3V Services JSC

SIMINTECH

10 BASIC TRAINING EXERCISES

illustrated by an example of creation the simplest thermohydraulics model

with automatic control system

Moscow, 2013

Table of Contents

[1 Creation of a new automatics diagram with a signals database 4](#_Toc382402601)

[1.1 Creation of a new automatics diagram 4](#_Toc382402602)

[1.2 Connection of signal database 5](#_Toc382402603)

[1.3 Signal database view and edit interface 6](#_Toc382402604)

[1.4 Addition of a category with database editor 7](#_Toc382402605)

[1.5 Project database saving 11](#_Toc382402606)

[2 Creation of thermohydraulics diagram file with signal database 12](#_Toc382402607)

[2.1 Creation of a new thermohydraulics diagram 12](#_Toc382402608)

[2.2 Connection of signal database 13](#_Toc382402609)

[2.3 Connection of diagram to available signal database 15](#_Toc382402610)

[3 Creation of a simple thermohydraulics model 17](#_Toc382402611)

[3.1 Creation of thermohydraulics diagram 17](#_Toc382402612)

[3.2 Setting design model parameters 19](#_Toc382402613)

[3.3 Linking parameters of design elements with database signals 21](#_Toc382402614)

[3.4 View of thermohydraulics diagram design parameters 24](#_Toc382402615)

[3.5 Addition of transducers to thermohydraulics elements 28](#_Toc382402616)

[3.6 Change of valve names on diagram 31](#_Toc382402617)

[4 Creation of a simple control algorithm 33](#_Toc382402618)

[4.1 Creation of the simplest control algorithm 33](#_Toc382402619)

[4.2 Link of blocks to database signals 34](#_Toc382402620)

[4.3 Creation of the simplest control diagram 37](#_Toc382402621)

[4.4 Check of interchange with signal database 38](#_Toc382402622)

[5 Creation of complex model 40](#_Toc382402623)

[5.1 Creation of complex model 40](#_Toc382402624)

[5.2 Calculation of complex model 41](#_Toc382402625)

[6 Creation of equipment control block 45](#_Toc382402626)

[6.1 Creation of equipment control block 45](#_Toc382402627)

[6.2 Addition of new signals to the database 46](#_Toc382402628)

[6.3 Creation of equipment control block 47](#_Toc382402629)

[6.4 Vector processing of signals 49](#_Toc382402630)

[6.5 Edition of “new” block parameters 52](#_Toc382402631)

[6.6 Creation of valve control model 54](#_Toc382402632)

[7 Creation of a simple control algorithm 59](#_Toc382402633)

[7.1 Creation of control algorithm 59](#_Toc382402634)

[7.2 First valve control algorithm 60](#_Toc382402635)

[7.3 Edition of “Signal reading” and “Algorithm output” blocks 60](#_Toc382402636)

[7.4 First valve control block diagram 64](#_Toc382402637)

[7.5 Check of first valve control algorithm operation 64](#_Toc382402638)

[7.6 Second valve control algorithm 67](#_Toc382402639)

[7.7 Check of model work 71](#_Toc382402640)

[8 Change of complex model 75](#_Toc382402641)

[8.1 Check of complex model 75](#_Toc382402642)

[8.2 Task for independent operation 78](#_Toc382402643)

[9 Creation of window of equipment control 79](#_Toc382402644)

[9.1 Manual control in project 79](#_Toc382402645)

[9.2 Creation of control window 79](#_Toc382402646)

[9.3 Creation of equipment control interface 81](#_Toc382402647)

[9.4 Creation of variables for window of valve control 83](#_Toc382402648)

[9.5 Programming of window of valve control 86](#_Toc382402649)

[9.6 Link of the valve and the control window 88](#_Toc382402650)

[9.7 Manual valve control in a complex model 90](#_Toc382402651)

[10 Creation of event log 93](#_Toc382402652)

[10.1 Event logging 93](#_Toc382402653)

[10.2 Creation of event log 93](#_Toc382402654)

[10.3 Adding parameters to “Event log” 95](#_Toc382402655)

[10.4 Setting of event logging parameters 96](#_Toc382402656)

[10.5 “Event log” window 98](#_Toc382402657)

[10.6 Application of event log for modeling 100](#_Toc382402658)

# Creation of a new automatics diagram with a signals database

## Creation of a new automatics diagram

To create an automatics diagram in SimInTech proceed as follows:

1. Select **“New Project”** button in the toolbar.
2. Select **“Automatics diagram”** item from a pop-up menu (Figure 1).

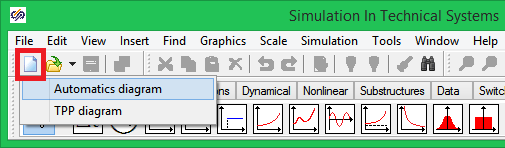


Figure 1. New project creation menu

Then a new diagram window will be opened, in which an automatics block diagram will be generated (Figure 2).

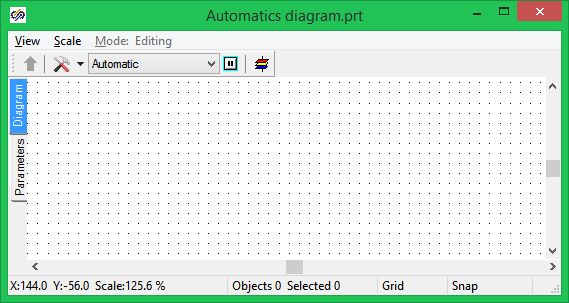


Figure 2. Diagram window for generating an automatics diagram

To proceed with the work save the given diagram in a file under a new name (e.g., “Automatics diagram 1”). To that end, perform the following operations:

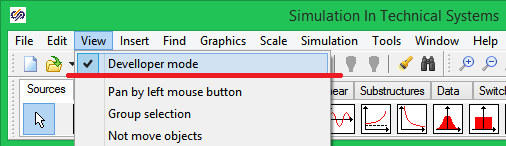
1. Select item **“File”** in the main menu and then select **“Save project as...”** item from the pop-up list.
2. Using the standard “save file” dialog, select a new name and catalog for saving. For example, **“C:\SimInTechEng\Projects\Automatics diagram 1.prt”**.

When the file has been saved, its name and full path are displayed in the diagram window title (Figure 2). When required, the User can change the size and location of the window on the computer screen using common methods and techniques for dialog control. Size and location of the diagram window will be stored in the project file.

## Connection of signal database

By default, diagrams of mathematical representations are intended for off-line debugging and do not contain any signal database. Nevertheless, a signal database can be connected to any diagram in SimInTech in order to arrange external interaction with other programs and for vector processing of signals as well.

**Note:** to proceed with the work the behavior and appearance of SimInTech shall be switched over to Developer mode:



Activation of Developer mode

Signal database is connected to the automatics diagram in the following manner:

1. Press button **“Simulation properties”** in the diagram window:

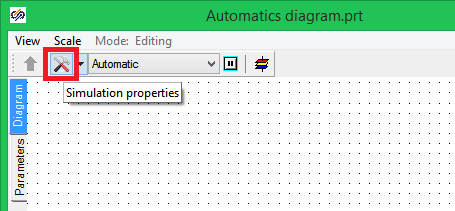


Figure 3. Simulation properties access button

1. Go to **“Settings”** tab in the popped up setting dialog. (Figure 4).
2. Enter the following text in **“Project database module”** edit bar: **$(Root)\sdb.dll** (sdb.dll is for the name of dynamic library of database program module).
3. Enter a random file name for saving the database in **“Project database name”** bar. For example, **“signals.db” (in case the full path to the file is not set, by default it is located in the same catalog as the project file is).**

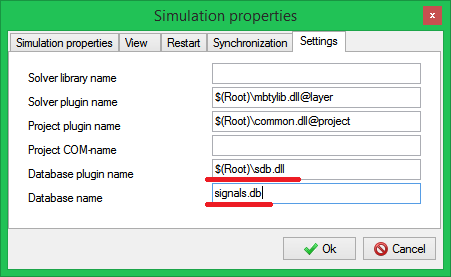


Figure 4. Project settings tab

1. Close the dialog by pressing **“Ok”** button (Figure 4).
2. To save the current project press **“Save project”** button in the program main window (Figure 5).

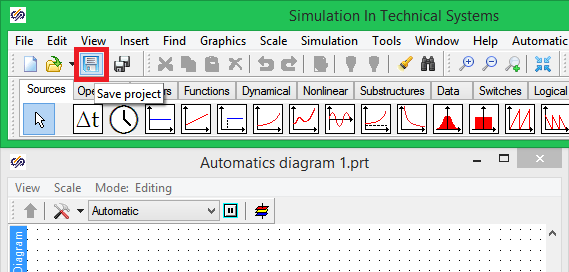


Figure 5. Current project save button

## Signal database view and edit interface

Operations related to filling and edition of the signal database are carried out by means **“Database Editor”** module. Signal database is activated via the program main menu (**“Tools”** menu item, **“Database…”** sub-item, see Figure 6).

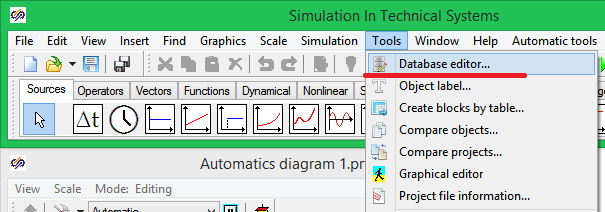


Figure 6. Database editor activation

1. **Attention!!!** This menu item is not accessible for projects, for which a database is not applicable. To get access to this menu item perform a sequence of actions as described in the section 2)Using the standard “save file” dialog, select a new name and catalog for saving. For example, **“C:\SimInTechEng\Projects\Automatics diagram 1.prt”**.

.

On selection of the menu item, the database editor dialog will be displayed, see Figure 7.

## Addition of a category with database editor

Database editor contains two tabs arranged in the upper part of the dialog: directly “**Editor**” and “**Settings**” (Figure 7). To set up signals “**Editor**” tab is used.

Editor window contains three main fields (panels): “**Categories**”, “**Signal** **groups**” and “**Signal** **and** **data** **for** **groups**”. “**Filter**” panel is also available for generating and saving user defined filters to display just parts of database signals filtered out as per any criterion. This mechanism is used for navigation and comfortable work with large databases.

“**Categories**” panel contains a list of types of objects, which can be loaded into the database. E.g., “**VCU**” – Valve control block.

**Groups of signal** panel contains a list of specific objects (specific occurrences of a selected type or category) available in the project database corresponding to a selected category. E.g., **1L01AS1\_2** is for the code of a particular valve available in the database and mathematical representation of an object.

**Signals and data for groups** panel contains a table with signals and data corresponding to a specific signal group selected in **Signal group** panel.

The user can independently generate, edit and delete new object categories in every project as well as a list and values of signals.

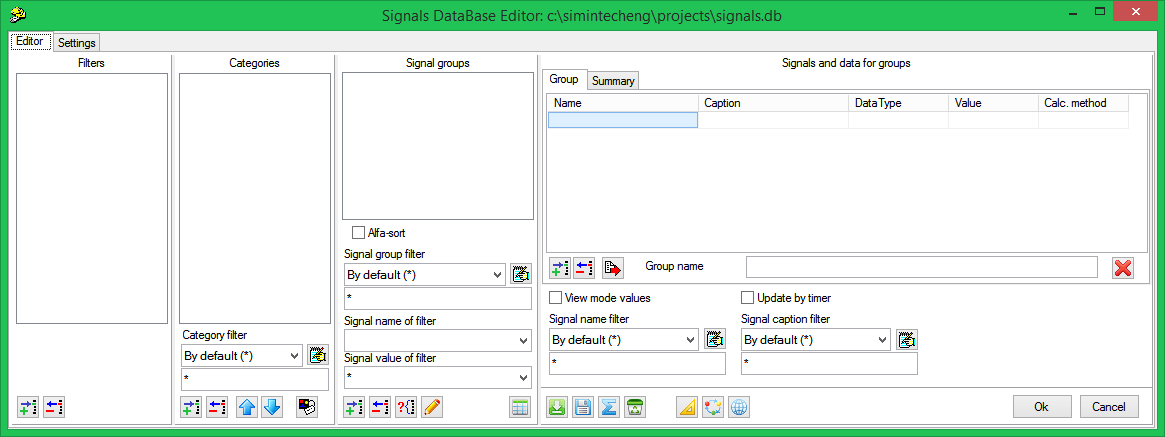


Figure 7. “Project database editor” dialog window

By default, all panels of the editor do not contain any elements for a project to be newly generated. The user can both independently fill the database and read out a previously prepared file with an available database.

A new category can be added as follows:

1. The user shall press “**Add category**” button in the lower part of the panel (Figure 7). After that a new “**New category**” entry will be displayed in the category list.
2. Double click on **“New category”** element will open the category edit dialog window (Figure 8). In this window the User fills the signal table that corresponds to the considered category. E.g., for a valve those can be real values of position of rod, binary value of “open” or “closed” signal, etc.

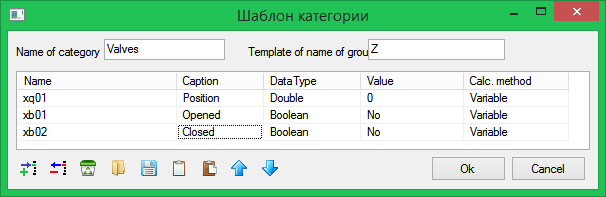


Figure 8. Category template edit dialog window

Dialog window represented in Figure 8 contains the simplest example for filling properties of a category of “**Valves**” type.

To save changes introduced close “**Category properties**” dialog window by pressing “**Ok**” button. After that, a new category under “**Valves”** name will appear in the list of categories.

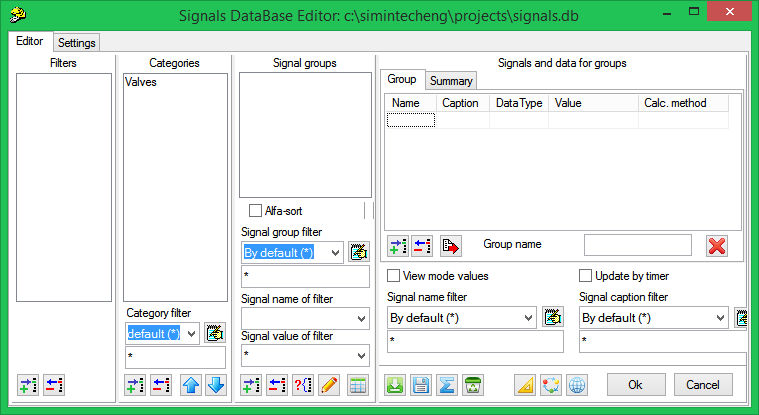


Figure 9. Database editor with added Valves category

For every created category the User can add indefinite number of signal groups to the database. E.g., on creating the category for a certain type of valves, we can add all valves belonging to this type and applicable in the design model to the database. For this purpose proceed as follows:

1. Select the category name in the category panel (**“Valves”** category in our example).
2. Press button **“Add table”** in the panel of signal groups (Figure 9).
3. Enter names of signal groups of the considered category to be loaded into the database of in “**Creating new groups**” dialog window: (Figure 10).
4. Close **“Creating new groups”** dialog window by pressing **“Ok”** button.

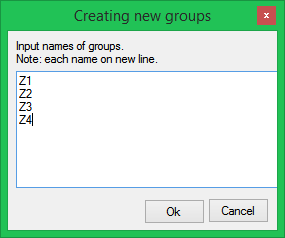


Figure 10. Creating new groups dialog window

Names of signal groups are desirably to be set using Latin characters and numerals. It is required when the database is used for processing a signal transmitted via OPC protocol.

Set signal groups are displayed in “**Signal Groups**” panel.

Figure 11 depicts “**Categories properties**” edit dialog window for **“Transducers”** categories in a test example. Enter this category into the database in accordance with the Figure.

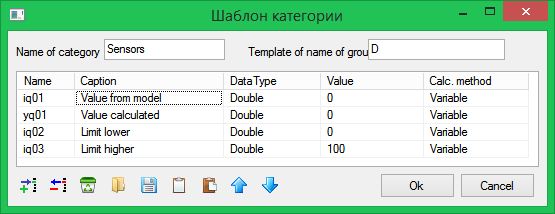


Figure 11. Categories properties dialog window for “Transducers” category

After creation of a new category, generate **“Transducers”** signal group following the above procedure for signals presented in Figure 12 to be enabled in the test database:

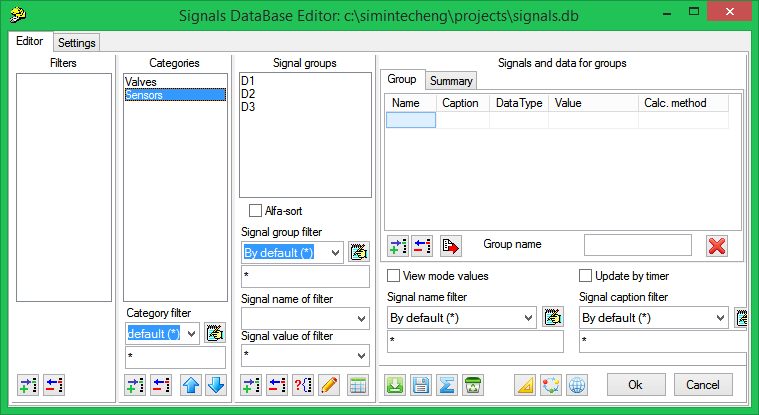


Figure 12. Database editor with added Transducers category

After addition of signal categories and groups to DB, one can switch between those to select a required signal category and group (Figure 12).

## Project database saving

Prior to closing the editor, one shall make sure that the editor settings involve automatically saving the database. To that end, go to “**Settings**” tab and tick **“Save base”** and **“Reserve DB”** items (Figure 13).

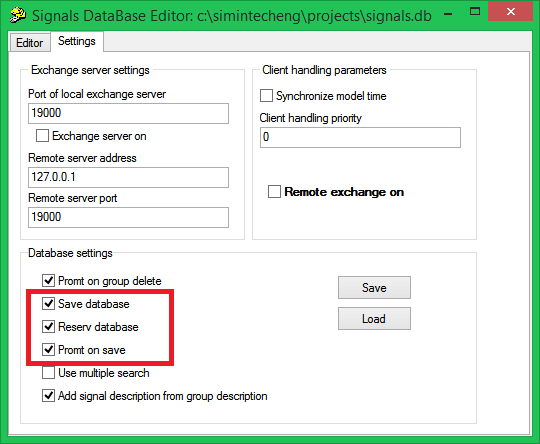


Figure 13. Save database settings

On setting **“Save database”** and **“Reserve DB”** options go to **“Editor”** tab and close **“Database editor”** dialog window by pressing **“Ok”** button.

On doing that, save the project.

These settings allow the project database (a file with the project database) to be automatically saved when the project file is being closed.

If the above described actions are correctly performed then, after closing the project, a file named as **“signals.db”** will appear in thedirectory the file has been saved in.

# Creation of thermohydraulics diagram file with signal database

## Creation of a new thermohydraulics diagram

To create a new thermohydraulics diagram proceed as follows:

1. Select **“New project”** key in the toolbar.
2. Select **“TPP diagram”** item from a pop-up menu (Figure 14)

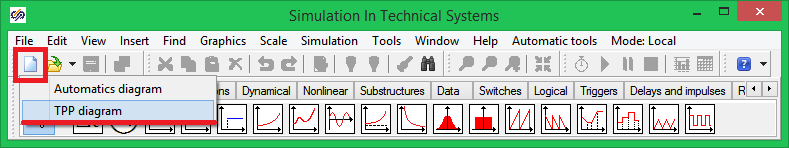


Figure 14. New project creation menu

Then a new diagram window will be opened, in which a thermohydraulics model diagram will be generated (Figure 15).

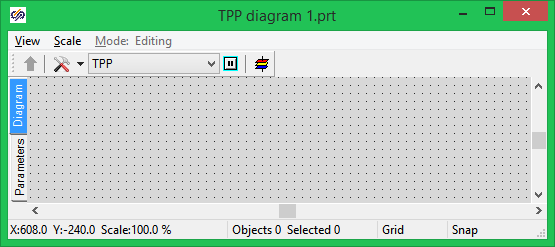


Figure 15. Diagram window for generating a thermohydraulics diagram

To proceed with the work save this diagram in a file under a new name in the same directory the automatics diagram has been saved in. For that purpose:

1. Select item **“File”** in the main menu, select **“Save project as...”** item from the pop-up list.
2. Use the standard “save file” dialog select a new name and catalog for saving. E.g., **“ТPP Diagram 1.prt”.**

When the file has been saved, its name and full path are displayed in the diagram window title (Figure 15). When required, the User can change the size and location of the window on the computer screen using common methods and techniques for dialog control.

## Connection of signal database

To enable interoperability of several design codes it is necessary to apply the same signal base (which in our case is located in the pre-generated file **“signals.db”**) for them.

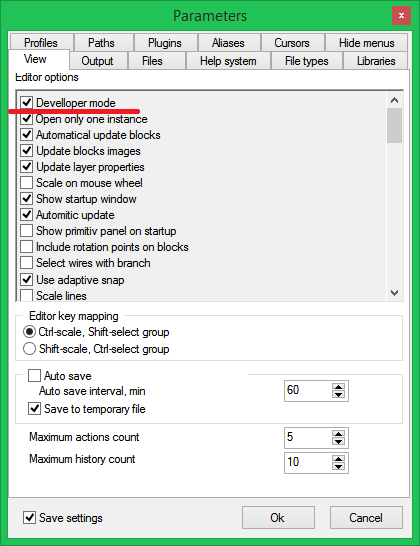


Figure 16. Software system parameters setting dialog window

To connect the database to a newly generated thermohydraulics project switch over the software system to the developer mode, for which purpose select **“File”** item in the main program menu and then **“Parameters”** sub-item. Go to **“View”** tab in the displayed **“Parameters”** dialog window and tick **“Developer mode”** option (Figure 16).

Signal database is connected to the thermohydraulics diagram in the following manner:

1. Press button **“Simulation properties”** in the diagram window:

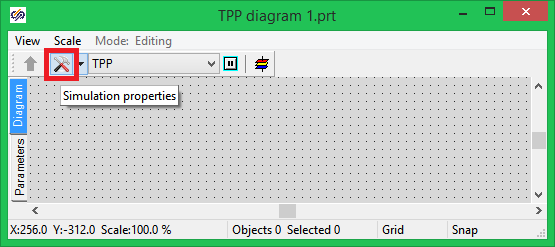


Figure . Simulation properties access button

1. Go to **“Settings”** tab in the popped up setting dialog (Figure 18).
2. Enter the following text in **“Project database module”** edit bar: **“$(Root)\sdb.dll@db”** (to be entered without quotation marks; sdb.dll is for the name of dynamic library of database program module).
3. Enter a random file name for saving the database in **“Project database name”** bar. To use the database file of pre-generated automatics diagram enter the file name (in the same manner as for the first training exercise, i.e., **“signals.db”** in our case, see Figure 18).

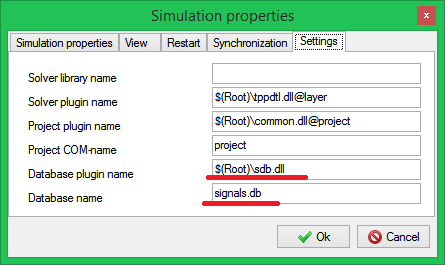


Figure 18. Project database setting tab

1. Close the dialog by pressing **“Ok”** button (Figure 18).

## Connection of diagram to available signal database

To use the available signal database generated during creation of the automatics diagram, follow the next procedure:

1. Activate the database editor via the program main menu: **“Tools”** menu item, **“Database…”** sub-item (Figure 19).

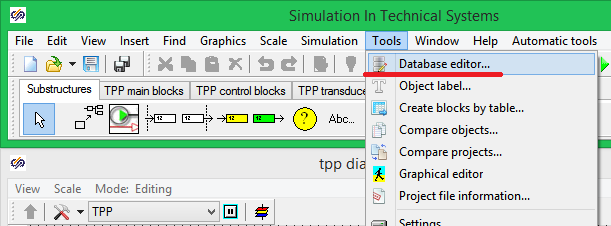


Figure 19. Database editor activation

1. Since this diagram will be connected to the available database, autosave of the database at the moment of thermohydraulics diagram saving shall be disabled. Thus, prior to closing the editor window, one shall make sure that the editor settings DO NOT involve automatically saving the database. To that end, go to “**Settings**” tab and untick **“Save base”** and **“Reserve DB”** items (Figure 20), if that has been ticked.
2. On unsetting **“Save database”** option go to **“Editor”** tab and close **“Database editor”** dialog window by pressing **“Ok”** button.
3. On doing that, save the diagram and close the project.

These settings allow the database not to be saved when the thermohydraulics diagram is being saved. It allows any erroneous change in the database during editions of the diagram to be excluded.

In case all above listed actions have been correctly executed then, after reopening of the thermohydraulics diagram, the signal database will be automatically loaded from **“signals.db”** file and will contain all signals generated during creation of the automatics diagram.

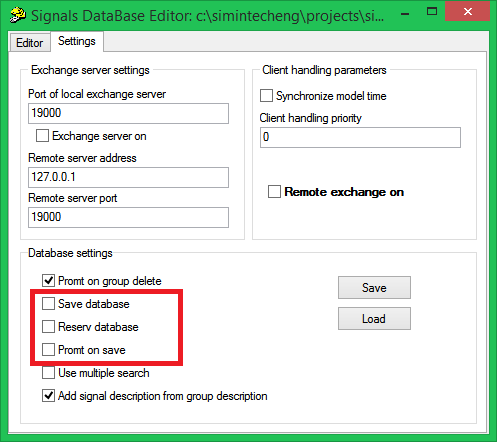


Figure 20. Settings for saving thermohydraulics diagram database save

# Creation of a simple thermohydraulics model

## Creation of thermohydraulics diagram

Open the file named **“Diagram ТPP 1.prt”.** This file was created during execution of the second training exercise and adjusted for operation with the database saved in **“signals.db”** file.

Make sure that the database contains signals created during execution of the first training exercise.

Blocks located in **“TPP process blocks”** tab of the block template are used to create the diagram. The blocks are grouped in pop-up lists (Figure 21).

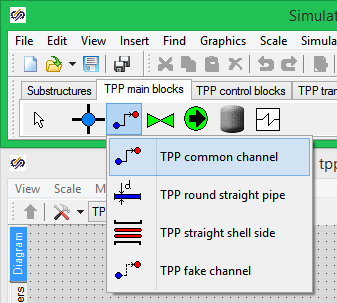
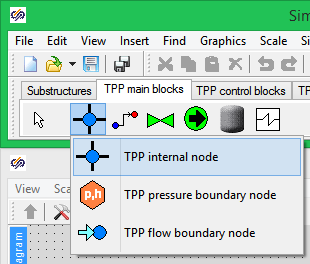


Figure 21. Thermohydraulics block pop-up list

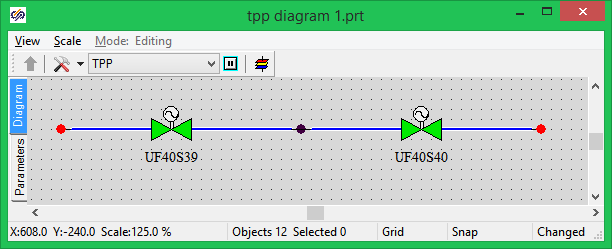
Then the following successive actions shall be performed:

1. Arrange the following design thermohydraulics blocks in the diagram window:

* **“Boundary node P”** (from “Nodes” list),
* **“Straight pipe”** (from “Channels” list),
* **“Internal node”** (from “Nodes” list),
* **“Straight pipe”** (from “Channels” list),
* **“Boundary node P”** (from “Nodes” list),

Image of the blocks of the diagram can consists of several interlinked graphic elements, which can be shifted relative to each other.

On setting an internal node on the diagram, two transducers are synchronously added: pressure transducer and temperature transducer bound to the given node.



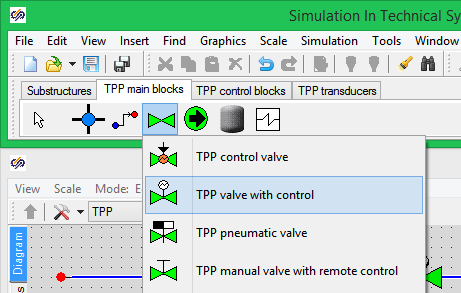


Figure 22. Thermohydraulics model test diagram

1. Unlink successively the elements so that two **“Straight pipe”** elements generate one hydraulic line with an internal node. Boundary nodes P will determine pressure at the boundaries of the given hydraulic line.
2. Place **“Adjusting valve”** element onto the first **“Straight pipe”** element.
3. Place **“Adjusting valve”** element onto the second **“Straight pipe”** element.

As a result, thermohydraulics model diagram shall resemble the Figure 22. **Note:** Images of adjusting valves can be different from the given Figure, while the internal node can be set up on the diagram without blocks of pressure transducer and temperature transducer type.

## Setting design model parameters

To ensure correct calculation of a thermohydraulics model geometrical and other properties of **each** diagram block (element) important for calculation shall be set up. For this purpose proceed as follows:

1. Select a required element.
2. Select **“Object properties”** item in the pop-up menu (Figure 23).

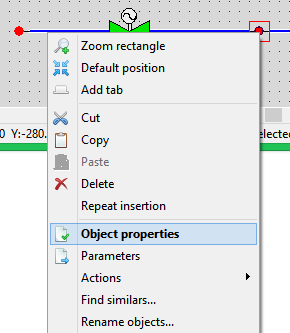


Figure 23. Diagram element contextual menu (upper part of the menu)

Then **“Properties”** dialog window will be displayed, in which properties of the element can be set up. Dialog window for **“Boundary node P”** object is presented below (Figure 24).

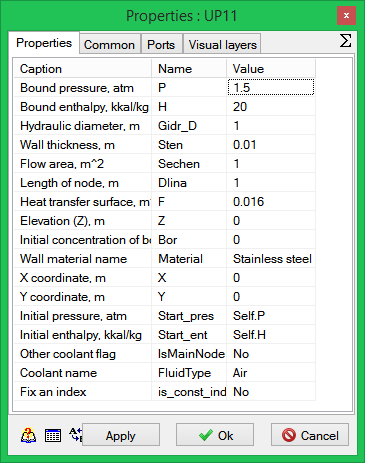


Figure 24. “Properties” dialog window for boundary node P

Set pressure **“1.5”** for the left boundary nodes and **“1”** for the right boundary node. These settings will set up a pressure difference constant at the boundaries of a pipeline being modeled and, thus, availability of flow rate in the pipeline.

To ensure correct operation of valves in the diagram the hydraulic pressure in the pipeline section, in which those are installed, shall not be equal to zero. In our example each pipeline has only one section. Set up the resistance for a pipe section as equal to 1 (for both left and right pipelines).

To this end, activate **“Properties”** window for **“Straight pipe”** element and enter **Self.Count#1** string in **“Direct local resistance”** and **“Reverse local resistance”** strings. This string indicates the resistance equal to 1 for all pipe-line sections (Figure 25).

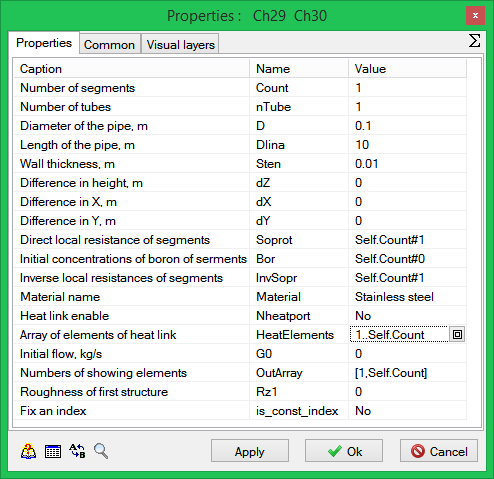


Figure 25. “Properties” dialog window for straight pipe

Now let us consider valves. We should remind that, when creating the automatics diagram, we used **“Z1”** and **“Z2”** as names of the valves.

## Linking parameters of design elements with database signals

After creating the simplest design diagram, parameters of design elements shall be linked with database signals.

Select the adjusting valve on the diagram and activate the contextual menu (right mouse button). Select **“Object properties”** item in the menu (Figure 23).

After that, parameter edit dialog window will be displayed for an element of **“Adjusting valve”** type as represented in the Figure below (Figure 26):

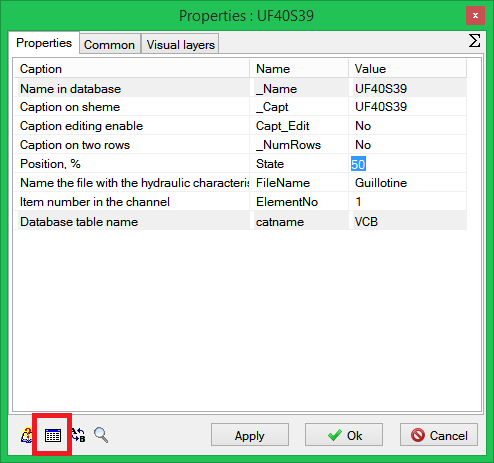


Figure 26. “Adjusting valve” element properties edit dialog window

Parameters presented in the edit window on **“Properties”** tab (Figure 26) can be set both by means of numerical values directly in the property editor and by importing from the signal database.

To link an object parameter to a database signal select the parameter value in the table (**“Position, %”** in this example, delete the numerical value (**“50”** in this example) and press **“Find value in base”** button (Figure 26).

On pressing this button, **“Database editor”** dialog window will be activated (Figure 27). Select successively in the dialog window:

1. **“Valves”** category.
2. Group of **“Z1”** signals for the left valve (**“Z2”** for the right valve).
3. **“Position”** signalname.

In this training exercise the properties of **“Position, %”** object shall be linked with **“Position”** signal in the database for the valve named **Z1** (Figure 27).

Select this signal and press **“Add”** button in “**Selected data”** panel (Figure 27). Delete available entries, when required.

Enter **“50”** value for **Position** signal (Figure 27). Signal setting in the database will set up the value for valve position as 50% open.

A unique name will be generated for this signal consisting of the signal group name and signal name separated by an underline (**Z1\_xq1** in this example). Press **“Ok”** button to close the editor window.

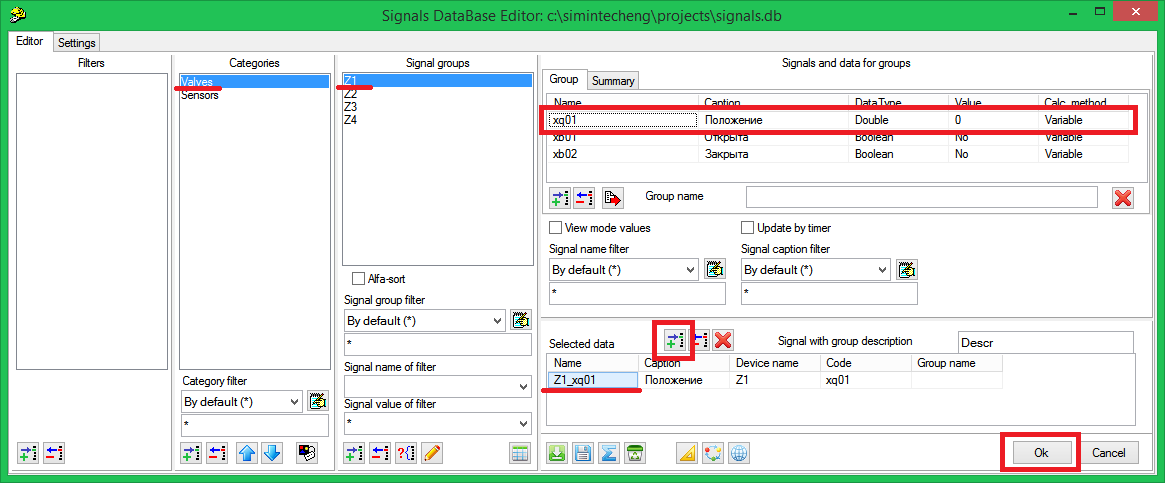


Figure 27. Selection of a signal in database for object properties

Redo the actions for the second valve, on linking that to a signal of the project database, which is similar but belongs to the signal group named **“Z2”**. Properties dialog window for this valve shall be like it is shown in the Figure 28.

Name of used signal (**Z1\_xq01** and **Z2\_xq01**) can be also manually entered…

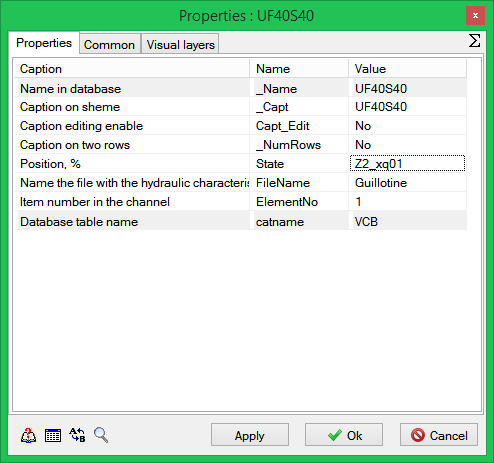


Figure 28. Valve properties after linking database Z1\_xq01 signal with “Position, %”

## View of thermohydraulics diagram design parameters

Each design diagram element contains a set of parameters, which are calculated by means of a code and can be used for analyzing transitional processes and for creating control points of an object (for being used in the control system). These parameters as tabular values as well as curves can be viewed directly during calculation of a thermohydraulics diagram. To view the list of parameters available for every element proceed as follows:

1. Select a thermohydraulics circuit element.
2. Press the right mouse button.
3. Select **“Object parameters”** item in the pop-up menu (Figure 29).

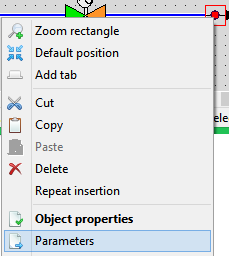


Figure 29. System elements pop-up menu

Then a window with the list of parameters, which can be obtained using the design code for the given system element, will be displayed.

Select the internal node on the thermohydraulics diagram (Figure 30)

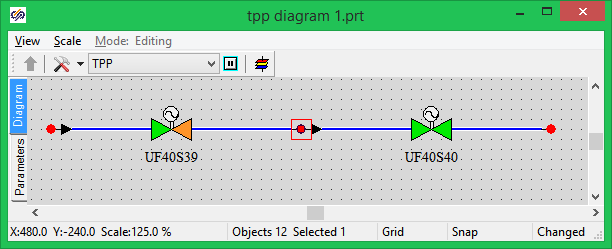


Figure 30. Selection of internal node of diagram

Select **“Object parameters”** item in the pop-up menu (Figure 29). Dialog window with parameters for the selected element will be displayed (for the internal node, see Figure 31). This window is displaying the list of parameters of an element currently selected on the diagram. While keeping the window open, you can select another element of the diagram and view the list of its parameters.

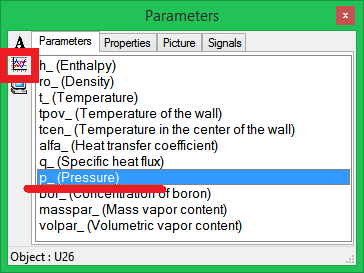


Figure 31. “Parameters” dialog window for internal node of diagram

Select **“Pressure”** parameter in the list and press **“Create graphic”** button in the same window (top left) (Figure 31). New **“Time Graphic”** window will be displayed, in which a change of a selected time parameter will be indicated (Figure 32).

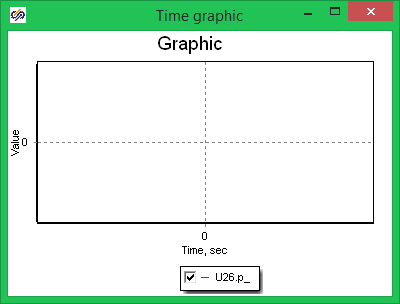


Figure 32. Time graphic window

Name of variable on the graphic is generated from the element name and parameter name, which are divided with a dot. For example, name of variable on the graphic for **“Internal node”** element named as **“U26”** and pressure parameter named as **“p\_”** is **“U26.p\_”** (Figure 32).

While keeping **“Parameters”** dialog window open, select the first (left) **“Straight pipe”** element on the thermohydraulics diagram (Figure 33).

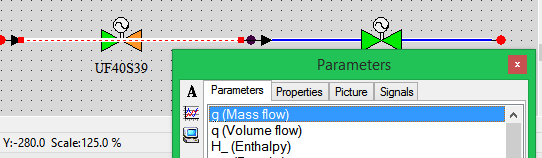


Figure 33. Diagram with “Straight pipe” element selected

Parameters dialog window, at the same time, will display the list of parameters related to the newly selected **“Straight pipe”** element (Figure 34).

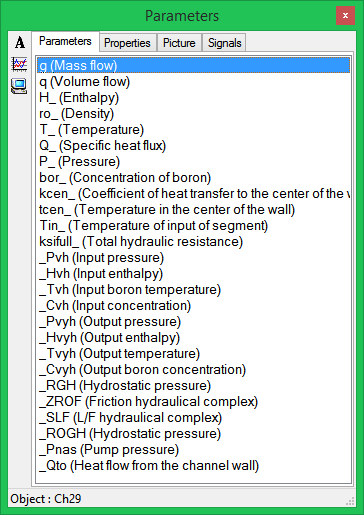


Figure 34. “Parameters” dialog window for straight pipe

Select **“q (Mass flow)”** parameter and press **“Create graphic”** button.

Activate the calculation task (“**Calculation**” item of the main window, **“Start”** sub-item). Calculation task can be also started by pressing **“F9”** key on the keyboard.

If the previous actions have been correctly executed then the following values for selected parameters will be set up in the created diagram after a short transitional process:

Pressure in the internal node (**“U26.p\_”**) = 1.25 (Figure 35),

Mass flow rate in pipe (**“Tube15.g”**) = 23.8 (Figure 36).

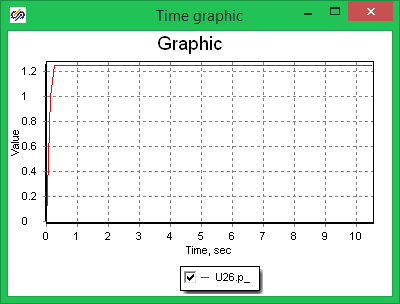


Figure 35. Pressure graphic for internal node of the system

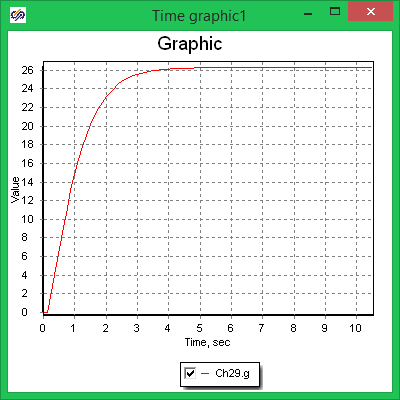


Figure 36. Graphic of mass flow rate in the pipe

## Addition of transducers to thermohydraulics elements

Thermohydraulics diagram can accommodate some additional elements, i.e., transducers, which will allow parameters calculated by means of a thermohydraulics code to be read out and then recorded in the database under a process name. Each element of the design diagram contains a set of parameters, which can be transmitted to the signal database by means of the transducers.

It allows a mathematical model of the control system receiving signals from the hydraulic system transducers to be created in full accordance with an actual object.

In order to add a transducer onto an element of the thermohydraulics diagram proceed as follows:

1. Select “**TPP channel flow meter**” element from “**TPP transducers**” block bar (Figure 37).

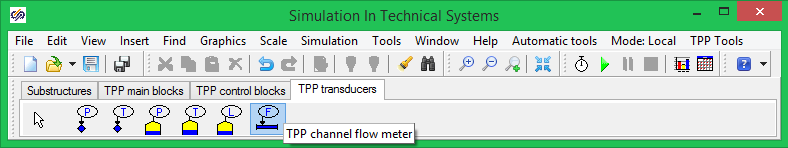


Figure 37. Element palette, “TPP channel flow meter” block

1. Set up the mouse cursor on **“Tube 15: Straight pipe”**.
2. Press the left mouse button.

If positioning is correct the newly added element will receive the name of the block owner, i.e. **“Tube15”**. Name of the block owner will be displayed in a pop-up help on navigating the mouse cursor upon that (Figure 38).

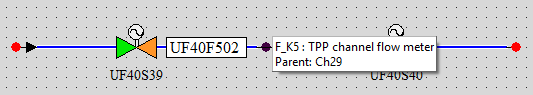


Figure 38. “TPP channel flow meter” block plugged into “Straight pipe” block

After the block is located on an element of the diagram it can be transferred to any point of the diagram window. The link between the block (flow meter transducer in this case) and the owner anyway will remain unchanged.

To change the block owner select **“Actions”** item in the pop-up menu and then “**Change owner”** (Figure 39).

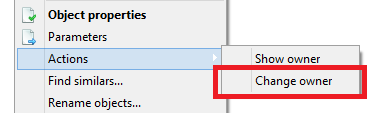


Figure 39. “Actions → Change owner” pop-up menu item

Double-click **“TPP channel flow meter”** element to edit the control point. Enter parent.g line in **“Value (source)”** field of displayed **“Control point editor”** dialog window (Figure 40). Value of the line is generated from the element name (parent) and parameter name (g) separated by a dot. In our case, **parent** word means and is replaced by the name of block owner. Tube15.g entry is the same.

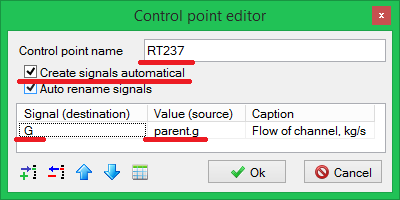


Figure 40. “Control point editor”

Tick **“Create signals automatical”** and **“Auto rename signals”** fields. Change the control point name for **“**RT237”.

Create a pressure transducer for “internal node” block.

1. Insert **“TPP node pressure transducer”** into **“Internal node”** element.
2. Set **“**RT238” line in “Control point name” field.
3. Set **“**parent.P\_” line in “Value (Source)” field.
4. Set **“Pressure of node, bar”** line in **“Signal (designation)”** field (Figure 41).

Thus, we have added two transducers on the diagram, one of which (**“RT237”**) measures flow rate in pipe and the second one (**“RT238”**) measures pressure in internal node.

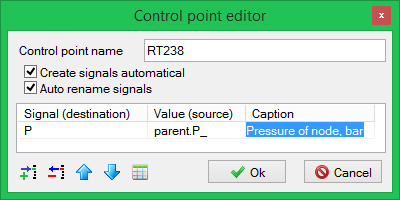


Figure 41. “Control point editor” window for internal node pressure

Start up a calculation task by pressing **“Calculation”** → “Run” item in the main menu.

If transducers are correctly added on the diagram then two new groups for “RT237” and “RT238” signals will appear in the database in “Transducers” category. Activate the database editor (**“Tools”**→ Database”item of the program main menu). Tick **“Signal view mode”** and **“Update with interval”** in **“Signal database editor”** (Figure 42). **“Database editor”** in this mode displays signal values calculated by the diagram. Make sure that the values given in the table for control points RT237 and RT238 meet the pipe flow rate and node pressure as per the diagram.

Since the database is saved in our example along with saving the automatics system the signal data base shall be saved manually. To that end, press button “Save” (Figure 42) and the database will be saved in a file named as **“**signals.db”.

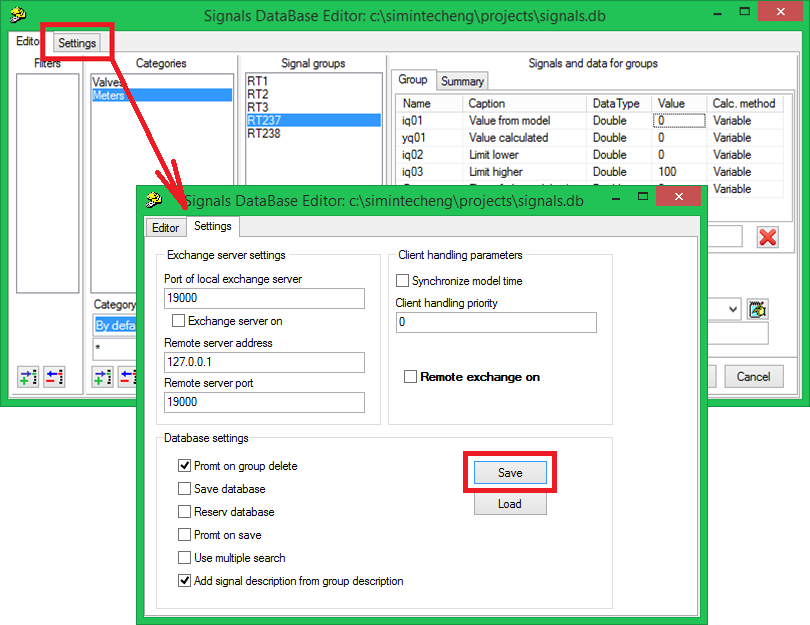


Figure 42. Database editor in signal value view mode for control point

## Change of valve names on diagram

Valves arranged on diagram have by default names corresponding to some equipment and instrument coding. Software complex facilities allow a template of names to be created by default for any elements. We use **“Z1”** and **“Z2”** namesin this training exercise and do not make any difference between valves and gates. Prior to proceeding with the next training exercise change valve names.

To change valve names proceed as follows:

1. Select the valve on the diagram.
2. Press the right mouse button.
3. Select **“Object properties”** item in the pop-up menu.
4. Go to **“Common”** tab in **“Properties”** dialog window.
5. Enter **“Z1”** in object name line (Figure 43).
6. Change the name of the second valve in the same way, **“Z2”**.

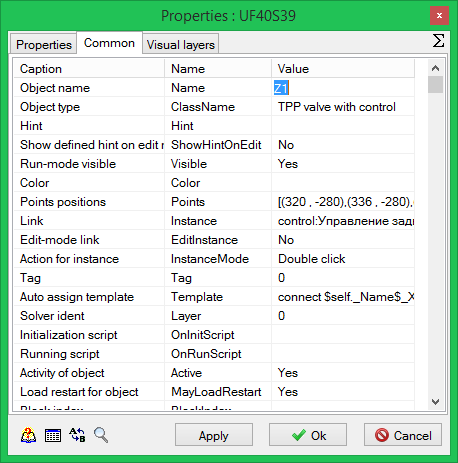


Figure 43. Change of valve name

# Creation of a simple control algorithm

## Creation of the simplest control algorithm

Open the file named **“**Automatics diagram 1.prt”. This file was created during execution of the first training exercise and adjusted for operation with the database saved in **“**signals.db” file. Make sure that the database contains signals created during execution of the first training exercise as well as **“Control points”** category and groups of **“RT237”** and **“RT238”** signals created and saved during generation of the thermohydraulics diagram in training exercise 3.

Go to **“Substructures”** tab of the block palette in the main program window and select **“Submodel”** block (Figure 44).

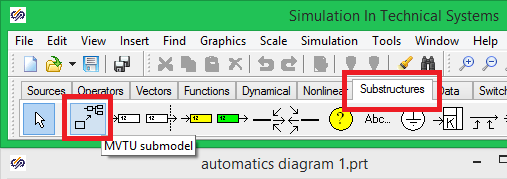


Figure 44. Selection of “Submodel” block in the block palette

Locate the selected block on the diagram window (Figure 45):

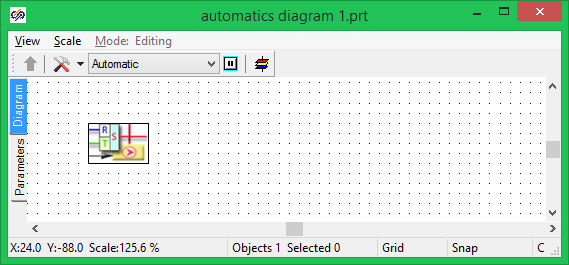


Figure 45. Automatics model diagram with “Submodel” block inserted

Double-click **“Submodel”** block. This action will lead to opening of the block internal structure. Since the new block still does not contain any elements, an empty diagram window will be activated (Figure 46).

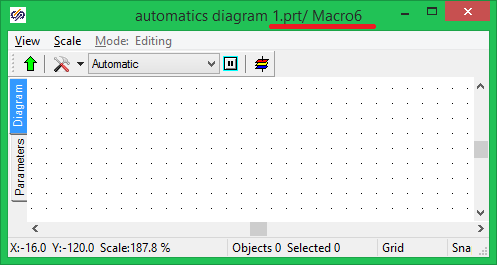


Figure 46. “Submodel” block internal structure diagram window

The window title after transferring to the submodel will contain the project file name and submodel name, within which the User is. This is **“…Automatics diagram** 1.prt/ Macro6” in this example (Figure 46).

Locate two **“Signal entering into the list”** blocks and one **“Signal reading from the list”** block from **“Data”** block bar of the main window block palette as shown in the next Figure (Figure 47):

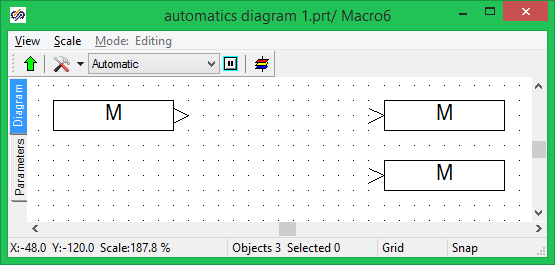


Figure 47. “Signal reading from the list” and “Signal entering into the list” blocks

## Link of blocks to database signals

These blocks will transmit data from control algorithms to the signal database and vice versa. To link a block to a particular database signal proceed as follows:

1. Select the block on the diagram window.
2. Press the right mouse button.
3. Select **“Object properties”** item in the pop-up menu (Figure 48).

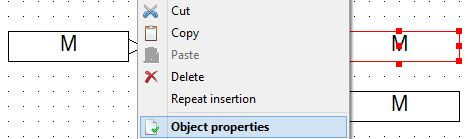


Figure 48. Diagram window element contextual menu

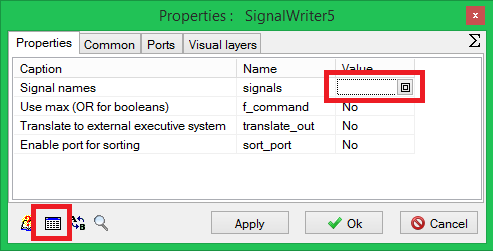


Figure 49. “Object properties” dialog window for “Signal entering into the list” block

After that **“Object properties**” dialog window for **“Signal entering into the list”** block will be displayed. Select **“Signal name”** line in this dialog window and press **“Fill from database”** button located at the foot of the dialog window (Figure 49).

On pressing this button, **“Database editor”** dialog window will be activated (Figure 50). Database signal and control system design diagram values are adjusted in this dialog window. This adjustment is completely the same as the one executed for the thermohydraulics model in training exercise 3: select successively in the dialog window:

1. **“Category”**.
2. **“Signal group”.**
3. **“Signal name**”.

In this training exercise the properties of **“State”** object shall be linked with **“Position”** signal in the database for the valve named Z1 (Figure 50). Select this signal and press **“Add”** button in **“Selected data”** panel (Figure 50). Delete available entries, when required.

A unique name will be generated for this signal consisting of the signal group name and signal name separated by an underline (Z1\_xq1 in this example). Press **“Ok”** button to close the editor window.

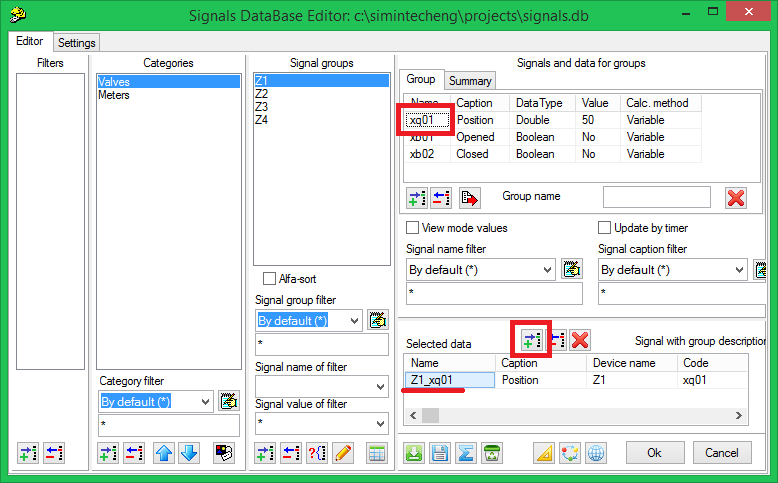


Figure 50. Selection of a signal in database for object properties

Signal name selected in the dialog window is displayed in the diagram window within a relevant block. Adjust the second **“Signal entering into the list”** block for the same database signal related, however, to the valve named as **“Z2”**.

Link **“Signal reading from the list”** block with **“Node pressure”** database signal belonging to **“Transducers”** categories and groups of RT238 signals.

If all actions are correctly executed then captions on blocks shall appear about in the same way as shown in the next Figure (Figure 51):

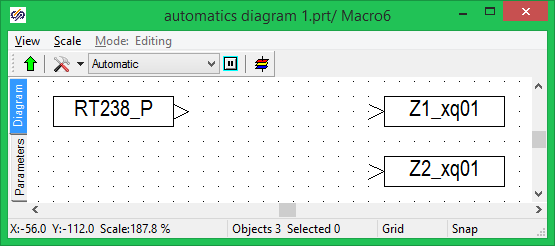


Figure 51. Submodel diagram after linking the block properties with the signal database

## Creation of the simplest control diagram

Locate additionally **“Constant”** block from **“Sources”** tab on the diagram. Set the value of block properties as equal to **“1.4”**.

Locate **“Step”** block from **“Sources”** tab on the diagram. Set the following values for the block properties:

* Response time **100**;
* Initial state **50**;
* Final state **10**.

Locate **“Comparator”** block from **“Operators”** tab on the diagram.

Locate **“Limited integrator”** block from **“Dynamic”** tab on the diagram. Set the following values for the block properties:

* Amplification ratio – **10**;
* Maximum value **100**;
* Minimum value **0**;
* Initial conditions **50**.

Link the blocks via lines to create a block diagram as it is shown in the Figure 52:

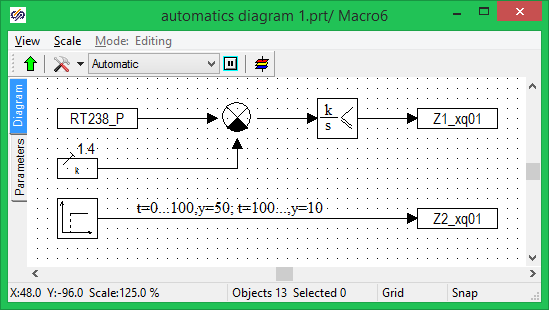


Figure 52. Control model diagram

Save the created project.

Created diagram of the simplest algorithm ensures pressure maintained at level 1.4 in the internal node due to opening of the valve at the first pipeline section.

**“Node pressure”** signal received from the database is compared with constant 1.4. Result of the comparison is transmitted to the limited integrator. If pressure exceeds the set-up value then positive mismatch will be generated at the comparator output and due to integrator amplification negative ration (amplification ration – 10) the valve will be closed if pressure is lower than set-up value and, thus, a negative signal is generated and the valve is opened.

Position of the second valve at the initial moment has value of 50 and is changed in stepwise manner for 10 in 100 seconds after commencement of calculation.

## Check of interchange with signal database

The control algorithm diagram created interchanges signals with the database in simulation mode. Since the thermohydraulics model is not connected to the database at the current stage then only signal entering and reading procedures can be checked.

Perform start of the diagram: **“Simulation”** menu item of the program main menu, **“Start”** sub-item (**Figure 53**).

|  |  |
| --- | --- |
| **Figure 53. Start of simulation** | **Figure 54. Activation of simulation** |

Enter the database editor and make sure that position of valves at the initial moment is 50.

Activate the diagram for simulation (“**Simulation**” item of the main window, **“Run”** sub-item (**Figure 54**)).

During simulation in the database a pressure signal applicable in the algorithm is equal to zero. The first valve, respectively, is opened immediately and its position is 100. The second valve is set to position 50. In 100 seconds of simulation **“Step”** block is activated and the second valve is set to position 10.

To watch such changes activate the database editor and tick **“View mode value”** and **“Upgrade with interval”** items (Figure 55).

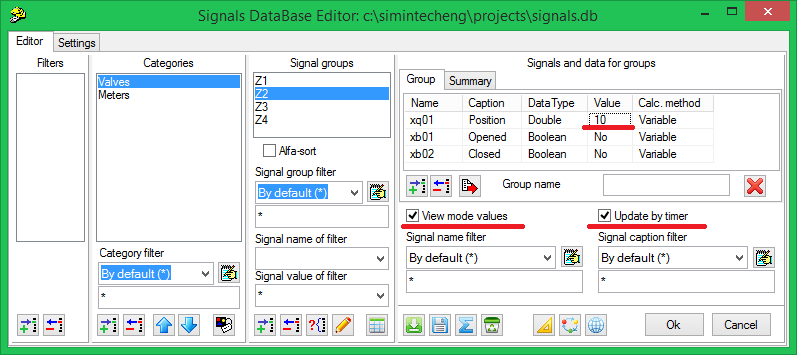


Figure 55. Database editor in signal value view mode

# Creation of complex model

## Creation of complex model

Complex model can consist of several mathematical models calculated by means of different calculation codes and operating together. To ensure data interchange the common signal database is used.

When executing the previous training exercises you have created two simplest mathematical models. Thermohydraulics model (**“TPP Diagram** 1.prt” file) and control system model (“Automatics diagram 1.prt” file) which both use the same signal database (”signals.bd” file).

To create the complex model proceed as follows:

1. Close all projects opened in SimInTech.
2. Select **“File”** item in the main menu of the program main window, then **“New project”**, then **“Pack”** (Figure 56).

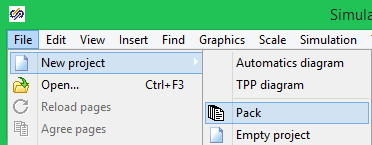


Figure 56. Complex model creation menu

1. **“Project pack”** window will be displayed, which is intended for creation and control of a complex mathematical model (Figure 57).

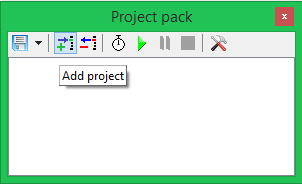


Figure 57. Project pack control window

1. Press **“Save Pack”** button and save the pack under name **“Pack1.pak”** in the same catalog, in which **“TPP diagram 1.prt”** and **“Automatics diagram 1.prt”** files are located.
2. Press **“Add project”** button on **“Project pack”** window. In the window that appears (Figure 58) select **“TPP Diagram 1.prt”** file with a hydraulic model created during previous sessions (file name can be entered manually or selected from a standard dialog window).

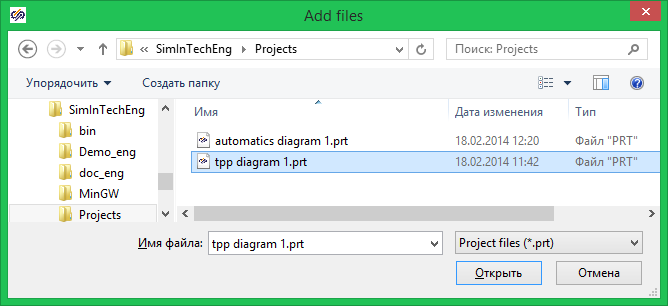


Figure 58. Add project window

1. Add **“**Automatics diagram 1.prt” file to the pack.

Every added project is automatically opened in SimInTech. Work with projects opened in such way (from the user’s point of view) is no different from work with projects opened separately. The user can switch over between windows and also edit and run every project individually.

## Calculation of complex model

Project control window shall appear as follows (Figure 59):

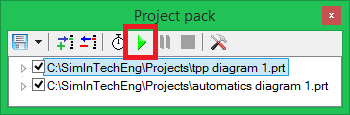


Figure 59. Project pack control window

Run the project rack using **“Run all”** button in the pack control window (Figure 59). In such a manner mathematical models of automatics and thermohydraulics included into the pack are started for calculation.

Wait until the model time reaches 30–50 seconds and press **“Pause”** button in **“Pack”** window. Model time can be watched in any diagram window. At the same time calculation is stopped in all opened mathematical models.

Since opened projects use the same signal database the end-to-end data exchange runs between applications, i.e., each design program at each step addresses the database for reading and/or recording signals.

In the created example the automatics diagram receives the pressure value for the intermediate node from the database and acts on the first valve so that the node pressure is maintained at level 1.4. Within the period of time between 30 and 50 seconds of calculation transitional processes of our simplest model are completed.

Go to the thermohydraulics model diagram window and click on a free space of the diagram window. On doing that, the window will be redrawn and captions above the valves will indicate their position in the given moment of calculation.

In Figure 60 one can see the state of the system, at which value 50 has been automatically received from the automatics diagram for the position of the second valve, while the position of the first valve is set by the simplest automatics controller at level 97.1 %.

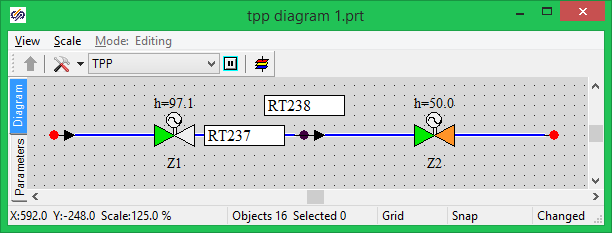


Figure 60. Thermohydraulics model diagram window at the moment of time of 45 sec

Press **“Run all”** button to continue, thus, the calculation of a complex mathematical model.

Stop the calculation on reaching the model time of 120–150 seconds.

In our model per 100 seconds of calculation a momentary change of position of the second valve from 50 to 10 takes place due to activation of **“Step”** block in the automatics system.

The momentary closing of the valve to position 10 results in stepwise rise of pressure in the intermediate node.

Our simplest controller close up the first valve to ensure the preset pressure value in the node. As a result, position of the first valve will be set up level 18.5 % (Figure 61).

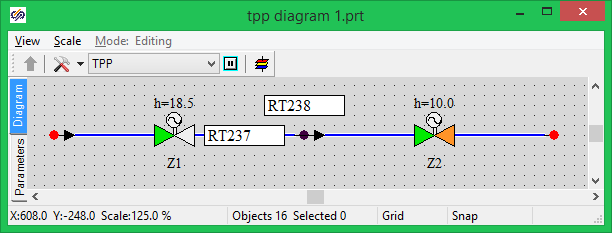
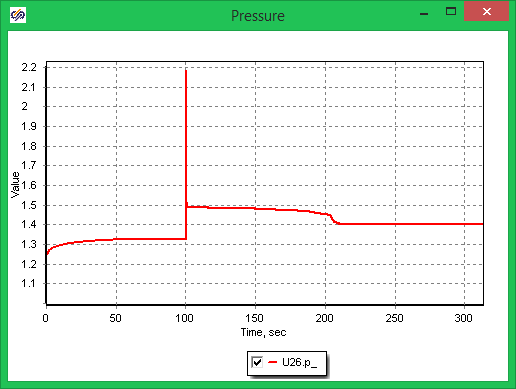


Figure 61. Thermohydraulics model diagram window, time – 125 sec

Pressure jump at the hundredth second of calculation can be seen on the pressure graphic in **“Internal node”** created in the process of creation of the hydraulics model (Figure 62).

Position of valves can be seen in the below Figures (Figure 63 and Figure 64).



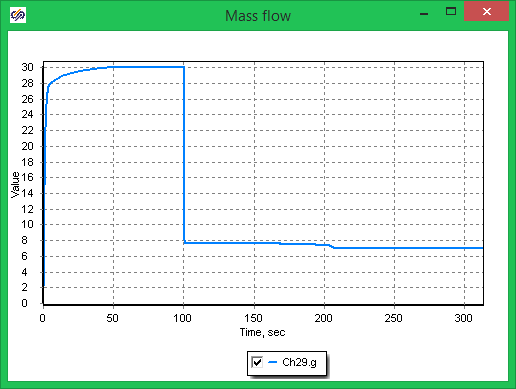


Figure 62. Internal node pressure and pipe flow rate graphic



Figure 63. Valve Z1 position graphic

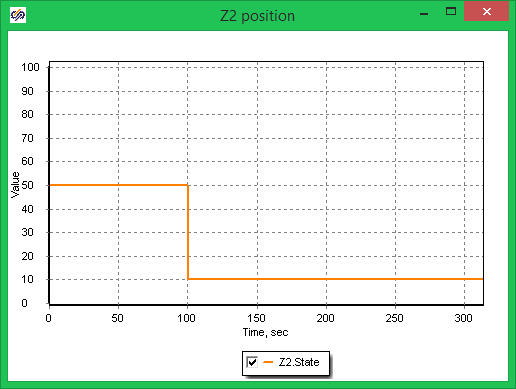


Figure 64. Valve Z2 position graphic

# Creation of equipment control block

## Creation of equipment control block

The simplest complex model was created during the previous training exercises consisting of thermohydraulics model calculated by means of TPP code, and automatics model interlinked via a common signal database.

This training exercise will demonstrate the option of creation of typical control blocks for SimInTech facilities.

During simulation of complicated systems a situation often occurs when the same typical mathematical model shall be repeatedly used.

The same mathematical model can be used in SimInTech for several objects of the same type using signal vector processing mechanism.

To demonstrate the capabilities of SimInTech we use the available database created during execution of previous training exercises. As the simplest example we will generate a valve control block and transducer processing block.

1. Open **“Automatics diagram 1.prt”** file created during execution of the previous training exercises.
2. Delete the existing **“Submodel”**.
3. Save the file under new name of **“**Automatics diagram 2.prt”.

To save the file select: **“File”** menu item of the program main menu, **“Save project as…”** sub-item (Figure 65).

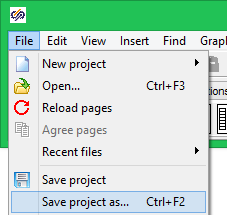


Figure 65. Menu for saving a file under a new name

Since the previously created file was linked with the database the file under a new name will be also linked with the same database.

Unlike the previous example when position of a valve was calculated in the automatics system and directly transmitted to the thermohydraulics model, we will try to generate a mathematical model of a valve, which will be more close to “actuality”. A new model of a valve will receive commands for opening and closing and, following signals received, change its position.

## Addition of new signals to the database

Commands for opening and closing of a valve will be transmitted via the signal database that is why existing categories shall be edited. We will use **“Valves”** category for data interchange with the valve in the training example.

1. Enter the **“Database editor”**. **“Tools”** menu item of the program main menu, **“Database”** sub-item (Figure 66).

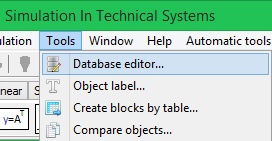


Figure 66. Data base editor activation menu

1. Select **“Valves”** category in the editor and press **“Set up category”** button (Figure 68).
2. Add two new signals in **“Category properties”** window (by means of **“Add signal”** button in the lower part of the window): **“Open command”** and **“Close command”** as shown in the Figure below (Figure 67).

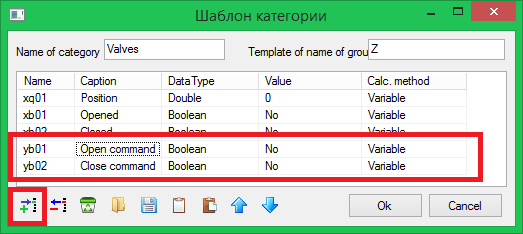


Figure 67. Category setting

1. Press **“Ok”** button to close **“Category properties”** window.

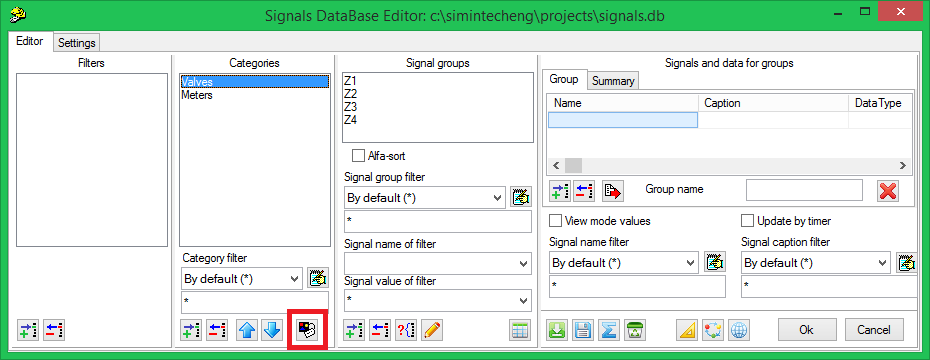


Figure 68. Database editor

1. Press **“Ok”** button to close **“Database editor”**.

## Creation of equipment control block

1. Locate the new **“Submodel”** block from **“Substructures”** tab on the diagram.
2. Double-click under the new block.
3. Enter **“Equipment control”** submodel name in a displayed field (Figure 69).

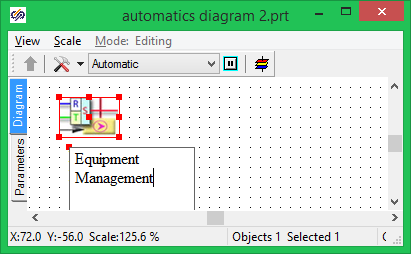


Figure 69. Creation of block caption

1. Double-click **“Equipment control”** block.
2. Locate the new “Submodel” block from “Substructures” tab on the diagram and subscribe that as **“VCU”**.

In this block we will create the simplest model of a valve. There will be two **“Open”** and **“Close”** commands at the input to the model. At the output we will obtain valve position, its state (open or closed valve).

1. Double-click **“VCU”** block.
2. Locate two **“Input port”** blocks and three **“Output port”** blocks on the diagram from **“Substructures”** tab (Figure 70).

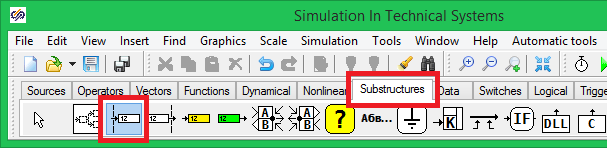


Figure 70. “Input port” block in block palette

1. Double-click **“Input Port”** block.
2. Enter **“Open command”** line for the first input port and **“Close command”** for the second one in **“Submodel port”** dialog window (Figure 71).

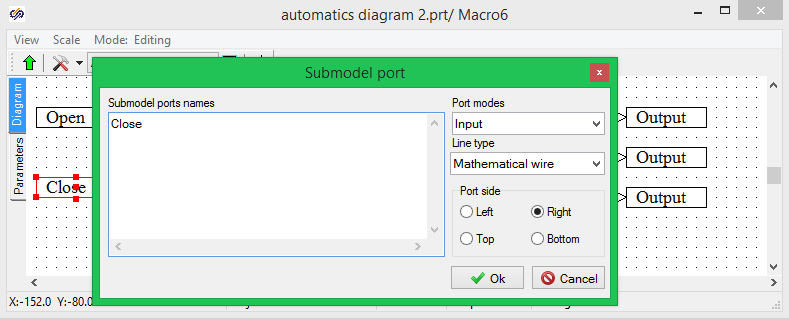


Figure 71. Edit submodel port name

1. Change names of submodel port blocks so that the diagram appears like it is shown in the Figure below (Figure 72).

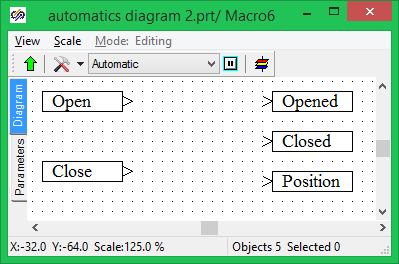


Figure 72. VCU submodel diagram with re-named ports

To ensure transition up to one nesting level double-click the empty space of the diagram. Please note that after addition of ports to the submodel diagram ports, which can be provided with communication links, have been added to its image on the diagram. On navigating the cursor on any port a help will be popped up with the port name assigned inside the model (Figure 73).

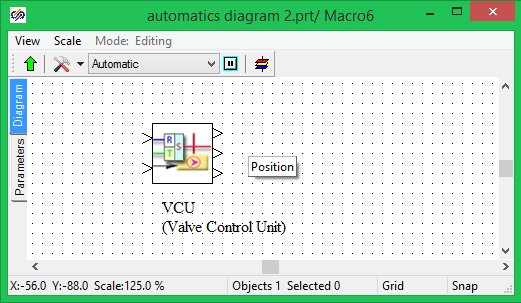


Figure 73. VCU submodel block after adding ports

## Vector processing of signals

Valve control block to be created by us will be then common one for all valves used in this training project. Thus, all signals processed by this block will be vector ones and every port will process as many signals as valves exist in the database.

1. Locate two **“Reading from signal list”** and three **“Recording in signal list”** on the diagram next to **“VCU block”.**

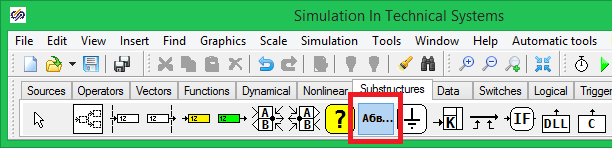


Figure 74. Note block in component palette

1. To make further work more comfortable locate also **“Note”** blocks on the diagram, which allowing captions to be entered into the diagram.
2. Double-click **“Text”** caption of **“Note”** block and enter a clarification text for every block into the editor window. The diagram shall appear like it is presented in the Figure 75.

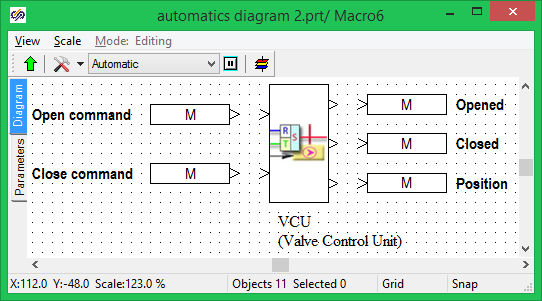


Figure 75. Equipment control block diagram

**“Signal reading from the list”** and **“Signal entering into the list”** blocks were used in the previous training exercises for reading and entering a block signal in the database. In the current training exercise we will receive not a single signal but a signal array for all valves available in the database. To this end, perform the following operations:

1. Double-click **“Signal reading from the list”** block.
2. In **“Properties”** displayed dialog window switch over to **“Signal names”** line and press the text editor activation button that is displayed in the process of edition of a line and arranged at the right from the text (Figure 76).

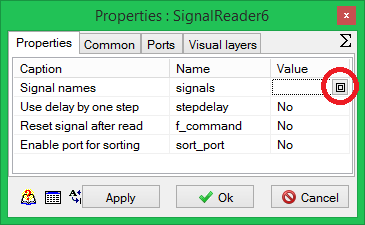


Figure 76. “Signal reading from the list” block properties editing window

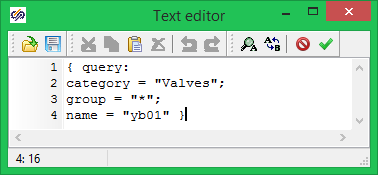


Figure 77. Text editor for a query to the database

1. A query to the database shall be generated in the editor text window.

The query is a text appearing as a single line or several lines in squiggle brackets:

**{**

query:

category = “Valves”;

group = “\*”;

name = “yb01”

}

, where query: is a keyword.

This query consists of three fields:

category = “Valves” – name of a category (“Valves”), from which we want to receive signals.

group = “\*” – name of a signal group to be included into the query. Sign \* means that the query shall be included into all groups of signals of the given category. Note that the database allows the group of signals to be selected using the filter.

For example, if there are valves whose names begin with letter D then to receive signals from such valves it will be enough to write group="D\*".

name="yb01" is the name of a signal that we desire to receive from the database. In this case this name corresponds to **“Command Open”** signal. We assigned this name was while generating the category properties (Figure 67).

1. Close the text editor by pressing **“Apply”** in the upper part of the window (Figure 77).
2. Press **“Ok”** button to close the block property editor window.
3. Redo items 1–5 for all signal reading and recording blocks on the diagram. At the same time query lines corresponding to assignment of signals in the database shall be entered:

**“Command Close”:**

{query: category = “Valves”; group = “\*”; name = “yb02”}

“Opened”:

{query: category = “Valves”; group = “\*”; name = “xb01”}

**“Closed”:**

{query: category = “Valves”; group = “\*”; name = “xb02”}

**“Position”:**

{query: category = “Valves”; group = “\*”; name = “xq01”}

1. Add **“Time schedule”** block to the diagram and connect blocks with communication links as shown in the Figure below (Figure 78):

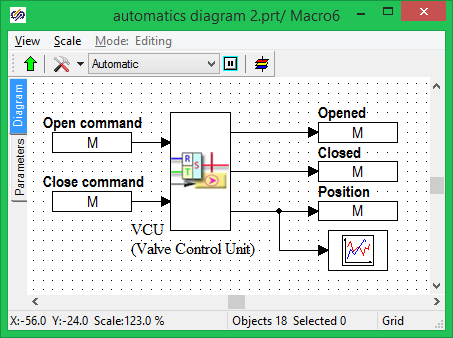


Figure 78. Equipment control submodel after linking the blocks

## Edition of “new” block parameters

Created diagram allows a signal vector to be received from the database for all valves and be sent for processing to VCU unit; processing results will be sent to the database also as a vector signal.

To correctly process vector signals their number, as a rule, shall be known. The number of signals in out model is equal to the number of valves in the signal database. In our example we will add the number of valves as a new parameter of VCU unit.

SimInTech allows new blocks with a random set of parameters on the basis of approved and tried submodels to be generated. The created model can be closed for edition and used as a black box.

To add a new parameter proceed as follows:

1. Select **“Submodel “VCU”** on the diagram.
2. Select **“Change block…”** sub-item of **“Edit”** menu item (Figure 79).

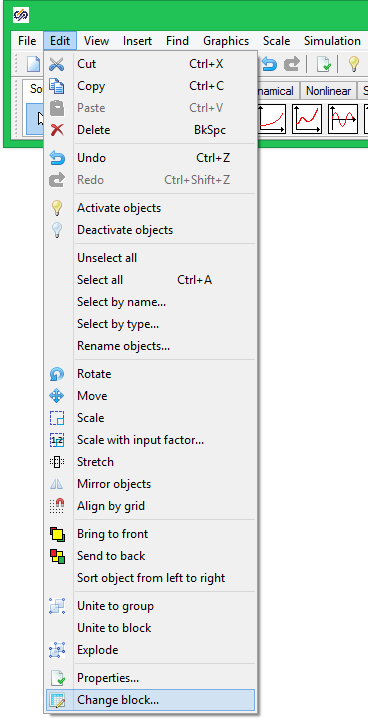


Figure 79. Change block menu

1. Press **“Add properties”** button in the lower part of **“New block editor”** dialog window. Enter the following values of fields:

Title – **Number of valves;**

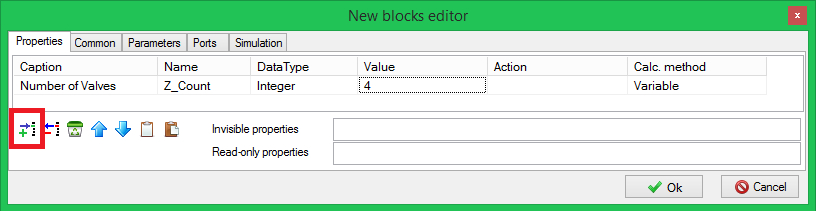
Name – Z\_Count;

Data type – **Integer**;

Value: {query: category=”Valves”; group=\*; name=”\*xb01”; what=count}

(Figure 80).

A query to the database, which returns the number of signals with **“**xb01” text in their name from “Valves” category, is generated in **“Value”** line. Since there is only one **“**xb01” signal in all valves then the returned value corresponds to the number of valves in the signal database.



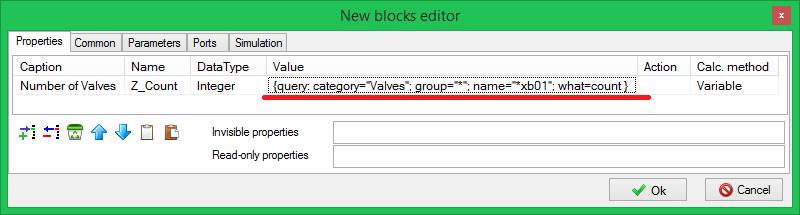


Figure 80. New block editor window

1. Close **“New block editor”** dialog window by pressing **“Ok”** button.

## Creation of valve control model

The simplest valve control model will operate as per the following principle. Model input receives two logic-type signals: **“Command “Open”** and **“Command “Close”.**

In case the both commands are equal to **“0”** (logical False) or the both commands are equal to **“1”** (logical True) the valve does not change its position.

Still **“Command Open”** is equal to **“1”** and **“Command Close”** is equal to **“0”** the control unit is changing the position of the valve increasing it at a constant speed until one of the following conditions is reached:

1 – **“Position”** of the valve is equal to **“100”** (fully open);

2 – **“Command Open”** becomes equal to **“0”**.

Still **“Command Open”** is equal to **“1”** and **“Command Close”** is equal to **“0”** the control unit is changing the position of the valve increasing it at a constant speed until one of the following conditions is reached:

1 – **“Position”** of the valve is equal to **“0”** (fully closed);

2 – **“Command Close”** is equal to **“0”**.

When **“Position”** values reaches **“100”** or **“0”** the block outputs named **“Open”** or **“Closed”**, correspondingly, take **“1”** values (logical True), otherwise those are equal to **“0”** (logical False).

Enter **“VCU”** submodeland make up a diagram as shown in the Figure 81:

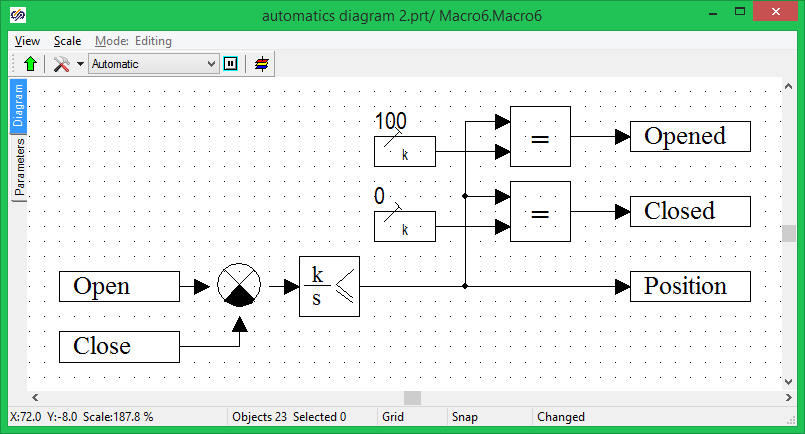


Figure 81. VCU block diagram

Set weight factor values [1,-1] for **“Adder”** block.

Thus, there will be **“1”** at the output of **“Adder”** block when **“Command Open”** is equal to **“1”** and **“Command Close”** is equal to **“0”**. There will be **“-1”** at the output of **“Adder”** block when **“Command Open”** is equal to **“0”** and **“Command Close”** is equal to **“1”**.

In all other cases there will be zero value **“0”** at the output of the adder.

Since all signals are vector ones then a signal array will be transmitted in every link.

To generate **“Open”** and **“Closed”** signals **“Logical operations”** logical block from **“Logical”** tab is used (Figure 82).

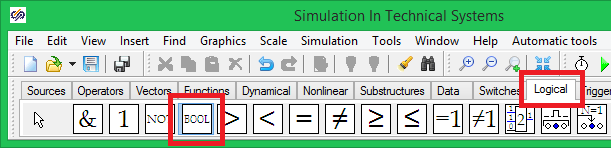


Figure 82. Logical operations block

This block allows a logical operation to be carried out for a vector signal. In our case we are comparing the vector signal with the constant. The unit checks every signals of a vector included into the block and returns a vector in which signals equal to the constant obtain **“1”** value (logical True).

The type of the second block, **“Scalar”**, shall be specified in the block settings.

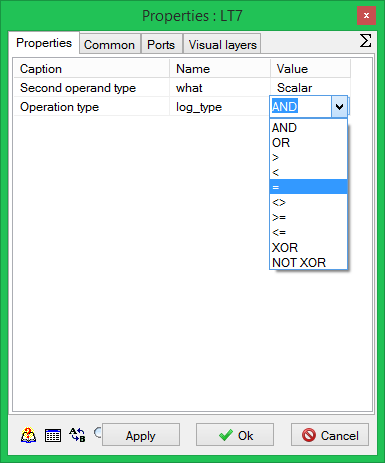


Figure 83. Edition of “Logical operations” block

**“LimitIntegrator”** block implements the calculation of position of valves using input signals: If the input is **“-1”** then it reduces the position, if the input is **“+1”** it increases the position.

In our simplest example we assume that the position is changing in a linear fashion at a constant rate, which is the same for all valves. The change rate is proportional to the integrator amplification ratio. To process the vector of signals a vector of values shall be set for every block parameter.

For example, if the block calculates 4 signals then four amplification ratios shall be introduced for each signal in the vector: [1,1,1,1]. 4#1 entry can be used for the same values.

Enter the editor window for **“LimitIntegrator”** block properties and set the following properties for the block:

– amplification ratios – Z\_Count#1;

– minimum value – Z\_Count#0;

– maximum value – Z\_Count#100;

– initial conditions – Z\_Count#50.

Press **“Ok”** button to close the property editor window.

Z\_Count#1 entry allows the array of values to be filled for the vector. Depending on **“**Z\_Count” value the different array of values is generated. Since **“**Z\_Count” is calculated by means of addressing to the database it will correspond to a number of signals coming from the database. Thus, this block will always correctly process the vector of signals in case of a change of a number of valves in the database.

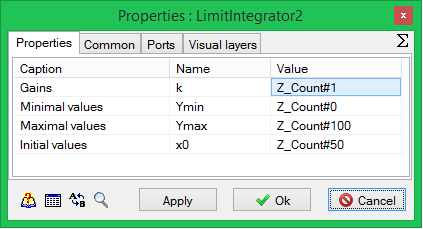


Figure 84. Edition of “LimitIntegrator” block

In case the diagram has been correctly selected then it can be activated for calculation to make sure that a signal vector consisting of four numerals corresponding to the number of valves in the database is being transmitted via all links (Figure 85).

Meanwhile, **“Closed”** signal value (on accordance with initial conditions of the integrator) are set in the database for all valves.

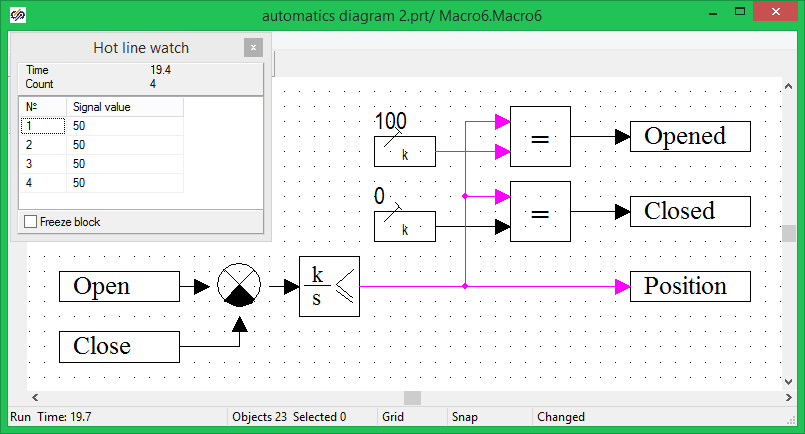


Figure 85. VCU diagram checking

Save the created model.

# Creation of a simple control algorithm

## Creation of control algorithm

In the process of development of control systems for complex objects the task of division of a complex system into relatively simple control algorithms is to be frequently solved. In this training exercise we will demonstrate the capabilities of SimInTech related to creation and formatting of such algorithms.

When executing the sixth training exercise, we simulated an equipment management block. In this training exercise we will make up two control algorithms using the created unit. In this training exercise we do not distinguish between operation of valves and gates.

Open **“Automatics diagram** 2.prj” project file created in the previous class.

Add two new **“Submodel”** blocks from **“Substructures”** tab to the diagram. Create captions under **“Valve Z1 control algorithm”** and **“Valve Z2 control algorithm”** blocks to make the diagram appear like the Figure below (Figure 86).

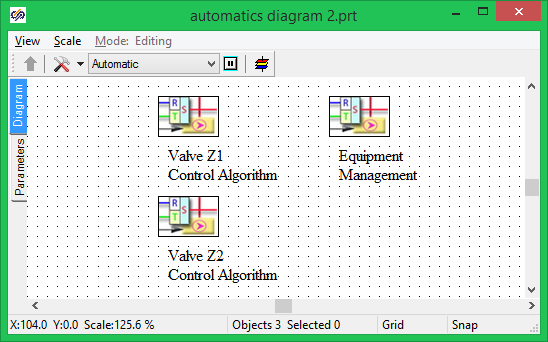


Figure 86. Control model diagram

These algorithms will control two valves in the thermohydraulics model created in the process of training exercise 3.

## First valve control algorithm

Enter **“Valve Z1 control algorithm”** and locate **“Signal reading”** block onto the diagram (Figure 87). This block allows signals to be received from the database in the same manner as **“Signal reading from the list”** block.

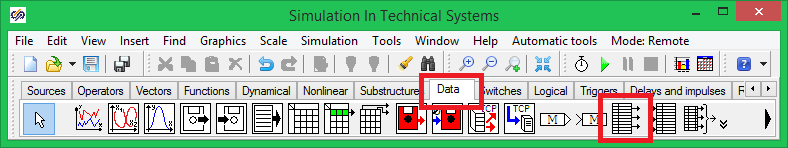


Figure 87. “Signal reading” block

Locate **“Algorithm output”** block onto the diagram (Figure 88). The considered block allows not only signal values to be entered into the data base but also new signals to be created.

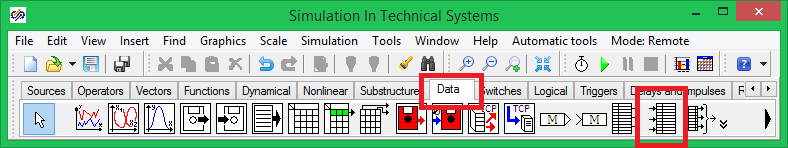


Figure 88. “Algorithm output” block

The diagram shall appear as it is presented in the following Figure (Figure 89).

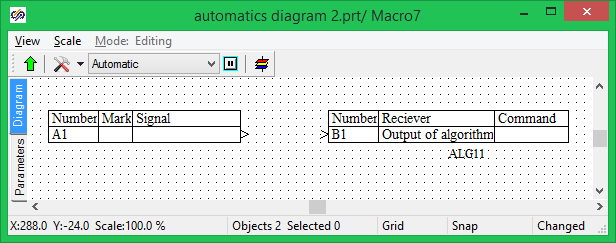


Figure 89. Algorithm diagram with added blocks

## Edition of “Signal reading” and “Algorithm output” blocks

Please note the format of these blocks. By default those are executed as a table, which allows a control algorithm to be created whose appearance is approximated to the appearance of functional plans applicable for designing control systems. The user can also set the appearance and content of tables.

Double-click **“Signal reading”** block. Reading/entering set-up dialog window will be displayed as shown in the Figure below (Figure 90).

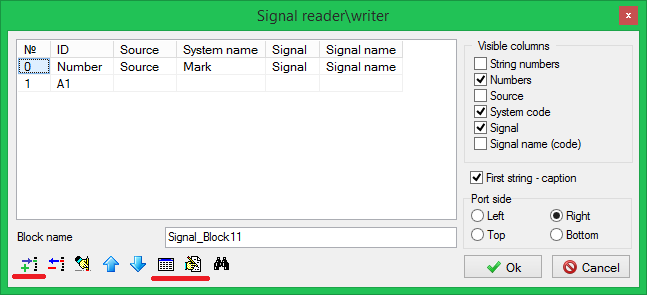


Figure 90. Signal reading block editor window

In this **“Signal reading/entering editor”** dialog window the appearance of a table and list of signals received by means of this block can be set. To set a number of displayed columns use “**Visible columns”** settings. To set lines use a set of buttons in the right lower corner.

To add lines press **“Add line”** button.

To add empty lines to the table add a line first, then remove the text from **“ID”** column.

To fill a line select a required line and press **“Fill selected from base”** button.

We will use **“Intermediate node pressure”** signal in the algorithm for controlling the first valve. To obtain the said signal perform the following operations:

1. Select the first line.

2. Press “Fill selected from base” button.

3. Select successively in the displayed **“Database editor”** window: Categories – “Transducers”; signal group – “RT238”; signals and data for groups – “Р” (Figure 91).

4. Press **“Add”** button in **“Selected data”** panel (Figure 91).

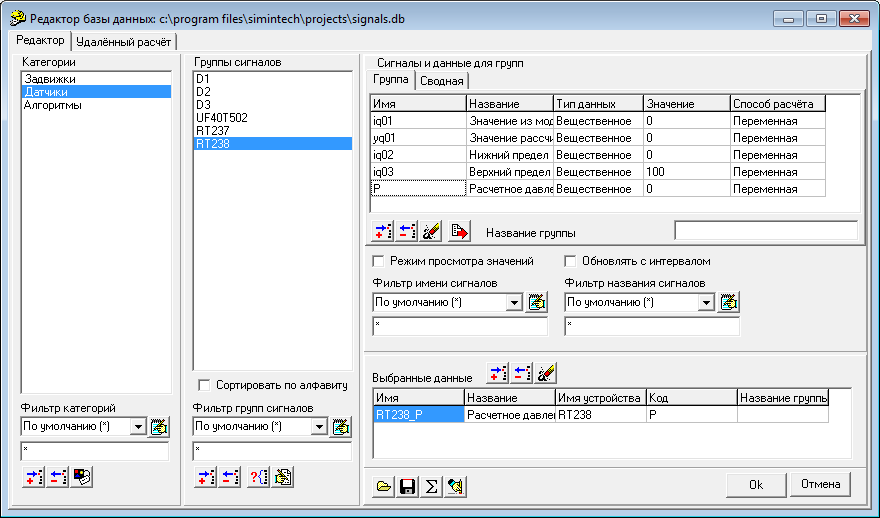


Figure 91. Database editor window

5. Press **“Ok”** button to close the database editor.

6. Set up the appearance using **“Signal reading/entering editor”** dialog window as it is shown in the following Figure (Figure 92).

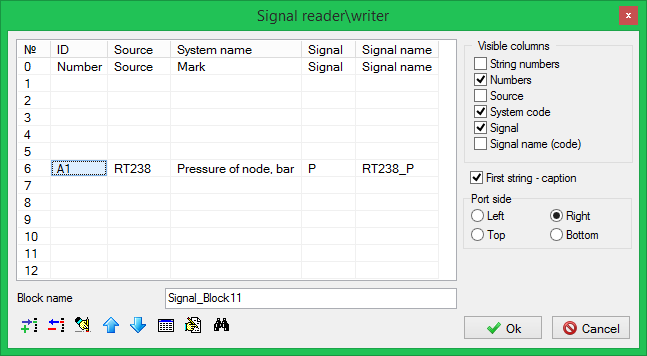


Figure 92. Setting of signal reading block

7. Press **“Ok”** button to close **“Signal reading/entering editor”** window.

Set up **“Algorithm output”** block by selecting first valve control signals from the database: **“Command Open”** and **“Command Close”.** Enter **“B1”** for **“Command Open”** line and **“B2”** for **“Command Close”** line in **ID** column. Enter **“a101”** line into **“Algorithm name”** field (Figure 93).

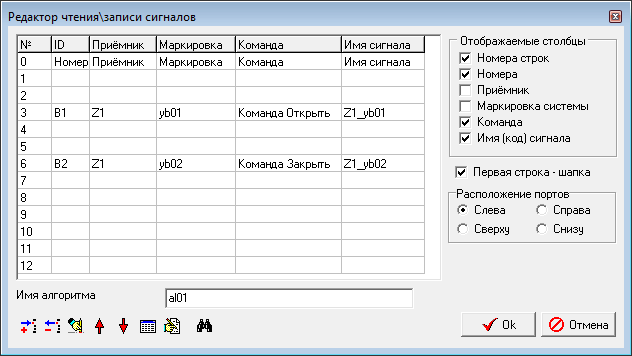


Figure 93. Setting of “Algorithm output” block

If all settings are correct the diagram window of the first valve control unit shall contain two tables in the way depicted in the Figure below (Figure 94):

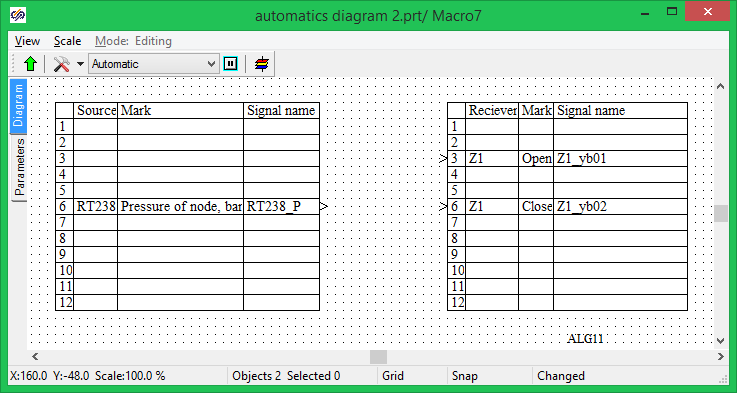


Figure 94. Appearance of the diagram after setting up “Signal reading” and “Algorithm output” blocks

## First valve control block diagram

In the first variant of the control system we calculated the required position of a valve and transmitted that directly to the thermohydraulics model. In the new variant we will control the valve using two commands **“Open”** and **“Close”**.

Our simplest algorithm will compare **“Pressure of node”** signal with a preset constant. In case pressure is higher than the preset value, a command to open the valve is generated, if pressure is lower than that a command to close the valve is generated. Block diagram of the algorithm made up of a standard blocks is presented in the Figure below (Figure 95). Now try yourself to make up the diagram for the considered algorithm.

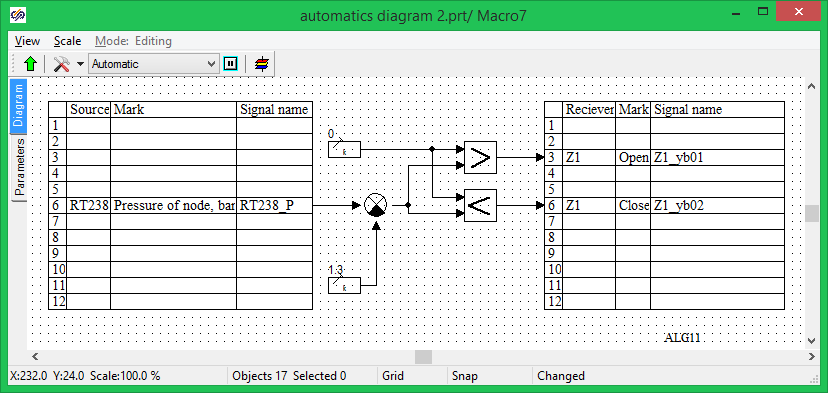


Figure 95. Appearance of the diagram after setting up “Signal reading” and “Algorithm output” blocks

## Check of first valve control algorithm operation

Let us run the created diagram for calculation to check the algorithm operation. The control algorithm diagram will have the same appearance as it is shown in the Figure below (Figure 96):

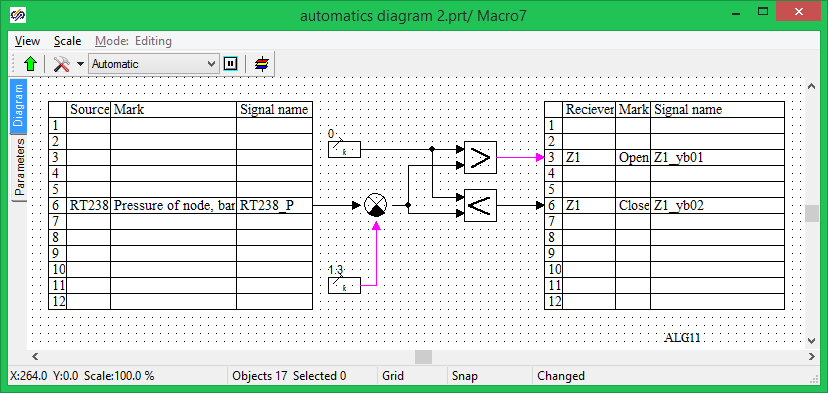


Figure 96. Algorithm diagram appearance in the process of modeling

During calculation links, whose value is above zero, are highlighted red (purple). Since the calculation runs without connecting the thermohydraulic model, the value of **“Pressure of node”** signal in the database is equal to zero. Comparator, thus, has **“-1,3”** at its output and comparing units generate **“1”** signal value (logical True) for **“Command Open”** signal.

Go to the equipment management block and double-click the link between **“Command Open”** and **“VCU”** blocks (Figure 97, such a line has an internal name in SimInTech – “MBTYWire : Mathematical wire”). The displayed **“Hot line watch”** indicates the list of values of **“Command Open”** signal for all valves entered into the database. Since control is implemented by means of the first valve then only the first element of the list has the value equal to **“1”** (Figure 97).

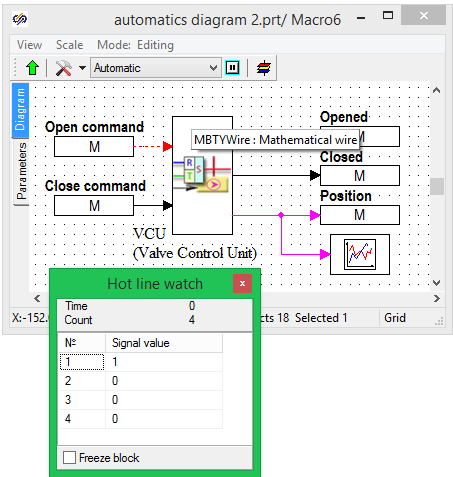


Figure 97. Appearance of “Equipment management” submodel in the process of modeling

Double-click the graphic the position of valves is transmitted to from VCU block. The created valve control model will process **“Command Open”** signal for the first valve and change the position of the valve at a constant rate. On reaching **“100”** position (completely open), the position is not changed; other valves do not move and stay in **“0”** position (completely closed, Figure 98):

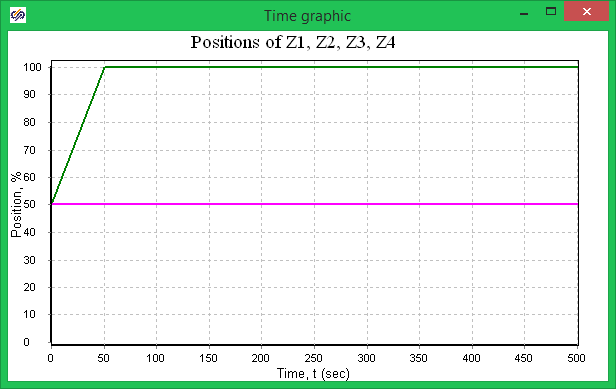


Figure 98. Gate valve Z1 repositioning graphic

## Second valve control algorithm

To control the second valve we will create the simplest algorithm, which will change the position of the valve in accordance with a preset sequence.

The sequence of repositioning will be set by **“Piecewise-linear”** (dependence) block from **“Sources”** tab (Figure 99):

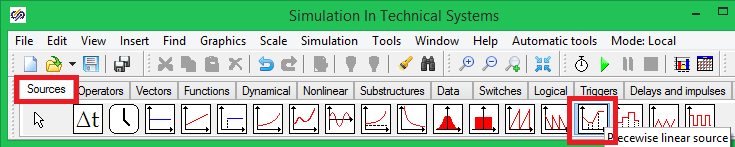


Figure 99. “Piecewise-linear” (dependence) block

The control algorithm compares a current position of the valve with the preset position obtained by means of **“Piecewise-linear”** block from **“Sources”** tab. In case of any discrepancy a command for opening or closing the valve will be issued.

To remove continuous activations and deactivations of the valve a relay link with a non-dead zone is used – **“Relay link with dead zone”** block from **“Nonlinear”** tab. This block allows the command not to be sent in case the position error is within the preset accuracy limit.

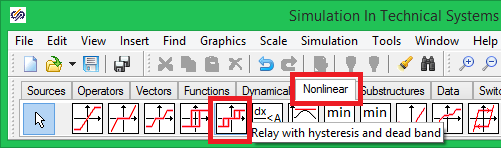


Figure 100. “Relay with dead zone” block

The second valve control algorithm diagram is given in the following Figure (Figure 101). The same **“Signal reading”** and **“Algorithm output”** blocks are used in the diagram of the considered algorithm as we used in the first valve control algorithm.

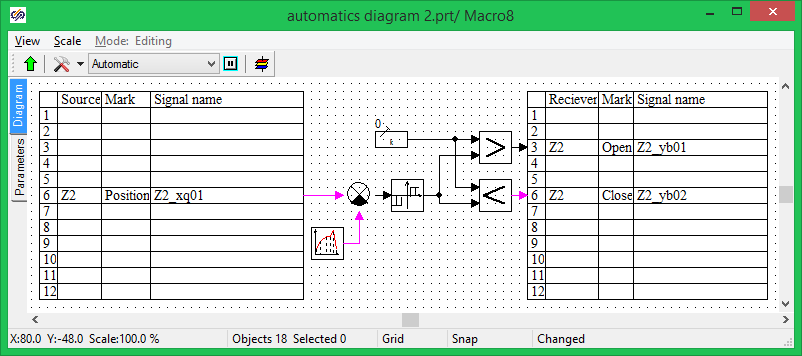


Figure 101. Second valve control diagram

Enter **“Valve Z2 control algorithm”** submodeland make up a diagram as shown in the Figure 101. Set parameters as shown in the following Figure for **“Signal reading”** block (Figure 102):

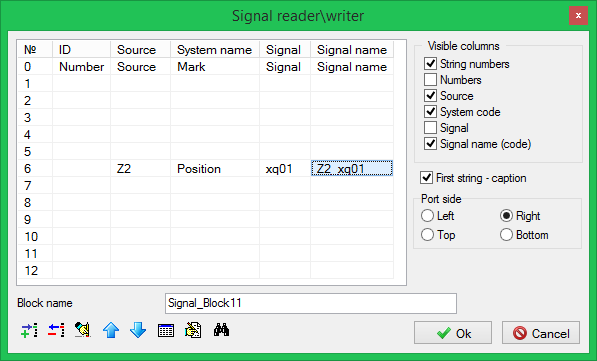


Figure 102. Signal reading block parameters

Set parameters as shown in the following Figure for **“Algorithm output”** block (Figure 103).

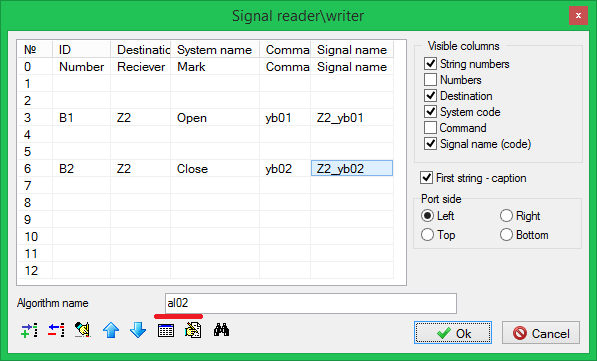


Figure 103. Algorithm output block parameters

Set parameters as shown in the following Figure for **“Relay with dead zone”** block (Figure 104):

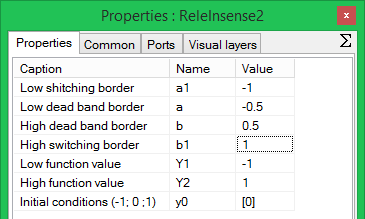


Figure 104. Relay with dead zone block parameters

The considered relay works as follows: **y(t)** block output takes one of three values Y1,0,Y2 or is not changed: y(t) = y(t – dt), where y(t – dt) is for the output value at the previous integration step (at the previous step of numerical calculation).

y(t) = Y1, if x(t) < a1 – the relay is switched over to a lower value on a drop of input action below the lower limit for switch-over.

y(t) = Y2, if x(t) > b1 – the relay is switched over to an upper value on an increase of input action above the upper limit for switch-over.

y(t) = 0, if a < x(t) < b – the relay is in zero position if the output value falls within the dead zone.

y(t) = y(t-dt), if a1 ≤ x ( t ) ≤ a or b ≤ x ( t ) ≤ b2 – the relay value is not changed, input action does not cross the limit for switch-over.

Preset block parameters (Figure 104) allows the valve position to be adjusted with accuracy up to ±0.01.

The second valve positioned uses the linear time dependence. **“Piecewise-linear”** block allows an array of function values to be set for different moments of time. Linear measurement of the block output is carried out between preset points.

Let us set the following algorithm for **“Piecewise-linear”** block:

within time interval 0–100 seconds the value is 10;

within time interval 100–200 seconds the value rises up to 40;

within time interval 200–300 seconds the value is 40;

within time interval 300–400 the value drops down to 20;

within time interval 400–500 seconds the value is 20.

To implement this algorithm set parameters Figure for **“Piecewise linear”** block as shown in the following (Figure 105):

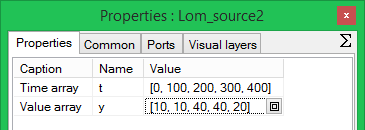


Figure 105. “Piecewise-linear” block parameters

## Check of model work

To ensure an implicit inspection of the algorithm work the modeling process can be decelerated. To this end, press **“Simulation properties”** button on the diagram window (Figure 106)

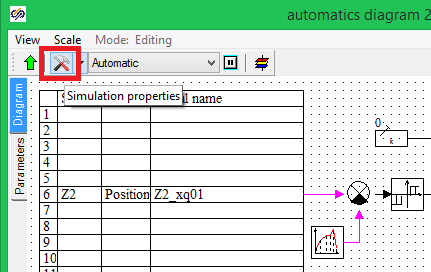


Figure 106. Simulation parameters activation button

In **“Solver properties”** dialog window go to **“Synchronization”** tab and tick **“Synchronize with real time”**. Set the acceleration ratio **1** (Figure 107).

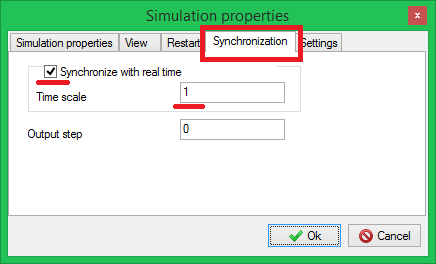


Figure 107. Setting of simulation rate

Go to **“Simulation properties”** tab and set up the final time of simulation as equal to 500 seconds (Figure 108).

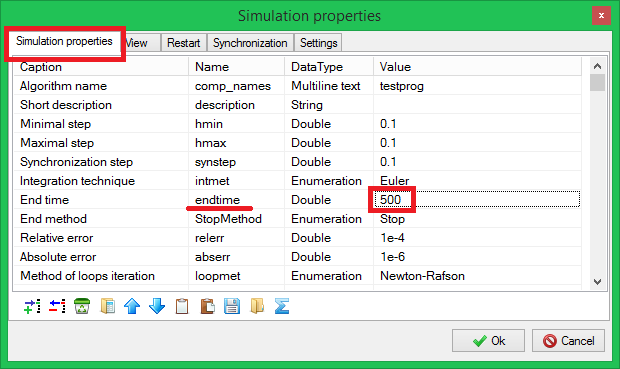


Figure 108. Setting of final simulation time

Run the created diagram for simulation. Stop the simulation in 3–5 seconds after its commencement by pressing **“Pause”** button in the main program window.

Go to **“Equipment management”** submodel and double-click the link between **“Command Open”** and **“VCU”** blocks (Figure 97). The displayed **“Hot line watch”** indicates the list of values of **“Command Open”** signal for all valves entered into the database. Since we added the second valve control algorithm at the initial moment of time two first elements of the list have the value equal to **“1”** (Figure 109).

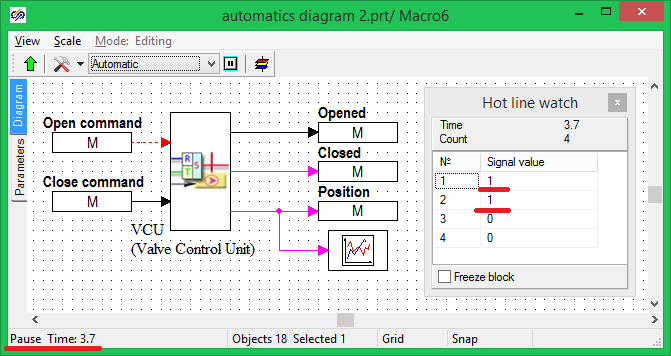


Figure 109. Appearance of “Equipment management” submodel in the process of modeling

Open **“Database editor”**, set **“View mode value”** and make sure that the value of signals for valves in the database corresponds to the value simulated in the model (Figure 110).

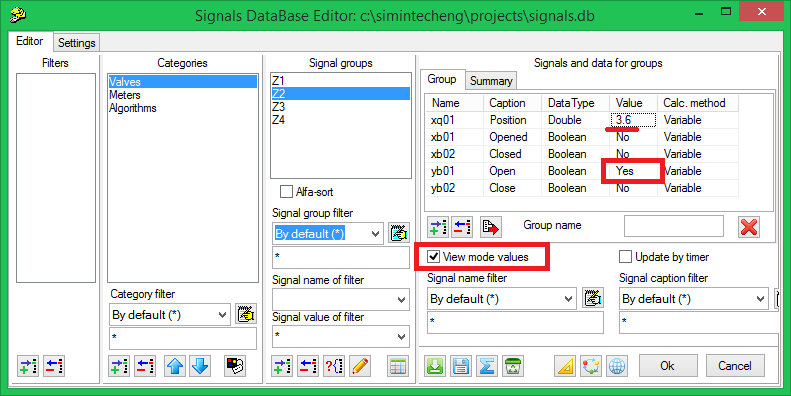


Figure 110. Database signal value

In the process of creation of an algorithm using **“Algorithm output”** block new algorithms are automatically added to the database to enable usability of algorithm outputs as input actions in other parts of the control system model.

In our case a new **“Algorithms”** category was generated in the database and two algorithms **“al01”** and **“al02”** were added whose signals correspond the values calculated in the mathematical model (Figure 111).

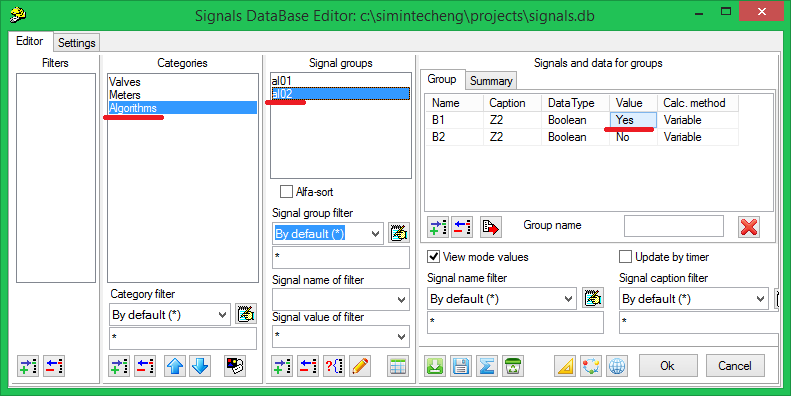


Figure 111. Database new signals

Proceed with calculation of the model (by pressing **“Start”** button of the main control panel). Go to **“VCU”** submodel, double-click the valve position graphic and make sure that the graphic is approximately corresponds to the one depicted in the Figure below (Figure 112):

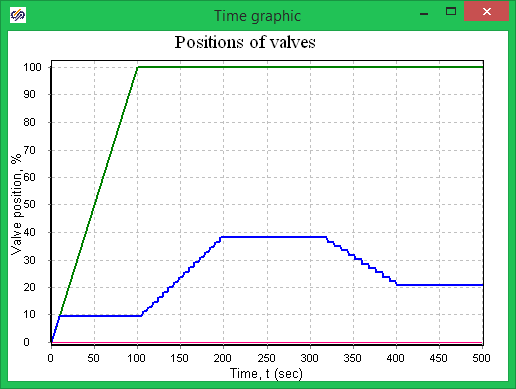


Figure 112. Positions of valves in the process of diagram modeling

The first valve is fully opened. The second valve will change its position according to the linear dependence set in the control algorithm. Since there is **“Relay with dead zone”** block in the second valve control algorithm the position will be changed in a stepwise manner.

# Change of complex model

Application of signal database in SimInTech allows complex models to be easily generated from several projects generated earlier. Application of the same database for signal interchange is the main condition for that.

The complex model generated in the process of execution of training exercise 5 can be easily changed by replacing one control system with another one. In this training exercise we will test the control system model (from training exercise 5) along with thermo-hydraulics model from training exercise 3.

Open **“**pack1.pak” file of the complex control model from training exercise 5. Select **“Automatics diagram 1.prt”** project in the pack control window and press **“Delete project”** button (Figure 113).

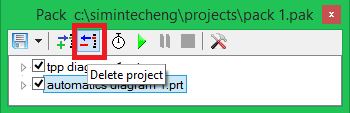


Figure 113. Project control window?

Press **“Add project”** button and select a model project created during execution of training exercise 7 in the standard open file menu – **“Automatics diagram 2.prt”**.

Make sure that the database contains all new signals created during execution of training exercise 7.

## Check of complex model

Run the complex model for calculation using **“Start”** button in **“Pack”** control window. If projects have been created without mistakes the calculation of a complex model shall be started.

Click the hydraulics model diagram.

Captions under the valves shall indicate their positions as received from the control system. For example, at the 20th second of calculation the second valve is in **9.3%** position, while the first one continues being opened/closed in accordance with the algorithm and stays within **10-12%.**

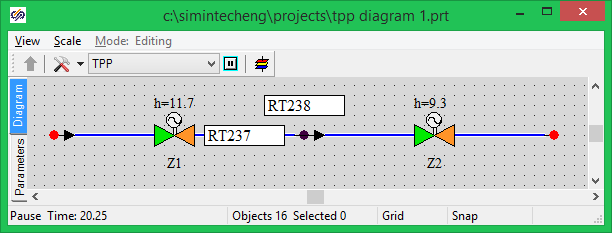


Figure 114. Hydraulics system diagram at the 30th second of calculation

After 100th second of the calculation the second valve starts being opened in accordance with control algorithm, at the same time, the first valve also starts being opened to keep pressure in the node.

After 200th second of the calculation **“Z2”** will be in **40%** position, and the first valve – **52–54%**.

After 400th second of the calculation the second valve will be in **20.7%** position, while the first one – in **28.5%**.

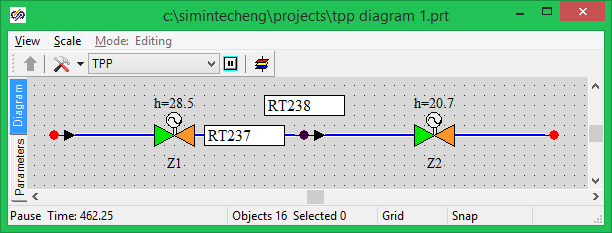


Figure 115. Hydraulics system diagram at the 450-th second of calculation

Go to the control model and **“Equipment management”** submodel. Double-click the valve position graphic block. The graphic shall appear like it is presented in the Figure below (Figure 116).

It can be seen from the graphic that the first valve operating to maintain pressure in accordance with the control algorithm work continuously performing oscillatory motions – this indicates that the created control algorithm is not good enough for the given conditions of system operation.

To “improve” the pressure controller a relay block similar to the block applied in **valve Z2 control algorithm** can be added to **valve** **Z1 control algorithm**.

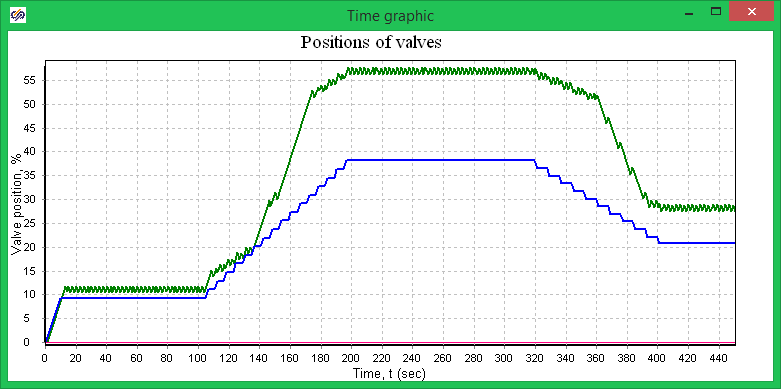


Figure 116. Positions of valve in complex model simulation mode

Go to the thermohydraulics model and open the internal node pressure graphic. Scale up the graphic as shown in the following Figure (Figure 117). While analyzing the pressure graphic, we can make a conclusion about a required width of **“Dead zone”** of the relay block for **“Valve Z1 control algorithm”**.

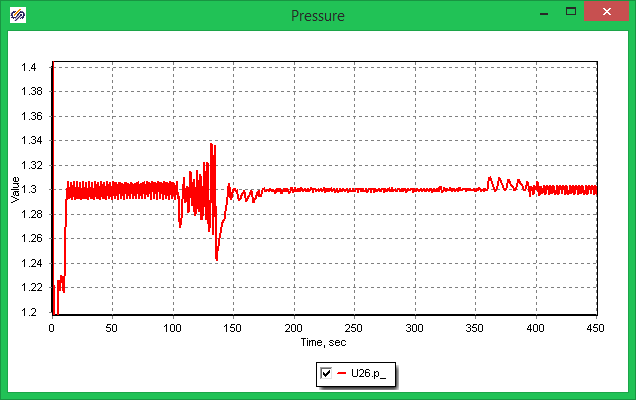


Figure 117. Pressure graphic for internal node

## Task for independent operation

1. Change the valve Z1 control algorithm so that maintenance of pressure at 1.3 level in the internal node do not lead to continuous displacement of valve Z1. Valve position graphic shall appear nearly the same as it is shown in the Figure 118:

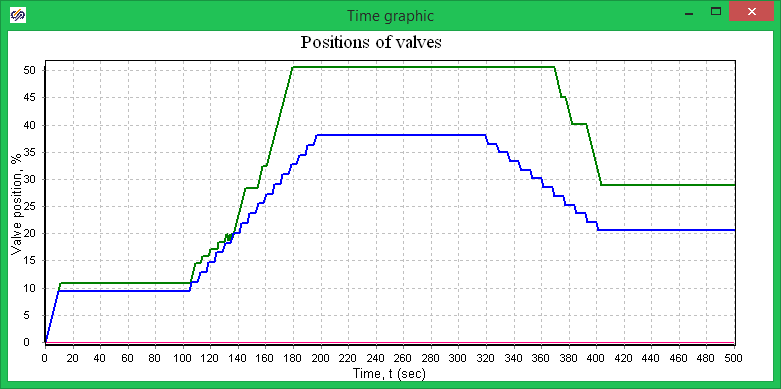


Figure 118. Valve position graphic

1. Create a new project of the control system, in which the Valve Z1 control algorithm will maintain the flow rate through the system at 20 level.

# Creation of window of equipment control

## Manual control in project

SimInTech is an open software environment, which allows data received from different sources to be applied in the process of mathematical modeling. One of possible sources include control windows, in which the user can set different actions on the mathematical model using interactive elements such as virtual buttons, switches, knobs, etc.

As an example of a control window we will make up a valve control window for thermohydraulics circuit created in former training exercises.

## Creation of control window

1. Open **“TTP block 1.prt”** file with a hydraulics model.
2. Press **“Data manager”** button in the SimInTech main window (Figure 119).

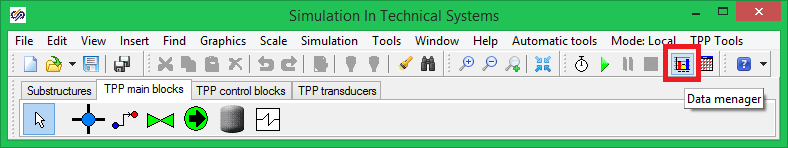


Figure 119. Data manager activation button

Pressing this button displays **“Data manager”** dialog window (Figure 120), which is designed to adjust different channels of actions on the mathematical model as well as to adjust data exchange.

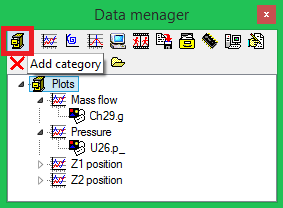


Figure 120. “Data manager” dialog window

1. Press **“Add category”** button (Figure 120). Enter the name of a new **“Windows of equipment control”** category. To rename categories as well as any object of **“Data manager”** click the object with the right mouse button and select **“Rename”** item in the pop-up menu.
2. Select the created category (selected category is highlighted blue) and press **“Animation window”** button, Figure 121):

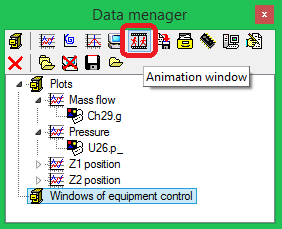


Figure 121. “Data manager” dialog window after adding a new category

New **“Animation window”** element will be displayed in **“Windows of equipment control”** category. If required, open the list of categories by pressing “+” sign at the left of the category name.

1. Enter a name for a newly created **“Window of valve control”** element (Figure 122):

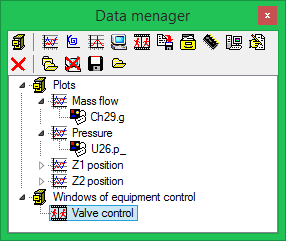


Figure 122. “Data manager” dialog window after adding a new category

1. Double-click **“Window of valve control”** with the mouse.

Upon that there shall be displayed an empty window (Figure 123), in which control panel and primitives panel will be generated (Figure 124), which will be used to generate equipment control elements.

Animation window created in the data manager is accessible and can be activated from any part of the mathematical model.

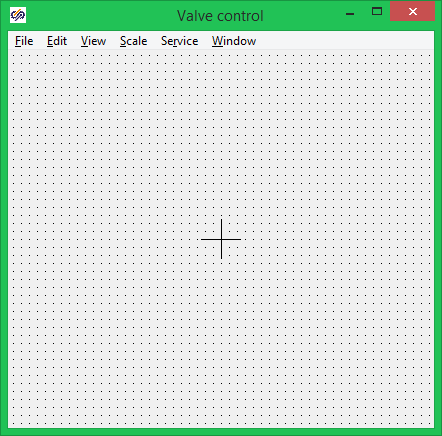


Figure 123. Empty window of valve control

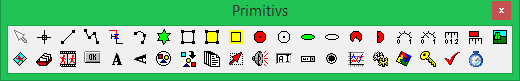


Figure 124. Primitives panel

## Creation of equipment control interface

The user can apply the set of primitives depicted in the Figure above to make up and adjust the appearance of the window of equipment control on the basis of the animation window. Animation window can contain both display elements, which change their appearances depending on effective signals, and interactive control elements that allow signals in the mathematical model database to be manipulated.

Window of valve control in our example will contain two buttons, one of which sends a command for opening the valve, while the other one sends a command for its closing.

To select a primitive click once a related button of the primitives panel with the left mouse button, then click the control window in the point the primitive is to be desirably located.

Locate the following elements on the window of valve control:

**“Button”** – two elements;

**“Text”** –three elements;

**“Bar”** – one element.

Arrange the primitives relative to each other about in the same way as depicted in the following Figure (Figure 125):

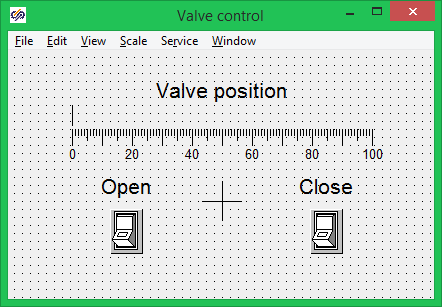


Figure 125. Window of valve control

Every primitive is an object with editable properties. To edit properties of a primitive arranged on the window proceed as follows:

Select the object, double-click it with the right mouse button, select **“Object properties”** menu item in the pop-up window (Figure 126).

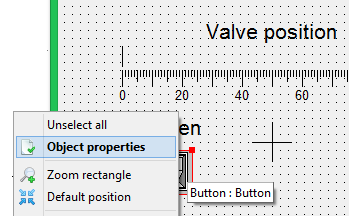


Figure 126. Edit primitive pop-up window

Then a dialog window for edition of object properties will be displayed. The user will be able to change properties of a selected primitive in the window (Figure 127).

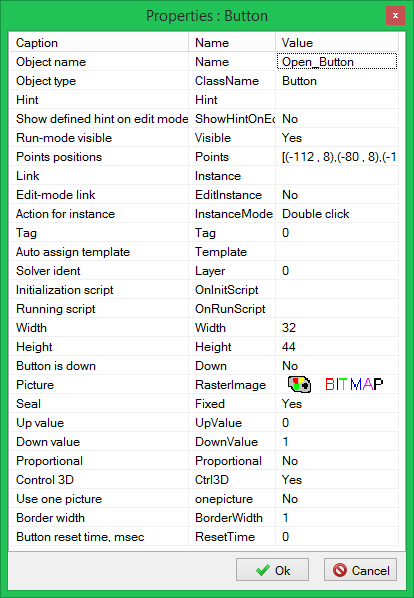


Figure 127. Primitive properties editor window

Change properties of primitives as follows:

1. **Open\_Button** and **Close\_Button** lines shall be set as the name of an object (upper line of the dialog window, see Figure 127) for buttons.
2. **Name\_TextLabel** – for the upper text caption.
3. **Position\_Bar** – for the linear device.
4. To improve the appearance of the window of valve control, scale up fonts up to **“15”** for text captions.

## Creation of variables for window of valve control

To ensure correct operation of the control window it is necessary to program transformations of user’s actions with primitives to signals applicable for a mathematical model. First the window of equipment control shall receive the name of a mathematical model object, for which it (the window) has been activated.

Go to **“Window of valve control”** main menu and select **“Service”** menu item, **“Global properties…”** sub-item (Figure 128). In a displayed **“Common properties”** dialog window (Figure 129) add a new property for the control panel.

Attention!!! If the name of a signal to be added in the control window is the same as the property name of an object, for which the window has been activated, then its value is automatically set as equal to the value of properties of the object.

For example, in this case we will add **“Name”** signal; on activation of the window, its value will become the same as the valve name.

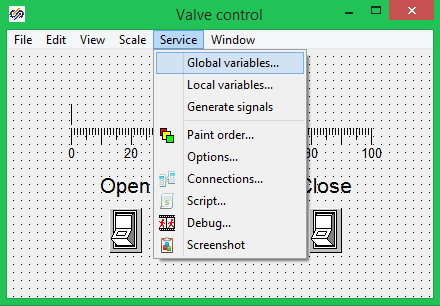


Figure 128. Activation of Add Properties window

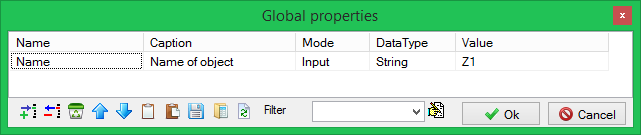


Figure 129. Add Global Properties window

Press the button, add a signal and enter the following values (Figure 129):

Name – Name

Caption – **Name of object**

Mode – Input

Data type – **String**.

To display states of the valves we will use a variable available in **“Common view valves”** object properties of TPP code – **“State”**:

1. **Add** a new signals and set up its properties as shown in the following Figure (Figure 130):

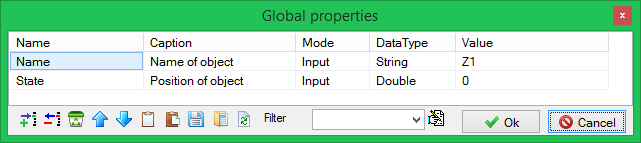


Figure 130. Add global properties window

1. Close the dialog window by pressing **“Ok”** button.

Thus, we have added two variables to transmit the name and position of a valve, for which we have activated the control window in the process of modeling.

1. Go to **“Window of valve control”** main menu and select **“Service”** menu item, **“Local variables…”** sub-item (Figure 128). In a displayed **“Local variables”** dialog window (Figure 131) add the signals that will be displayed (and formatted) by this control window. In our case we will display **“Position”** of the valve and send **“Command Open”** and **“Command Close”.**

In our case, signals created during generation of the database in the process of execution of the previous training exercises will be used to control the valve. Let us remind the rules for generating signals in the database:

A database signal consists of a name of object and name of signal separated by an underline. For example, **“Position”** signal of valve **“Z1”** in the database is named as **“**Z1\_xq01”.

To generate a correct name of a control signal the names in local variables shall be created using an underline in the beginning of a name (Figure 131).

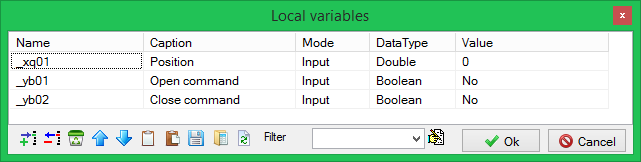


Figure 131. Window of local variables

1. Add local variables as shown in the above Figure (Figure 131) and close the window by pressing **“Ok”** button.

## Programming of window of valve control

Interface elements created before and a set of local and global variables allow the window of valve control to be programmed so that the mathematical model could be “manually” manipulated in the process of modeling. To complete the creation of the control panel the logic of behavior of displaying and controlling interface elements shall be programmed.

Go to **“Window of valve control”** main menu and select **“Service”** menu item, **“Global Script…”** sub-item (Figure 132).

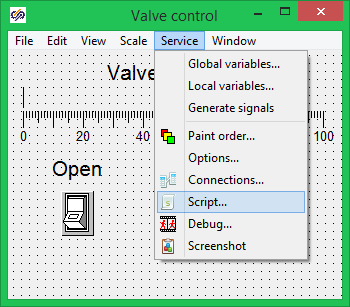


Figure 132. Programming window activation

In a displayed **“Programming language”** text window (Figure 133) enter the following text of the program:

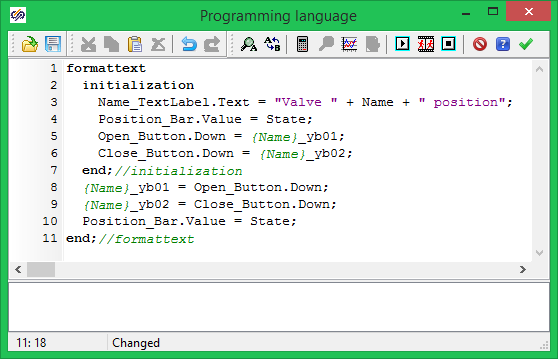


Figure 133. Script of valve control

The whole program is included between the key words of **“**formattext” and **“**end;”. Use of the given key word allows the name of a variable to be generated using a template. Instead of an expression collected in squiggle brackets its value will be set. In our case, when activating the dialog window, the value of **“Name”** global variable will be the same as the value of this property of the valve. For valve **“Z1”**, after activation of the control window, expression of {Name}\_yb02type will be transformed, in accordance with a template, to Z1\_yb02 expression – of the database signal name. Strings included between keywords “initialization” and “end;” are executed once on activation of the window.

Name\_TextLabel.Text = “Position of valve ” + Name; –

assignment of an upper caption with the name of a valve, for which the control window has been activated, to the text.

Position\_Bar.Value = State; – display of a rate of valve opening by means of the linear device.

Open\_Button.Down = {Name}\_yb01; – matching the appearance of **“Open”** button with the database signal.

Close\_Button.Down = {Name}\_yb02; – matching the appearance of **“Close”** button with the database signal.

Strings of the program main text are executed at every step of modeling until the control window is active.

{Name}\_yb01 = Open\_Button.Down; – sending a valve opening signal to the database.

{Name}\_yb02 = Close\_Button.Down; – sending a valve closing signal to the database.

Position\_Bar.Value = State; – display of a rate of valve opening by means of the linear device.

Close the window by pressing **“Apply”** button in the left upper corner.

## Link of the valve and the control window

1. Go to the hydraulic diagram.
2. Select valve **“Z1”** and activate the property editor window (Figure 134).

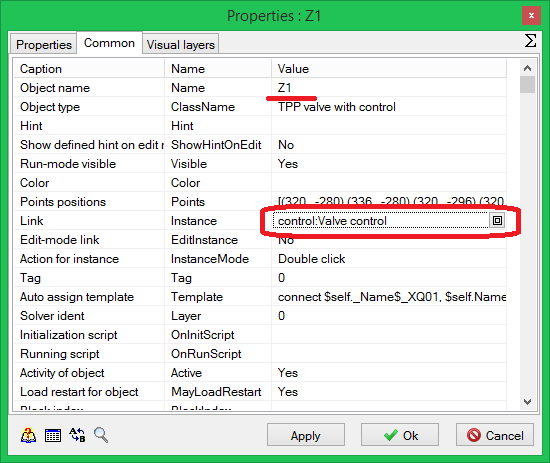


Figure 134. Editor window for Valve Z1 properties

1. Set the name of the valve in compliance with the name in the database.
2. Press the edition button in **“Reference”** string.

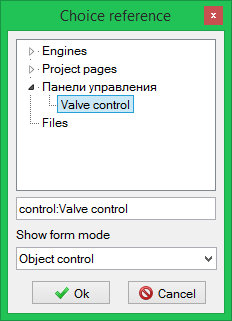


Figure 135. Valve editor window

1. In the appeared dialog window select **“Control panels”** category and **“Window of valve control”** element (Figure 135).
2. Select **“Object control”** in **“Show form mode”** string (Figure 135).
3. Set (in a similar manner) a link of the second valve **“Z2”** with the control window.
4. Go to the thermohydraulics diagram, set **“Indication”** mode using the button in the upper part of the window (Figure 136):

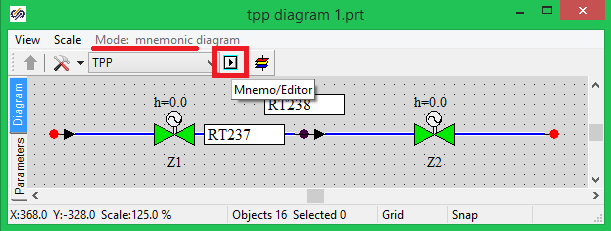


Figure 136. Thermohydraulics model diagram window

1. Start up the model for calculation.
2. Make sure that, after double-clicking, the window of valve control will be displayed on the valve.
3. Make sure that, after pressing buttons on the valve control panel the value of **“Command Open”** and **“Command Close”** signals in the database are changed.
4. Save **“TPP diagram 1.prt”** project.
5. Close the project.

## Manual valve control in a complex model

1. Open **“pack1.pak”** complex model created during execution of training exercise 8.

The considered complex model includes two projects: **“TPP diagram 1.prt”** – thermohydraulics model and **“Automatics diagram 2.prt”** – control system model.

The both models are automatically loaded when the pack is loaded.

1. Make sure that the thermohydraulics model contains the control window generated earlier. For that purpose press **“Data manager”** button in the SimInTech main window (Figure 119).

Since “**Automatics diagram 2.prt”** constantly controls the gates we will deactivate the gate **“Z2”** control algorithm to exclude any interference between automatic and manual controls.

1. For that purpose go to the automatics model and select **“Valve Z2 control algorithm”** submodel.

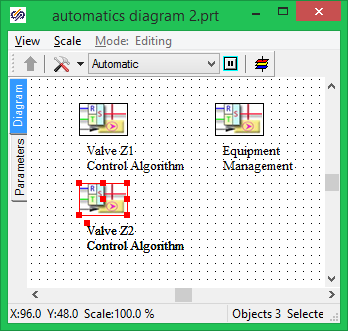


Figure 137. Automatics model diagram window

1. Go the program main window.
2. Select **“Edit”** menu item, **“Deactivate objects”** sub-item (Figure 138). Blocks excluded from the calculation will be displayed black on the diagram and are not used for signal interchange in the process of modeling.

Thus, we have deactivated the second valve control algorithm on the diagram and excluded the situation that can arise when signals set by the user via **“Window of valve control”** would be conflicting with control system signals.

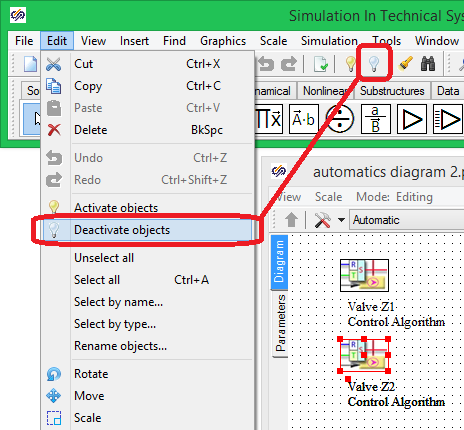


Figure 138. Exclude block menu

1. Save **“Automatics diagram 2.prt”** project.
2. Start up the complex model for calculation.
3. Double-click on the second valve.
4. Issue commands for valve activation and deactivation in the displayed control window. While doing so, make sure that the mathematical model is correctly processing valve opening and closing signals (Figure 139).

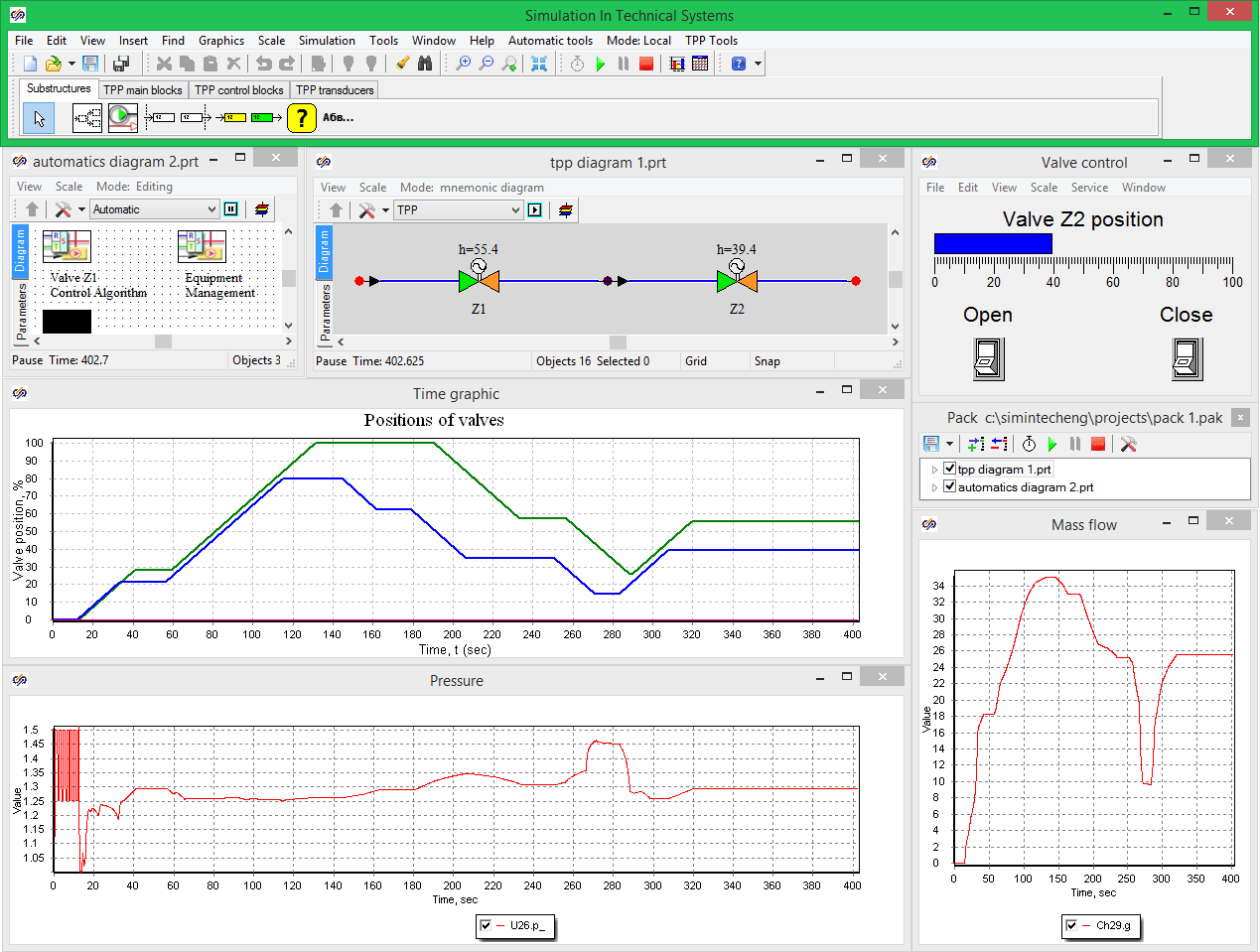


Figure 139. Second valve control in “manual mode”

# Creation of event log

## Event logging

As a rule, in the process of debugging of a complex mathematical model a great number of parameters, both analog and discrete ones, have to be analyzed. SimInTech enables system parameter monitoring in the form of time schedules, phase portraits, text tables and virtual instruments.

Analysis to be carried out for automatic control systems is greatly facilitated by means of an event log, which allows sequence of any events in the mathematical model to be recorded. Analysis of these records allows the sequence of events to be reconstructed.

There is an event logging system in SimInTech that allows one or more event logs to be created for the whole mathematical model or for any of its parts.

## Creation of event log

Open **“TPP diagram 1.prt”** file with the hydraulic model created during execution of the previous training exercises.

Make sure that there is a valve control panel in the model, for which purpose:

1. Start up the model for calculation. When required, switch over the diagram window to **“Indication”** mode.
2. Double-click valve **“Z2”**. Make sure that the valve control panel created during execution training exercise 9 is displayed.
3. Stop the calculation.
4. Press **“Data manager”** button in the program main window (Figure 140):

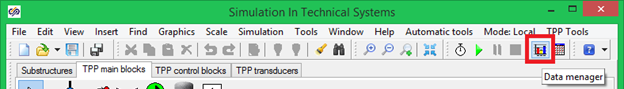


Figure 140. Data manager activation button

1. Pressing this button displays **“Data manager”** dialog window (Figure 141), which is designed to adjust different channels of actions on the mathematical model as well as to adjust data exchange and information display system.

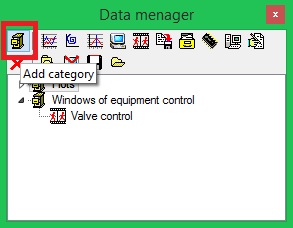


Figure 141. “Data manager” dialog window

1. Press **“Add category”** button (Figure 141). Enter the name of a new **“Logs”** category.
2. Select the created category (selected category is highlighted blue) and press **“Event log”** button (Figure 142).

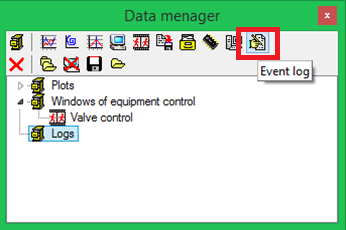


Figure 142. “Data manager” dialog window after adding a new category

1. A new **“Event log”** element will be displayed in **“Logs”** category (Figure 143). If required, open the list of categories by pressing “+” sign at the left of the category name:

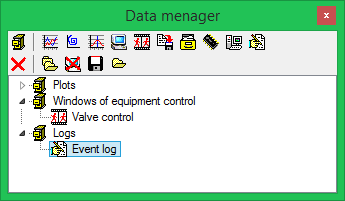


Figure 143. “Data manager” dialog window after adding “Event log”

Any change of a calculation parameter is considered as an event in the mathematical model. To create an event select a parameter, whose change will be considered as an event, and set up its properties.

## Adding parameters to “Event log”

To add a new parameter proceed as follows:

1. Click **“Event log”** item with the right mouse button.
2. Select **“Add parameter”** item from the pop-up menu (Figure 144):

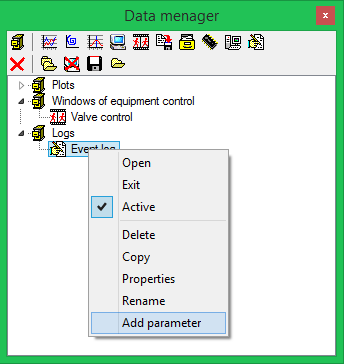


Figure 144. Adding parameters to event log

1. Enter block name “Z1” (first valve) and “State” parameter name (valve position) into the displayed dialog window (**Ошибка! Источник ссылки не найден.**):

|  |  |
| --- | --- |
| Figure 145. Change of parameter for logging | Figure 146. Database parameter search activation button |

1. Close the window by pressing **“Ok”** button.

New **“Z1.State”** parameter will be displayed in **“Data manager”** window under **“Event log”** item.

Besides adding a new block name parameter, signals from the project database can be added as parameters. To this end, redo the above items 1–2 and press **“Find value from database”** button in “**Changing of parameter”** dialog window (Figure 146) .

**“Database editor”** dialog window will be displayed by pressing **“Find value from data base”** button.

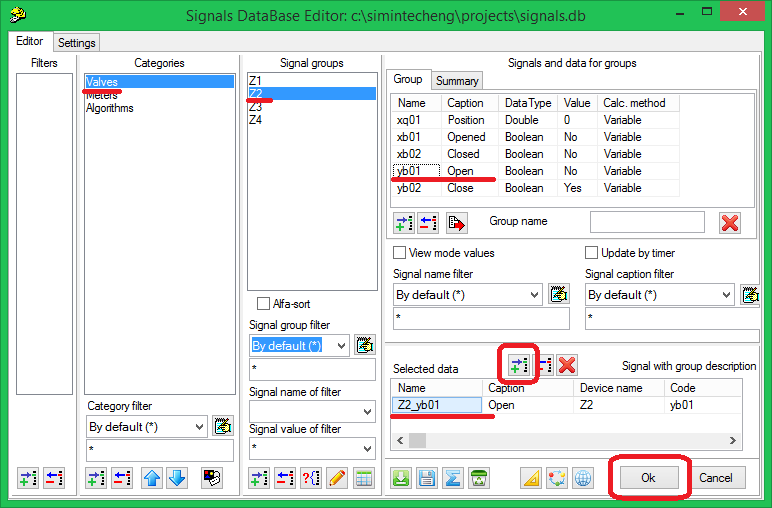


Figure 147. Selection of signals from database

Select **“Command Open”** signal in the database for valve **“Z2”**. For that purpose, select successively **“Valves”** item in the list of **“Categories”** panel, **“Z2”** in **“Signal groups”**, **“**yb01” signal in “Signals and data for groups” table (“Command Open”). Press **“Add”** button. The selected signal will be displayed in **“Select data”** table in the right lower corner (Figure 147). Press **“Ok”** button to close **“Database editor”** window.

Add **“yb02”** parameter (**“Command Close”**) for the second valve in the same manner.

## Setting of event logging parameters

Beside selection of a mathematical model parameter for event logging, conditions for event origination shall be selected. For example, a parameter value violation in the process of modeling above a specified value, i.e. setting, can be an event.

By default the log is set for changing values of logical parameters from **“0”** (logical **“No”**) to **“1”** (logical **“Yes”**).

To change the conditions for event activation proceed as follows:

1. Click the parameter name with the right mouse button in **“Event log”** section.
2. Select **“Additional”** item in the pop-up menu:

|  |  |
| --- | --- |
| Figure 148. Event setting dialog window activation | Figure 149. Setting of event logging  parameters |

Then the event log setting window will be displayed (Figure 149). The following parameters shall be set in this dialog window:

**– Logging mode** – defines the parameter change leading to the event origination. The following options are provided:

* **Increase of value;**
* **Decrease of value;**
* **Change of value;**
* **Setting exceeding;**
* **Reduction below setting.**

**– Priority** – defines the sequence of event logging in the log for events that have occurred simultaneously; an event with a higher priority will be recorded first.

**– Setting** – numerical value of a setting the parameter value is to be compared with.

**– Event description** – event message text to be recorded in the event log.

1. Set the following values for **“Z1.State”** parameter: **“Register mode” – “Setting violation”, “Priority” – “0”, “Setting” – “99.99”,** **“Event description” – “Valve Z1 fully open”**.

For events linked with “Command Open” and **“Command Close”** parameters for valve **“Z2”** the event logging parameters are set by default, thus, origination of the commands will automatically result in origination of events.

1. Save **“TPP diagram 1.prt”** project.
2. Close **“TPP diagram 1.prt”** project.

## “Event log” window

To activate **“Event log”** window double-click the due item in **“Data manager”** window. A window the same as depicted in the Figure below will appear (Figure 150). The window contains two tabs:

* **“Journal”** – contains a table for entering the list of mathematical model event, and a control button panel for this list.
* **“Settings”** – are intended for setting “Event log”.

Control panel on **“Journal”** tab contains the following buttons:

* **“Always on top”** activates and deactivates this mode for **“Event log”**;
* **“Clean”** deletes all records available in **“Event log”**;
* **“Open”** allows a previously saved event list to be loaded;
* **“Save”** allows an event list to be saved in a text file;
* **“Delete”** is intended for deletion of a selected event from the signal list;
* **“Copy”** allows an available event list to be copied in the Windows clipboard.

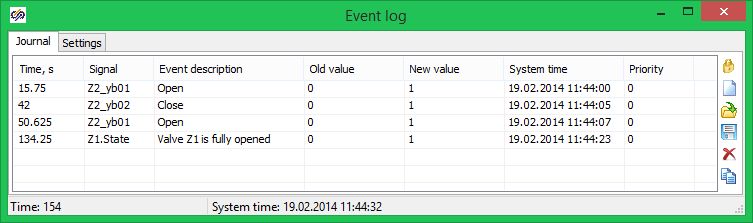


Figure 150. “Event log” window

Table on **“Journal”** tab contains the following columns:

* **“Time”** – mathematical model estimated time of origination of the event;
* **“Signal”** – parameter name in the format of a programming internal language, for which events are being logged;
* **“Event description”** – a text string set in the process of event setting or for database signals this string corresponds to the value in the “Name” field of database editor;
* **“Old value”** – value of the parameter before the event;
* **“New value”** – value of the parameter after the event;
* **“Priority”** – priority value.

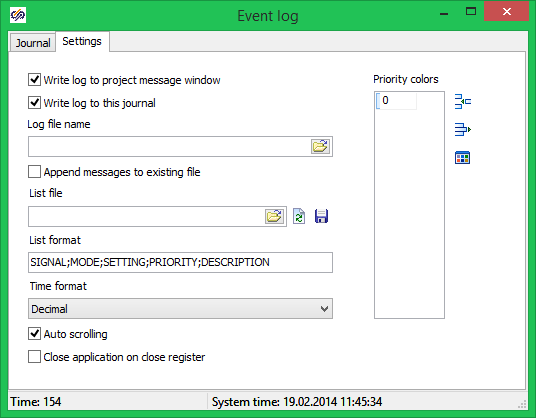


Figure 151. “Event log” window, “Settings” tab

**“Settings”** tab contains the following control elements:

* **“Record events in project message common window”** flag – when this flag is set all messages in **“Event log”** will be duplicated in the project message window;
* **“Record events if log window”** flag – when this flag is set event messages are not displayed in the log window;
* **“Event log file name”** edit bar – allows the name of a file intended for storing event messages to be set.
* **“Add events to existing file”** flag – when this flag is set records existing in the file are not deleted;
* **“Signals list file”** bar – allows a file to be set for storing the parameter list applicable for creation of the event log;
* The buttons nearby are intended for:
* opening the file selection dialog window;
* upgrading the signal list from the file;
* saving the signal list into the file;
* **“Signals list format”** bar – allows parameters to be set as well as their sequence for recording the signals into a signal list text file;
* **“Modal time display format”** pop-up window allows a modal time display format to be set:
* **“Decimal”** – time is displayed in seconds;
* **“Hours:** **minutes:** **seconds”** – time is displayed using hours, minutes and seconds.

## Application of event log for modeling

Perform the following actions:

1. Open **“pack1.pak”** complex model created during execution of training exercise 9. The considered complex model includes two projects: **“TPP diagram 1.prt”** – thermohydraulics model and **“Automatics diagram 2.prt”** – control system model. The both models are automatically loaded when the pack is loaded.
2. Go to the file named **“**ТPP Diagram 1.prt”. For that purpose, you can use the SimInTech main menu, **“Window”** item (Figure 152).

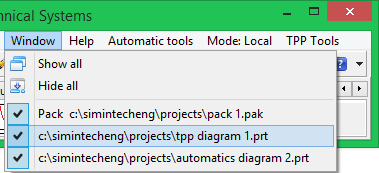


Figure 152. Change-over among complex model windows

1. Make sure that the thermohydraulics model contains the previously created event log, for which purpose press **“Data manager”** button in the program main window.

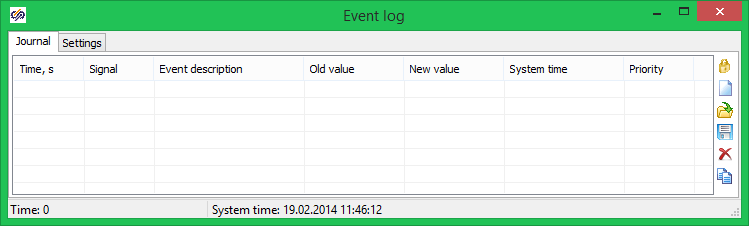


Figure 153. “Event log” window

1. Double-click **“Event log”** item. A window indicating a list of events registered in the process of modeling will be displayed (Figure 153).
2. To ensure comfortable view of the events **“Always on top”** mode can be set (upper button at the right of the window).
3. Start up the complex model for calculation.
4. Double-click the second valve in the thermohydraulics model.
5. Issue valve opening and closing commands in the displayed control window.
6. Make sure that valve opening and closing commands are registered in the event log.

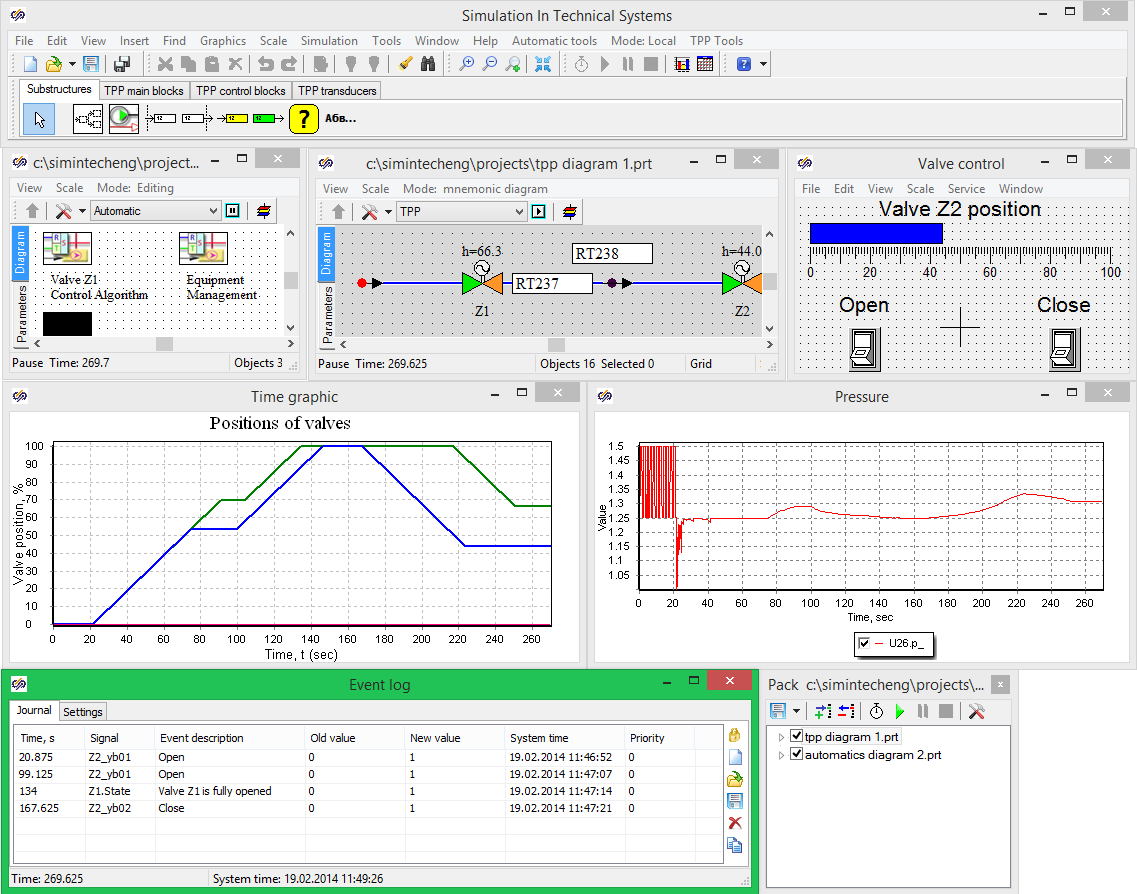


Figure 154. Event logging in complex model

1. Set valve **“Z2”** to fully open position. Wait until the valve **“Z1”** control algorithm fully opens of the valve. Make sure that the fact of the full opening of the valve is logged by the event log (Figure 154).

Now the training exercises (from one to ten) are over. Thank you!