

操作指南 • 5 月/2016

WinCC V7.3 中如何使用结构变量组态画面模板

WinCC, 结构类型, 画面模板, 脚本

摘要

在实际的工业级现场及应用中，经常会遇到多个设备需要显示相同的参数（组）的情况，如现场有多个电机，每个电机需要显示和控制的参数（组）都相同，只是不同电机的参数数值不同。使用画面模板，能够避免多册重复组态相同的画面，减少组态人员的工作量，又能避免日后项目维护时期的大量工作，提高系统的可维护性。画面模板和结构变量配合使用，能够快速高效地组态模板，满足现场要求。

本文档实例的组态环境为 SIMATIC WinCC V7.3SE Update9，操作系统环境为 64 位 Windows 7 Ultimate SP1。

关键词 WinCC，结构类型，画面模板，脚本

Key Words WinCC, construct, picture template, script

目录

1	示例说明	4
1.1	示例任务.....	4
1.2	解决方案.....	4
2	建立结构变量	5
3	使用变量前缀建立画面窗口	6
3.1	创建画面模板.....	6
3.1.1	新建画面.....	6
3.1.2	组态 IO 域.....	6
3.1.3	组态按钮.....	8
3.1.4	组态滚动条和量表.....	10
3.1.5	组态趋势控件.....	11
3.2	创建起动画面，调用画面模板.....	12
3.2.1	静态设置 TagPrefix.....	13
3.2.2	通过变量修改 TagPrefix.....	14
3.2.3	通过脚本修改 TagPrefix.....	14
4	使用动态向导建立画面模板	16
4.1	创建基准画面.....	16
4.2	利用动态向导创建模板.....	16
4.2.1	启动动态向导.....	16
4.2.2	选择结构类型.....	16
4.2.3	连接对象属性.....	17
4.2.4	完成向导.....	17
4.3	调用模板.....	18
4.3.1	添加模板.....	18
4.3.2	连接模板和结构变量.....	19
4.3.3	完成组态.....	19
4.3.4	运行效果.....	20
5	两种画面模板建立方法的比较	21
	附录 - 推荐网址	22

1 示例说明

1.1 示例任务

假设某生产现场有 3 台电机，每台电机都有电机名字、速度实际值、速度设定值、启停状态等参数需要显示在 WinCC 画面中，需达到的效果如图 1-1 所示。



图 1-1

1.2 解决方案

在该任务中，每台电机参数类型都一致，可以使用自定义结构类型来存储电机的参数；在画面中显示，每个电机的显示仅是数据不同，这可以通过建立画面模板来实现，画面模板只需建立一次，然后多次调用画面模板并传入不同的电机参数，如图 1-2 所示。画面模板的建立方式有两种：1.利用前缀变量建立画面窗口；2.利用动态向导建立画面模板。

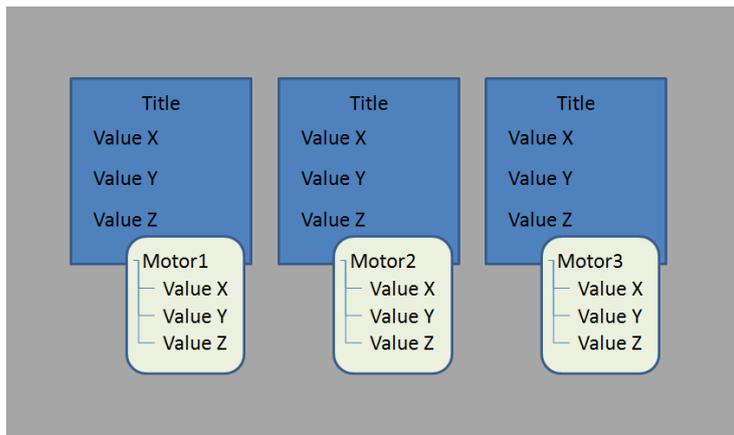


图 1-2

2 建立结构变量

在 WinCC 项目中，双击“ Tag Management” 打开变量管理窗口，如图 2-1 所示。右键点击“ Structure tags”，选择“ New structure type” 建立结构变量，本例命名为 Motor，如图 2-2 所示。

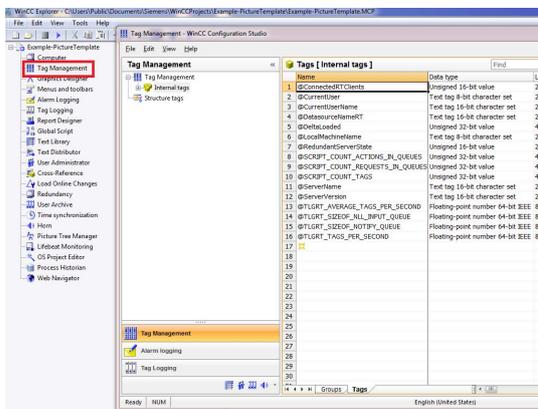


图 2-1

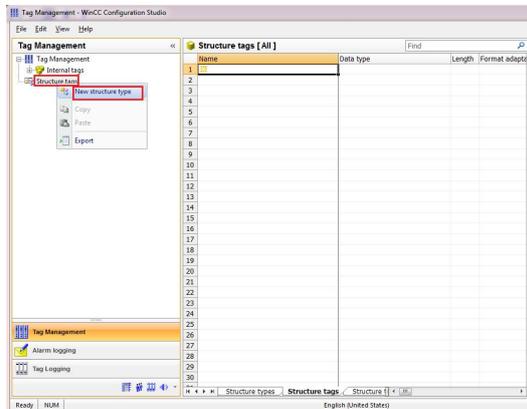


图 2-2

选中 Motor，在“ Structure type elements” 标签页下，添加结构变量的元素及其各元素数据类型，完成后，如图 2-3 所示。进入“ Structure tags” 标签页，添加三个 Data type 为 Motor 的变量，如图 2-4 所示。至此，完成结构类型的定义和结构变量的添加。

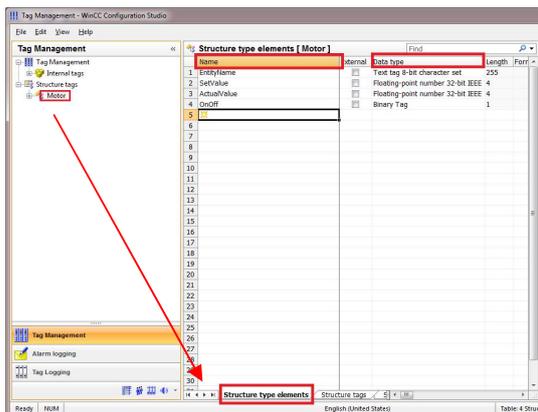


图 2-3

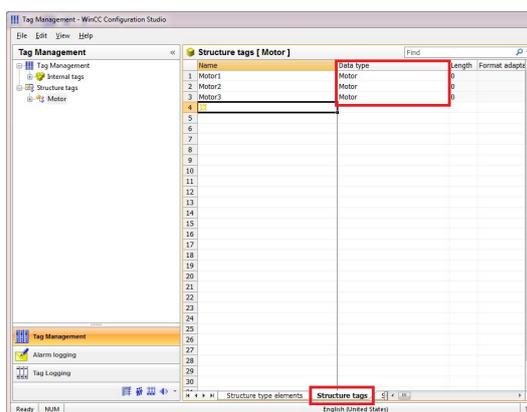


图 2-4

3 使用变量前缀建立画面窗口

该方式是先组态一个画面作为画面模板，令画面中的对象与结构变量的元素相连，在运行画面中，通过添加画面窗口调用该画面，并将画面窗口的前缀变量设置为结构变量的名称。

3.1 创建画面模板

3.1.1 新建画面

WinCC Explorer 下右键点击“ Graphics Designer”，选择“ New picture” 建立一个画面模板，并命名为 PictureTemplate.Pdl，如图 3-1 所示。

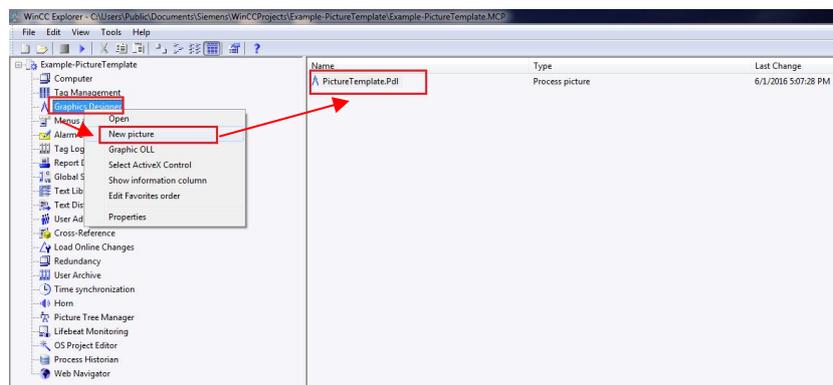


图 3-1

打开画面，进入 Object Properties 标签页，在 Properties Object->geometry 中设置画面模板的长宽，根据实际需求设定，如图 3-2 所示。

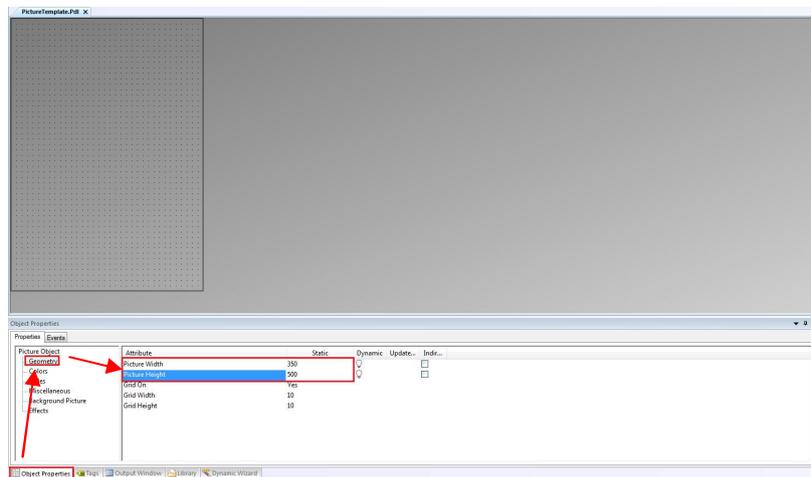


图 3-2

3.1.2 组态 IO 域

在画面中添加一个 IO 域，随即弹出 IO Field Configuration 对话框。将 Tag 关联为 Motor1.EntityName，如图 3-3 所示。

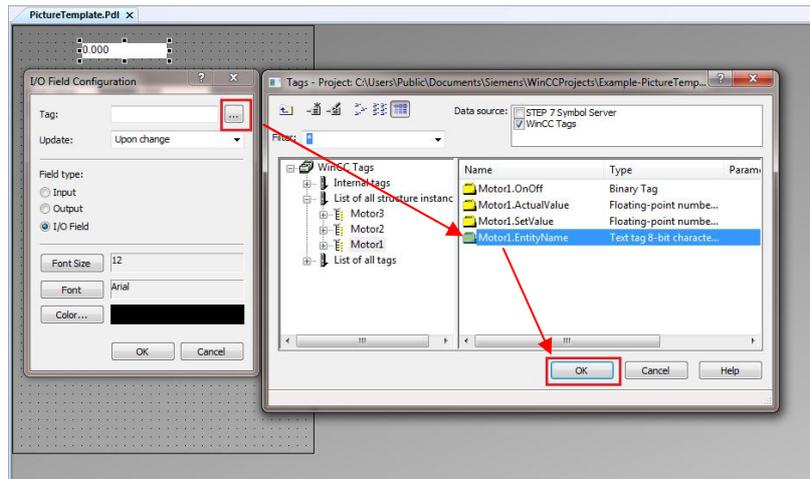


图 3-3

然后去掉 Tag 的变量前缀，仅保留 EntityName（即结构变量的元素名），再将 Update 选为 Upon change，如图 3-4 所示。

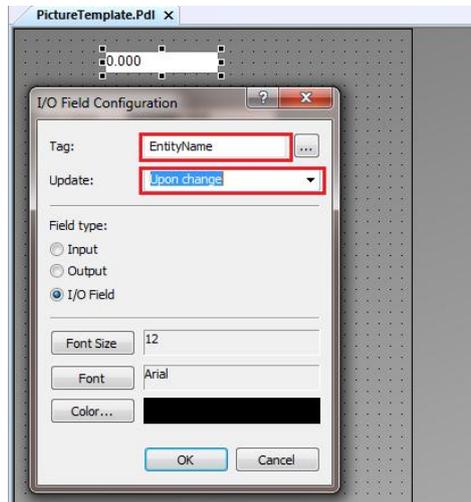


图 3-4

该 IO 域用于字符显示，还需要将 Properties->Output/Input->DataFormat 设置为 String，如图 3-5 所示。至此，用于显示 EntityName 元素的 IO 域组态完成。

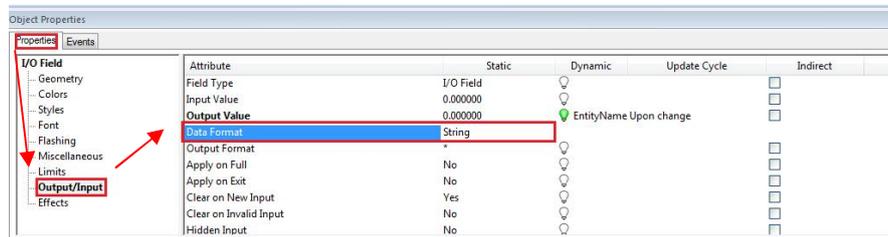


图 3-5

继续组态两个 IO 域，用于显示 SetValue 和 ActualValue 的变量值，操作方法参考上文，变量连接效果如图 3-6 所示。

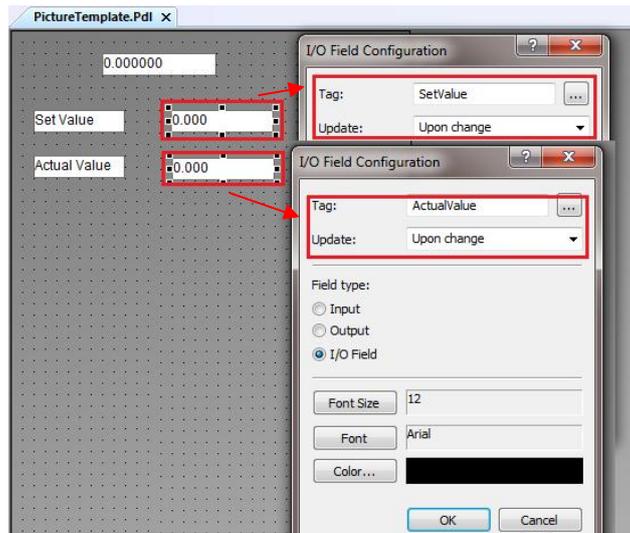


图 3-6

3.1.3 组态按钮

1. 添加按钮。在画面中添加一个按钮，随即弹出 **Button Configuration** 对话框。将 **Text** 设置为 **Start**，表示初始显示的文本，如图 3-7 所示。

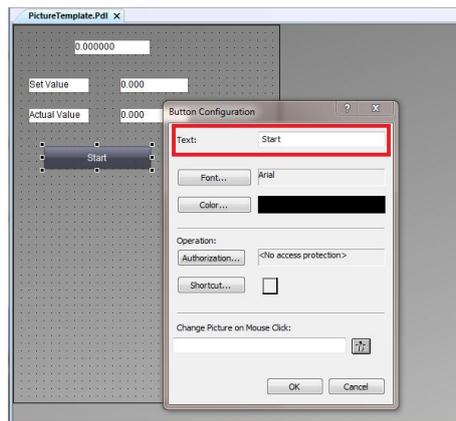


图 3-7

2. 添加点击操作的 C 脚本。选中按钮，在 **Events->Button->Mouse** 下的 **MouseClicked** 右边，右键点击闪电图标，选中 **C-Action** 为点击操作编写 C 脚本，如图 3-8 所示。在弹出的脚本编辑器中添加如图 3-9 所示的代码，代码实现的功能是将 **OnOff** 元素的值取反。

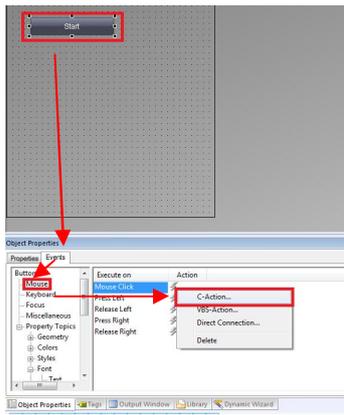


图 3-8

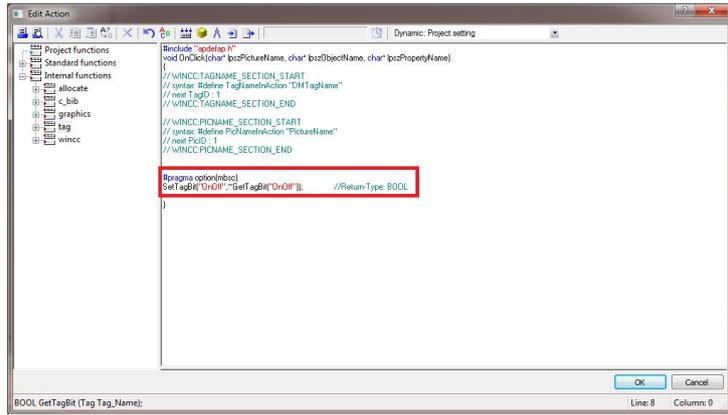


图 3-9

3. 按钮文本动态变化的设置。选中按钮，在 Properties->Button->Font 中，右键单击 Text 的小灯泡图标，选择 Dynamic Dialog，如图 3-10 所示。在弹出的 Value Range 对话框中，单击 Event Name 右边的浏览图标进行动作触发器的设置，Event 选为 Tag，Tag name 为用以触发动作的变量，此处为 OnOff，将 Standard cycle 选为 Upon change，点击 OK 保存设置，如图 3-11 所示。回到 ValueRange 对话框，将 Expression 填' OnOff'，变量类型选择 Boolean，并编辑对应状态下的文本，点击 OK 后会弹出 Missing tags 提示框，忽略即可，如图 3-12 所示。说明：会弹出提示框是因为我们关联的变量 OnOff 只是一个结构变量的元素名，系统目前无法在变量表中找到，故目前先忽略该提示。

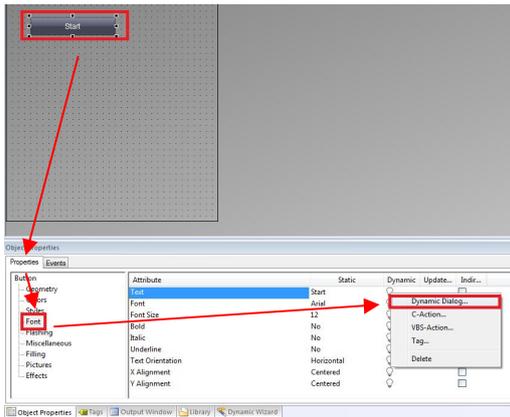


图 3-10

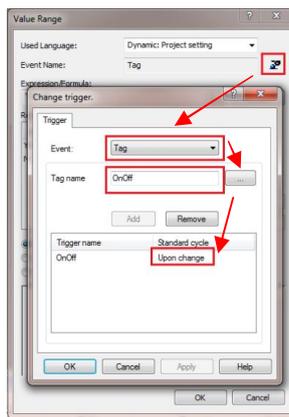


图 3-11

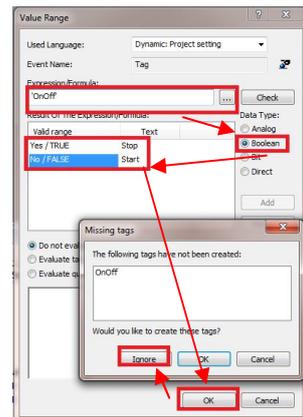


图 3-12

4. 按钮颜色动态变化的设置。选中按钮，在 Properties->Button->Color 中，将 BackgroundColor 的静态颜色改为绿色（作为初始颜色），右键单击小灯泡，选择 Dynamic Dialog，设置动作触发器和动态颜色的对应关系。操作可以参考文本的动态设置，如图 3-13 所示。在 WinCC V7.3SE 中，所有对象都默认使用“ Global Color Scheme”，为了能够让按钮颜色变化，我们还需在 Properties->Button->Effects 中将该属性选为 No，如图 3-14。

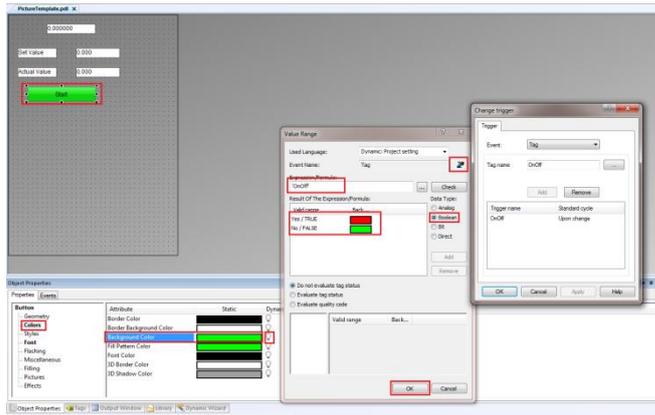


图 3-13

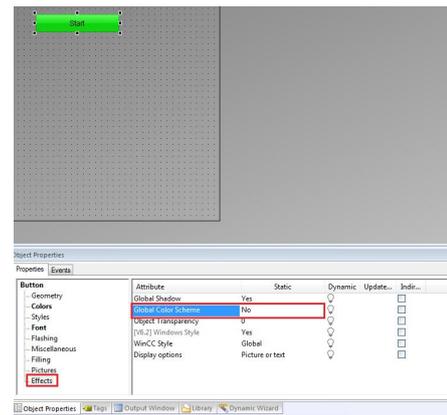


图 3-14

3.1.4 组态滚动条和量表

1. 组态滚动条控件。在画面中添加滚动条控件，在 Properties->SliderCtrl->ControlProperties 中，设置 Position 的 Static 值为 0，将 Dynamic 值设为 SetValue，UpdateCycle 为 Upon change，并将 RangMin 和 RangMax 属性设为合适的值，如图 3-15 所示。

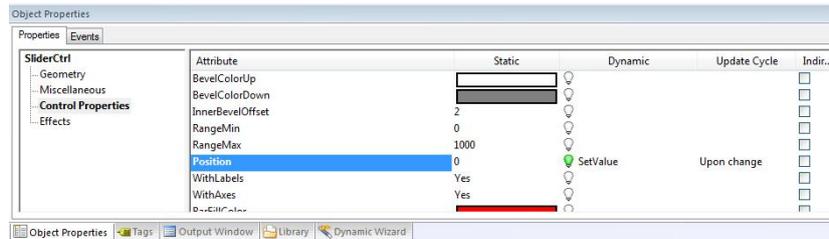


图 3-15

2. 组态量表控件。在画面中添加一个量表控件，在 Properties->XGaugeCtrl->ControlProperties 中，设置 Value 的 Static 值为 0，将 Dynamic 值设为 ActualValue，UpdateCycle 为 Upon change，并将 RangMin、RangMax 和 Delta 属性设为合适的值，如图 3-16 所示。

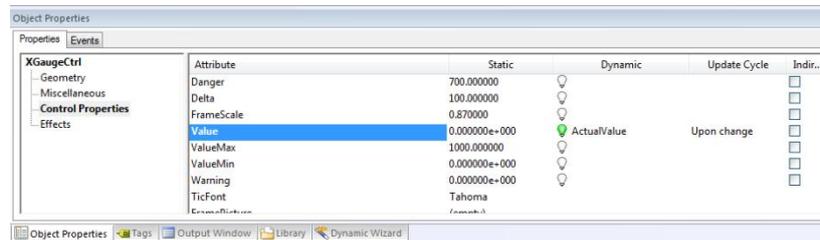


图 3-16

画面效果如 3-17 所示。

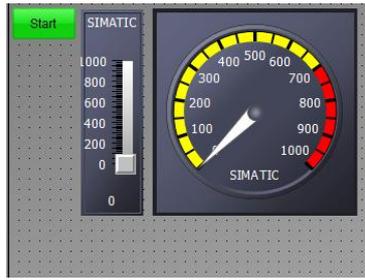


图 3-17

3.1.5 组态趋势控件

上述对象都是带有 TagPrefix 属性，故可以将其直接连接到结构变量的元素上。对于没有 TagPrefix 属性的对象的处理方式由本小节进行介绍，以趋势控件为例，分别针对在线变量和归档变量进行说明。

(1) 趋势控件关联在线变量

添加趋势控件，在弹出的 WinCC OnlineTrendControl Properties 对话框中，将 Data Source 选择为 Online tags，Tag name 保持为空，如图 3-18 所示。

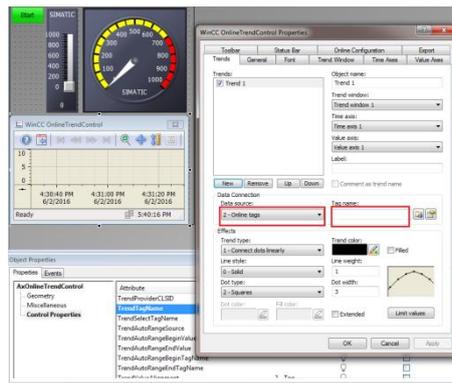


图 3-18

然后，需要在 Properties->Control->TrendTagName 中，在 Dynamic 处添加 C-Action，如图 3-19 所示。

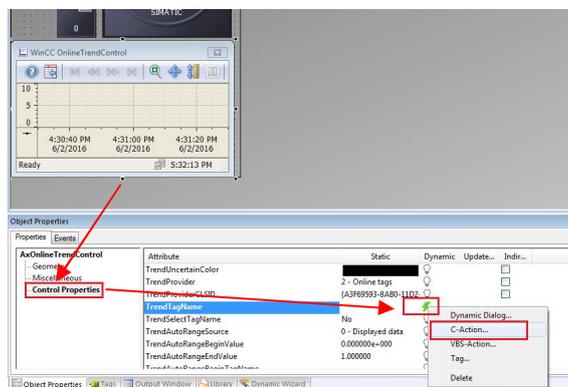


图 3-19

在脚本编辑器中添加如图 3-20 所示的代码，并添加动作触发器，如图 3-21 所示。

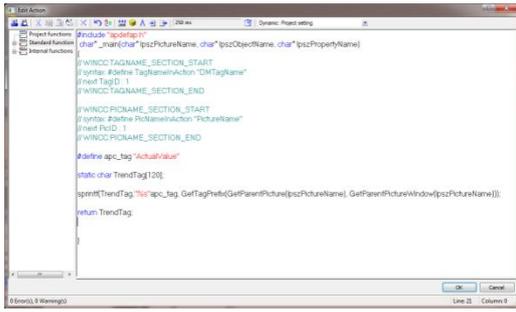


图 3-20

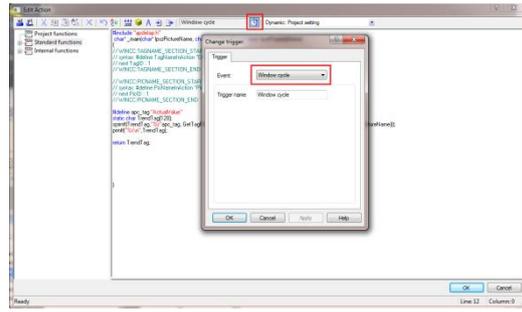


图 3-21

代码文本: (本段代码由网友“玄极道人”提供)

```
#define apc_tag " ActualValue" //元素名称
static char TrendTag[120];
sprintf(TrendTag,"%s"apc_tag, GetTagPrefix(GetParentPicture(lpszPictureName),
GetParentPictureWindow(lpszPictureName)));
return TrendTag;
```

(2) 趋势视图控件关联归档变量

添加趋势控件，在弹出的 WinCC OnlineTrendControl Properties 对话框中，将 Data Source 选择为 Archive tags，Tag name 保持为空，如图 3-22。

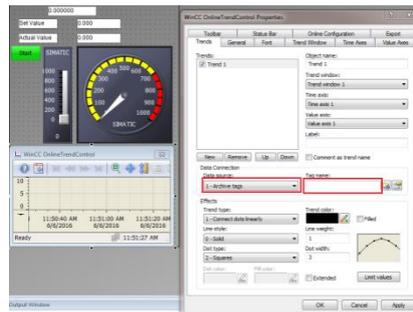


图 3-22

同上述方法，需要给该控件的 TrendTagName 属性添加 C-Action，不同于在线变量，归档变量的名称还包含有归档名称，假设归档名称为“ MotorArchive”，则代码文本如下(本段代码由网友“玄极道人”提供):

```
#define arch_tag " MotorArchive\\" //归档名称
#define apc_tag " ActualValue" //元素名称
static char TrendTag[120];
sprintf(TrendTag, arch_tag"%s"apc_tag, GetTagPrefix(GetParentPicture(lpszPictureName),
GetParentPictureWindow(lpszPictureName)));
return TrendTag;
```

3.2 创建起动画面，调用画面模板

在 WinCC Explorer 中右键点击 Graphic Designer 添加一个名为“ PictureStart.pdl” 的画面，右键点击该画面选择 Define screen as start screen 使之成为起动画面。如图 3-23 所示。

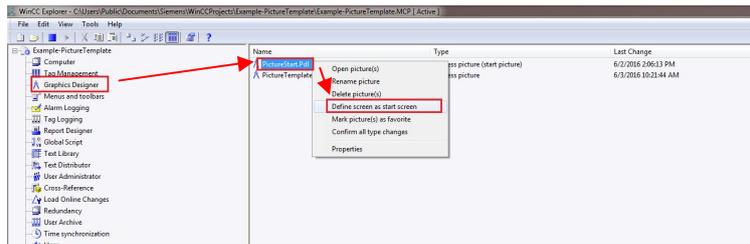


图 3-23

打开 PictureStart.pdl 画面，添加三个 PictureWindow 对象，并在 Properties->Picture Window->Geometry 中设置对象的长和宽，设置为与之前建立的画面模板一致，以保证画面模板显示完整，如图 3-24 所示。

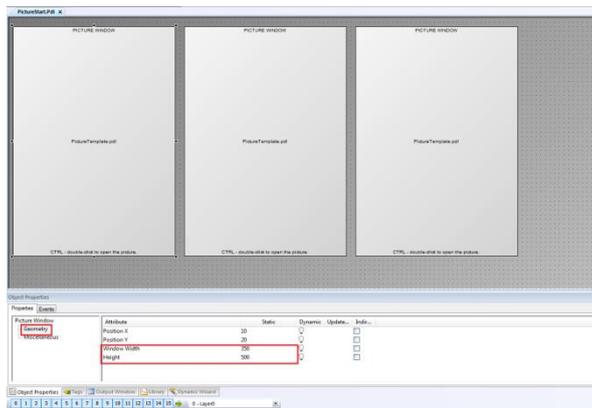


图 3-24

将三个对象的 PictureName 属性均设置为画面模板的名称 PictureTemplate.pdl，这样就能调用之前画面模板的内容了，如图 3-25。

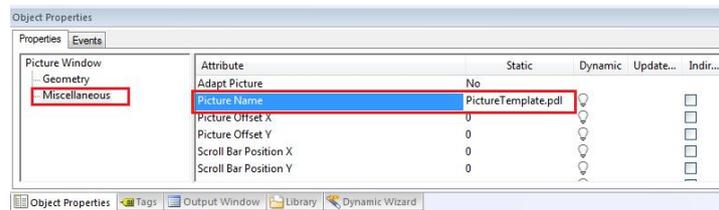


图 3-25

为了能让 PictureWindow 中显示不同的数据，还需要进行 PictureWindow 对象的 TagPrefix 属性设置。设置 TagPrefix 可以静态的，也可以通过变量或脚本进行动态设置，下面分别介绍。

3.2.1 静态设置 TagPrefix

选中画面中的 PictureWindow 控件，在 Properties->Picture Window->Miscellaneous 中，将 TagPrefix 的 Static 值设为 Motor1.或 Motor2.或 Motor3.，如图 3-26 所示。这样，PictureWindow 三个对象中显示的数据就是来自于三个不同的结构变量。

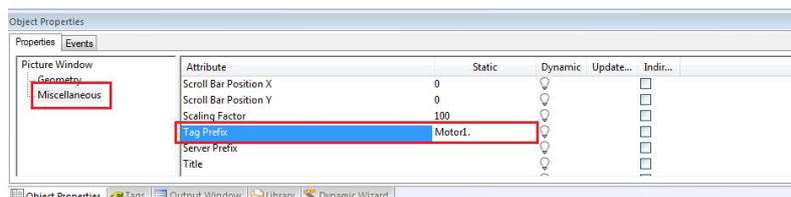


图 3-26

3.2.2 通过变量修改 TagPrefix

选中画面中的 PictureWindow 控件，在 Properties->Picture Window->Miscellaneous 中，将 TagPrefix 的 Dynamic 关联为 Tag（请自己在变量管理器中定义一个字符串变量用于控制 TagPrefix），Update change 设为 Upon change，如图 3-27 所示。

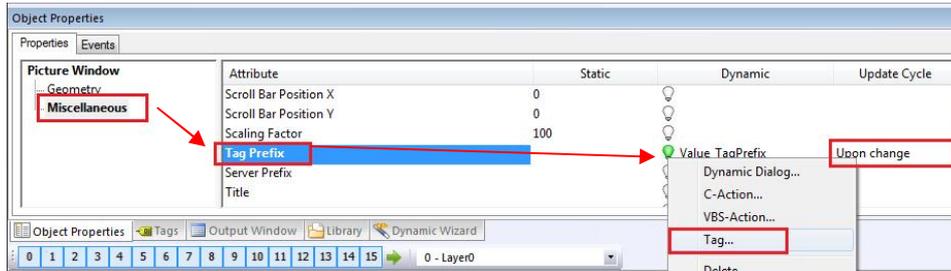


图 3-27

然后在 Event->Picture Window->Property Topics->Miscellaneous->Tag Prefix 中，右击闪电图标，将 change 关联为 Direct Connection，如图 3-28 和图 3-29。这样，就能在 TagPrefix 值变化后，重新加载画面名称，实现画面内容的更新。



图 3-28

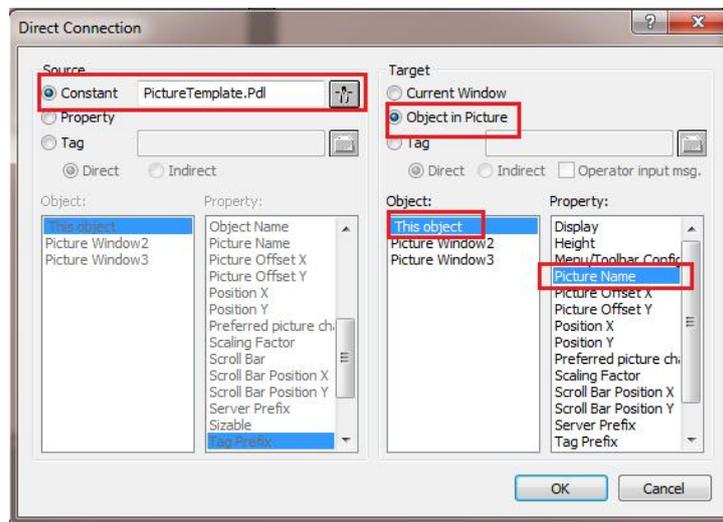


图 3-29

3.2.3 通过脚本修改 TagPrefix

使用该方式的话，可以在画面中再添加一个按钮，在该按钮的 click 事件中编 C-Action 脚本，可以实现对 TagPrefix 的动态修改，如图 3-30 所示。使用其他类型对象的事件动作脚本也可。

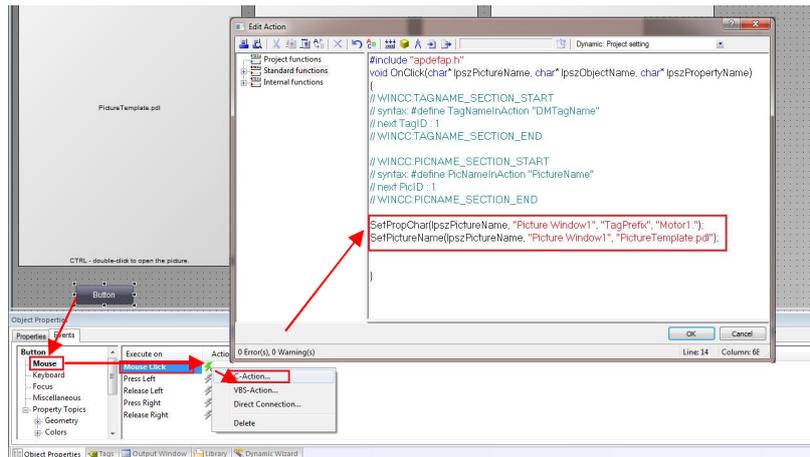


图 3-30

主要代码就两句。第一句为设置 TagPrefix 属性值，其中“ Picture Window1” 请用您实际的对象名称代替，“ Motor1.” 为你要设置的变量前缀；第二句的作用为设置画面名称，画面就会重新加载一次，这样画面内容才会刷新。

4 使用动态向导建立画面模板

该方式利用动态向导建立画面模板，将模板内的对象的属性和结构元素相连；然后再利用动态向导建立模板实例，并关联结构变量。

4.1 创建基准画面

建立一个新画面，添加一些对象，如静态文本、输入输出域、按钮等，具体操作参考 3.1 节的内容，(注意：这些对象都不用连接变量，并为其 **Object Name** 设置便于识别的名称，下文 4.3.4 节中的操作会用到)，保存画面，如 **DynPicModule.pdl**，如图 4-1 所示。

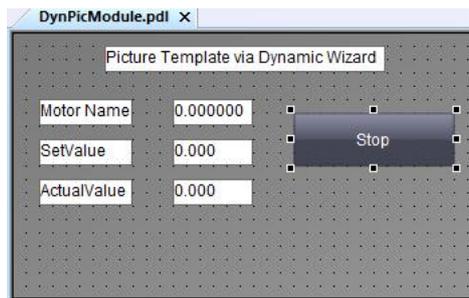


图 4-1

4.2 利用动态向导创建模板

4.2.1 启动动态向导

进入 Dynamic Wizard->Picture-Modules 选项卡，双击 **Picture-Module template-V1.14**，打开动态向导，如图 4-2 所示。

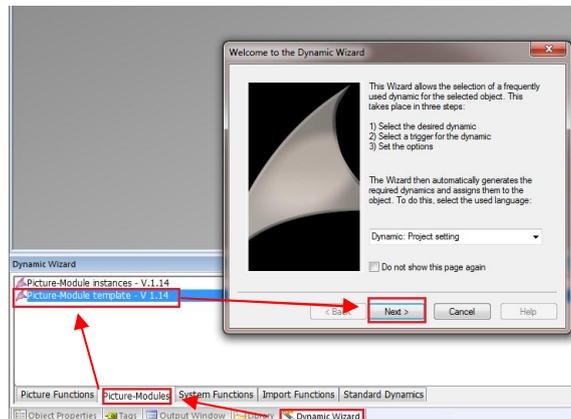


图 4-2

4.2.2 选择结构类型

选择先前建立的结构类型，用于连接画面模板中对象的属性，这里选择 **Motor**，如图 4-3 所示。

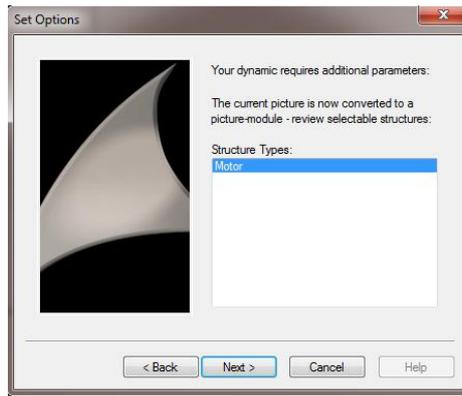


图 4-3

4.2.3 连接对象属性

将对象和结构的元素相连。在 **Object** 中选择对象，选择需要进行控制的属性，连接所关联的元素，并设置刷新时间，然后点击 **Link** 进行连接，完成所有关联之后，点击 **OK** 进入下一步，例如按钮可见性的变量关联如图 4-4 所示。

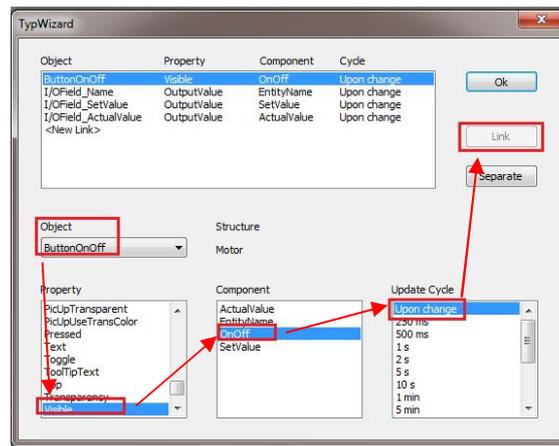


图 4-4

4.2.4 完成向导

完成向导后，WinCC 会自动创建一个新的画面@Type_DynPicModule.pdl，即模板画面，并且画面上方自动添加了一个 IO 域，其 **Object Name** 为 InstanceName，且自动组态了 C-Action，如图 4-5 和图 4-6 所示。

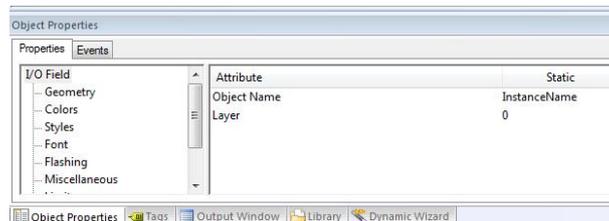


图 4-5

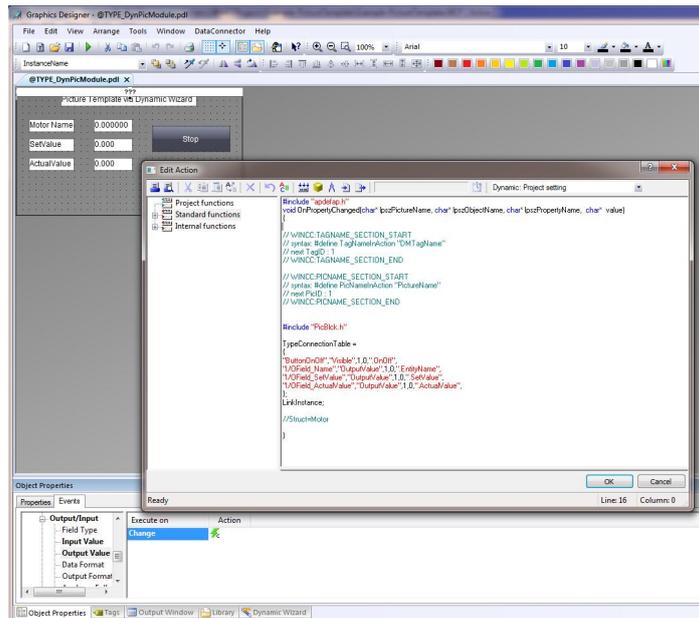


图 4-6

不难看出，该 C-Action 脚本的内容是设置画面中对象属性和结构元素的连接关系，这段脚本用户可以根据自己的需求做修改，**但是请不要修改 IO 域的 Object Name。**

请注意如下情况：

1. 建议在 WinCC 界面语言为英文时调用动态向导建立模板，这样自动生成的 IO 域的 Object Name 就是“ InstanceName”，如果在中文界面语言下调用该向导，则生成的 IO 域的 Object Name 就是”实例名称”，这时需要手动将其修改为“ InstanceName”。
2. 如果已经通过 template 产生了模板@Type_***.pdl 但又需要修改，那么如果是修改和结构元素的链接关系，则需修改 InstanceName 的“输出值更改”组态的 C 动作；如果需要修改画面中的对象或者添加/删除对象，则直接在模板画面中修改即可。

4.3 调用模板

4.3.1 添加模板

在起动画面中，进入 Dynamic Wizard->Picture-Modules 选项卡，双击 Picture-Module instance-V1.14，打开动态向导，选择 4.2 节中建立的模板，模板的显示方式请根据实际需求选择，本例选用 fixed module in picture（其他显示方式的效果用户可以自己测试，这里不做说明），进入下一步，如图 4-7 所示。

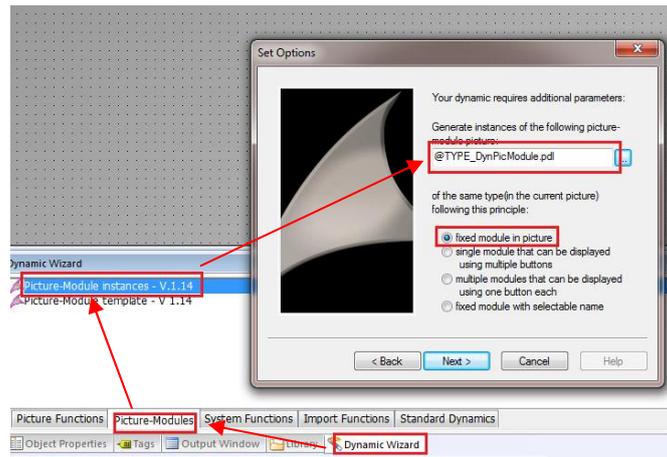


图 4-7

4.3.2 连接模板和结构变量

将模板分别和各结构变量相连，并设置模板在画面中的位置，如图 4-8 所示。

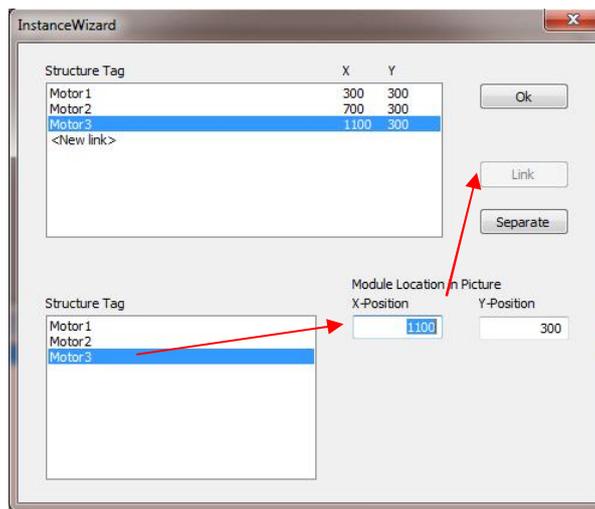


图 4-8

4.3.3 完成组态

完成组态后，画面中会出现相关的画面窗口，如图 4-9 所示。

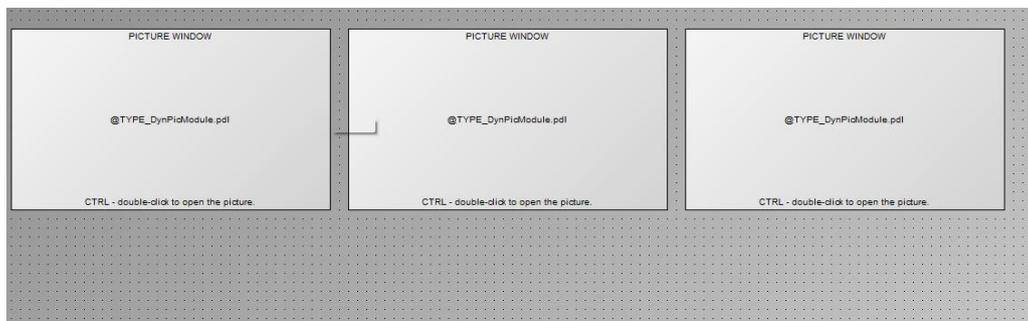


图 4-9

4.3.4 运行效果

运行 WinCC 后，可见各画面窗口中显示的是各个结构变量元素的数值如图 4-10 所示。

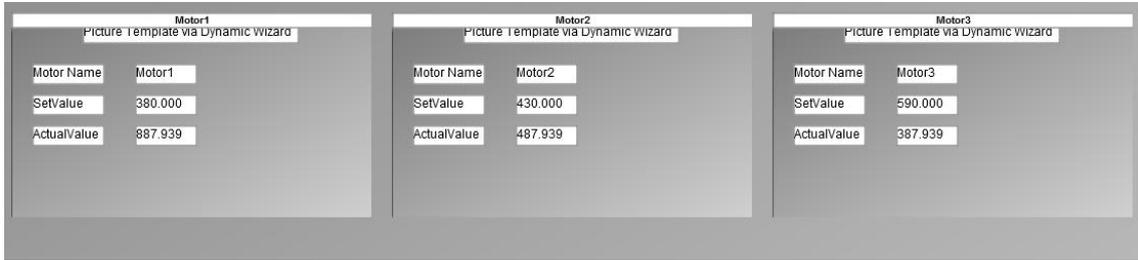


图 4-10

5 两种画面模板建立方法的比较

利用变量前缀，可以将结构元素关联到画面中对象的属性中，也能关联到对象的事件中，包括 C 动作、VBS 动作和直接连接。

利用动态向导生成的模板，只能将结构元素关联到画面中对象的属性里，不能关联对象的事件。但是并不是说模板里就不能为对象组态事件，只是事件的动作不是直接操作某个变量，而是间接地操作，例如用 C 函数 `GetLinkedVariable(lpszPictureName, " IOField1" , " OutputValue")` 即可定位到 " IOField1 " 所连接的变量。

声明：本文所描述的是如何利用结构变量组态画面窗口从而降低工作量，这并不表示使用画面窗口一定得结合结构变量！请明确区别。

附录一 推荐网址

HMI

西门子（中国）有限公司

工业自动化与驱动技术集团 客户服务与支持中心

网站首页: <http://www.ad.siemens.com.cn/Service/>

WinCC 下载中心:

<http://www.ad.siemens.com.cn/download/DocList.aspx?Typeld=0&CatFirst=1&CatSecond=9&CatThird=-1>

专家推荐精品文档: <http://www.ad.siemens.com.cn/Service/recommend.asp>

“找答案” WinCC 版区:

<http://www.ad.siemens.com.cn/service/answer/category.asp?cid=1032>

版权© 西门子（中国）有限公司 2001-2008 版权保留

复制、传播或者使用该文件或文件内容必须经过权利人书面明确同意。侵权者将承担权利人的全部损失。权利人保留一切权利，包括复制、发行，以及改编、汇编的权利。

西门子（中国）有限公司