**Screen1**到**画面管理员**区域中任意位置的连接。此时会出现一个新画面（**Screen2**）。标有**Screen2**的按钮会出现在**Screen1**的左上角。

6.1.3 画面跳转

- 在画面管理员中单击 Screen2，然后拖动出一个从 Screen2 到 Screen1 的连接。
标有 Screen1 的按钮会出现在 Screen2 的左上角。



6.2 背景和前景画面

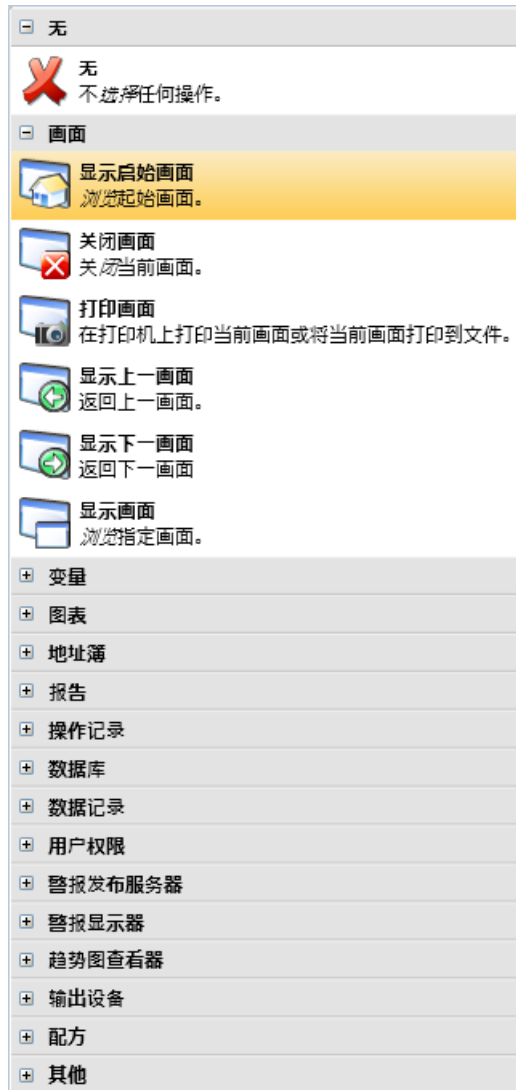
6.2.1 添加画面

- 单击插入功能区标签页中的画面。
项目中会创建一个新画面 (Screen3)，并且会将它打开以便进行编辑。

6.2.2 背景和前景画面

- 确保 Screen3 在桌面上呈打开状态以便进行编辑。
- 从元件组 (位于主页功能区标签页中) 中选择一个按钮，然后将其置于 Screen3 的左下角区域。将此按钮标注为启动画面。

3. 使该按钮保持选中状态，然后单击**操作**功能区标签页。从**单击操作**下拉列表的**画面**组中选择**显示启动画面**。



4. 在**项目浏览器**中单击打开**Screen2**进行编辑。
5. 选择**主页**功能区标签页。
6. 打开**父画面...**下拉列表。
7. 点击**Screen3**。
8. 选定**背景**。
9. 单击**确定**。
10. 尝试更改启动画面按钮在**Screen2**中的位置。无法执行此操作。
 ． 请注意，**Screen2**中反映的是**Screen3**所做的更改。
 现在，要从**Screen2**浏览到**Screen1**（启动画面）便产生了两种途径。
 设置前景画面可通过同样的步骤完成，除非您在步骤8中选择了**前景**。

6.3 画面浏览运行测试

1. 运行项目。
2. 确认 **Screen2** 中的每个按钮是否均可执行到 **Screen1** 的跳转操作。
由于未将任何画面设置为启动画面，因此 **Screen1** 仍将作为该项目的启动画面。
3. 关闭运行窗口。

7 趋势图查看器

趋势图查看器功能用于存储操作面板中控制器的寄存器信息。本产品可提供实时趋势图查看器和历史趋势图查看器。

目标：

- 添加带有两条曲线的趋势图查看器元件。

7.1 添加实时趋势图查看器

1. 在桌面中打开 **Screen1** 进行编辑。
2. 从元件组中选择 **趋势图查看器** 元件，然后将其置于画面中。

7.1.1 曲线

1. 单击 **项目浏览器** 中的 **变量**。
2. 单击 **添加** 以添加新的变量。
3. 在 **名称** 字段中输入 **Counter** 并将其连接至 **控制器 1** 下的 **C0**。
C0 是一种频率为 1 Hz、计数从 0 到 100 然后再返回到 0 的计数器。
4. 打开 **Screen1**，确保趋势图查看器元件处于选中状态，然后单击 **常规** 功能区标签页中的 **编辑曲线**。
5. 单击 **添加** 以添加新的曲线。
6. 选择变量字段中的 **MeterAndSlider**，并将曲线连接至您曾在 **Screen1** 中用于滑动器的相同变量。
7. 单击 **添加** 以添加第二条新曲线。
8. 选择 **变量** 字段中的 **Counter**，并为此曲线选择另一种颜色。



9. 单击**确定**。

7.2 运行实时趋势图查看器测试

- 运行项目，然后检查两条曲线在趋势图查看器中是否可见。测试 **Curve 1** 是否随滑动器的变化而变化。

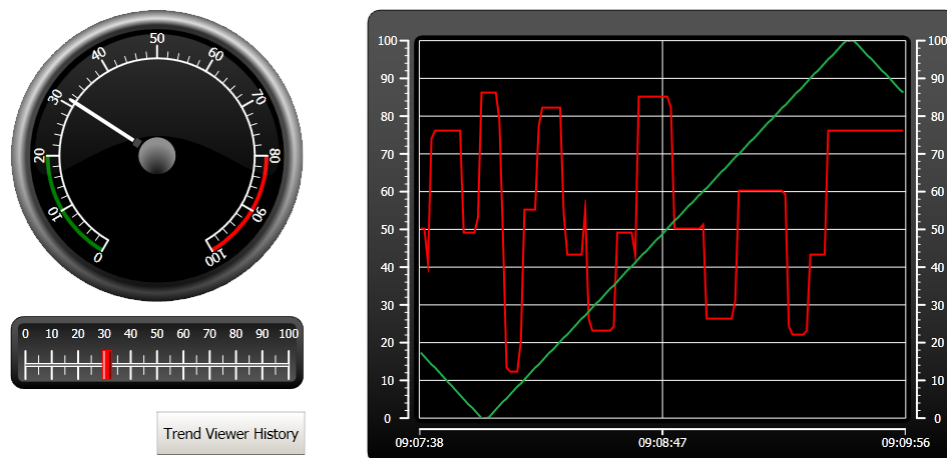
7.3 趋势图查看器历史记录

7.3.1 操作

1. 在桌面中打开 **Screen1** 进行编辑。
2. 单击**主页**功能区标签页。
3. 在趋势图查看器元件的左侧放置一个按钮。将此按钮标注为**趋势图查看器历史记录**。
4. 使该按钮保持选中状态，然后单击**操作**功能区标签页。
5. 单击**单击**组右下角的小箭头。
6. 选择**触发器**下拉列表中的**单击**。
7. 单击**选择操作...**下拉列表，展开**趋势图查看器**并选择**趋势图查看器历史记录**。
8. 单击**选择趋势图查看器...**下拉列表并选择**趋势图查看器**。
9. 选择最后下拉列表中的**打开**。
- 10 单击**确定**。
- .
- 11 选择该趋势图查看器元件，然后单击**操作**功能区标签页。
- .
- 12 单击**鼠标按下**组右下角的小箭头。
- .
- 13 选择**触发器**下拉列表中的**鼠标按下**。
- .
- 14 单击**选择操作...**下拉列表，展开**趋势图查看器**并选择**趋势图查看器历史记录**。
- .
- 15 单击**选择趋势图查看器...**下拉列表并选择**趋势图查看器**。
- .
- 16 选择最后下拉列表中的**关闭**。
- .
- 17 单击**确定**。
- .

7.4 运行历史趋势图查看器测试

1. 运行项目。



2. 测试是否可以通过**趋势图查看器历史记录**按钮切换到历史趋势图查看器。
3. 通过单击趋势图查看器元件返回到实时趋势图查看器。

8 报警管理

报警用于通知操作员需要立即进行处理的事件。满足某种特定条件后即可触发报警。报警条件被设计为一种对变量值的逻辑评估。可通过对报警分组来创建优先顺序。

目标：

- 配置报警列表并设计报警元件。

8.1 报警指示器

报警触发时，无论当前处于哪一画面，均会显示报警指示器通知操作员。

报警指示器的外观取决于当前的报警状态。

选择报警设定属性页面中的**常规**设置可决定在哪些报警状态下显示报警指示器。

报警指示器将根据下列指示显示报警列表中最严重的报警：

- 存在任何已触发的未确认报警时闪烁红色。
- 不存在已触发的报警但存在未触发的未确认报警时闪烁绿色。
- 仅存在已触发的已确认报警时闪烁绿色。

所有报警均已确认并恢复到未触发状态时，报警指示器会消失。

8.2 报警服务器

- 单击项目浏览器中的报警设定，打开报警设定配置页面。

8.2.1 报警组

报警组标签页用于建立多个报警的组合，适用于项目需要分别管理各独立功能的报警等情况。本示例中使用的是默认报警组。

8.2.2 报警项目

- 单击报警项目标签页。
- 单击添加。
一个名为AlarmItem0的新变量已添加至列表。
- 单击空白文本字段并输入滑动器最大值。
- 选择变量字段中的MeterAndSlider。
- 选择条件字段中的GreaterThan。
- 将触发器值字段中的值更改为99。
- 再次单击添加。
一个名为AlarmItem1的新变量已添加至列表。
除非连接到变量配置页面中的控制器，否则该变量将是内部变量，并且使用内部变量恰好适用于此示例项目。有关信息，请参阅[内部变量](#)一章。
- 单击空白文本字段并输入Boolean变量。
- 单击变量字段并输入BoolAlarmTag。
- 选择条件字段中的EqualTo。
- 将触发器值字段中的值更改为1。
- 根据下图添加两个报警项目。



- 使历史记录和需要确认两列处于选中状态。其他字段保持原样。

- 14 确保所有报警变量均可通过项目画面控制，或确保这些报警可通过其他机制触发（10 秒和 20 秒后，计数器将相继触发 **AlarmItem2** 和 **AlarmItem3**）。
- 15 在趋势图查看器元件的左侧放置一个按钮。将此按钮标注为**设定报警**。
- 16 保持此按钮为选中状态。在**操作功能区**标签页中选择**切换**，位于**变量组**，**单击**操作下拉列表中。
- 17 在**选择变量**字段中选择 **BoolAlarmTag**。

8.3 报警显示器

- 1. 在桌面中打开 **Screen2** 进行编辑。
- 2. 单击**元件组**左下角的小箭头并选择**HMI 控制下的报警查看器**。
- 3. 单击画面放置报警元件。
在报警元件中可以自定义各列以及按钮放置位置。
- 4. Select the alarm object on the screen, and click on the **General** ribbon tab.
- 5. 在**按钮组**中，选择**位置**下拉列表中的**顶部**。
- 6. 调整大小，使报警元件中的所有按钮控件均可见。
- 7. 单击**显示设置组**中的**配置列**在报警元件中自定义报警信息和列的先后顺序。

8.4 运行测试报警

- 1. 运行项目。
- 2. 进行触发报警测试。

Ack Selected

Ack All

Clear

Filter

Info

||

名称	状态	文本	活动时间	无效时间	正常时间	确认的时间
报警项1	Active	BooleanTag	2011-01-05 15:51:37			
报警项3	Inactive	计数器 20	2011-01-05 15:51:11	2011-01-05 15:51:13		
报警项2	Normal	计数器 10	2011-01-05 15:50:57	2011-01-05 15:50:58	2011-01-05 15:51:05	2011-01-05 15:51:05
报警项0	Acknowledge	滑块最大值	2011-01-05 15:50:51			2011-01-05 15:51:28
报警项1	Inactive	BooleanTag	2011-01-05 15:50:45	2011-01-05 15:50:49		

Active: 1 Inactive: 2 Ack: 1 Normal: 1 [5 / 5]

- 3. 单击**全部确认**按钮，然后观察报警指示器。
- 4. 按**清除 (Clear)**移除处于正常状态（已确认和未触发）的所有报警。

9 配方

配方用于设定或保存单个操作中一组预定义的变量。

目标：

- 创建配方并使用配方更改多个值。

9.1 创建配方变量

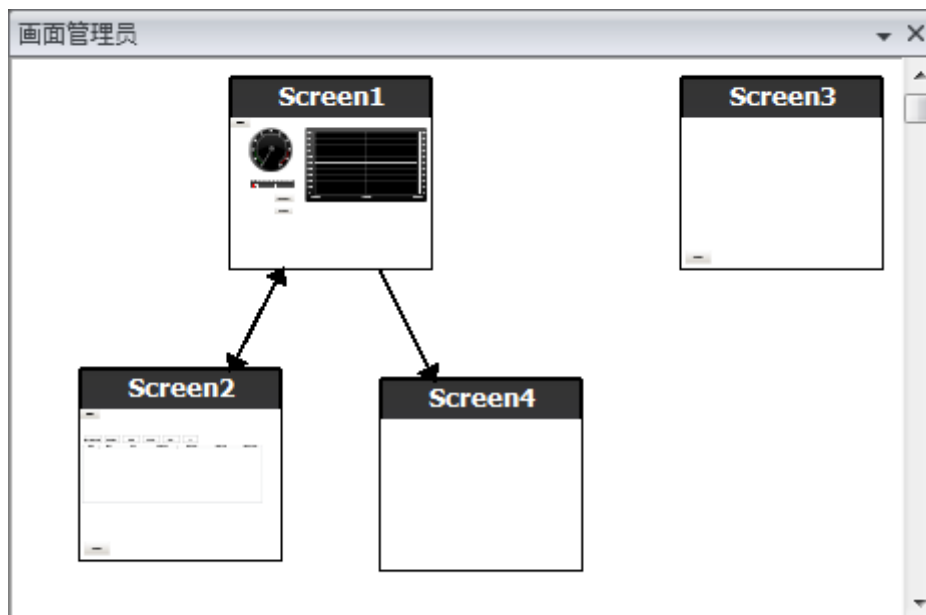
- 创建一组配方所要施加作用的控制器变量。根据下图创建三个整型值设置某假想物的重量、长度和宽度。

变量			控制器	
名称	数据类型	访问权限	数据类型	Controller 1
MeterAndSlider	DEFAULT	ReadWrite	INT 16	D0
Tag2	DEFAULT	ReadWrite	INT 16	D1
Counter	DEFAULT	ReadWrite	INT 16	C0
BoolAlarmTag	DEFAULT	ReadWrite	DEFAULT	
重量	DEFAULT	ReadWrite	INT 16	D2
长度	DEFAULT	ReadWrite	INT 16	D3
I 宽度	DEFAULT	ReadWrite	INT 16	D4

9.1.1 添加画面

1. 单击功能区中的视图功能区标签页。单击画面管理员。

2. 在画面管理员中，指向**Screen1**并拖动出一个到画面浏览区域中空白位置的连接。



项目中会创建一个新的画面 (**Screen4**)。

3. 打开**Screen4**，然后选择**主页**功能区标签页。
 4. 在**画面组**中，单击**父画面**下拉列表。
 5. 选择**Screen3**，然后再选择**背景**。
 6. 单击**确定**。
- 这样便可以从**Screen4**浏览至**Screen1**。

9.1.2 调整浏览按钮

- 打开**Screen1**。
- 选择标注为**Screen4**的按钮（位于左上角），然后将其移开，以使下面的按钮 (**Screen2**) 完整显示。

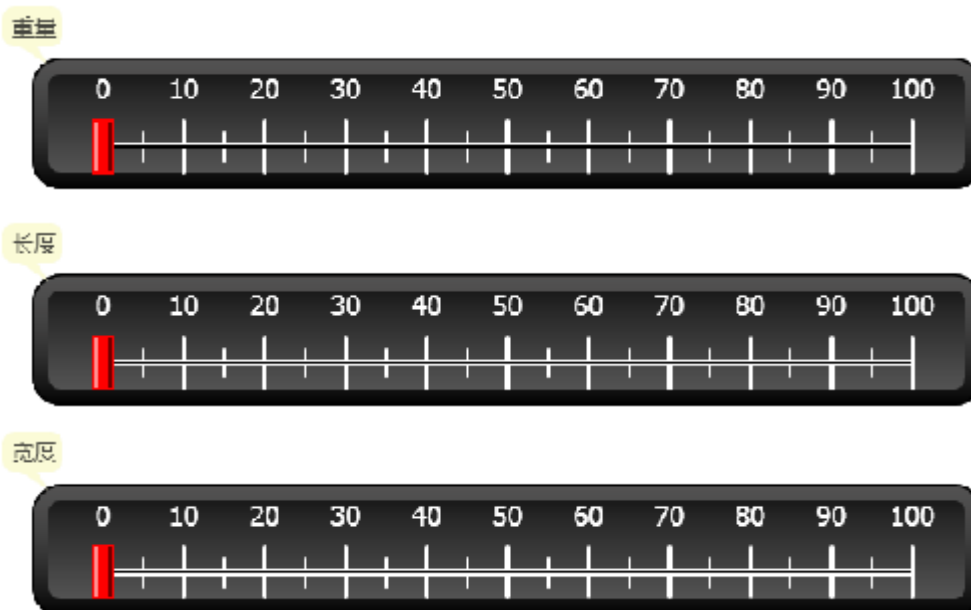
9.2 新建元件

1. 在桌面中打开 **Screen4** 进行编辑。
2. 添加滑动器，请见主页功能区标签页上的**元件组**。
3. 右击滑动器并打开**选择变量...**下拉列表，打开控制器下拉列表。
4. 选择**重量**，然后点击**OK**。
这会将**重量**变量连接至元件。
5. 按住 **Ctrl** 键，然后拖动滑动器到画面中的其他位置以便复制出一个滑动器。定位新的滑动器，然后将它连接到**长度**。
6. 重复步骤5，但将滑动器连接至**宽度**。

9.2.1 显示信息

可以显示有关每项元件所连接的变量信息，以及是否为元件配置了动态和操作。这可以通过单击桌面区域中的**显示/隐藏信息**或使用键盘快捷键**Ctrl+D**完成。

- 按键盘上的**Ctrl+D**可检查这些变量是否正确绑定到画面中对应的滑动器。



9.3 配方项

1. 单击**插入**功能区标签页中的**配方**。
这样便会在桌面上打开配方配置页面。在**项目浏览器**中也可找到新建的配方。
2. 在**变量配置**标签页中，单击**添加**三次以添加三个新配方。

- 根据下图将项目连接至变量。

名称	变量
RecipeItem1	重量
RecipeItem2	长度
> RecipeItem3	宽度

9.4 保存配方

- 在桌面中打开 **Screen4** 进行编辑。
- 将按钮放置在配方变量滑动器组边上，并将此按钮标注为**保存配方**。
- 使该按钮保持选中状态，然后单击**操作功能区**标签页
- 打开用于单击操作的**选择操作...**下拉列表。
- 选择**配方组**中的**保存配方**。
- 确保下一下拉列表中选择了**Recipe1**。
- 将**选择配方数据 (可选)...**保持为空。

9.5 载入配方

- 在桌面中打开 **Screen4** 进行编辑。
- 按住 **Ctrl** 键，然后拖动**保存配方**按钮到画面中的其他位置以便复制出一个保存配方按钮。
- 将标签更改为**加载配方**。
- 使该按钮保持选中状态，然后单击**操作功能区**标签页
- 打开用于单击操作的**选择操作...**下拉列表。
- 选择**配方组**中的**加载配方**。
- 将**选择配方数据 (可选)...**保持为空。

9.6 配方数据

在配方配置页面的**运行时数据**标签页中定义值，从而创建预定义配方。

- 在**项目浏览器**中单击**Recipe1**，打开配方配置页面。
- 单击**运行时数据**标签页。
- 单击**添加**按钮两次。
- 根据下图更改配方项的值。

运行时配方标题	RecipeItem1	RecipeItem2	RecipeItem3
Book	2	25	15
I TV	30	45	60

5. 打开 **Screen4** 进行编辑。
6. 在这组滑动器旁边放置一个新按钮。将此按钮标注为**载入 Book**。
7. 使该按钮保持选中状态，然后单击**操作功能区**标签页
8. 打开用于单击操作的**选择操作...**下拉列表。
9. 选择**配方组**中的**加载配方**。
- 10 打开**选择配方数据 (可选)...**下拉列表并选择 **Book**。
- .

9.7 运行测试配方

1. 运行项目。
2. 将滑动器设置为各种值进行测试，然后将这些值保存在配方中。
3. 进行配方载入测试。
检查滑动器是否随配方值的变化而变化。

10 动态

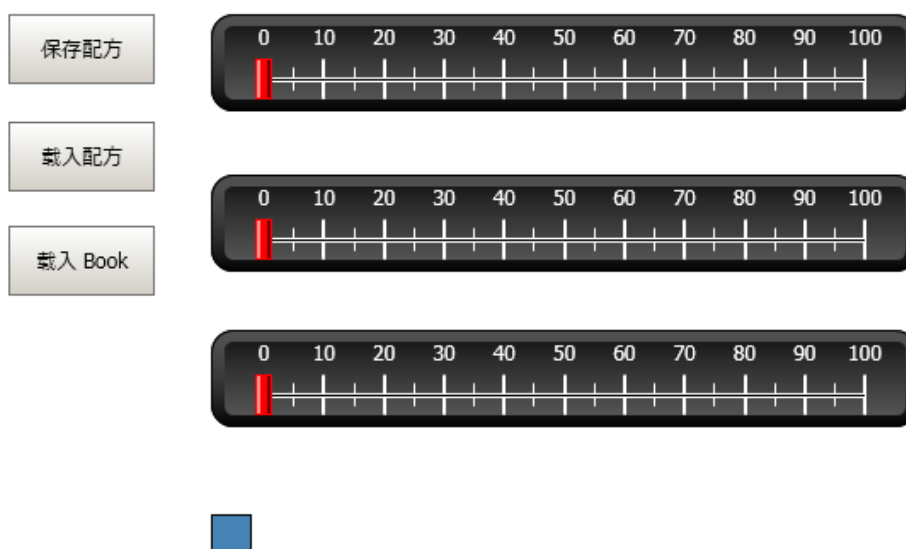
动态元件属性用于根据控制器变量值移动元件和调整元件大小。

目标：

- 根据变量值的变化更改元件的大小和颜色。

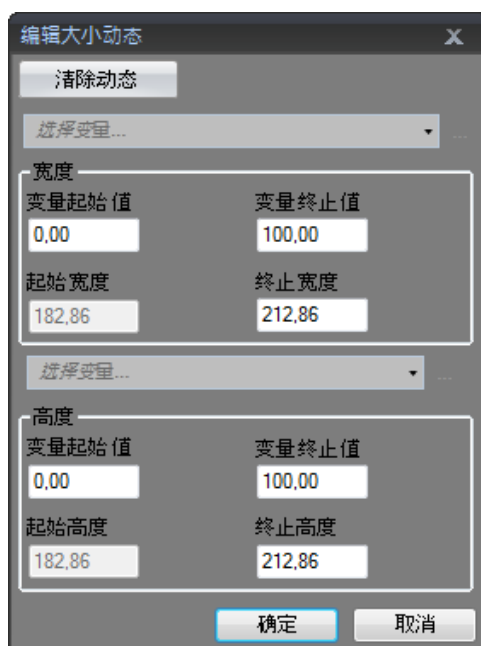
10.1 创建元件

1. 打开 **Screen4** 进行编辑。
2. 在主页功能区标签页中选择位于**元件组**的红色矩形。
3. 在这组滑动器下方画一个小正方形。



10.2 调整元件大小

1. 选中该正方形。
2. 在**动态**功能区标签页中单击位于**布局**组中的**大小**。
3. 为**宽度**选择**宽度**变量，为**高度**选择**长度**变量。
4. 单击**确定**。
5. 直接在画面中调整正方形放大后的大小，并在**编辑大小动态窗口**中记录值的变化。



10.3 为元件设定颜色

1. 选中该正方形。
2. 在**动态**功能区标签页中单击位于**颜色**组中的**填充**。
3. 在**选择变量...**下拉列表中指定**重量**变量。
4. 单击**添加**四次。
5. 调整变量值，正方形的颜色将根据**重量**变量值的不同而变化。下图中的示例使用的是结合了渐变色的填充颜色。



6. 单击**确定**。

10.4 运行测试动态

1. 运行项目。
2. 用滑动器以及载入配方的方式更改变量值，以便进行测试。观察小正方形的大小和颜色如何变化。

11 脚本

脚本用于管理元件的功能。使用 C# 编写。

目标：

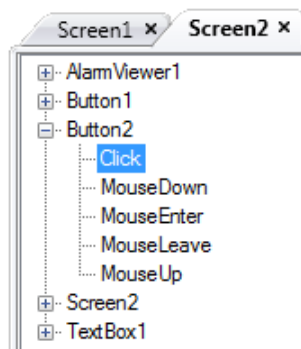
- 插入按钮和文本框。
- 为此按钮编写一个修改文本框中文本的脚本。

11.1 添加元件

1. 打开 **Screen2** 进行编辑。
2. 单击**主页**功能区标签页。
3. 单击**元件组**右下角的小箭头。
4. 在 **Windows 控制** 下选择**文本框**。
5. 在屏幕上画一个**文本框**。
6. 在画面中放置一个按钮，然后将其标注为**写入测试**。

11.2 脚本标签页

1. 选择**写入测试**按钮。
2. 单击位于桌面区域左下方的**脚本**标签页。
从布局到脚本的视图模式更改。
3. 展开 **Button2** 节点。



4. 双击**单击**节点。
您现在可以开始为 **Button2** 活动**单击**输入脚本代码。
在输入 **Ctrl + [空格]** 时，可激活上下文相关名称补全功能 (IntelliSense)，而在代码元素后输入句号 (".") 时可自动触发该功能。

5. 输入以下click事件代码：

```
TextBox1.Text= "Test";
```

“脚本” 标签页中的C#代码：

```
public partial class Screen2
{
    void Button2_Click(System.Object sender, System.EventArgs e)
    {
        TextBox1.Text= "Test";
    }
}
```

11.3 运行脚本

1. 运行项目。
2. 单击**写入测试**按钮，然后查看通过脚本代码指定的文本字符串是否出现在文本框中。

12 内部变量

内部变量可用于计算不需要在控制器中表示的值，例如仅提供给内部判断使用的信息。

目标：

- 编写脚本以便通过长度和宽度变量执行面积计算。
- 通过内部变量显示结果。

12.1 添加内部变量

- 单击项目浏览器中的变量。
这样便会在桌面上打开变量配置页面。

12.1.1 面积变量

1. 添加一个变量，然后将其标注为 **Area**。
2. 将数据类型更改为 **FLOAT**。

12.1.2 计算变量

1. 添加一个变量，然后将其标注为 **Calc**。
2. 将数据类型更改为 **BIT**。

变量			控制器	
名称	数据类型	访问权限	数据类型	Controller 1
MeterAndSlider	DEFAULT	ReadWrite	INT 16	D0
Tag2	DEFAULT	ReadWrite	INT 16	D1
Counter	DEFAULT	ReadWrite	INT 16	C0
BoolAlarmTag	DEFAULT	ReadWrite	DEFAULT	
重量	DEFAULT	ReadWrite	INT 16	D2
长度	DEFAULT	ReadWrite	INT 16	D3
宽度	DEFAULT	ReadWrite	INT 16	D4
Area	FLOAT	ReadWrite	DEFAULT	
Calc	BIT	ReadWrite	DEFAULT	

将 **Controller1** 列保留为空会使该变量成为内部变量，不会使它连接到控制器。

3. 切换到脚本视图模式。
4. 展开 **Calc** 节点。
5. 双击 **ValueChange**。

为使用 C# 控制代码访问数据和方法，可使用关键字 “Globals” 。本例中使用的是显式类型 casting (“(double)”)，这在运算量过大的情况下很有必要。

- 在 **ValueChange** 节点中计算面积：

```
Globals.Tags.Area.Value =
    (double) Globals.Tags.Length.Value *
    (double) Globals.Tags.Width.Value / 100;
```

12.2 创建模拟数值

- 打开 **Screen4** 进行编辑。
- 单击**主页**功能区标签页。
- 在**元件组**中选择**模拟数值**元件。
- 在那组滑动器下方画出模拟数值元件，与矩形元件保持一段距离。
- 右键单击模拟数值元件并将其连接到**Area**变量。
- 保持模拟数值元件为选中状态并单击**常规**功能区标签页。
- 在**显示格式**组下拉列表中选择**小数**。
- 将小数位数设置为**2**。



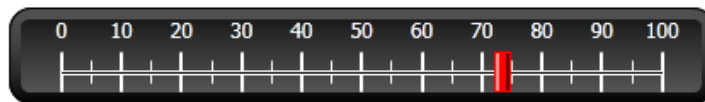
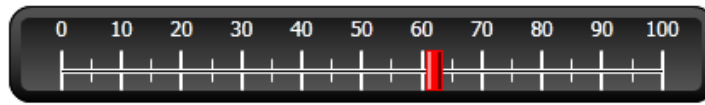
- 从**元件组**中选择一个**文本**元件，并将文本元件置于画面上。
- 在画面上放置一个与模拟数值元件相关的文本元件，并将其命名为**Area**：

12.2.1 变量操作

- 打开 **Screen4** 进行编辑。
- 选择矩形元件。
- 前往**操作**功能区标签页。
- 在**鼠标按下**组中，打开**选择操作...**下拉列表。
- 展开**变量**节点并选择**切换**。
- 打开**选择变量...**下拉列表并选择**Calc**变量。

12.3 内部变量运行测试

1. 运行项目。
2. 将滑动器设置为各种值进行测试。单击动态矩形区域并观察模拟数值控件的变化。



Area:

45,26

13 元件浏览器

在**元件浏览器**中可以总体浏览某一画面中包含的所有元件。

目标：

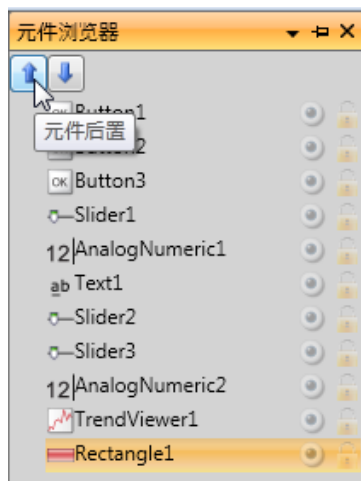
- 深入定位元件
- 锁定元件
- 使元件变为不可见

13.1 添加图形元素

1. 打开**Screen4**进行编辑。
2. 单击**主页**功能区标签页。
3. 从**元件组**中选择**矩形**。
4. 画一个矩形，使其与滑动器和按钮组合的背景大小一致。
之后此矩形便可完全遮盖住其他元件。

13.2 使用元件浏览器

1. 在**视图**功能区标签页中选择**元件浏览器**。
2. 在画面中选择矩形。
3. 单击**元件后置**箭头按钮直至所有按钮和滑动器可见。



4. 单击**元件浏览器**中选定矩形元件边上的**锁定**图标。
5. 尝试通过拖动的方式在画面中移动矩形。
元件处于锁定状态，无法移动。任何情况下均无法选中。
6. 选择某一按钮，然后单击**可见性**按钮。
按钮被隐藏。但如果项目在 iX Runtime 中运行，元件将正常显示。

7. 再次单击矩形**锁定**按钮并删除矩形。

14 文本库

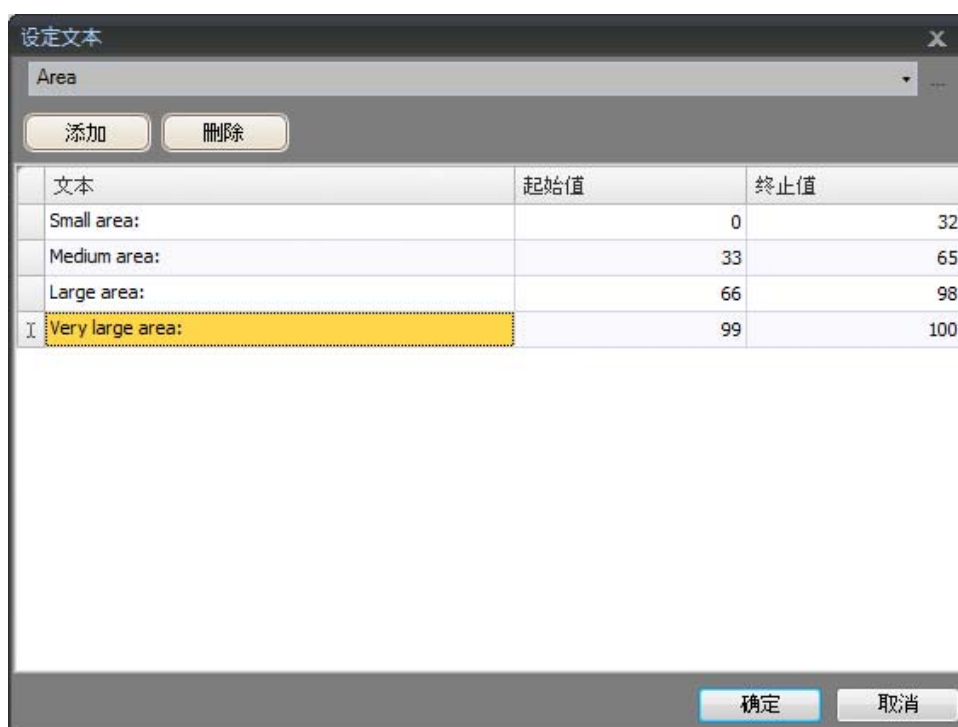
文本元件可用于显示文本信息，还可反映控制器变量的变化。

目标：

- 显示能够反映计算所得面积的变化可变文本消息。

14.1 配置文本

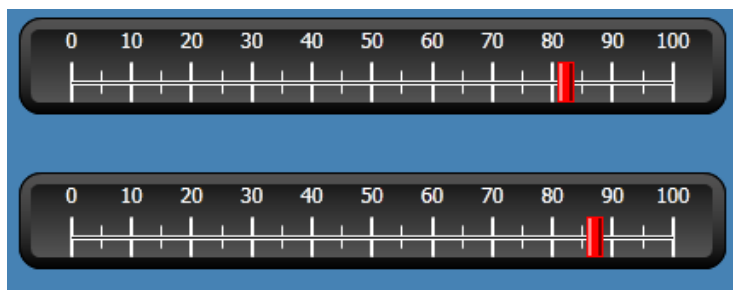
1. 打开 **Screen4** 进行编辑。
2. 选择标注为 **Area:** 的文本元件。
3. 单击**常规**功能区标签页。
4. 单击**文本组**中的**设定文本**。
5. 将文本连接到 **Area** 变量。
6. 按照下方示例添加文本字符串并编辑间隔。



由于文本元件的默认设置为**自动调整**，因此无需为了能够在运行时容纳最长的字符串而调整元件。

14.2 文本库运行测试

1. 运行项目。
2. 将滑动器设置为各种值进行测试。单击动态矩形并观察模拟数值控件的变化。同时也要检查文本是否已更新。



Large area:

71,34

15 用户权限

通过使用用户权限组和用户密码，可以限制对项目元件和操作的访问。

目标：

- 添加用户名并设置密码。
- 设置登录控件和注销控件。
- 限制对配方处理的访问。

15.1 用户权限设定

- 单击项目浏览器中的用户权限，打开配置页面。

15.1.1 用户权限组

用户权限的处理方式是将用户划分到各个用户权限组。这需要在组标签页中进行配置。在此例中，会使用两个默认的用户权限组，即Administrators (管理员)和Operators (操作员)。

15.1.2 用户

1. 在用户权限配置页面中选择用户标签页。
2. 添加一个用户，然后将其命名为超级用户。
3. 将超级用户连接至组下的管理员和操作员。
4. 再添加一个用户，将其命名为User1并连接至操作员组。
5. 输入用户密码。

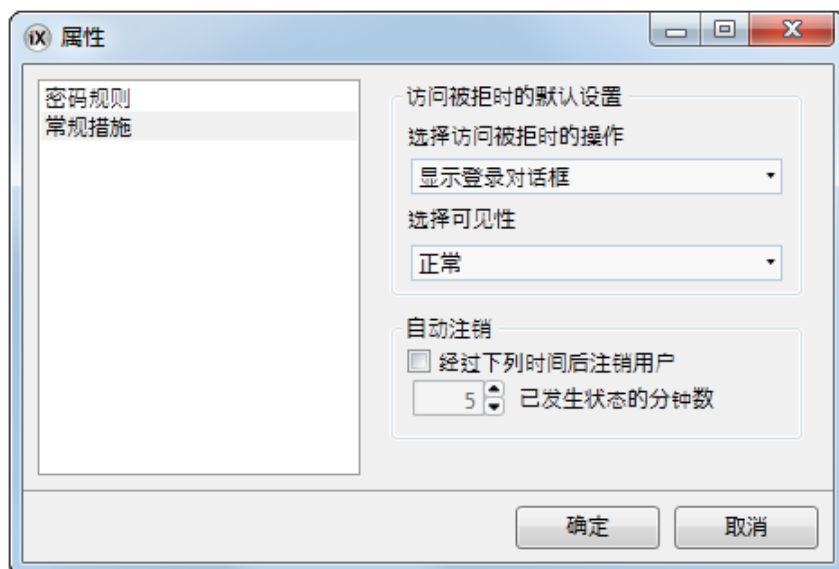
用户名	密码	说明	组
Administrator	*****		Administrators
SuperUser	*****		Administrators, Operators
> User1	oxpy		Operators

注意：
离开密码输入单元格时，密码会变为星号。

15.2 访问被拒绝时的登录行为

只要用户试图访问对于其所属用户组设有限制的元件，就会显示一个登录对话框。

1. 选择**用户**标签页，然后单击**设置**按钮。
2. 为访问被拒绝时的操作选择**显示登录对话框**，为可见性选择**正常**。



3. 单击**确定**。

15.3 创建注销按钮

1. 打开**Screen3**（背景画面）进行编辑。
2. 在**启动画面**按钮旁边放置一个按钮。将此按钮标注为**注销**。
3. 使该按钮保持选中状态，然后单击**操作功能区**标签页。
4. 打开单击组中的**选择操作...**下拉列表。
5. 展开**用户权限**节点并选择**注销**。

15.4 元件用户权限

1. 打开 **Screen4** 进行编辑。

15.4.1 管理员

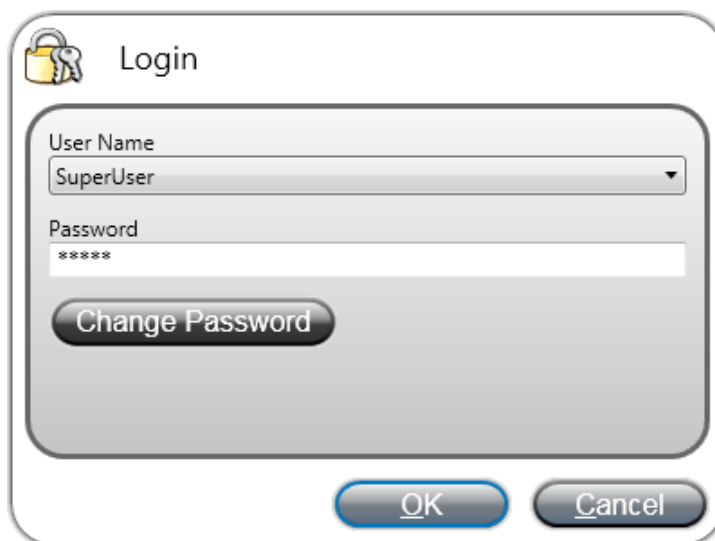
1. 右键单击**保存配方**按钮，然后为**选择用户权限组**选择**管理员**。

15.4.2 用户

1. 右键单击**载入配方**按钮，然后为**选择用户权限组**选择**操作员**。

15.5 运行测试用户权限

1. 运行项目。
2. 单击 **Screen4** 按钮。
3. 进行测试以确保如果未登录则无法载入或者保存配方，并在按任意按钮时都会打开登录对话框。
4. 以**管理员**身份登录并保存配方。
5. 进行配方载入测试。
登录对话框打开。
6. 以**User1**身份登录并载入配方。
7. 进行配方保存测试。
登录对话框打开。
8. 以**超级用户**身份登录。进行配方保存和载入测试。



9. 注销。
10. 测试是否无法载入或保存配方。

16 功能键

功能键可用于执行操作和运行脚本。无论当前为哪一画面，操作员均可通过功能键控制数据和画面的功能。

功能键还可配置为局部功能键，即它们的适用范围仅限于各自的画面。本例中使用的是全局功能键。



目标：

- 通过对功能键的操作进行编程来更改画面、设置控制器变量值、配方管理和显示登录对话框。
- 通过对功能键进行编程执行脚本。

16.1 定义功能键操作

1. 单击**插入**功能区标签页中的**功能键**。

16.1.1 显示画面

1. 单击功能键**F1**对应行中**操作**下的... 可打开**属性**对话框。
2. 单击**选择触发器...**并选择**向下键**。
3. 展开**画面**节点并选择**显示启动画面**。
4. 单击**确定**。
5. 单击功能键**F2**对应的行。
6. 单击**操作**下的...。
7. 单击**选择触发器...**并选择**向下键**。
8. 展开**画面**节点并选择**显示画面**。
9. 选择下一个下拉列表中的**Screen2**。
- 10 单击**确定**。
- .
- 11 单击功能键**F3**对应的行。
- .
- 12 单击**操作**下的...。
- .
- 13 单击**选择触发器...**并选择**向下键**。
- .

- 14 展开画面节点并选择**显示画面**。
.
- 15 选择下一个下拉列表中的**Screen4**。
.
- 16 单击**确定**。
.

16.1.2 用户权限

1. 单击功能键**F4**对应的行。
2. 单击**操作下的...**。
3. 单击**选择触发器...**并选择**向下键**。
4. 展开**用户权限**节点并选择**登录**。
5. 单击**确定**。

16.1.3 设置控制器变量值

1. 单击功能键**F5**对应的行。
2. 单击**操作下的...**。
3. 单击**选择触发器...**并选择**向下键**。
4. 展开**变量**节点并选择**设定数值**。
5. 在**选择变量...**下拉列表中选择**重量**变量。
6. 将模拟值指定为**50**。
7. 单击**确定**。
8. 重复上述步骤设置功能键**F6**和**F7**以控制**长度**和**宽度**变量。

16.1.4 配方

- 设置功能键**F8**用于加载**Recipe1**，并设置功能键**F9**用于保存**Recipe1**。将**选择配方数据(可选)...**保持为空。

16.1.5 设定时区、地区和夏令时

- 设置功能键**F10**用于设置时区、地区和日光节约时间。此操作位于**其他**组中。

功能鍵	操作
F1	Show Start Screen
F2	Show Screen
F3	Show Screen
F4	Login
F5	Set Analog
F6	Set Analog
F7	Set Analog
F8	Load Recipe
F9	Save Recipe
> F10	Set Time Zone, Region and Daylight Saving ...

Show Start Screen

16.2 定义功能键脚本

功能键还可用于触发脚本。

16.2.1 面积计算

编写一个用于进行矩形元件面积计算的功能键：

1. 单击功能键F11对应的行。
2. 单击画面底部的脚本，然后选择脚本视图模式。
3. 展开F11节点，然后双击其KeyDown节点。
4. 通过如下代码在KeyDown事件中计算面积：

```
Globals.Tags.Area.Value =
    (double) Globals.Tags.Length.Value *
    (double) Globals.Tags.Width.Value / 100;
```

使用功能键脚本便无需使用单独的触发器变量 (Calc)。

16.2.2 确认所有警报

编写一个功能键用来确认所有警报：

1. 展开F12节点，然后双击其KeyDown节点。
2. 输入以下KeyDown事件代码：

```
Globals.AlarmServer.Acknowledge();
```

脚本标签页中的C#代码：

```
public partial class FunctionKeys
{
    void F11_KeyDown(System.Object sender, System.EventArgs e)
    {
        Globals.Tags.Area.Value =
            (double) Globals.Tags.Longitud.Value *
            (double) Globals.Tags.Ancho.Value / 100;
    }

    void F12_KeyDown(System.Object sender, System.EventArgs e)
    {
        Globals.AlarmServer.Acknowledge();
    }
}
```

16.3 功能键运行测试

1. 运行项目。
2. 测试定义的功能键（PC 键盘上）是否能够用于执行相应操作。

17 对象使用浏览

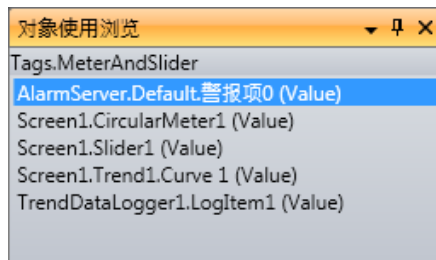
通过交叉引用工具可以总体浏览特定变量在当前项目中的使用位置。

目标：

- 快速定位出某特定变量的所有出现位置。

17.1 使用元件使用浏览工具

- 单击项目浏览器中的变量。
- 如果您在脚本视图模式中，那么单击底部的设计。
- 选择 **MeterAndSlider** 行。
- 单击交叉引用按钮。
这时将显示交叉引用工具。



- 双击交叉引用列表中的警报设定. 默认. 警报项0 (值)。
这样便会在桌面上打开警报设定配置页面。
- 双击交叉引用列表中的 Screen1. 滑动器 (值)。
这时会在桌面上打开 Screen1，并且 Slider1 呈选中状态。