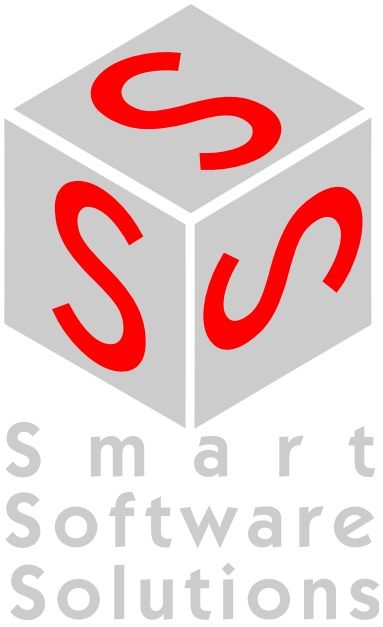
CoDeSys入门

更新日期: **05.03.2004**



目录

1 启动CoDeSys

启动 CoDeSys 编程系统：
开始 -> 所有程序 -> 3S Software -> CoDeSys V2.3 -> CoDeSys V2.3

2 编写第一个程序

- **任务：**

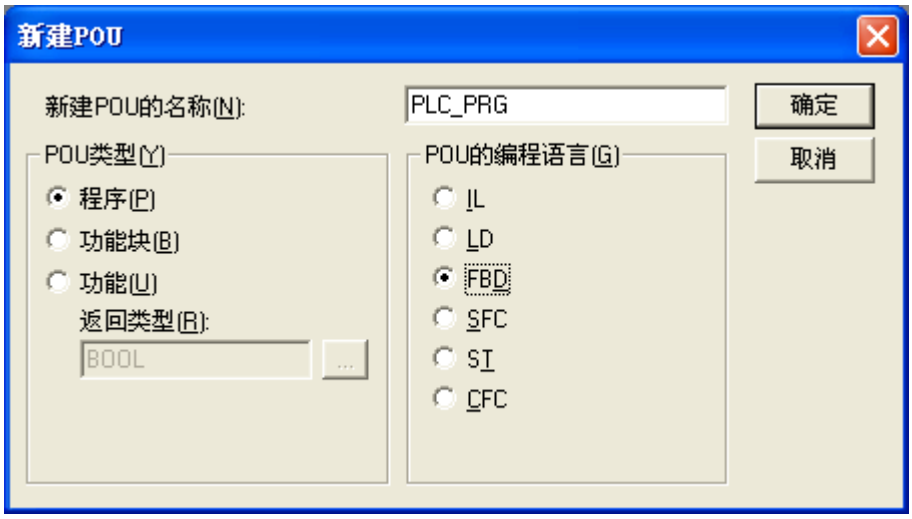
  一个机器操作工正在监控一台运行的机器。正确的运行必须是在规定的时间间隔内完成。如果超过运行时间，就会产生一个警告，过一会儿机器停止运行。

  机器的动作：手臂沿着一个矩形路径运动，每完成一周计数器加一。

- **创建一个新项目**
  启动很容易. 点击菜单**文件 -> 新建**.

- **目标系统设置**
  在“配置”的列表选项中选择适合的运行系统作为目标系统，如： 3S CoDeSys SP RTE

- **PLC_PRG POU**
  在**新建 POU** 对话框中选择 **FBD** (功能块图) 为 **POU** 的编程语言，POU 类型是**程序**和名称是**PLC_PRG**。
  PLC_PRG 是特殊的 POU，它将被循环调用并在实时系统中执行。



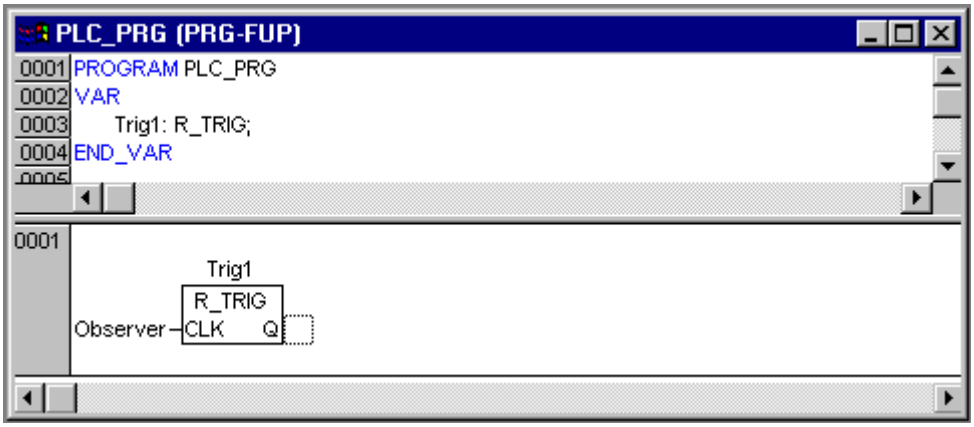
- **声明确认开关**
  我们从确认开关开始。可以看到第一个网络中有三个问号**???**，输入开关的名称(例如 **Observer**). 按右箭头键或回车键，弹出声明变量对话框：

- 将‘类别’改为 **VAR_GLOBAL** (定义成全局变量)。 点击‘**确认**’，下面文字将自动输入到全局变量对象中：

```
VAR_GLOBAL
        Observer: BOOL;
END_VAR
```

- **确认开关的上升沿**
  如果开关从关的状态(FALSE)变为开的状态(TRUE)，称为上升沿；如果从开到关称为下降沿。我们从定义上升沿（从 FALSE 到 TRUE）开始。因此，我们返回到 PLC_PRG POU。

  在 Observer 变量后点击鼠标，则出现一个小正方形。通过快捷菜单（鼠标右键）执行**框**命令，将插入一个带 **AND** 操作符的框，点击选中 AND 后，按 F2（输入助手）打开一个包含可选操作符的对话框，首先选择“标准功能块”项，然后选择 standard.lib 中的 **R_TRIG**（上升沿触发器）。此时一个 R_TRIG 实例被创建，然后把出现在 R_TRIG 框上面的**???**改一个名称(例如 **Trig1**)。 之后无论点击鼠标左键或右箭头键，弹出声明变量对话框现。在**类别**，**名称**，**类型**中已经分别输入 **VAR**（局部变量），**Trig1** 和 **R_TRIG** 。 按确认后变量被写到此 POU 的声明部分。

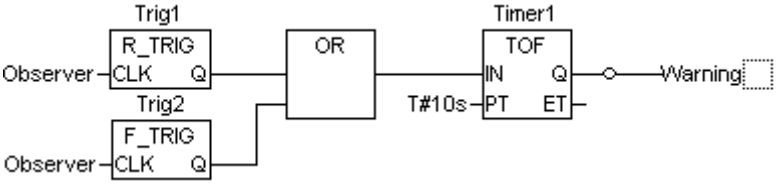


- **确认开关的下降沿**
  在功能块后点击出现小正方形，通过快捷菜单执行**框**命令，将 **AND** 改为 **OR** (逻辑或)；点击 OR 框的第二个输入插入 **F_TRIG**（下降沿触发器）框，声明实例名为 **Trig2**.
  点击 Trig2 功能块前的三个问号**???**， 按**<F2>** 键打开输入助手对话框，在全局变量选项中选择 **Observer**。

- **时间控制, 第一部分**
  在 **OR** 功能块后插入 **TOF** (延时闭合)功能块，命名为 **Timer1**。在 **PT** 输入端将三个**???**替换成 **T#10s** (延时 10 秒，以后可以修改这个时间)。

- **发出 Warning 信息**
  使用快捷菜单在 Time1 功能块的 Q 后面插入**赋值**。将**???**改为 **Warning**。在变量声明中将它设置成类别 **VAR_GLOBAL** 和 **BOOL** 类型。

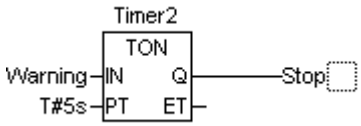
  为了使 warning 正确执行，使用快捷菜单在 warning 前插入**取反**命令，它使布尔型变量的输出取反（即 TRUE 变为 FALSE 或 FALSE 变为 TRUE），**取反**用小圆圈表示。



- **在超出第二个时间限制后设置停止信号**
  用菜单命令**插入->网络(插入在当前行后)**在当前行后插入一个新网络。
  在这个网络中添加类型为 **TON**(延时打开功能块)的框，声明实例名为 **Timer2**。
  使用**<F2>**键将变量 **Warning** 分配给 TON 的 **IN** 输入端, 然后将时间常量 **T#5s** 分配 **PT** 输入端。

  在 Timer2 功能块后面再次使用**赋值**命令，将 TON 的 Q 输出赋值到变量 **Stop (类别 VAR_GLOBAL，类**

型 BOOL)。



- **新建名为 Machine 的 POU**
  在对象管理器（CoDeSys 界面左边区域）中的 POUs 选项页面下，点击鼠标右键执行**添加对象**命令新建一个 POU，命名为 **Machine**, 类型为**程序**，编程语言为 **SFC** (顺序功能图)。
  新建的 SFC 由步"Init"，转换"Trans0"和跳转回"Init"组成。

- **定义机器的运动顺序**
  机器操作的每阶段都需要一**步**。
  点击转换 **Trans0** 后 Trans0 四周出现一个矩形框，借助快捷菜单执行命令**步-转换（插入在当前行后）**。此命令执行 5 次。如果直接点击在步或转换的名称上，它们将用蓝色标记，可以改变它们的名称。在 **Init** 后面的步骤依次命名为 **Go_Right**，**Go_Down**, **Go_Left**, **Go_Up** 和 **Count**.。

- **编写 Go_Right 步中的程序**
  双击 **Go_Right** 步后弹出选择编程语言对话框，选择 **ST** (结构化文本)编程语言，按“确定”后弹出一个程序编辑窗口。

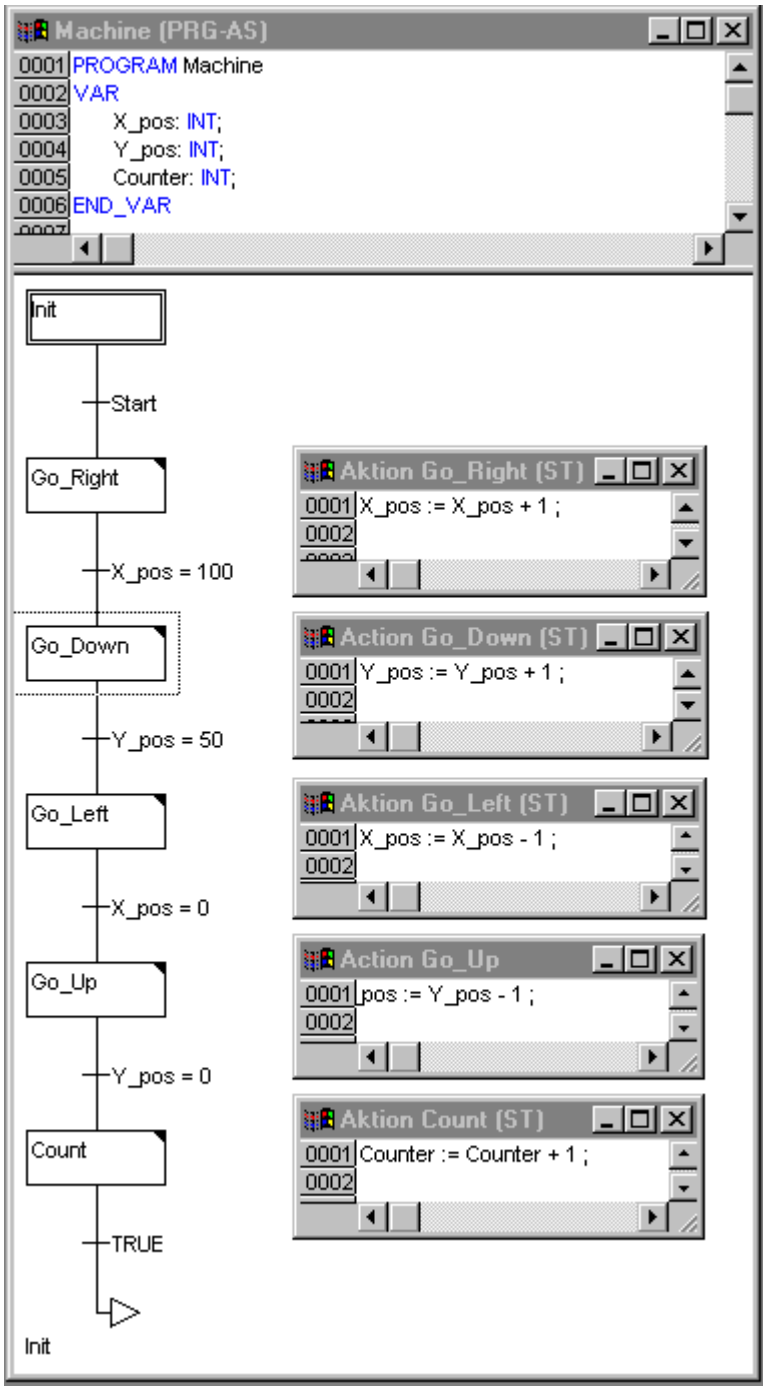
  机器臂沿 X 方向.程序如下：**X_pos := X_pos + 1 ;**

  输入完成后按回车键，声明变量 **X_pos** 的类型为 **INT** （整型）。
  在步的右上角将出现一个小三角，它表明此步中有程序。

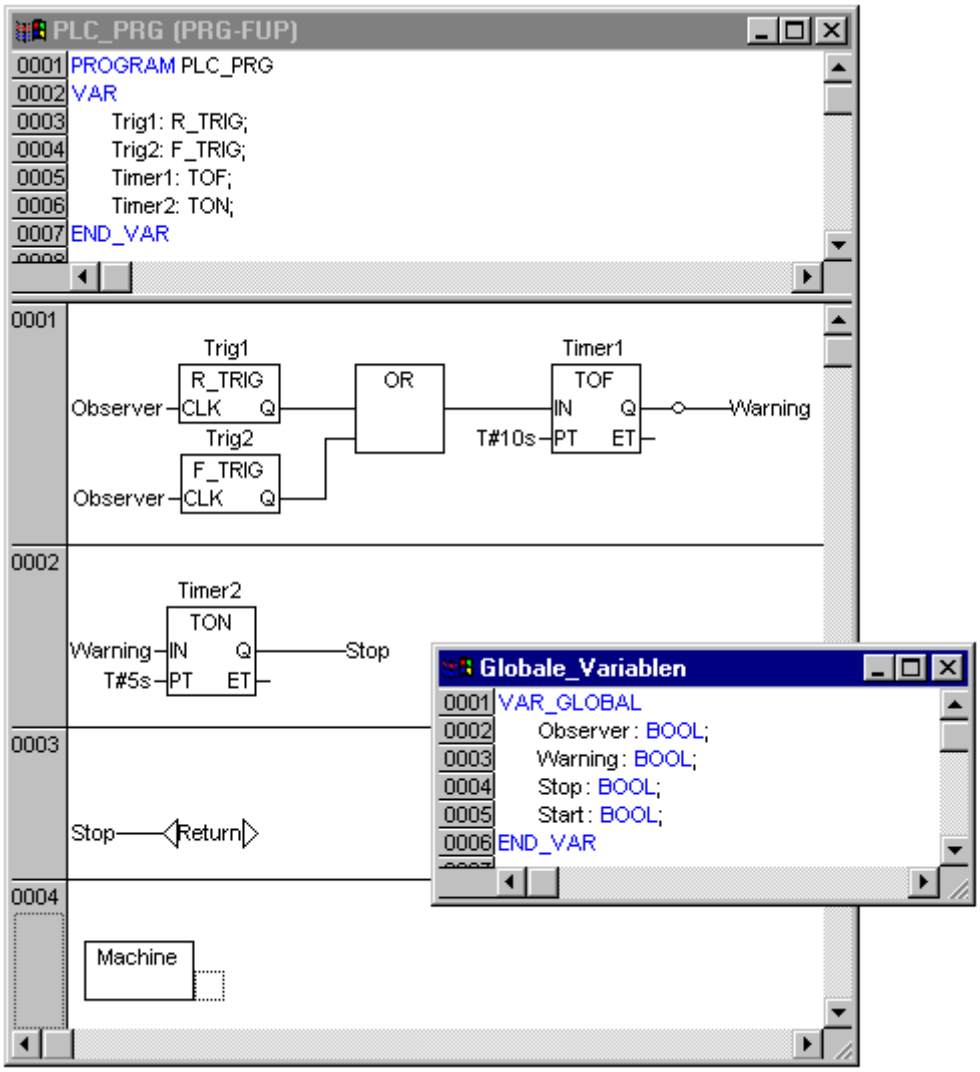
- **编写后续步**
  重复上面的步骤，声明变量 **Y_pos** 和 **Counter** 的类型 为 **INT**。
  在 **Go_Down** 步中程序　　**Y_pos := Y_pos + 1 ;**
  在 **Go_Left** 步中程序　　**X_pos := X_pos - 1 ;**
  在 **Go_Up** 步中程序　　**Y_pos := Y_pos - 1 ;**
  在 Count 步中程序　　Counter := Counter + 1 ;

- **编写转换条件**

  转换条件是程序从一个阶段转到下一个阶段运行的条件。将 Init 后面的转换条件 Tran0 改为变量 **Start**。
  Start 变量的类别是 **VAR_GLOBAL**，类型是 **BOOL**。当 start 开关按下时机器开始工作。
  第二个转换条件为 **X_Pos = 100**，即当 x 位置达到 100 是转到下一个阶段运行。
  第三个转换条件为 **Y_pos = 50**,
  第四个转换条件为 **X_pos = 0**,
  第五个转换条件为 **Y_pos = 0**
  第六个转换条件为 **TRUE** (一次循环结束后继续运行，表示程序循环运行)。



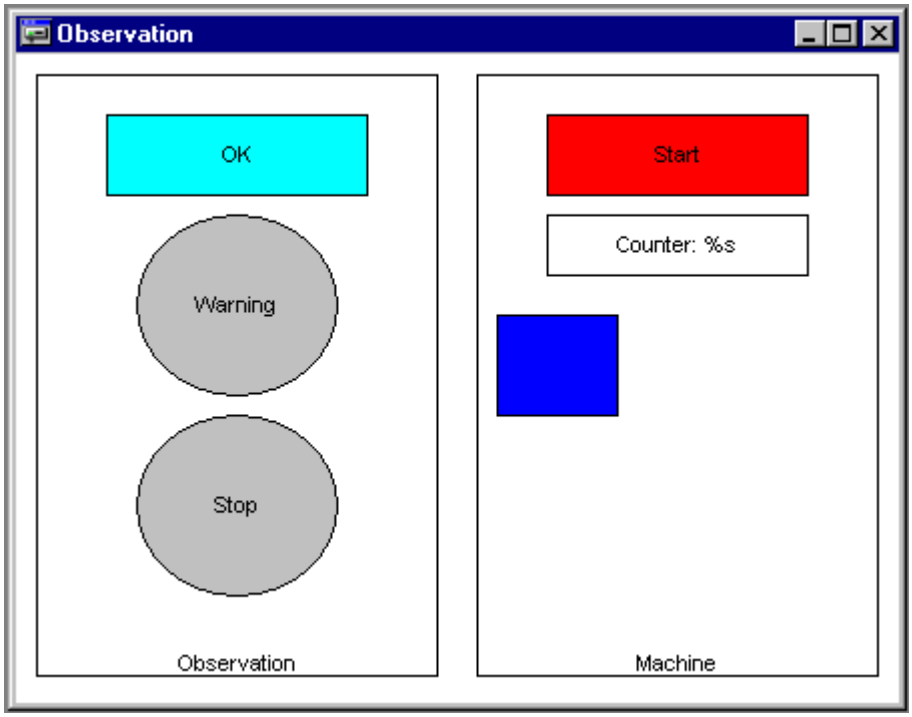
- **在停止时的处理**
  返回到 **PLC_PRG** POU，然后插入第三个网络。
  用变量 **Stop** 替换**???**, 通过快捷菜单插入**返回**命令。当 Stop 为 TRUE 时，执行**返回**命令将退出 PLC_PRG POU。
- **调用 Machine POU**
  添加一个新网络，使用快捷菜单插入一个**框**，按<F2>键打开输入助手对话框，在**用户定义程序**选项中选择 **machine** POU。 完整的程序如下：



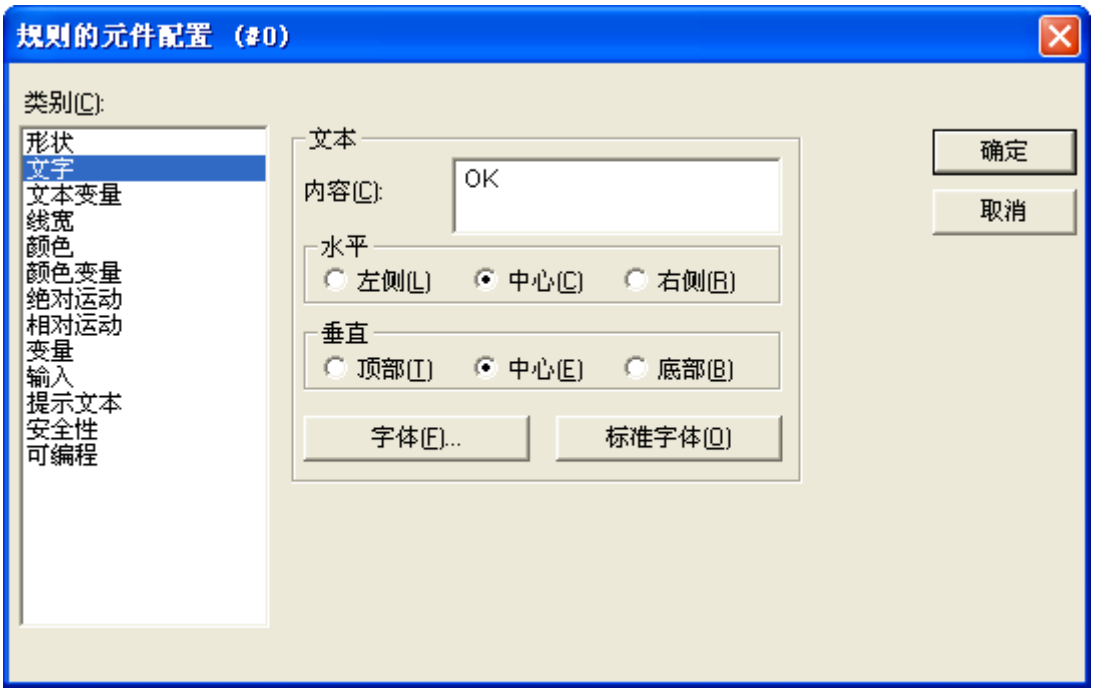
- **编译生成工程**
  使用菜单**工程->全部重新编译生成或<F11>**功能键编译工程。编译生成后在信息窗口的右下角显示**„0 错误 0 警告„**。如果有错误，根据错误提示修改错误。

3 可视化界面

- **创建可视化界面**
  选择对象管理器中左下角第三个（从左边数）页面“可视化界面”。
- 使用对象管理器中的快捷菜单命令**添加对象**。
  给可视化对象命名，如 **Observation**.
  完整的可视化界面如下所示：

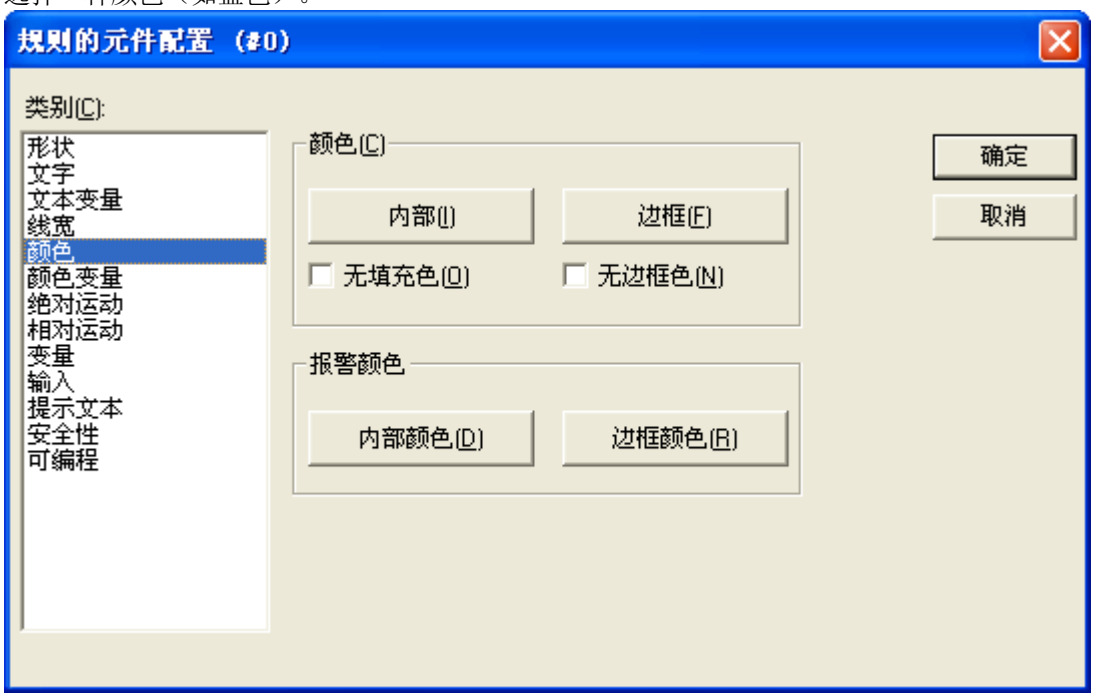


- **添加可视化界面中的元件**
  从确认开关开始设计(上图中带有 OK 的矩形)。
  在工具栏中选择**矩形**元件。
  在可视化编辑器中按住鼠标左键拖拽一个矩形。
- **配置第一个可视化元件**
  在矩形上双击鼠标打开配置对话框。

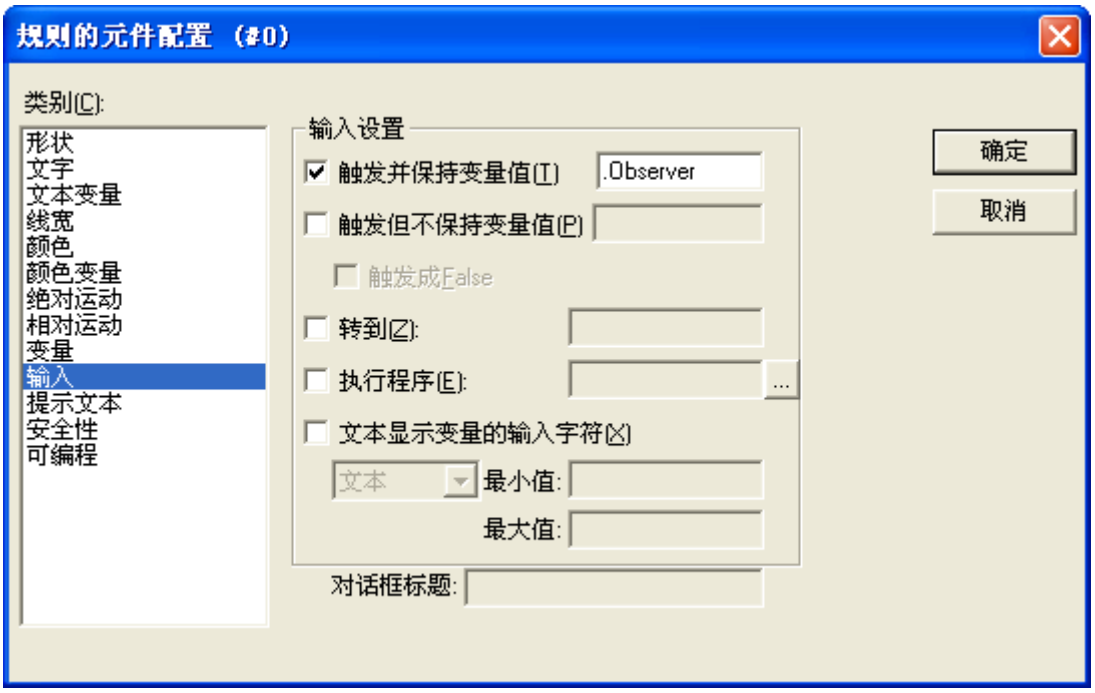


In the 在**文字**选项**内容**字段中输入 **OK**.
选择**变量**选项，在**改变颜色**字段中点击鼠标，然后按**<F2>** 键打开输入助手对话框，在对话框中右侧的 Global_Variables 上双击将列出所有全局变量，选择 Observer 变量，则在字段中显示**.Observer**。

选择**颜色**选项，点击**颜色**下的**内部**按钮选择一种颜色（如，浅兰色）；点击**报警颜色**下的**内部颜色**选择一种颜色（如蓝色）。



在**输入**选项中，选中“**触发并保持变量值**”，使用**<F2>**功能键在后面的输入项中输入变量.Observer 。



经过上述设置，在程序运行过程中当 Observer 变量为 FALSE 时矩形的颜色是浅蓝色；当 Observer 变量为 TRUE 时，矩形的颜色为蓝色。点击一下矩形，Observer 变量从 TRUE 变为 FALSE，再点击一次 Observer 变量从 FALSE 变为 TRUE 。

- **添加其它可视化元件**

画一个圆，作如下配置：
文字选项, **内容**字段中输入 **Warning**。
变量选项, **改变颜色**字段中输入 **.Warning**。
颜色选项，“**颜色**”“**内部**”设置成灰色，“**报警颜色**”“内部颜色”为红色。

复制并粘贴一个新圆，修改下面的配置：
文字选项, **内容**字段中输入 **Stop**。
变量选项, **改变颜色**字段中输入**.Stop**。

画一个矩形，用于机器启动，并作如下配置：
文字选项, **内容**字段中输入 **Start**。
变量选项, **改变颜色**字段中输入**.Start**。
在**输入**选项中，选中“**触发并保持变量值**”，使用**<F2>**功能键在后面的输入项中输入变量**.Start**。
颜色选项，“**颜色**”“**内部**”设置成红色，“**报警颜色**”“内部颜色”为绿色。

画一个矩形，用于计数器，并作如下配置：
文字选项, **内容**字段中输入**: %s** (%s 表示变量值的占位符)
变量选项,**, 文本显示**字段中输入 **Machine.Counter**

画一个矩形，用于表示机器运动，并作如下配置：
绝对运动选项**, X-偏移量**字段中输入 **Machine.X_pos**。
绝对运动选项**, Y-偏移量**字段中输入 **Machine.Y_pos**。
颜色选项，“**颜色**”“**内部**”设置成蓝色。

也可以画两个大矩形框，在**文字**选项，**内容**字段中分别输入 **Observation** 和 **Machine** 。同时选中这两个矩形，使用快捷菜单中的命令“**对齐**”“**底部**”以底部为基准对齐它们，并执行命令“**置于后面**”将它们放在其它元件后面显示。

下面 4,5,6 步只有计算机上安装了运行系统时才被运行。运行系统要与 CoDeSys 中“目标系统设置”中的一致。否则程序只能运行于仿真模式，见第 6 步登录和运行工程。

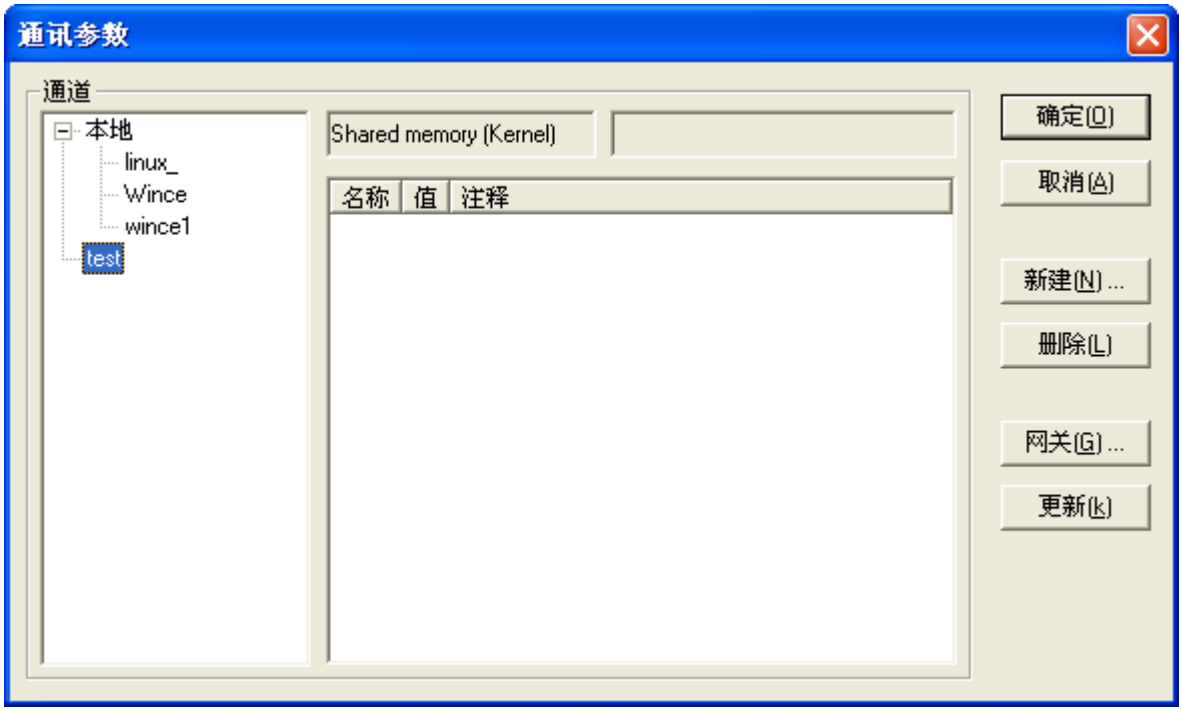
4 启动目标系统

启动目标运行系统。(请注意如果使用 CoDeSys SP RTE 运行系统，只能使用 Windows NT 4.0，Windows 2000 或 Windows XP 操统。) 执行“开始 -> 所有程序 -> 3S Software -> CoDeSys SP RTE -> Start CoDeSys SP RTE”，在状态栏中会出现 CoDeSys SP RTE 图标，鼠标移动到图标处点击鼠标右键，弹出命令菜单，点击命令“启动系统”。

5 进行连接设置

- 在与目标计算机建立连接前，必须进行一些设置。

  执行命令 **联机 -> 通讯参数.**，弹出下面对话框：



- 点击**新建**按钮配置与目标系统的连接。在新的对话框中选择一种连接方式并输入名字。在 CoDeSys SP RTE 下选择 **Shared memory (Kernel)**.
  如果目标计算机就是本机，那么点击**确认**关闭对话框。如果目标计算机不是本机，而是局域网中的其他计算机，必须将'localhost'替换成目标计算机的 IP 地址或目标计算机名。设置完成点击**确认**关闭此对话框。

6 运行工程

- 通过命令**联机->登录**建立 **CoDeSys** 开发环境与运行系统（目标计算机）的连接；

- 执行**联机-> 运行**，程序将在运行系统（目标计算机）上运行。（如果在仿真模式下运行，激活'联机' '仿真模式' 选项）。
- 也可以利用可视化界面启动机器并操作确认开关。

7 从这儿继续

- 现在您可以使用 CoDeSys 更多功能了。更多的信息请参看在线帮助和 CoDeSys 编程手册。
- 我们祝您成功！