Learn-/Training Document

Siemens Automation Cooperates with Education (SCE) | As of Version V9 SP1

PA Module P01-06
SIMATIC PCS 7 – Control loop and other control functions

siemens.com/sce

SIEMENS
Global Industry Partner of WorldSkills International

worldskills

Matching SCE Trainer Packages for this Learn-/Training Document

- **SIMATIC PCS 7 Software Package V9.0 (set of 3)**
  Order No.: 6ES7650-0XX58-0YS5
- **SIMATIC PCS 7 Software Package V9.0 (set of 6)**
  Order No.: 6ES7650-0XX58-2YS5
- **SIMATIC PCS 7 Software Upgrade Packages (set of 3)**
  Order No.: 6ES7650-0XX58-0YE5 (V8.x→ V9.0)
- **SIMIT Simulation Platform with Dongle V10**
  (contains SIMIT S & CTE, FLOWNET, CONTEC libraries) – 2500 simulation tags
  Order No.: 6DL8913-0AK00-0AS5
- **Upgrade SIMIT Simulation Platform V10**
  (contains SIMIT S & CTE, FLOWNET, CONTEC libraries) from V8.x/V9.x
  Order No.: 6DL8913-0AK00-0AS6
- **Demo Version SIMIT Simulation Platform V10**
  Download
- **SIMATIC PCS 7 AS RTX Box (PROFIBUS) only in combination with ET 200M for RTX –**
  Order No.: 6ES7654-0UE23-0XS1
- **ET 200M for RTX Box (PROFIBUS) only in combination with PCS 7 AS RTX Box –**
  Order No.: 6ES7153-2BA10-4AB1

Note that these trainer packages are replaced with successor packages when necessary.
An overview of the currently available SCE packages is available at: siemens.com/sce/tp

Continued training

For regional Siemens SCE continued training, get in touch with your regional SCE contact
siemens.com/sce/contact

Additional information regarding SCE

siemens.com/sce

Information regarding use

The SCE Learn-/Training Document for the integrated automation solution Totally Integrated Automation (TIA) was prepared for the program "Siemens Automation Cooperates with Education (SCE)" specifically for training purposes for public educational facilities and R&D institutions. Siemens does not guarantee the contents.

This document is to be used only for initial training on Siemens products/systems, which means it can be copied in whole or part and given to those being trained for use within the scope of their training. Circulation or copying this Learn-/Training Document and sharing its content is permitted within public training and advanced training facilities for training purposes.

Exceptions require written consent from the Siemens. Send all related requests to scesupportfinder.i-ia@siemens.com.

Offenders will be held liable. All rights including translation are reserved, particularly if a patent is granted or a utility model or design is registered.

We wish to thank the TU Dresden, particularly Prof. Dr.-Ing. Leon Urbas and the Michael Dziallas Engineering Corporation and all other involved persons for their support during the preparation of this Learn-/Training Document.

Table of contents

Control loop and other control functions

1 Goal

In this chapter, students become familiar with the essential components and requirements for a block for continuous closed-loop control of process tags and can create and configure a temperature control using the PIDConL and PULSEGEN blocks.

2 Prerequisite

This chapter builds on chapter 'Functional safety'. To implement this chapter, you can use an existing project from the previous chapter or the archived project 'p01-05-exercise-r1905-en.zip' provided by SCE. The download of the project(s) is stored on the SCE Internet for the respective module.

The (optional) simulation for the SIMIT program can be retrieved from the file 'p01-04-plantsim-v10-r1905-en.simarc'. It can be run in Demo mode.

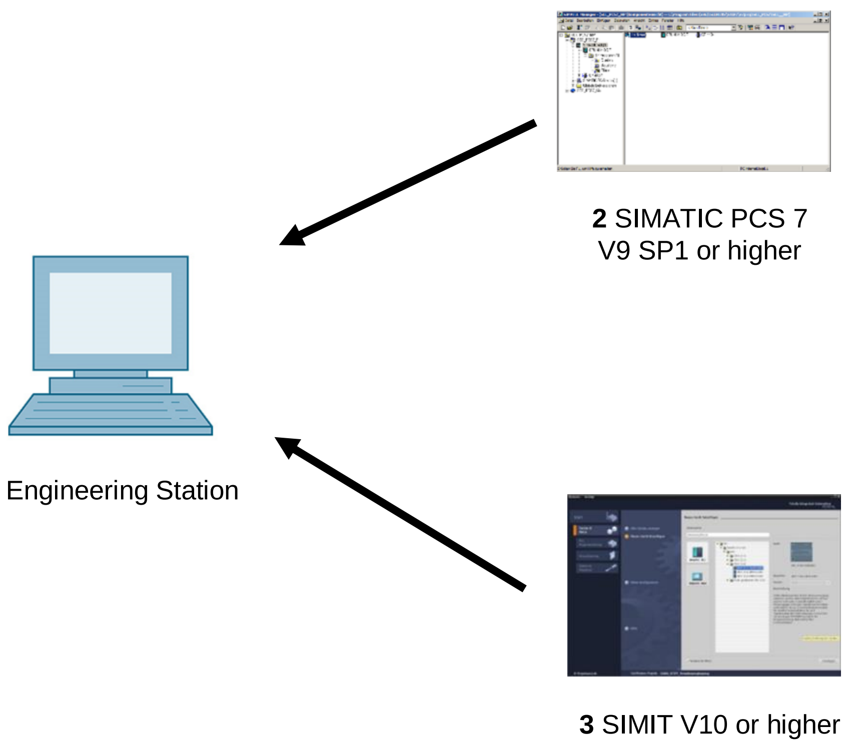
3 Required hardware and software

1 Engineering station: Requirements include hardware and operating system (for further information, see Readme on the PCS 7 installation DVD)

2 SIMATIC PCS 7 software V9 SP1 or higher

- Installed program packages (contained in SIMATIC PCS 7 Software Trainer Package):
  - *Engineering → PCS 7 Engineering*
  - *Engineering → BATCH Engineering*
  - *Runtime → Single Station → OS Single Station*
  - *Runtime → Single Station → BATCH Single Station*
  - *Options → SIMATIC Logon*
  - *Options → S7-PLCSIM V5.4 SP8*

3 Demo Version SIMIT Simulation Platform V10



4 Theory

4.1 Theory in brief

In the process industry, certain process tags must be kept at a specific value despite disturbances (***disturbance response***) and process tags must be adjusted to specified setpoints in a stable manner (***response to setpoint change***). Control loops, as shown in Figure 1, are used for this.

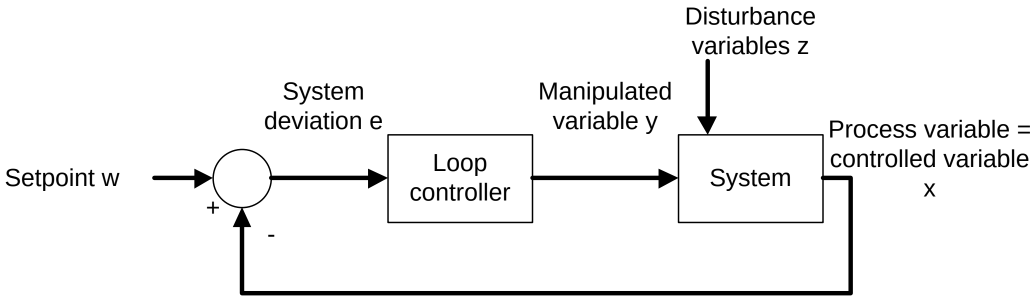


Figure 1: Control loop

For the plant used in these training curriculums, the reactor temperature is set to a certain value for controlling a reaction according to specification. Disturbance variables are the ambient temperature and the raw materials with different temperatures. In order for the temperature to be controlled, it first has to be measured. This measured value, which corresponds to the ***actual value*** of the process tag, is then compared with the desired value (***setpoint***). The difference between the actual value and setpoint is called the ***(control) deviation***.

When the control deviation is known, it is possible to identify counter measures. In the case of temperature control, the heater is switched on when the measured actual value is less than the specified setpoint. For the process to react automatically, a loop controller is needed. A loop controller that calculates the manipulated variable based solely on the current deviation is called a proportional controller (or P controller for short).

In practice, so-called ***PID controllers***, which can be used for a wide variety of processes with the help of just a few parameters, have prevailed.

The ***PCS 7 Advanced Process Library V90*** contains proven blocks that implement this functionality. The PIDConL block will be used in the following.

4.2 Introduction

The above-mentioned P controller represents the simplest loop controller. It operates according to the principle: the more the actual value differs from the setpoint, the larger the manipulated variable will be. Its behavior is thus derived directly from the control deviation at the given moment, which makes it fast and relatively dynamically favorable. However, certain disturbances are not fully compensated for, which means there is a sustained control deviation.

Not every process tolerates a sustained control deviation so that additional measures have to be taken. One possibility consists of adding an integral-action component, which turns the P controller into a PI controller. The effect of the integral-action component is that a sustained control deviation is added up. This means that manipulated variable becomes larger and larger despite the constant control deviation.

If abrupt disturbances occur in a system, they can be counteracted quickly with an additional derivative-action component. The derivative-action component calculates the manipulated variable from the time derivative of the control deviation. However, this behavior also leads to an increase of stochastic disturbances (noise). It is necessary to strike a balance here.

A controller that combines proportional, integral-action and derivative-action components is called a PID controller. In the process industry, 95% of applications are implemented with a PID controller, because it is tuned with only three parameters (gain, integral-action time and derivative-action time). These few parameters allow for a good adaptation to a many different dynamic processes.

However, tuning the parameters requires knowledge of the system that is being controlled. Knowledge of the system can be gained by experience, determined experimentally or calculated by modeling the system. For a wide variety of processes that are not dominated by dead times and that respond similarly to positive and negative changes of the manipulated variable interventions, several practical tuning rules have been identified. Examples are the tuning rules according to Chien, Hrones and Reswick [1], the Ziegler and Nichols method [2] and the T-sum rule [3].

The process control system ***PCS 7*** supports the tuning of parameters using a ***PID Tuner***.

In the controller block PIDConL, the parameters for the gain, integral-action component and derivative-action component are called GAIN, TI and TD, respectively. The times must be specified in seconds in each case. The input variables of the controller are the process tag PV and the setpoint SP, which yield the control deviation ER. The manipulated variable MV is the output variable for the controlled system; it is calculated according to the following formula:

$$MV = GAIN \cdot \left(1 + \frac{1}{TI \cdot s} + \frac{TD \cdot s}{1 + \frac{TD}{DiffGain} \cdot s}\right) \cdot ER$$

4.3 Suitability of loop controllers in industry

For a loop controller to also work in everyday industrial applications, additional functions have to be implemented. These include above all:

– Bumpless changeover

– Anti-reset windup

– Support of different closed-loop control structures

The purpose of bumpless changeover is to prevent an abrupt change of the manipulated variable on a changeover between manual and automatic mode or between an internal and external setpoint setting or when a parameter changes. A bumpless changeover between manual and automatic mode is required, for example, when a process in the process industry runs semi-automatically, i.e. startup is performed in manual mode and a switch is then made to automatic mode during regular operation. In manual mode, the operator specifies the manipulated variable directly; in automatic mode, the control algorithm calculates the manipulated variable.

The purpose of the anti-reset windup (ARW) function is to prevent the integral-action component (reset) of the manipulated variable from continuing to increase (figuratively: windup), because control deviation cannot be corrected due to the manipulated variable limitation, for example.

The support of different closed-loop control structures enables optimization of the loop controller without having to replace the controller. Some of these closed-loop control structures are explained in greater detail in section 'Expanded closed-loop control structures'. With PIDConL from the ***SIMATIC PCS 7 Advanced Process Library V90***, the following closed-loop control structures can be implemented:

– Fixed setpoint control

– Cascade control

– Ratio control

– Feedforward control

– Split-range control

– Smith predicator control

– Override control

4.4 Expanded closed-loop control structures

In some applications, single-loop control loops are inadequate, so that expanded closed-loop control structures must be used to reach the desired goal.

If the response to setpoint changes and the disturbance response is not simultaneously optimized satisfactorily for a process tag, a feedforward control/auxiliary value injection or a cascade control can be used.

If the disturbance is measured and its point of application is known, there is the option to apply feedforward compensation for disturbances at the controller input or controller output. ***Feedforward control*** can be used to fully compensate the disturbance variable, so that the controller can be tuned for optimal response to setpoint changes.

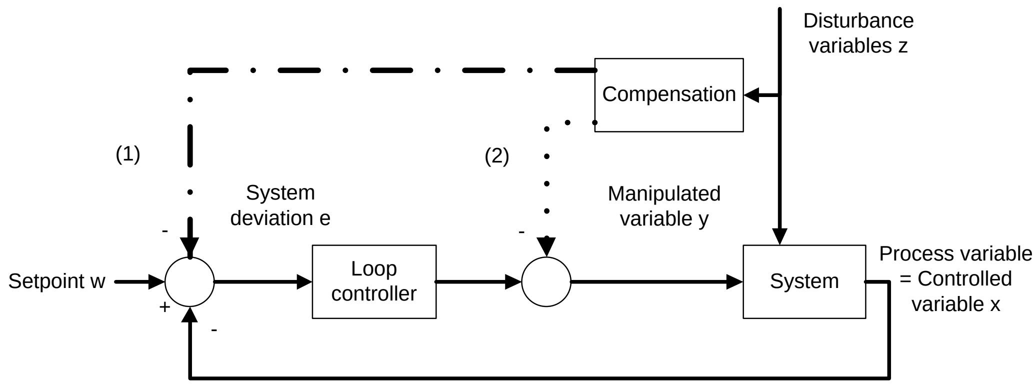


Figure 2: Feedforward control at the controller input (1) or controller output (2)

If the disturbance variable cannot be measured, but another variable in the system can be measured in its place, this auxiliary variable is fed to the controller input with a loop controller. The ***auxiliary variable injection*** reduces the effect of the disturbance variable but does not compensate for it completely.

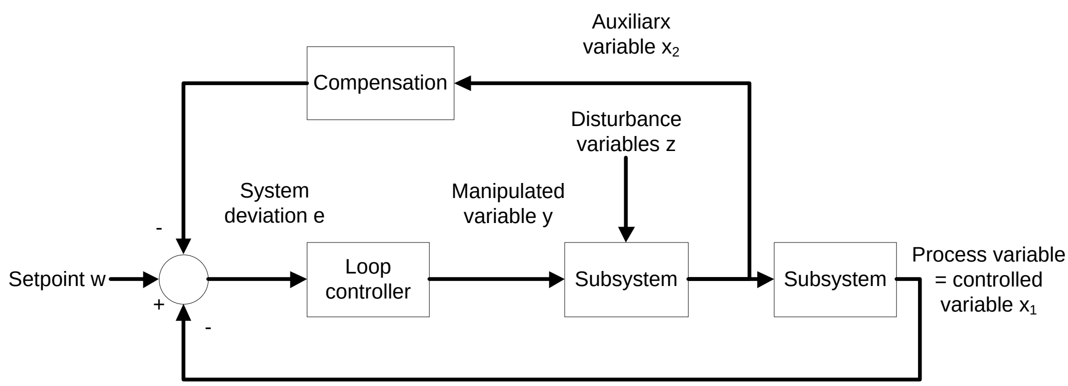


Figure 3: Auxiliary variable injection

If the auxiliary variable is injected at the controller input, the compensation and the loop controller do not act independently of each other. This means that if the controller parameters are adjusted, the compensation has to be adjusted as well.

If the feedforward control and auxiliary value injection are inadequate, the point of application of the disturbance variables cannot be determined with sufficient accuracy or the subsystems cannot be modeled with sufficient accuracy, a two-loop or multi-loop ***cascade control*** is used.

When designing a cascade control, it is assumed that the lower-level control loops (Loop controller 2 in Figure 4 – a so-called follower controller) respond faster in each case than the higher-level control loops (Loop controller 1 in Figure 4 – a so-called master controller). The closed-loop control system is thus always optimized from the inside out.

The cascade control reduces the effect of the disturbance variable and speeds up control of the setpoint. In order for the cascade control to be used, measurable variables appropriate for this type of control must exist.

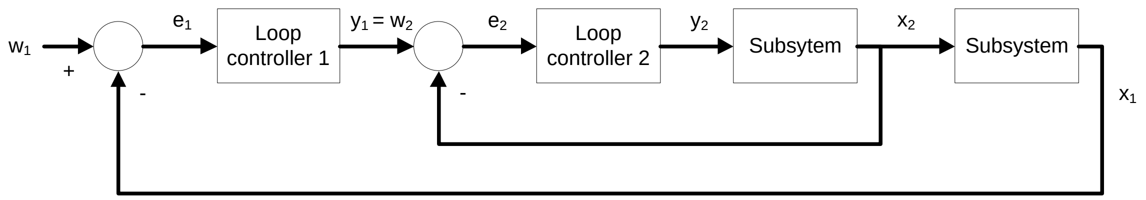


Figure 4: Cascade control with two loops

Ratio control is used if the process tag is determined in dependence on another variable, e.g. ratio control of two liquid flows that are to be mixed. This means closed-loop control of the composition of the mixture or ratio control of combustion gas and fresh air in a gas burner for optimal combustion. The setpoint of the process tag w_2 is calculated from the ratio V_w and the process tag x_1.

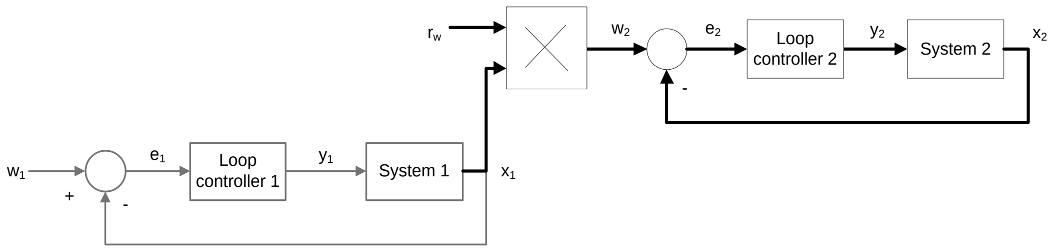


Figure 5: Ratio control

4.5 Interfacing to processes

The continuous output signal of the loop controller is not always output directly to the process. Especially when large forces or flows are involved, this direct output is not advisable, and a binary interface is therefore implemented. To this end, the analog signal is converted into a binary signal by means of ***pulse width modulation***. The elementary block PULSEGEN [4] is available in the ***CFC Library*** for this.

The PULSEGEN function uses pulse width modulation to transform the input variable INV (= LMN manipulated variable of the PID controller) into a pulse train with a constant period. It corresponds to the cycle time used to update the input variable and must be assigned in PER_TM.

The duration of a pulse per period is proportional to the input variable. The cycle assigned with PER_TM is not identical to the processing cycle of the PULSEGEN function block. As shown in Figure 6, a PER_TM cycle ② consists of several processing cycles ① of the function block PULSEGEN. In this context, the number of PULSEGEN calls per PER_TM cycle provides a measure of the precision of the pulse width modulation.

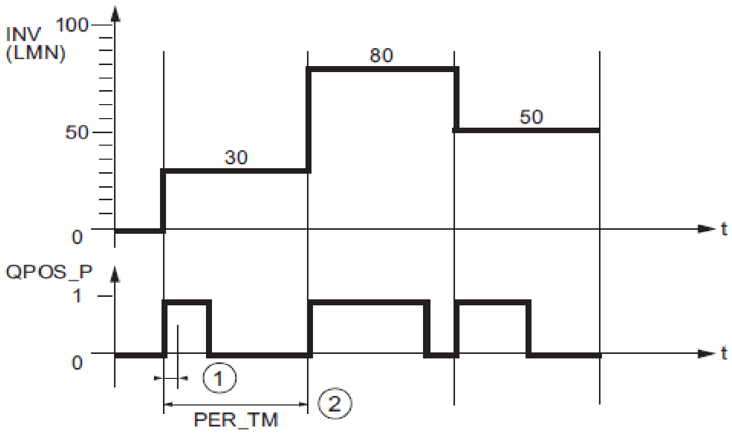


Figure 6: Timing diagram of input INV and output QPOS_P of PULSEGEN [4]

An input variable of 30% with 10 PULSEGEN calls per PER_TM means the following:

– 1 at output QPOS for the first three calls of PULSEGEN (30% of 10 calls)

– 0 at output QPOS for the remaining seven calls of PULSEGEN (70% of 10 calls)

The pulse width is recalculated at the beginning of each period. Through a sampling ratio of 1:10 (CTRL_PID calls to PULSEGEN calls), the manipulated value precision is limited in this example to 10%. Specified input values INV can only be mapped to a pulse width at the QPOS output in a grid of 10%. Correspondingly, the precision increases with the number of PULSEGEN calls per PIDConL call. If PULSEGEN is called 100 times and PIDConL only once, a resolution of 1% of the manipulated variable range is achieved.

Note:

– *You must program the reduction ratio of the call frequency yourself.*

4.6 References

[1] Chien, Kun Li; Hrones, J. A.; Reswick, J. B. (1952): On the Automatic Control of Generalized Passive Systems. In: Transactions of the American Society of Mechanical Engineers, Vol. 74, Cambridge (Mass.), pg. 175-185.

[2] Ziegler, J. G. and Nichols, N. B (1942): Optimum settings for automatic controllers. In: Trans. ASME, 64, pg. 759-768.

[3] Kuhn, U.: Eine praxisnahe Einstellregel für PID-Regler: Die T-Summen-Regel. Automatisierungstechnische Praxis, Nr. 5, 1995, pg. 10-16.

[4] SIEMENS (2017): Process Control System PCS 7: CFC Elementary Blocks. A5E41367308-AB. (support.automation.siemens.com/WW/view/en/109755019)

5 Task

In accordance with the specifications in chapter 'Process Description', the temperature control and the associated manual control of Reactor R001 is to be added to the CFCs from chapter 'Functional safety'. The heating of the reactor will be implemented using a PID controller with downstream pulse generator.

The following CFCs will be created here:

– A1T2H008 (manual local control for heating reactor R001)

– A1T2T001 (temperature control reactor R001)

When implementing the temperature control, the following interlock conditions must be taken into account in the CFC.

– An actuator may only be operated when the main switch of the plant is switched on and the Emergency Stop switch is unlocked.

– The temperatures in the two reactors must not exceed 60 °C.

– The heaters of the two reactors may only be started up if they are covered with liquid (here: a minimum of 200 ml in the reactor).

6 Planning

The manual control A1T2H008 (see Figure 8) for controlling the heating consists of three parts:

– A digital input for the Start command: 'A1.T2.A1T2H008.HS+.START' / I7.0

– A digital input for the Stop command: 'A1.T2.A1T2H008.HS-.STOP' / I7.1

– A digital output for the status feedback: 'A1.T2.A1T2H008.HO+-.0+' / Q4.1

An analog measured value is available for the temperature measurement:

– 'A1.T2.A1T2T001.TIC.M' / IW76 / actual temperature value R001

A digital output is available for the heating control:

– 'A1.T2.A1T2T001.TV.S' / Q4.

The signals are already contained in the symbol table and only still have to be linked.

The analog measured value must still be scaled to an actual temperature. For this, the user specifies the high limit 100°C and low limit 0°C of the measured value.

The manual control influences the A1T2T001 temperature control (see also Figure 7), which must be expanded accordingly.

The interlocks mentioned in the task description can all be implemented with previously created sensors and actuators.

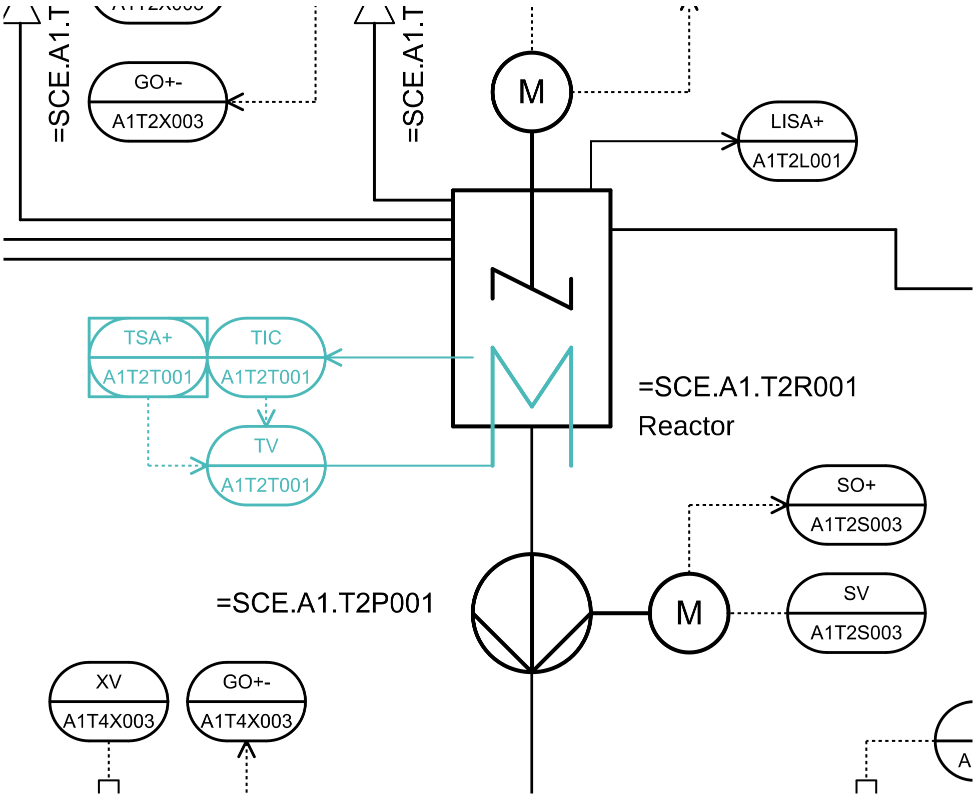


Figure 7: Portion of the P&ID flow chart to be programmed

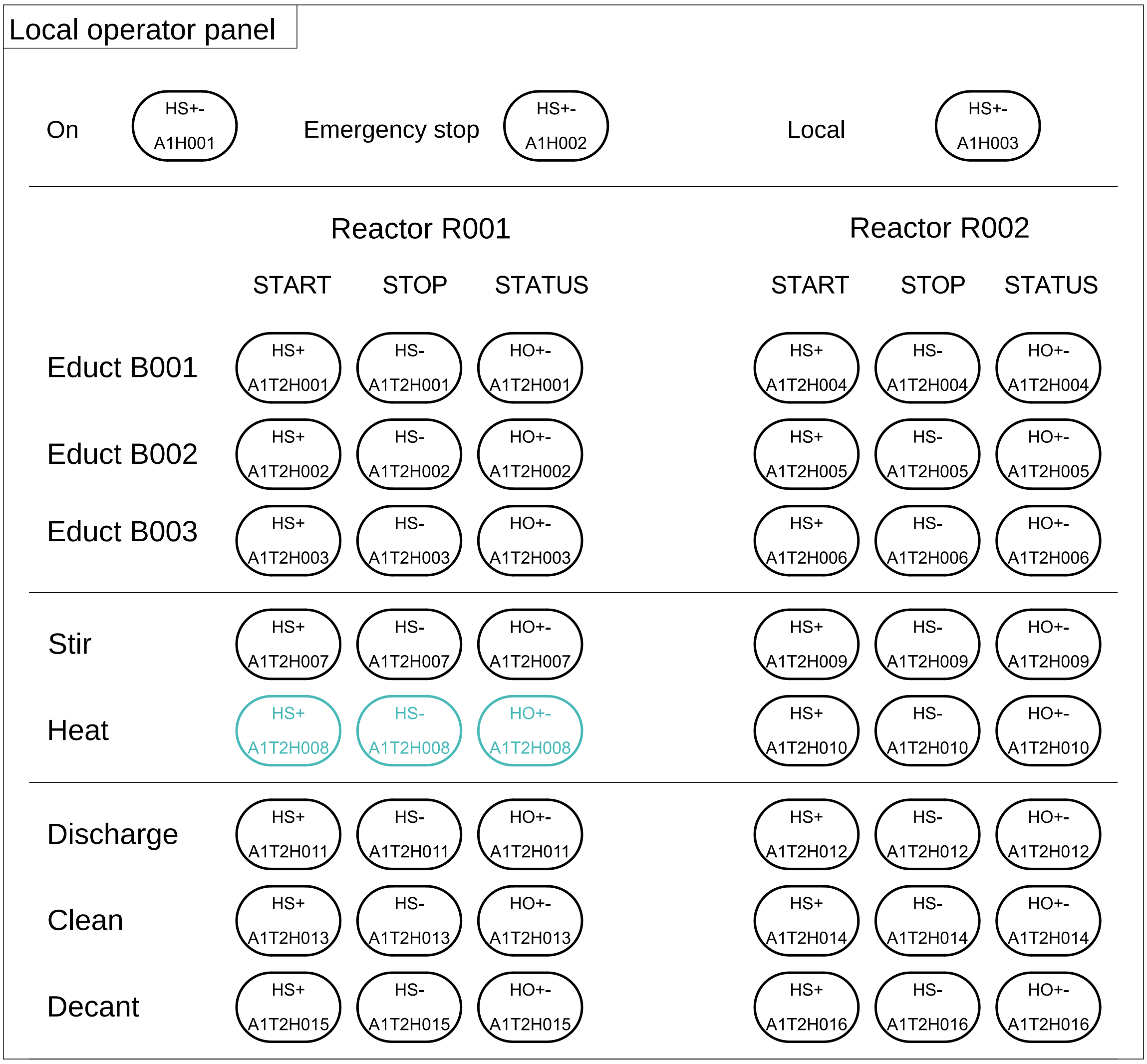


Figure 8: Local operator station

7 Learning objective

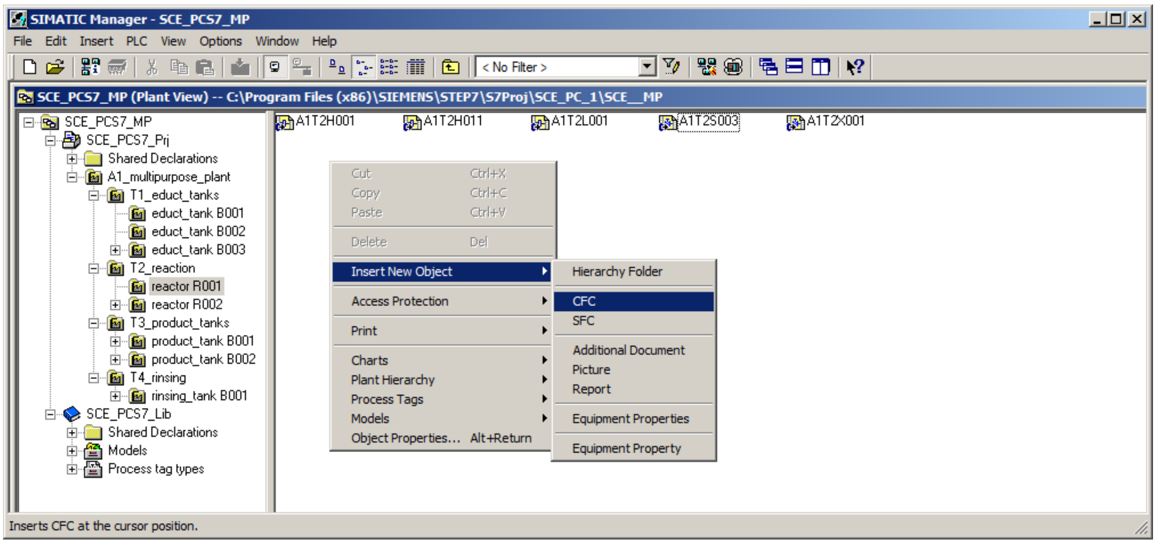
In this chapter, students gain the following:

– Knowledge of how to program a continuous loop controller with pulse output and interlocks

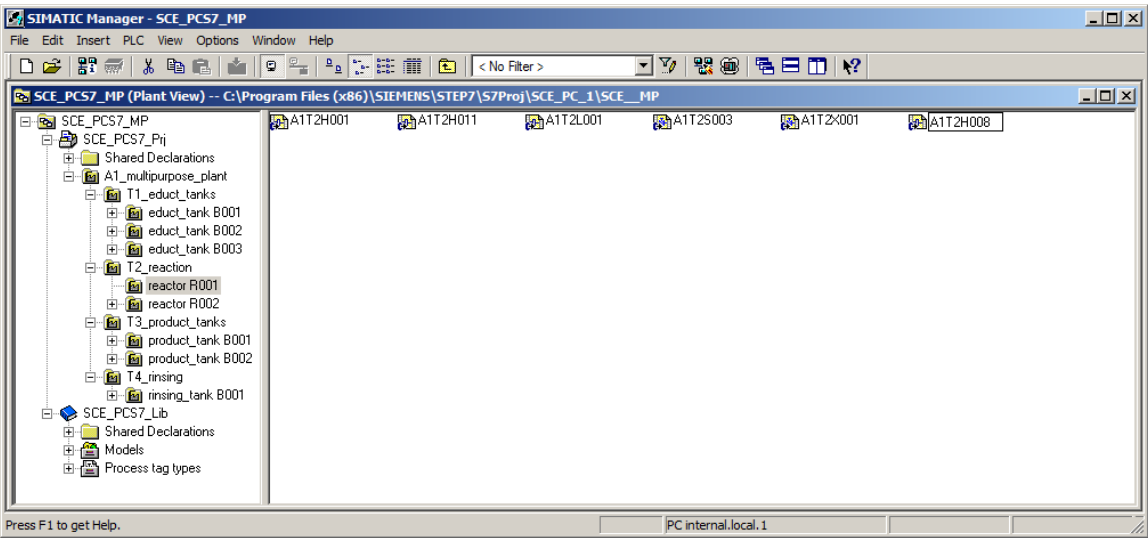
8 Structured step-by-step instructions

8.1 Creating the manual control A1T2H008

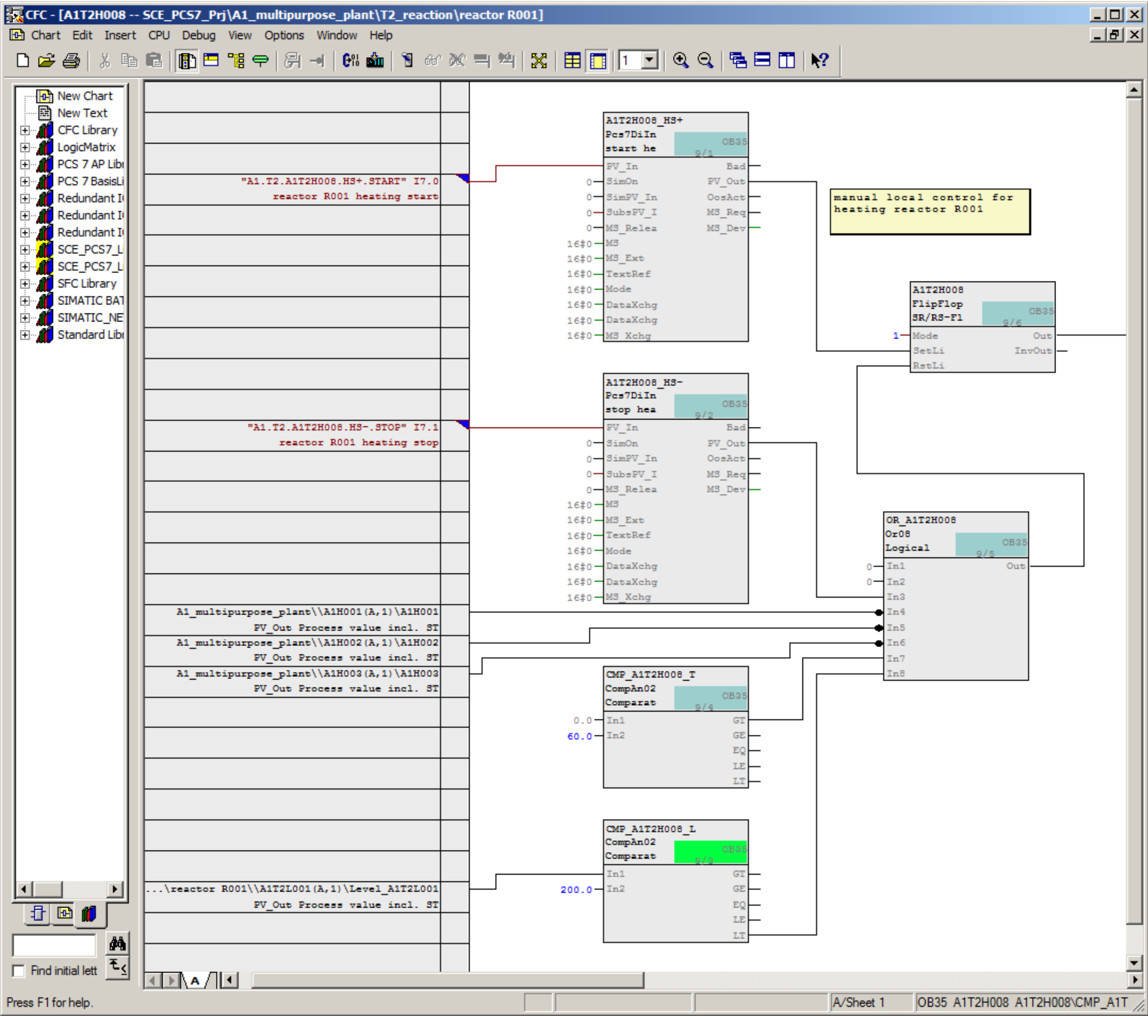
1. First, insert a new CFC in the reactor R001 folder. You will implement the manual control for the heating in this folder.

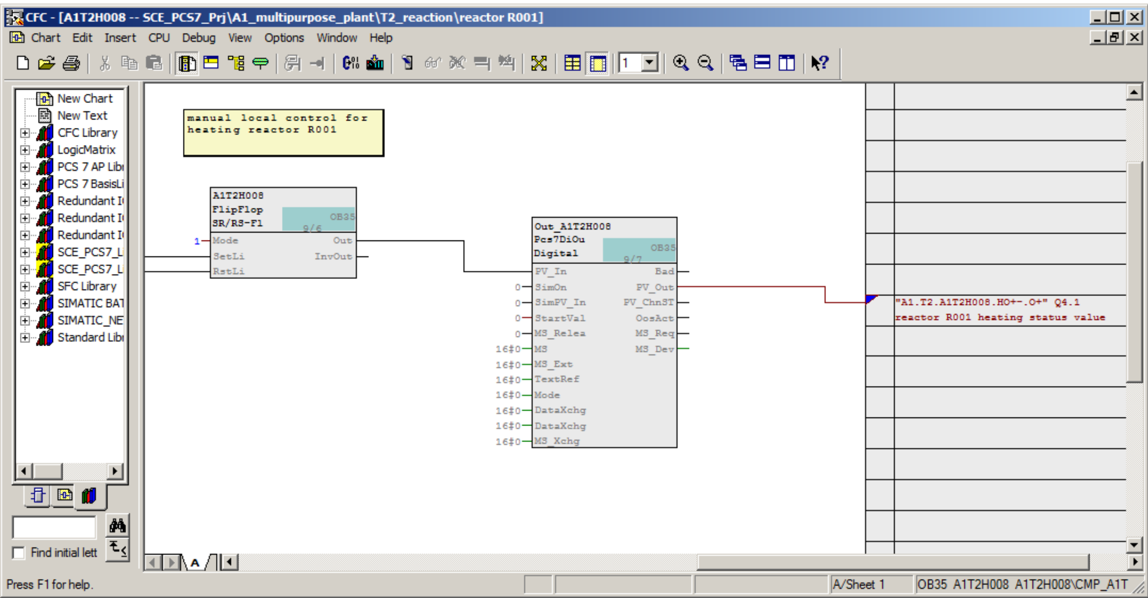


2. The newly created chart is renamed A1T2H008.



3. The interconnections of A1T2H008 differ from those of A1T2H011 only in the input and output signals (Pcs7DiIn and Pcs7DiOu) and in the last two reset conditions (block 'Or08'). The conditions relate to the minimum level of 200.0 ml and the maximum temperature of 60.0 °C.





Block:	Catalog/Folder:
Pcs7DiIn (2x)	Blocks/Channel
Or08	Blocks/LogicDi
CompAn02 (2x)	Blocks/LogicAn
FlipFlop	Blocks/LogicDi
Pcs7DiOu	Blocks/Channel

Table 1: New blocks in chart 'A1T2H008'

Input:	Interconnection to:	Inverted
Pcs7DiIn.HS+.PV_In	'A1.T2.A1T2H008.HS+.START' / I7.0 / reactor R001 heating start	No
Pcs7DiIn.HS-.PV_In	'A1.T2.A1T2H008.HS-.STOP' / I7.1 / reactor R001 heating stop	No
Or08.In4	A1H001(A,1) / A1H001 PV_Out Process value incl. ST	Yes
Or08.In5	A1H002(A,1) / A1H002 PV_Out Process value incl. ST	Yes
Or08.In6	A1H003(A,1) / A1H003 PV_Out Process value incl. ST	Yes
CompAn02.T.In2	60.0	
CompAn02.L.In1	A1T2L001(A,1) / Level_A1T2L001 PV_Out Process value incl. ST	
CompAn02.L.In2	200.0	
FlipFlop.Mode	1	

Table 2: Input interconnections in chart 'A1T2H008'

Input:	Output:	Inverted
FlipFlop.SetLi	Pcs7DiIn.HS+.PV_Out	No
FlipFlop.RstLi	Or08.Out	No
Or08.In3	Pcs7DiIn.HS-.PV_Out	No
Or08.In7	CompAn02.T.GT	No
Or08.In8	CompAn02.L.LT	No
Pcs7DiOu.PV_In	FlipFlop.Out	No

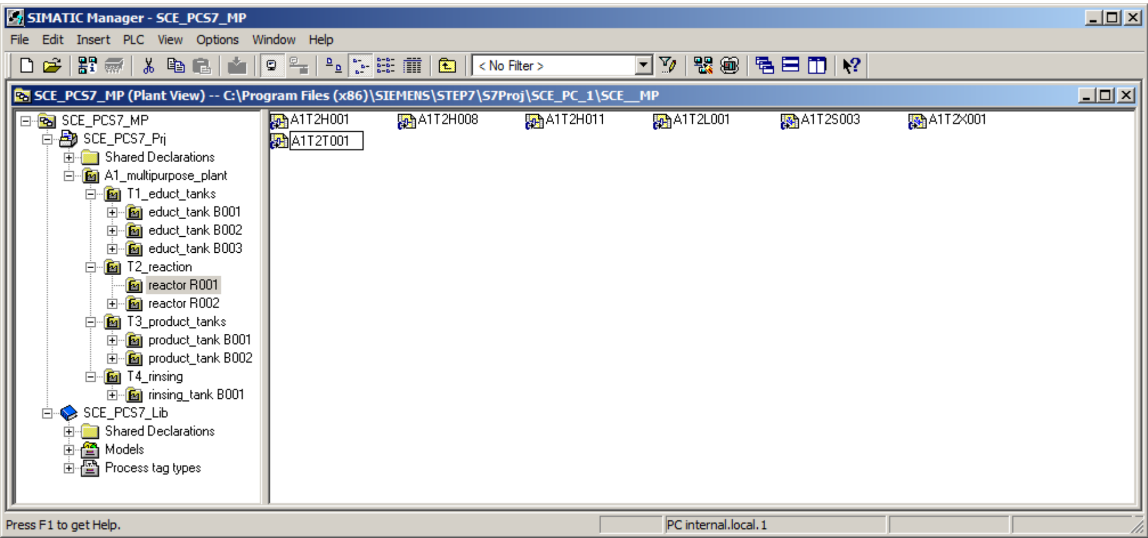
Table 3: Block interconnections in chart 'A1T2H008'

Output:	Interconnection to:	Inverted
Pcs7DiOu.PV_Out	'A1.T2.A1T2H008.HO+-.0+' / Q4.1 / reactor R001 heating status value	No

Table 4: Output interconnections in chart 'A1T2H008'

8.2 Creating the A1T2T001 temperature control

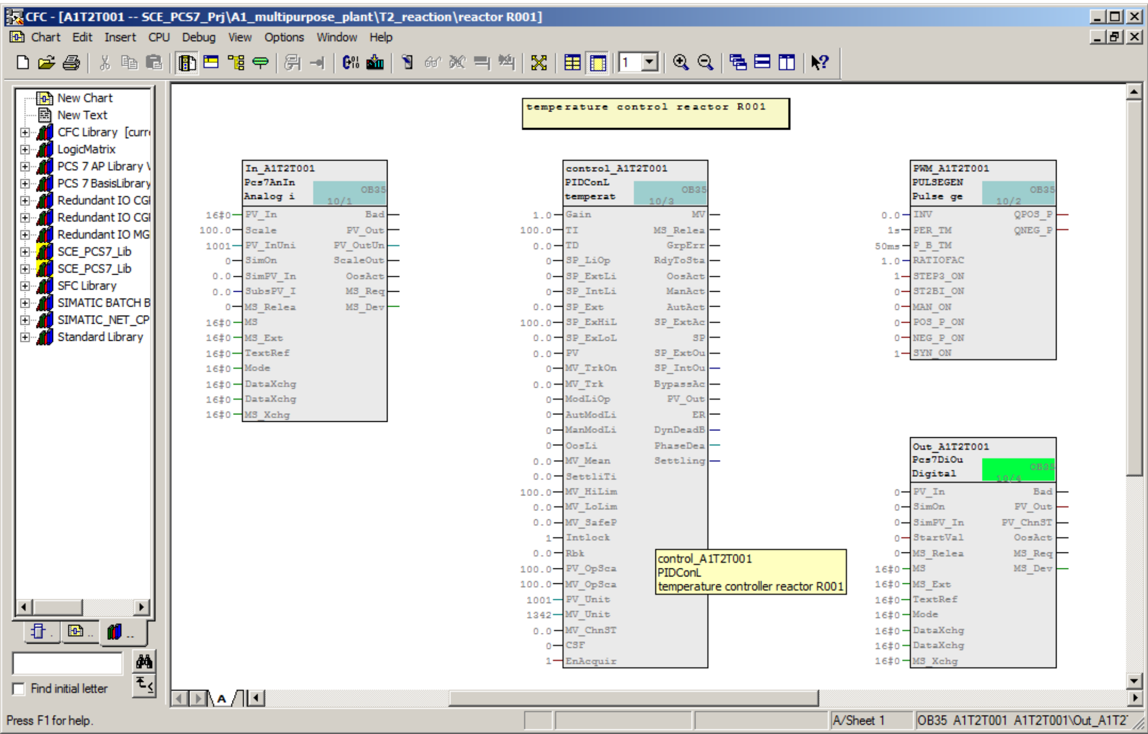
1. First, a new CFC named 'A1T2T001' is created. In this chart, you will implement the actual closed-loop control of the heating of reactor R001.



2. Add the following blocks and name them appropriately.

Block:	Catalog/Folder:
Pcs7AnIn	Blocks/Channel
PIDConL	Libraries/PCS7 APL V90/ Blocks + Templates\ Blocks/Control
PULSEGEN	Libraries/CFC Library/ELEM400\Blocks/CONTROL
Pcs7DiOu	Blocks/Channel

Table 5: New blocks in chart 'A1T2T001'



3. Next, implement the basic interconnections as shown in the table below. Compare your result with the figure.

Input:	Interconnection to:	Inverted
Pcs7AnIn.PV_In	'A1.T2.A1T2T001.TIC.M' / IW76 / actual temperature value R001	
Pcs7AnIn.Scale	High value = 100.0, Low value = 0.0	

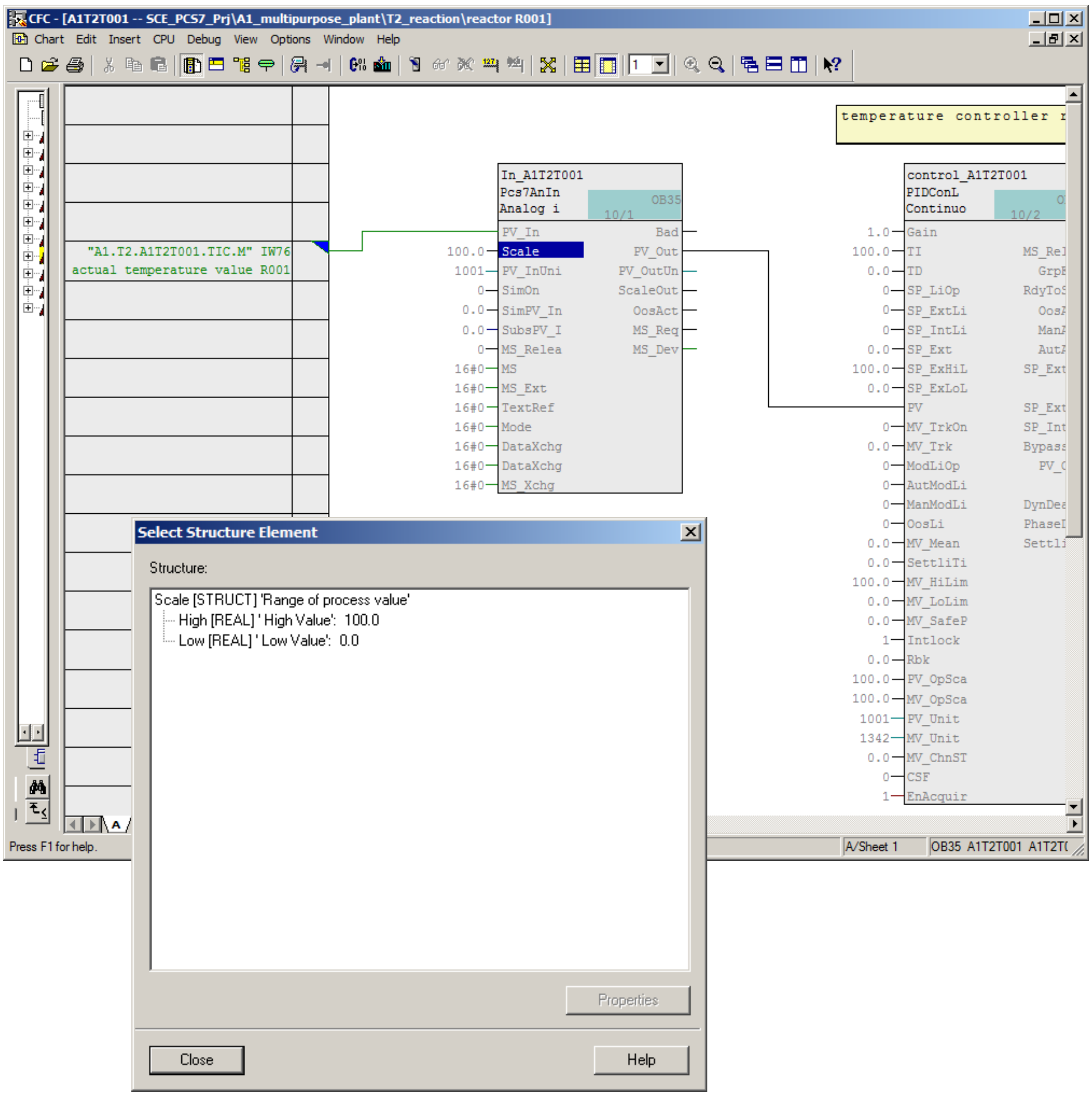
Table 6: Input interconnections in chart 'A1T2T001'

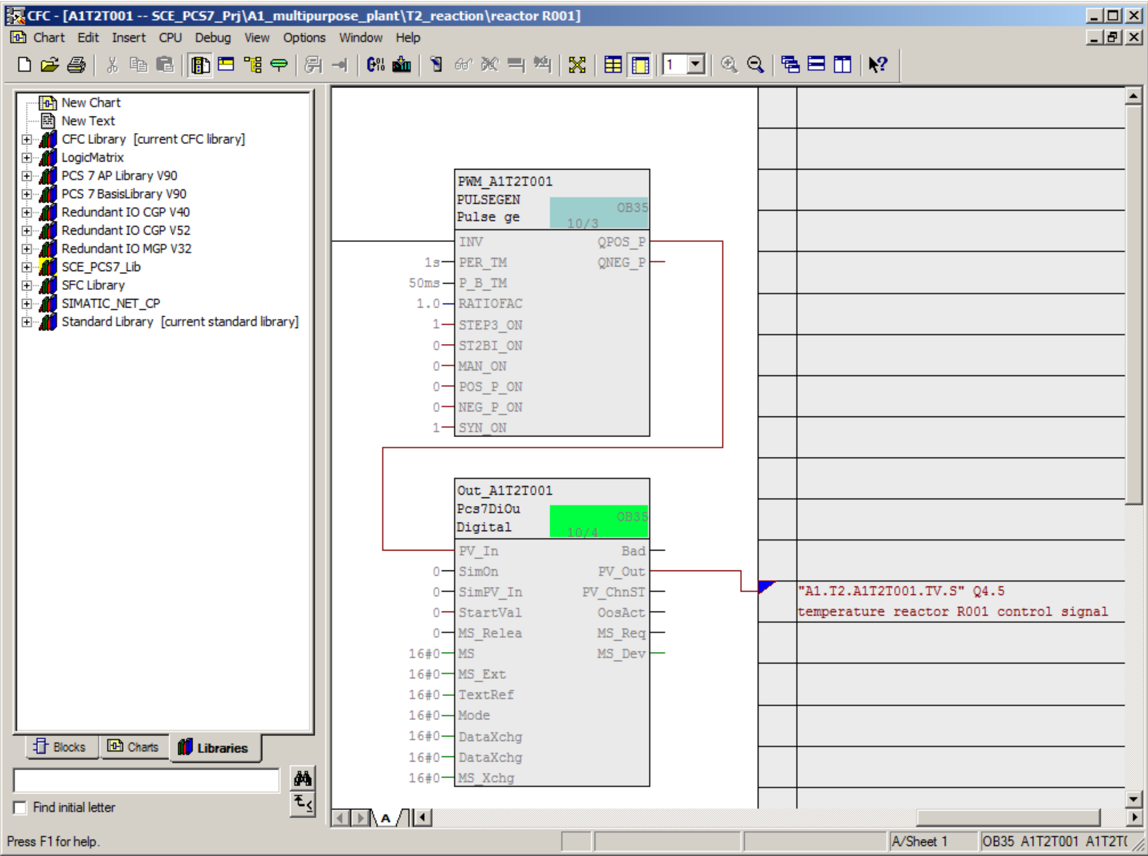
Input:	Output:	Inverted
PIDConL.PV	Pcs7AnIn.PV_Out	
PULSEGEN.INV	PIDConL.MV	
Pcs7DiOu.PV_In	PULSEGEN.QPOS_P	No

Table 7: Block interconnections in chart 'A1T2T001'

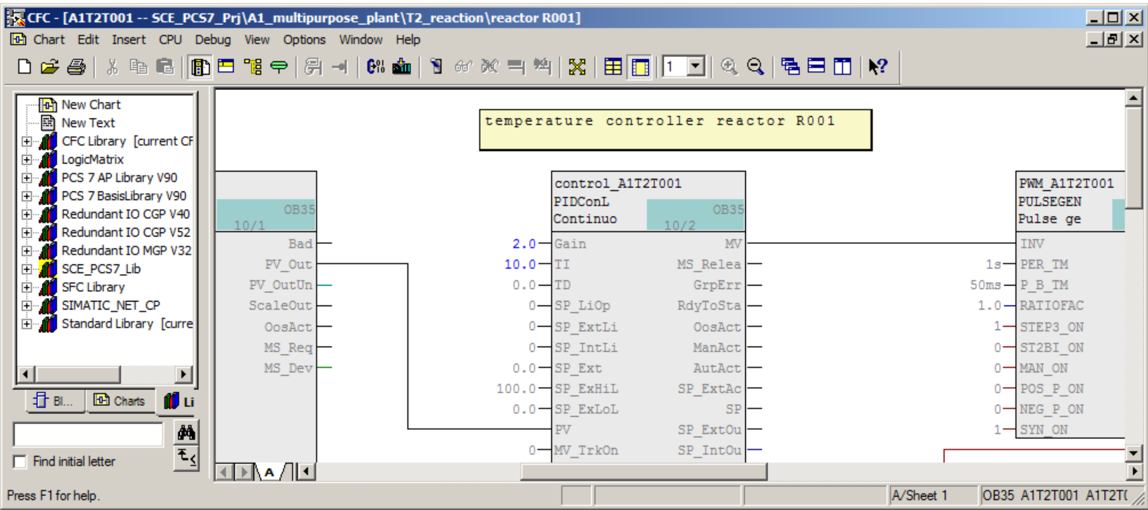
Output:	Interconnection to:	Inverted
Pcs7DiOu.PV_Out	'A1.T2.A1T2T001.TV.S' / Q4.5 / temperature reactor R001 control signal	No

Table 8: Output interconnections in chart 'A1T2T001'





4. Now, configure the gains and the integral-action time of the PID controller by setting PIDConL.Gain = 2 and TI = 10.0.



5. Change to Sheet 2 and create the interlocks shown below:

Block:	Catalog/Folder:
Or04	Blocks/LogicDi
CompAn02 (2x)	Blocks/LogicAn

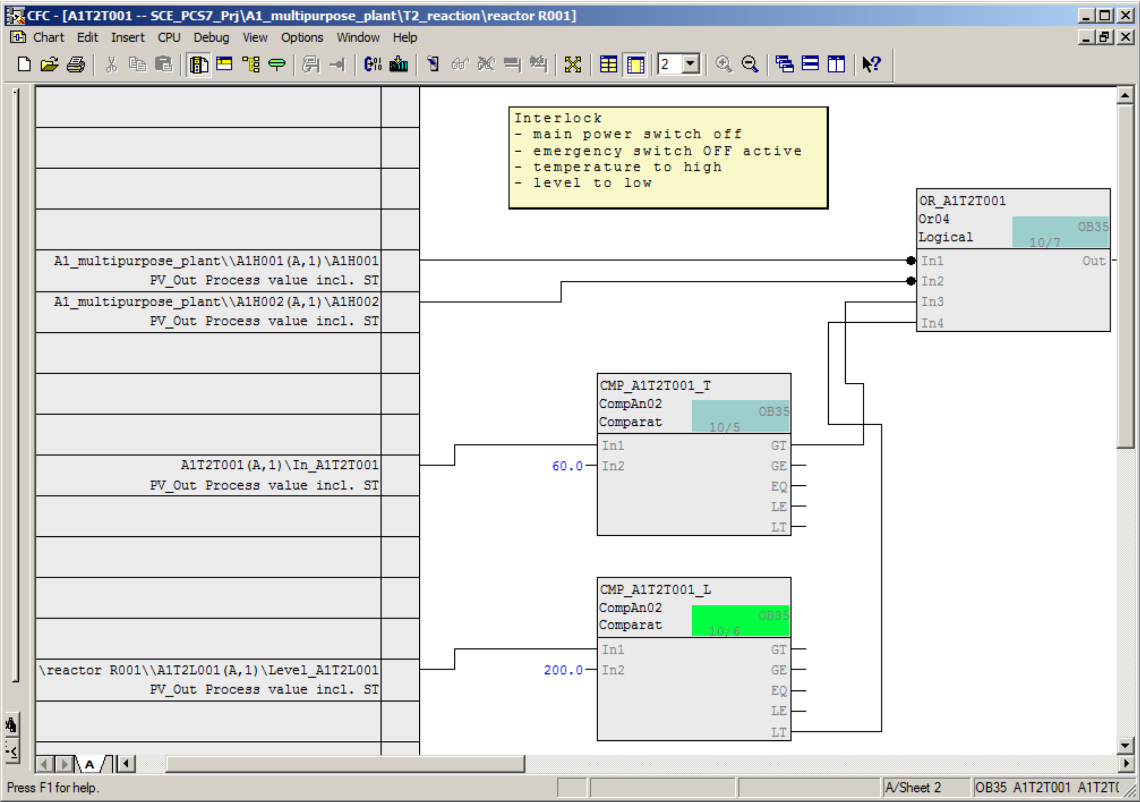
Table 9: New blocks in chart 'A1T2T001/Sheet2'

Input:	Interconnection to:	Inverted
Or04.In1	A1H001(A,1) / A1H001 PV_Out Process value incl. ST	Yes
Or04.In2	A1H002(A,1) / A1H002 PV_Out Process value incl. ST	Yes
CompAn02.T.In1	A1T2T001(A,1) / In_A1T2T001 PV_Out Process value incl. ST	
CompAn02.T.In2	60.0	
CompAn02.L.In1	A1T2L001(A,1) / Level_A1T2L001 PV_Out Process value incl. ST	
CompAn02.L.In1	200.0	

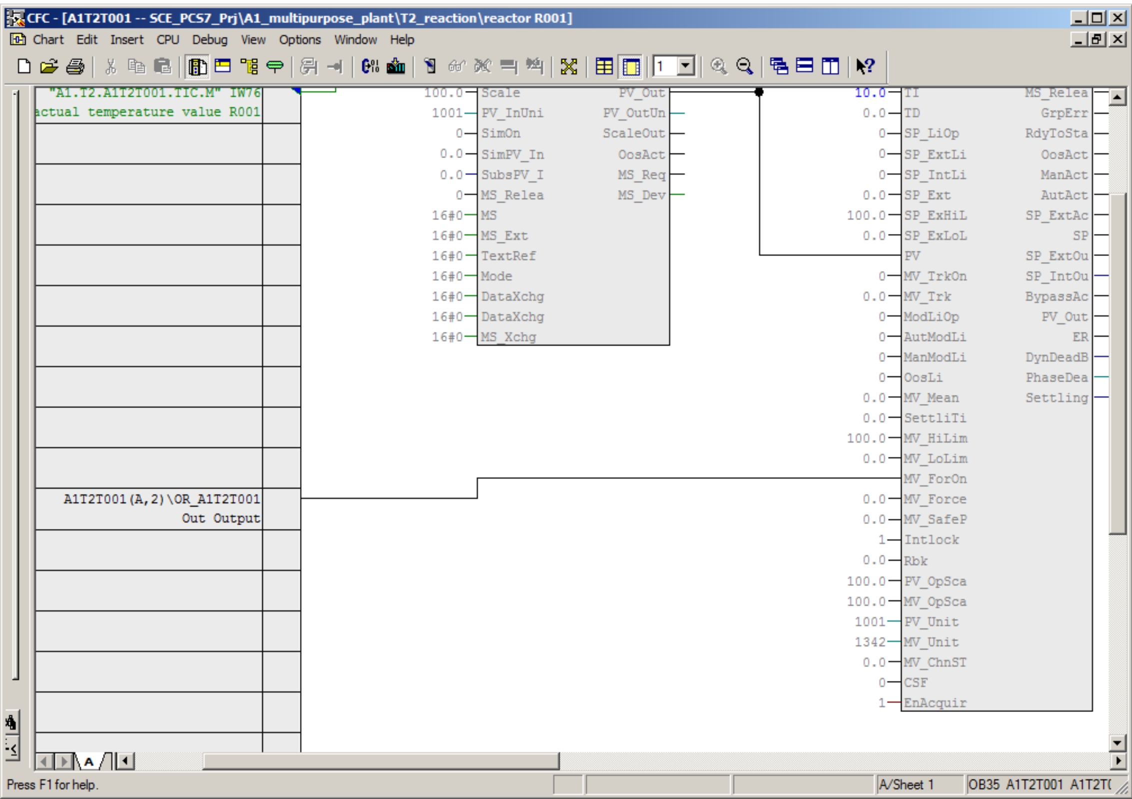
Table 10: Input interconnections in chart 'A1T2T001/Sheet2'

Input:	Output:	Inverted
Or04.In3	CompAn02.T.GT	
Or04.In4	CompAn02.L.LT	

Table 11: Block interconnections in chart 'A1T2T001'



6. Now, interconnect output 'Out' of block 'Or04' with input 'MV_ForOn' of block 'PIDConL' and check that 'MV_Force' = 0.0. As a result of this, the value 'MV_Force' will be applied at output 'MV' of the PID controller (manipulated variable of controller) as soon as the interlocking conditions are met.

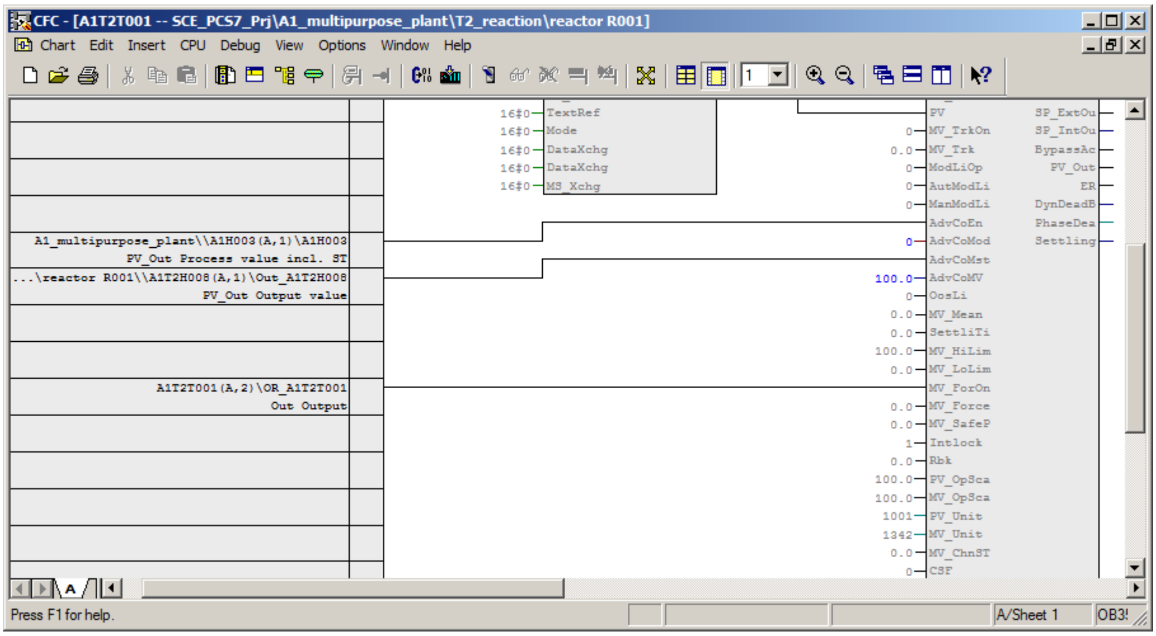


8.3 Manual control of A1T2T001 temperature control

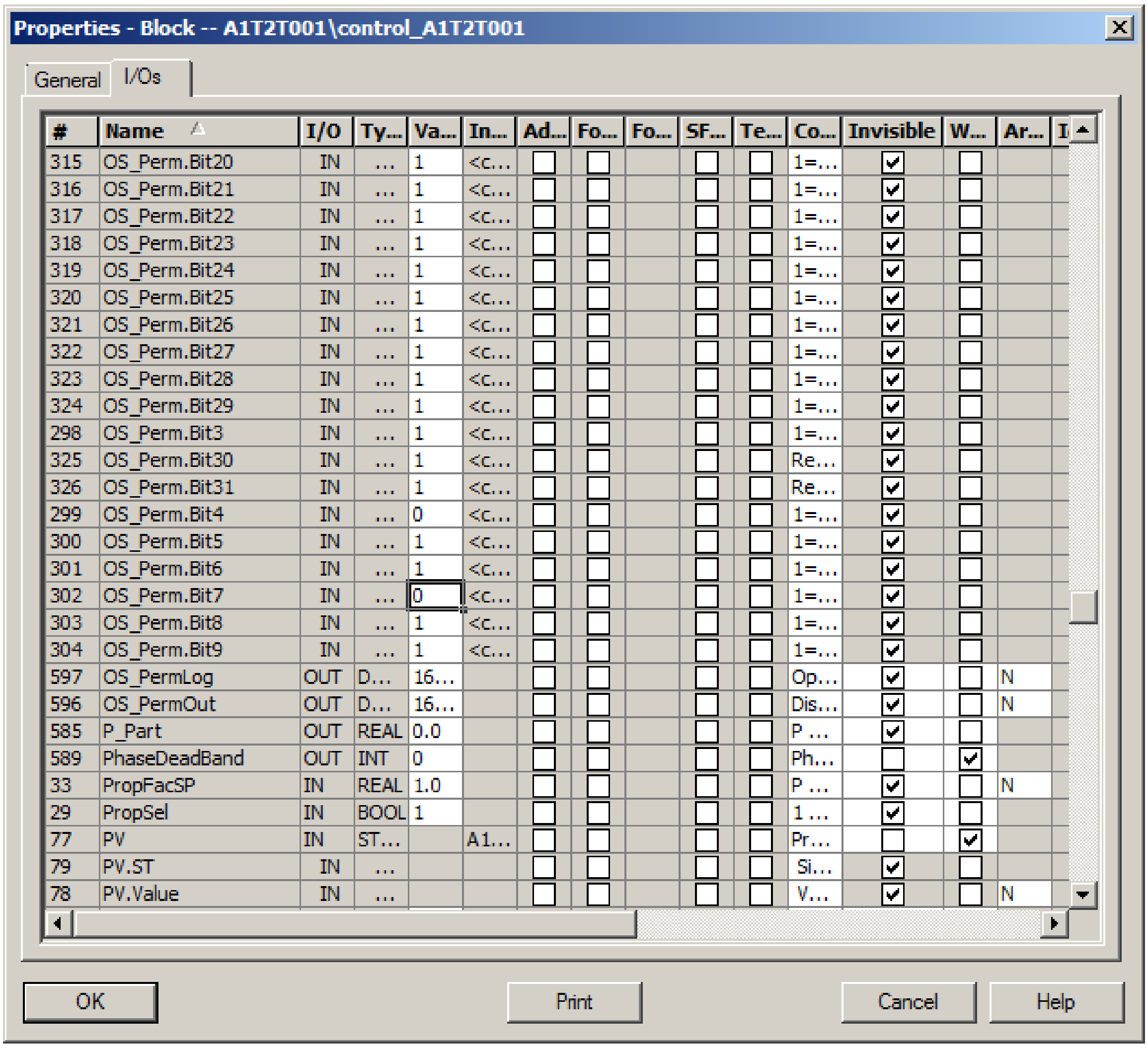
1. Now you will assign parameters for local control. You will use programming mode for this. Programming mode is enabled using the 'AdvCoEn' input and activated using 'AdvCoMst'. Set the 'AdvCoMod' parameter to '0' so that in local mode the 'AdvCoMV' input is interpreted as the manipulated variable and not as the setpoint. Make 'AdvCoMV' visible and set it to 100. As a result of this, the heating will be controlled in local mode without closed-loop control.

Input:	Interconnection to:	Inverted
PIDConL.AdvCoEn	A1H003(A,1) / A1H003 PV_Out Process value including ST	No
PIDConL.AdvCoMod	0	
PIDConL.AdvCoMst	A1T2H008(A,1) / A1T2H008 Out Output	No
PIDConL.AdvCoMV	100.0	

Table 12: Input interconnections in chart 'A1T2T001/Sheet1'



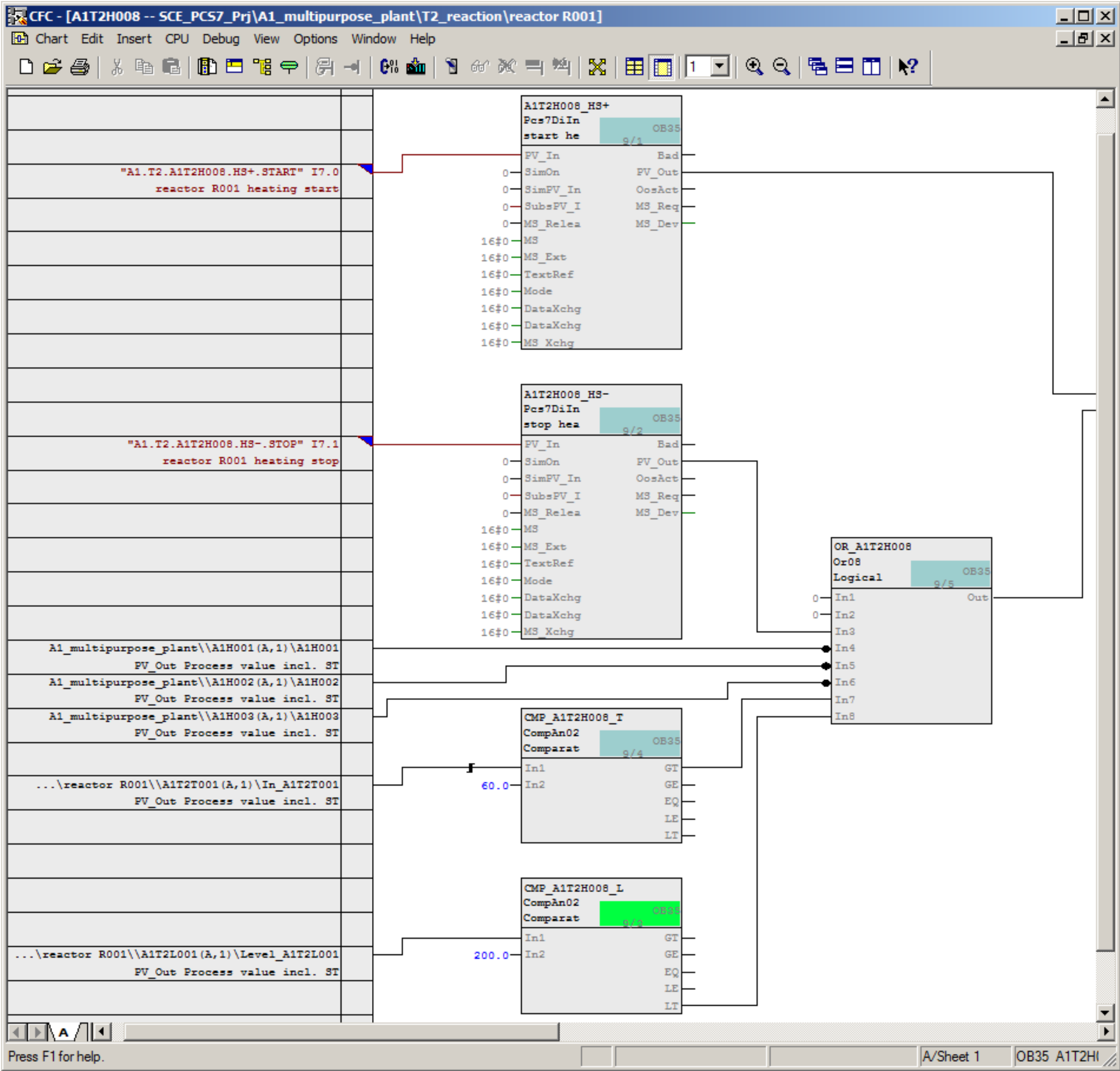
2. With parameter assignment of OS_Perm (bits 0 to 31) of the PIDConL block, operator authorizations can be restricted. Set Bit 4 and Bit 7 to zero, so that the operator cannot enable programming mode and cannot change the manual setting for the manipulated variable ('Man').

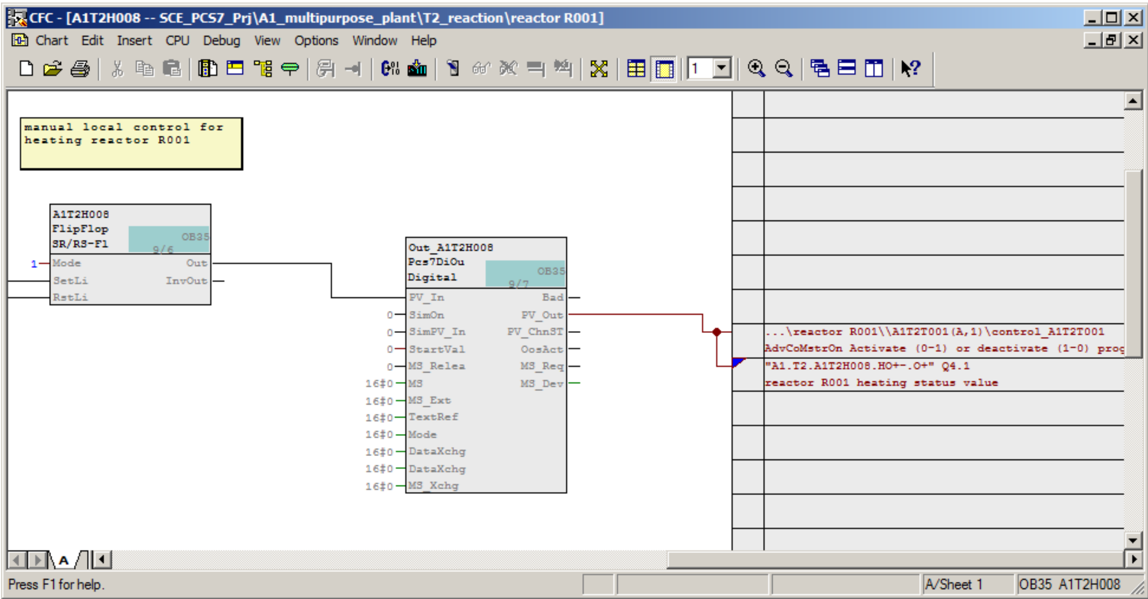


3. Finally, you can interconnect chart 'A1T2H008' and chart 'A1T2T001' as follows.

Input:	Output:	Inverted
CompAn02.T.In1	A1T2T001(A,1) / In_A1T2T001 PV_Out Process value incl. ST	

Table 13: Block interconnections between chart 'A1T2H008/Sheet1' and 'A1T2T001/Sheet1'


CFC - [A1T2H008 -- SCE_PCS7_Prj\A1_multipurpose_plant\T2_reaction\reactor R001]
Chart Edit Insert CPU Debug View Options Window Help
"A1.T2.A1T2H008.HS+.START" I7.0
reactor R001 heating start
A1T2H008_HS+
Pcs7DiIn
start he
"A1.T2.A1T2H008.HS-.STOP" I7.1
reactor R001 heating stop
A1T2H008_HS-
Pcs7DiIn
stop hea
OR_A1T2H008
Or08
Logical
A1_multipurpose_plant\\A1H001(A,1)\A1H001
PV_Out Process value incl. ST
A1_multipurpose_plant\\A1H002(A,1)\A1H002
PV_Out Process value incl. ST
A1_multipurpose_plant\\A1H003(A,1)\A1H003
PV_Out Process value incl. ST
CMP_A1T2H008_T
CompAn02
Comparat
...\reactor R001\\A1T2T001(A,1)\In_A1T2T001
PV_Out Process value incl. ST
60.0
CMP_A1T2H008_L
CompAn02
Comparat
...\reactor R001\\A1T2L001(A,1)\Level_A1T2L001
PV_Out Process value incl. ST
200.0
Press F1 for help.
A/Sheet 1
OB35 A1T2H


CFC - [A1T2H008 -- SCE_PCS7_Prj\A1_multipurpose_plant\T2_reaction\reactor R001]
Chart Edit Insert CPU Debug View Options Window Help
manual local control for
heating reactor R001
A1T2H008
FlipFlop
SR/RS-Fl
Out_A1T2H008
Pcs7DiOu
Digital
...\reactor R001\\A1T2T001(A,1)\control_A1T2T001
AdvCoMstrOn Activate (0-1) or deactivate (1-0) prog
"A1.T2.A1T2H008.HO+-.O+" Q4.1
reactor R001 heating status value
Press F1 for help.
A/Sheet 1
OB35 A1T2H008

8.4 Checklist – step-by-step instruction

The following checklist helps students to independently check whether all steps of the step-by-step instruction have been carefully completed and enables them to successfully complete the module on their own.

No.	Description	Checked
1	Manual control A1T2H008 created and completely interconnected	
2	Temperature control A1T2T001 created, configured and interlocked	
3	Manual control A1T2H008 combined with the A1T2T001 temperature control	
4	Interlocks and manual control successfully tested (optional)	
5	Project successfully archived	

Table 14: Checklist for step-by-step instructions

9 Exercises

In the exercises, you apply what you learned in the theory section and in the step-by-step instructions. The existing multiproject from the step-by-step instructions (p01-06-project-r1905-en.zip) is to be used and expanded for this. The download of the project is stored as zip file "Projects" on the SCE Internet for the respective module.

To prepare for the next chapter, you will implement the final missing function of reactor R001 – the stirrer and the manual control of the stirrer. The interlock conditions are as follows:

– An actuator may only be operated when the main switch of the plant is switched on and the Emergency Stop switch is unlocked.

– The stirrers of the two reactors should only be started up when they are in contact with liquid (here: a minimum of 300 ml in the reactor).

In addition, you can learn more about the PID controller, how it works and what parameters can be set. However, this is not needed here for the functionality of the control.

9.1 Tasks

1. Implement stirrer A1T2S001 in the chart folder 'reactor R001'. Use the same process tag type for the stirrer as for the pumps. Connect the feedback signal and actuating signal. Assign appropriate names to the blocks. Then add the interlocks as explained above.
2. Then implement the manual control A1T2H007 for the stirrer you just created. Implement the interlocking conditions here as reset conditions.
3. Inform yourself about the inputs 'ModLiOp', 'AutModLi' and 'ManModLi' of the 'PIDConL' block. To do so, open the help of the 'PIDConL' block with the 'F1' key. Select 'PIDConL operating modes' and then manual mode or automatic mode.
4. If you want to learn more about the inputs 'SP_LiOp', 'SP_ExtLi', 'SP_IntLi', etc., enter setpoint setting in the 'Search' tab of the help. You will receive information under the suggested title 'Setpoint setting – Internal/external'.
5. What is the purpose of parameters MV_HiLim and MV_LoLim? Search for information on these inputs in the help on your own.

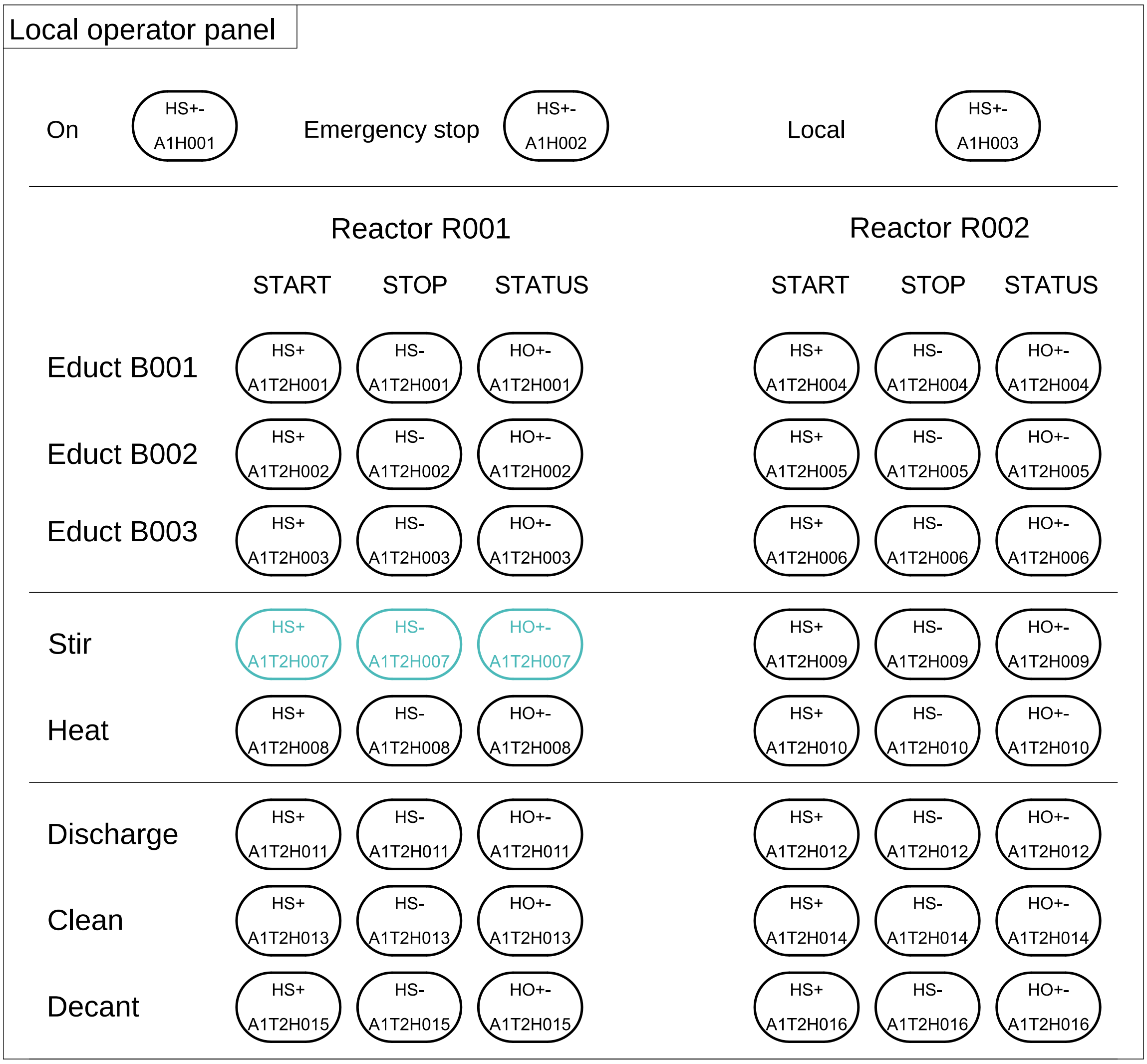


Figure 9: Excerpt from the local operator station

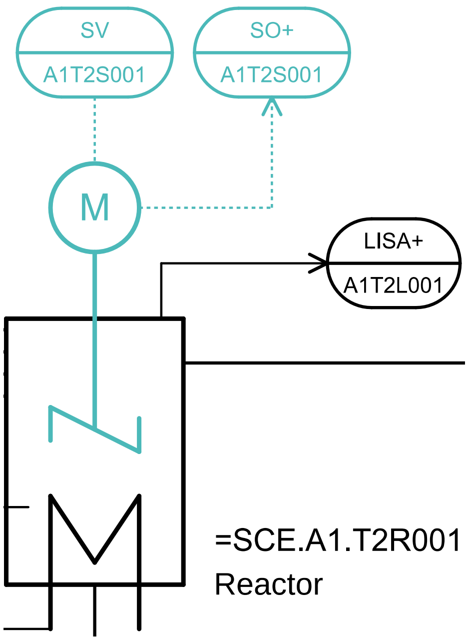


Figure 10: Excerpt from P&ID flowchart

9.2 Checklist – exercise

The following checklist helps students to independently check whether all steps of the exercise have been carefully completed and enables them to successfully complete the module on their own.

No.	Description	Checked
1	Stirrer 'Reactor R001\A1T2S001' created, configured and interlocked	
2	Manual control 'Reactor R001\A1T2H007' created and configured	
3	Manual control A1T2H007 combined with temperature control A1T2S001	
4	The functionality of operating modes and setpoint setting in the PIDConL block is known.	
5	Function of the manipulated variable limitation in the PIDConL block is known	
6	New elements successfully tested (optional)	
7	Project successfully archived	

Table 15: Checklist for exercises

10 Additional information

More information for further practice and consolidation is available as orientation, for example: Getting Started, videos, tutorials, apps, manuals, programming guidelines and trial software/ firmware, under the following link:

siemens.com/sce/pcs7

Preview "Additional information"

Getting Started, Videos, Tutorials, Apps, Manuals, Trial-SW/Firmware

- SIMATIC PCS 7 Overview
- SIMATIC PCS 7 Videos
- Getting Started
- Application Examples
- Download Software/Firmware
- SIMATIC PCS 7 Website
- SIMATIC S7-400 Website

Further Information

Siemens Automation Cooperates with Education
siemens.com/sce

Siemens SIMATIC PCS 7
siemens.com/pcs7

SCE Learn-/Training Documents
siemens.com/sce/documents

SCE Trainer Packages
siemens.com/sce/tp

SCE Contact Partners
siemens.com/sce/contact

Digital Enterprise
siemens.com/digital-enterprise

Industrie 4.0
siemens.com/future-of-manufacturing

Totally Integrated Automation (TIA)
siemens.com/tia

TIA Portal
siemens.com/tia-portal

SIMATIC Controller
siemens.com/controller

SIMATIC Technical Documentation
siemens.com/simatic-docu

Industry Online Support
support.industry.siemens.com

Product catalogue and online ordering system Industry Mall
mall.industry.siemens.com

Siemens
Digital Industries, FA
P.O. Box 4848
90026 Nuremberg
Germany

siemens.com/sce