First Steps with CoDeSys

Last update: 05.03.2004

Smart
Software
Solutions

CONTENT

1 STARTING CODESYS 3

2 WRITING THE FIRST PROGRAM 3

3 A VISUALIZATION FOR THIS 7

4 START THE TARGET SYSTEM 9

5 SETTINGS FOR ESTABLISHING THE CONNECTION 9

6 START THE PROJECT 9

7 CONTINUING FROM HERE 10

1 Starting CoDeSys

Start the CoDeSys programming system with
Start -> Programs -> 3S Software -> CoDeSys V2.3 -> CoDeSys V2.3

2 Writing the First Program

- **Task:**
  A running machine is to be observed by a machine operator. Correct running must be confirmed at specified intervals. If there is no confirmation, a warning is first issued, shortly after which the machine will be stopped.
  The machine moves an arm along a rectangular path, and completed circuits are to be counted.

- **Creating a New Project**
  Starting is easy. Create a new project with the command **File -> New**.

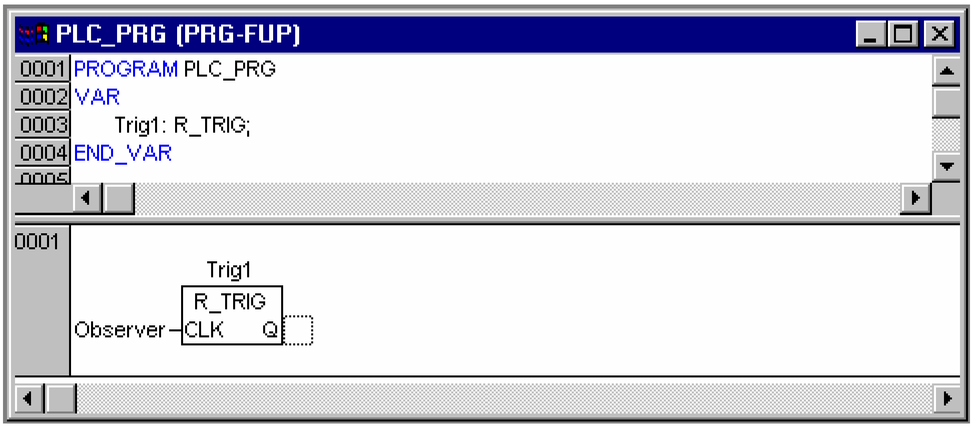
- **Target Settings**
  For our first project please select the entry which fits to your runtime system as target in the combo box ‘Configuration’, e.g. CoDeSys SP for Windows NT Realtime

- **The PLC_PRG POU**
  In the **New POU** dialog select the **language of the POU FBD** (Function Block Diagram) and retain the settings **Type Program** and **Name PLC_PRG**.
  PLC_PRG is a special POU which is cyclically called and executed by the run-time system in operation.

- **Declaration of the Confirmation Switch**
  We will start with the confirmation switch. Mark the three question marks **???** in the first network, and type in the name of the switch (e.g. **Observer**). Now press the right-arrow key.
  The dialog which appears for the declaration of variables contains the **Name Observer** and the **Type BOOL**. Alter the **Class** to **VAR_GLOBAL** (for global variables). The variable is entered in the object for global variables with **OK**. The following text appears there:

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

- **Rising Edge of the Confirmation Switch**
  Everything is OK if the operator is there and confirms correct sequence of operation. That is, if the switch changes from the off state (FALSE) to the on state (TRUE), known as the rising edge, or from on to off, called the falling edge. We will start by determining the rising edge, i.e. whether the switch changes from FALSE to TRUE.
  For this, we return to the PLC_PRG POU. Mark the location behind the Observer variable, so that a rectangular marking appears. Execute the **Box** command through the quick menu (right mouse button).
  A box for the **AND** operator is inserted and the name is marked. Now call the input assistant with F2. You get a dialog, which offers all available operators. Select first the **Standard Function Blocks** category and there the trigger function block **R_TRIG**, which belongs to the standard.lib. R_TRIG reports the rising edges of boolean value, which in this case is our switch.
  An instance (data copy) of R_TRIG must be created. To do this, replace the three question marks **???** which appear on top of the inserted box by the instance name (e.g. **Trig1**). After that either click with the left mouse button behind the function block or press the right-arrow key.
  The **Declare variables dialog** will appear, where the **Class VAR** (for local variables), the **Name Trig1** and the **Type R_TRIG** are already entered. Press **OK** to enter the variable into the declaration segment of the POU.

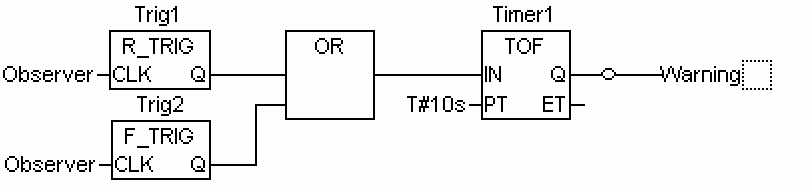


```
PROGRAM PLC_PRG
VAR
    Trig1: R_TRIG;
END_VAR
```

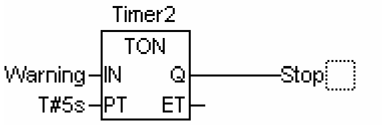
- **Falling Edge of the Confirmation Switch**
  Mark the location behind the function block (rectangular marking), execute the **Box** command via the quick menu, and alter the designation **AND** to **OR** (logical or).
  Mark the line of the second input of the OR-operator and insert a **Box** of **Type F_TRIG** for the falling edge, having the instance **Name Trig2**.
  Mark the three question marks in front of the Trig2 function block, press the **<F2>** function key (input assistance) and select the **Observer** variable in the **Global Variables** category of the dialog which appears.

- **Time Monitoring, First Part**
  Insert a **Box** of **Type TOF** (switch-off delay) with the **Name Timer1** behind the OR-operator using the quick menu.
  Replace the three question marks at the **PT** input with the time constant **T#10s** for 10 seconds. (The time can later be increased, after successful tests.)

- **Issue Warning**
  Insert an **Assignment** behind the Q output of the Timer1 function block using the quick menu. Replace the three question marks with the variable name **Warning**. In the declaration of variables, enter the **Class VAR_GLOBAL** and the type BOOL.
  So that the warning is indeed properly given, mark the first position on the assignment and execute the **Negate** command via the quick menu. This causes the boolean signal to be inverted, and is displayed by means of a small circle.



- **Set the Stop Signal After Exceeding the Second Time Limit**
  Create another network with the menu command **Insert->Network (after)**.
  Use the quick menu to add a **Box** of **Type TON** (function_block**,** switch-on delay) with the **Name Timer2** into the second network.
  Allocate the variable **Warning** to the **IN** input with the aid of the **<F2>** key, and the time constant **T#5s** to the **PT** input.
  After the Timer2 function block we need an **Assignment** again, this time to the variable **Stop (Class VAR_GLOBAL)**.



- **Insert POU for the Machine**
  Insert another POU with the **Name Machine**, of **Type program** and with the **Language SFC** (Sequential Function Chart) into the object organizer (the left hand area of CoDeSys), under

POUs, using the quick menu command **Add object**.
The initial diagram of a POU in SFC consists of a step "Init", and of a subsequent transition "Trans0" followed by a jump back to Init.

- **Specify the Machine's Sequence**
  Each phase in the operation of our machine requires a step.
  Mark the transition **Trans0** in such a way that a dotted frame appears round Trans0, and use the quick menu to select the command **Step-Transition (after)**. Repeat this procedure five times. If you click directly on the name of a transition or of a step it will be marked in blue and can be altered.
  The step after the **Init** step should be called **Go_Right**, and the following steps **Go_Down**, **Go_Left**, **Go_Up** and **Count**.

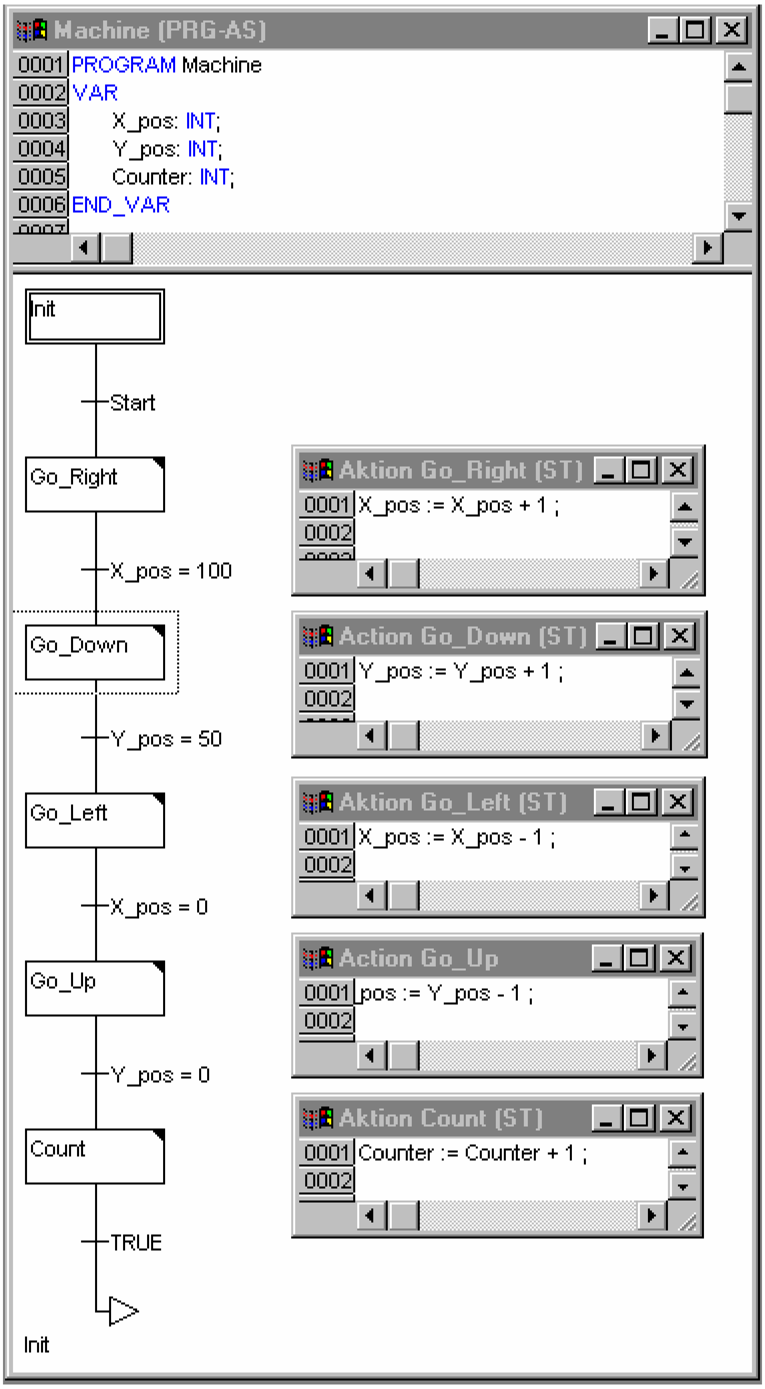
- **Programming the First Step**
  After a double click on the **Go_Right** step you will be asked about the desired **Language**. Select **ST** (structured text) and a separate window appears for the action associated with this step. The machine arm should be moved in the X-direction. The program line reads: **X_pos := X_pos + 1 ;**

  Complete the line with a return, and declare the variable **X_pos** having **Type INT** for integer.
  A small triangle appears in the upper left hand corner of the step to indicate that this step has been programmed.

- **Programming the Next Steps**
  Repeat this procedure for the other steps with altered program lines, and as you do so declare the variables **Y_pos** and **Counter** of **Type INT**.

In the **Go_Down** step	**Y_pos := Y_pos + 1 ;**
In the **Go_Left** step	**X_pos := X_pos - 1 ;**
In the **Go_Up** step	**Y_pos := Y_pos - 1 ;**
In the **Count** step	**Counter := Counter + 1 ;**

- **Specify Transitions**
  Transitions are the conditions for progressing from one phase to the next. Overwrite the first transition after Init with the variable **Start**. In the declaration of variables enter the **Class VAR_GLOBAL** and the **Type BOOL**. The machine thus starts to run when the start switch is pressed.
  The second transition contains the condition **X_Pos = 100**, so that when the X-position has reached the value 100 the next phase starts.
  The third contains the condition **Y_pos = 50**,
  the fourth contains the condition **X_pos = 0**,
  the fifth contains the condition **Y_pos = 0 and**
  the sixth contains the condition **TRUE** (continue immediately after a single execution)



Machine (PRG-AS)

```
PROGRAM Machine
VAR
	X_pos: INT;
	Y_pos: INT;
	Counter: INT;
END_VAR
```

Aktion Go_Right (ST)

```
X_pos := X_pos + 1 ;
```

Action Go_Down (ST)

```
Y_pos := Y_pos + 1 ;
```

Aktion Go_Left (ST)

```
X_pos := X_pos - 1 ;
```

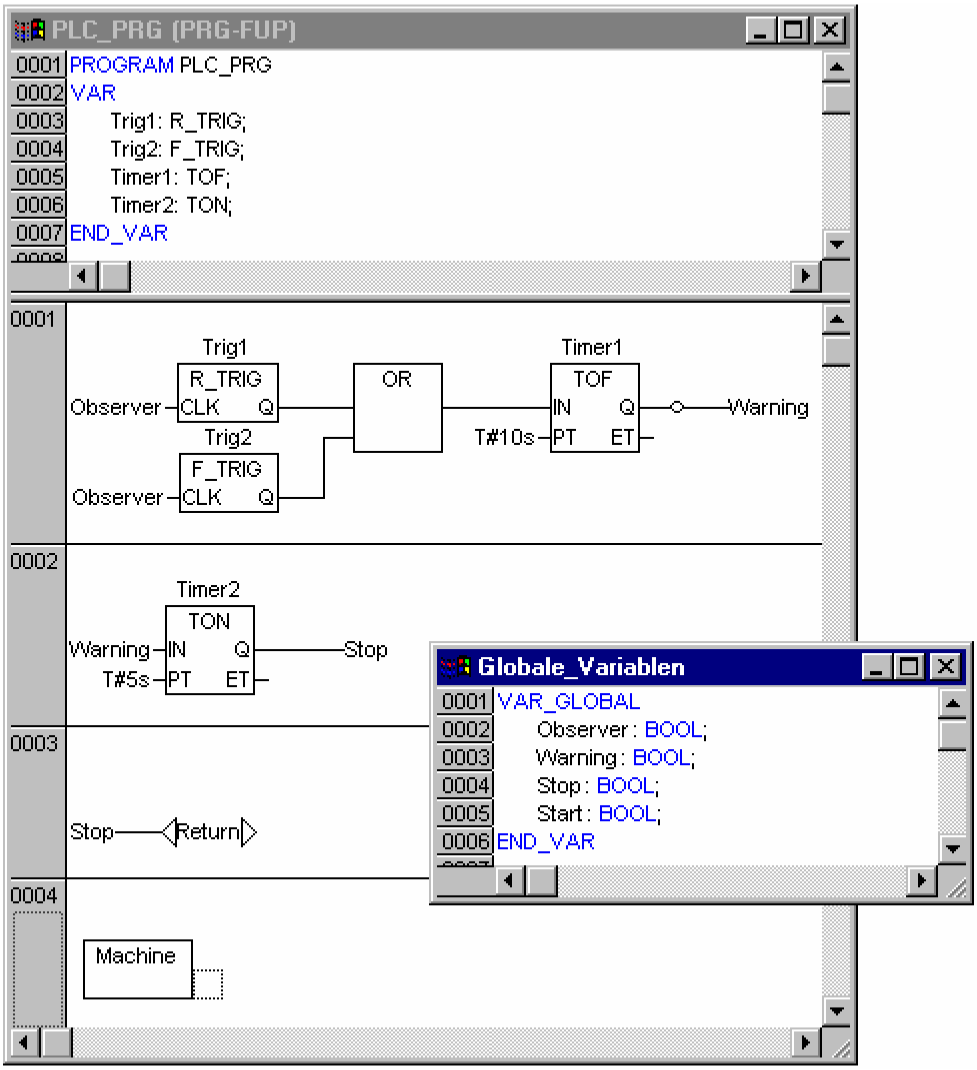
Action Go_Up

```
pos := Y_pos - 1 ;
```

Aktion Count (ST)

```
Counter := Counter + 1 ;
```

- **Halt Processing at Stop**
  Go back to the **PLC_PRG** POU and insert a third network.
  Replace the question marks by the variable **Stop**, and after that select the **Return** command from the context menu. The Return has the effect of leaving the PLC_PRG POU when Stop is set.
- **Call Machine POU**
  Insert another network, set the focus there and execute in the command **Box** from the context menu. The "AND" in the box is marked already. Press <F2> and in the input assistant dialog which appears, select the **machine** POU in the **User defined Programs category**. This completes the programming.



PLC_PRG (PRG-FUP)

```
PROGRAM PLC_PRG
VAR
    Trig1: R_TRIG;
    Trig2: F_TRIG;
    Timer1: TOF;
    Timer2: TON;
END_VAR
```

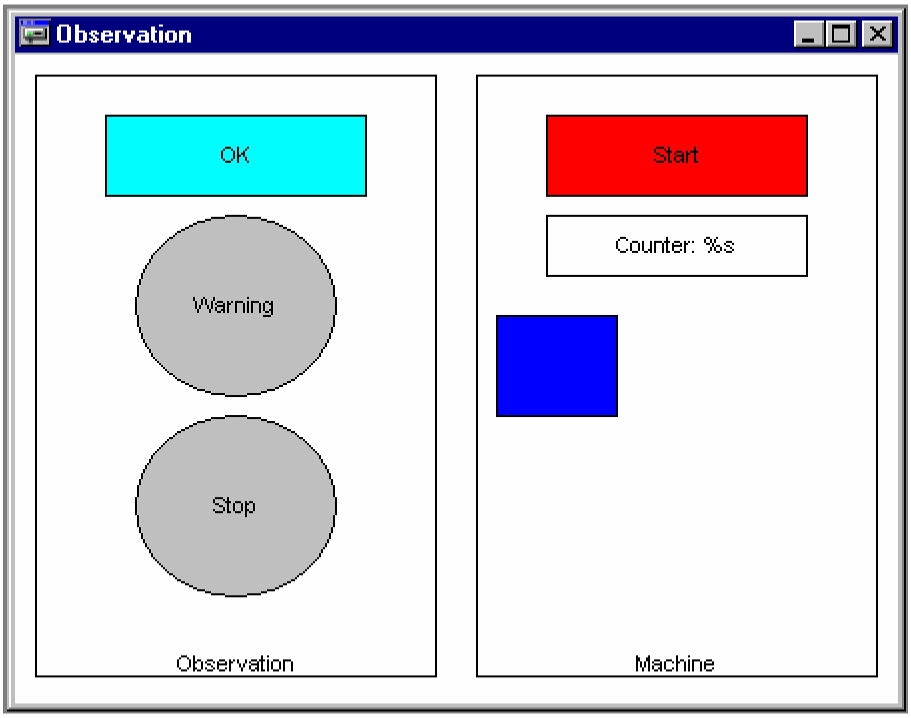
Globale_Variablen

```
VAR_GLOBAL
    Observer : BOOL;
    Warning : BOOL;
    Stop : BOOL;
    Start : BOOL;
END_VAR
```

- **Build Project**
  Build the project with the menu command **Project->Rebuild all**, or the **<F11>** function key. In the message window at the bottom right, **„0 errors„** should be reported after the build procedure. If not, check that all the entries have been correctly made. The reported error may also help.

3 A Visualization for This

- **Create Visualization**
  In the object organizer at the bottom left select the third tab from the left, named Visualization.
- Use the object organizer's quick menu to call the **Add object** command.
  Give the visualization object a **Name**, such as **Observation**.
  The visualization should finally look like this:



- **Draw Visualization Element**
  Let's start with the confirmation switch (the rectangle with the text OK in the figure). Select the symbol for **Rectangles** from the toolbar.
  Press the left hand mouse key in the visualization editor, hold it down, and draw a rectangle with it. Release the mouse key when it has reached the desired form and size.

- **Configure the First Visualization Element**
  With a double click on the rectangle, the configuration dialog will appear. In the **Contents** field of the **Text Category** write the text **OK**.
  Select the **Variables Category**, click in the **Change Color** field and call up the input assistance with the **<F2>** function key. Give a double click to the point in the right hand half of the dialog box, and the global variable will appear. Select the Observer variable. The field now contains **.Observer**

  After this, select the **Colors Category** in the configuration dialog, and click on the **Inside** button in the **Color** area. Select a color (e.g. light blue). Then select another color with the **Inside** button in the **Alarm color** area (e.g. blue).
  In the **Input Category** select the **Toggle variable** point and again enter the variable Observer into the field behind it with **<F2>**. Leave the configuration dialog.
  This causes the rectangle to be colored light blue during running operation when the Observer variable is in the FALSE state, and blue when it is in the TRUE state. With each mouse click the state is changed from TRUE to FALSE, or the other way around.

- **Expand Visualization**
  Draw a circle for the warning, and configure it with the following values:
  Text Category, **Contents** field with **Warning**
  Variable Category, **Color change** field with **.Warning**
  Colors Category, **Color** area using **Inside** with color grey, and **Alarm color** with red.

  Copy the circle you have created using the menu command **Edit -> Copy** and insert another one with the menu command **Edit -> Paste**. Alter the following values in the new circle:
  Text Category, Contents field with the new entry **Stop**

  Variable Category, Color change field with the new entry **.Stop**

  Draw a rectangle for the start of the machine and configure it with the following values:
  Text Category, Contents field with **Start**

Variable Category, Color change field with **.Start**
Input Category, select the point **Toggle variable**
Input Category, behind **Toggle variable,** enter **.Start** in the field
Colors Category, Color area using **Inside** with color red, and **Alarm color** with green.

Draw a rectangle for the counter and configure it with the following values:
Text Category, Contents field with **Counter : %s**
(%s functions as a place holder for the variable value)
Variable Category, Textdisplay field with **Machine.Counter**

Draw a rectangle for the machine and configure it with the following values:
Absolute movement Category, X-Offset field with **Machine.X_pos**
Absolute movement Category, Y-Offset field with **Machine.Y_pos**
Colors Category, Color area using **Inside** with color blue.

If you like, you can draw two large rectangles around the two areas for observation and for the machine, enter **Observation** (or **Machine**) as the **Contents** in the **Text Category,** and select the **Vertical** alignment **bottom**. Using the quick menu command **Send to back** place the rectangle behind the existing elements.

The following steps 4,5,6 only can be executed, if a runtime system has been installe, which fits to the target settings in CoDeSys. Otherwise you have the possibility to work in simulation mode. In this case continue now with Step 6 for logging in and starting the project.

4 Start the Target System

Start the target runtime system. (Please regard that in case of using the runtime system CoDeSys SP RTE this is only possible on Windows NT 4.0, Windows 2000 or Windows XP. Here you will get the icon of CoDeSys SP RTE in the status bar. Click on this with the right mouse button and choose the command 'Start System'.)

5 Settings for Establishing the Connection

- A small number of settings must be made before we can establish the connection to the target computer.
  For this purpose carry out the menu command **Online -> Communication parameters**.
- Select the button **New** to configure the connection to your target system. Enter an appropriate name for the connection at the new dialog and select a connection; in case of CoDeSys SP RTE **Shared memory (Kernel)**.
  If the target computer is your local computer you can now close the dialog with **OK.** If the target computer is not your local computer, you have to change the value 'localhost' to the Internet address or the target name of the target computer. After that also close the communications parameters dialog with **OK**.

6 Start the Project

- The connection to the target computer will be established using the menu command **Online -> Login** from the **CoDeSys** development environment, and
- with **Online -> Run** your program will be started on the target computer. (If you want to work in simulation mode, the option 'Online' 'Simulation' must be activated).
- By means of the visualization you can now start the machine and operate the confirmation switch.

7 Continuing From Here

- It is now up to you to create and use further applications. For further information please see the Online Help and the Users Manual for the CoDeSys Programming System.
  We wish you every success!