

HOMA VIRTUAL PLANT

User Manual

Version 1.1

Desktop SIL tool for virtual commissioning of PLC software

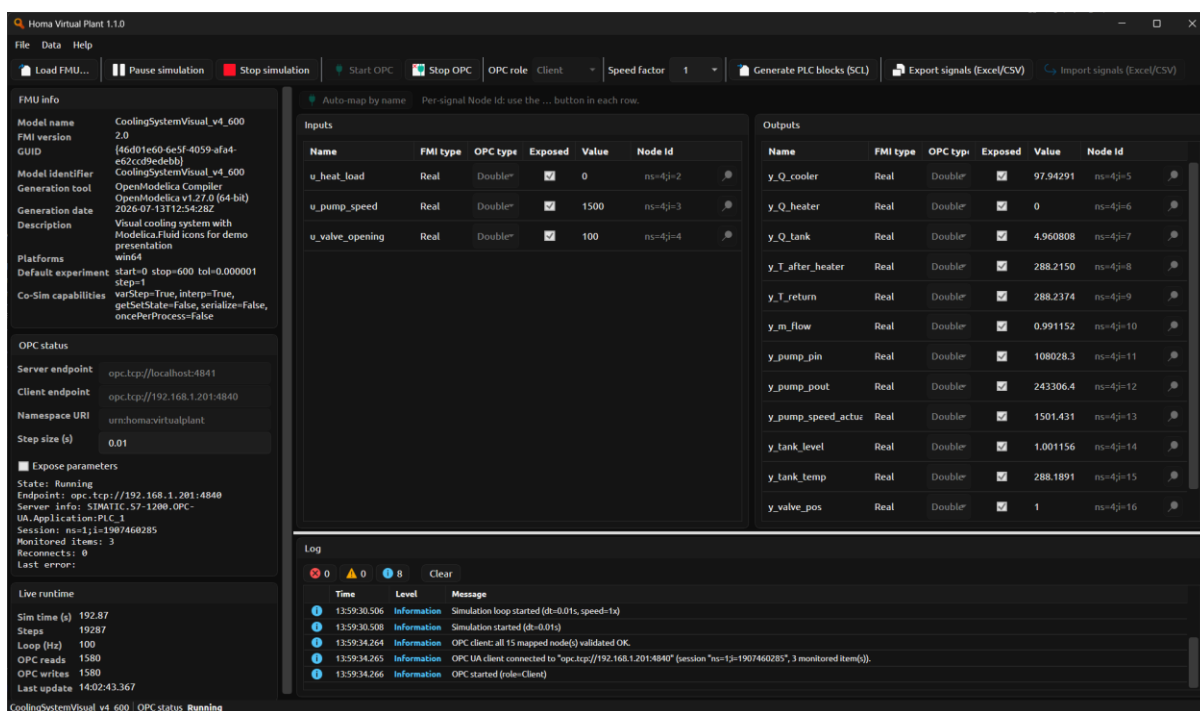


Figure 1: Cover illustration – main application window

Homa inženjering

www.homa.hr · info@homa.hr

© 2026 Homa inženjering. All rights reserved.

Contents

1.	Introduction	4
1.1.	What is Homa Virtual Plant	4
1.2.	Who this manual is for.....	4
1.3.	How HVP works	4
1.4.	Two ways of connecting	5
1.5.	Limitations of this version	6
2.	Installation	7
2.1.	System requirements.....	7
2.2.	Installation	7
2.3.	First launch	8
2.4.	Where settings and logs are stored.....	8
2.5.	Upgrade and uninstallation	8
3.	Interface overview	9
3.1.	Main window	9
3.2.	Menu.....	9
3.3.	Toolbar.....	10
3.4.	Left panel	11
3.4.1.	FMU info.....	11
3.4.2.	OPC status	11
3.4.3.	Live runtime	12
3.5.	Signal tables.....	14
3.6.	Log	15
3.7.	Status bar.....	15
4.	Working with projects.....	16
4.1.	What is a project.....	16
4.2.	Basic actions	16
5.	Basic workflow	17
5.1.	Step 1 – loading the model.....	17
5.2.	Step 2 – selecting signals	17
5.3.	Step 3 – checking OPC types.....	18
5.4.	Step 4 – simulation step and speed factor	18
5.5.	Step 5 – starting the simulation.....	18
5.6.	Checking whether the simulation runs in real time	19
6.	Server mode.....	20
6.1.	Setup.....	20
6.2.	Address space structure	21
6.3.	Connecting the PLC.....	22
6.4.	Checking the connection	22

7.	Client mode.....	24
7.1.	Setting up the connection	24
7.2.	Browsing the address space	24
7.3.	Automatic mapping by name	25
7.4.	Starting communication	26
8.	Signal table in Excel.....	27
8.1.	Export	27
8.2.	Editing.....	28
8.3.	Import	28
9.	Siemens: generating PLC blocks (SCL).....	29
9.1.	What it is for	29
9.2.	What is generated	29
9.3.	Data type mapping	29
9.4.	Rules for variable names	30
9.5.	Generating the file.....	30
9.6.	Layout of the generated file	31
9.7.	Import into TIA Portal	32
9.8.	Using the blocks in the program.....	33
9.9.	Connecting the blocks to the simulation	33
9.10.	When you change the signals	34
9.11.	If you do not use a Siemens PLC.....	34
10.	Application options	35
10.1.	Interface language	35
10.2.	About	36
11.	Troubleshooting	37
11.1.	Loading the model	37
11.2.	Starting the simulation	37
11.3.	OPC communication	37
11.4.	Mapping and browsing	38
11.5.	Other.....	39
12.	Glossary	40
13.	Appendices.....	41
13.1.	Default values	41
13.2.	Summary table of type mappings.....	41
13.3.	Files and locations	41
13.4.	Keyboard shortcuts.....	42
13.5.	Support	42

1. Introduction

1.1. What is Homa Virtual Plant

Homa Virtual Plant (HVP for short) is a desktop tool that enables testing of PLC programs without a real machine or plant. Instead of physical equipment, the PLC communicates with a simulation model of the process that runs on a computer and behaves like a real plant.

In industry, this approach is called SIL (Software-in-the-Loop) or virtual commissioning. The process model is called a digital twin, and HVP is the bridge between that model and the PLC.

HVP loads a simulation model in **FMU** format (FMI 2.0 Co-Simulation), runs it in real time and exposes its inputs and outputs over the **OPC UA** protocol, which practically every modern PLC supports.

1.2. Who this manual is for

This manual is written for automation and PLC engineers. Knowledge of PLC programming, a development environment (e.g. TIA Portal) and the basics of OPC UA communication is assumed. The simulation side (FMU, FMI, co-simulation) is explained from the ground up because it is usually new to this group.

1.3. How HVP works

In each simulation step, HVP performs three actions, always in the same order:

1. It transfers the input values (what the PLC sends) from the signal table into the simulation model.
2. It advances the model by one communication step (the Step size (s) parameter).
3. It reads the model outputs and returns them to the signal table, from where the PLC picks them up.

This loop repeats continuously and synchronizes with real time according to the set speed factor. The result is that the PLC „sees“ a process taking place in real time, exactly as if it were connected to real equipment.

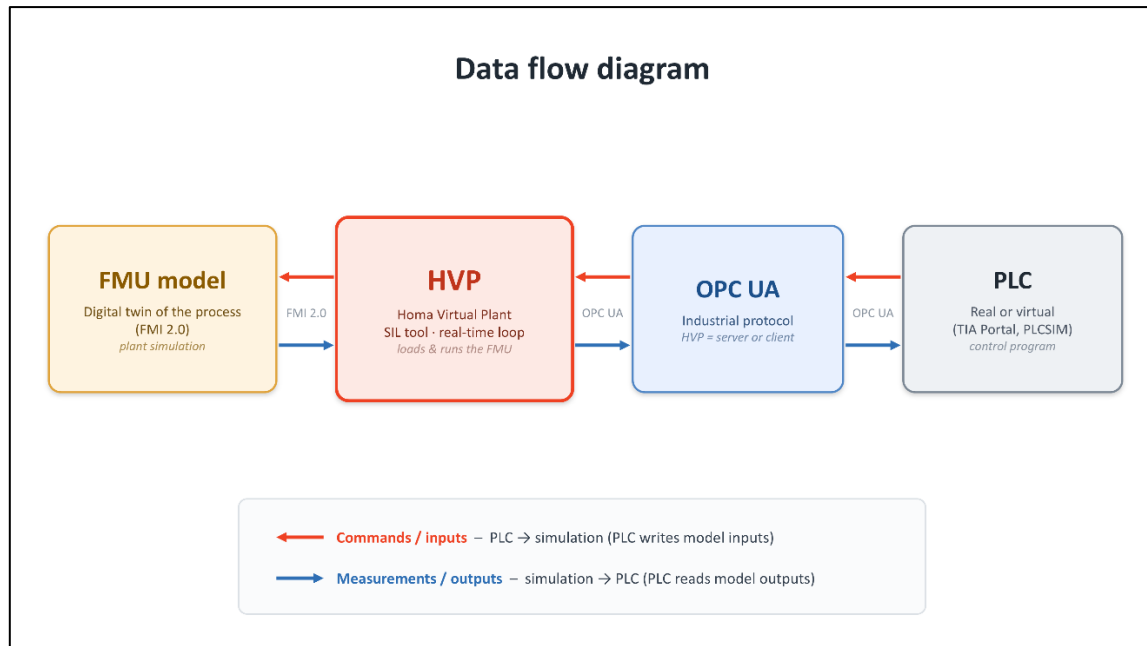


Figure 2: Operating principle – PLC, HVP and the simulation model

1.4. Two ways of connecting

HVP can connect to the PLC in two ways, depending on who is the server and who is the client:

Mode	Who is the server	When it is used
Server	HVP	HVP exposes the model signals as an OPC UA server, and the PLC (or SCADA, or a PLC simulator) connects to it as a client. The most common case.
Client	PLC / another device	HVP connects to an existing OPC UA server (e.g. the OPC UA server in the PLC) and reads/writes its existing variables.

NOTE The mode is selected in the toolbar under OPC role. Both address fields (Server endpoint and Client endpoint) are always visible, but only the one matching the selected role is used.

1.5. Limitations of this version

Version 1.1 is a free version and has the following limitations that are worth knowing in advance:

- The FMI 2.0 Co-Simulation format is supported. FMI 3.0 and Model Exchange are not supported.
- 64-bit Windows FMUs (win64) are supported. The FMU must contain a binary file for that platform.
- Generating PLC blocks works only for Siemens PLCs (SCL format). See chapter 9.
- The security settings of the OPC UA connection use no encryption and anonymous login, which is intended for use on a development computer and in a closed laboratory network.

IMPORTANT HVP is not intended for connecting to production systems or to equipment in operation. Use it on a development computer, in an isolated network, or against a virtual PLC (e.g. PLCSIM Advanced).

2. Installation

2.1. System requirements

Item	Requirement
Operating system	Windows 10 or newer, 64-bit
Processor / memory	Any modern computer; at least 8 GB of RAM is recommended because simulation models can be demanding
.NET	No installation needed – the installation package contains everything required (self-contained)
Network access	Needed only for OPC UA communication (local computer or local network)
Additional tools	Not required. TIA Openness is not required for generating PLC blocks.

2.2. Installation

1. Download the installation file HomaVirtualPlant-Setup.exe.
2. Run it by double-clicking.
3. If a Windows warning appears (see the note below), select More info, then Run anyway.
4. Accept the suggested installation folder or choose another.
5. Optionally, select creating a desktop icon.
6. Click Install and wait for it to finish.

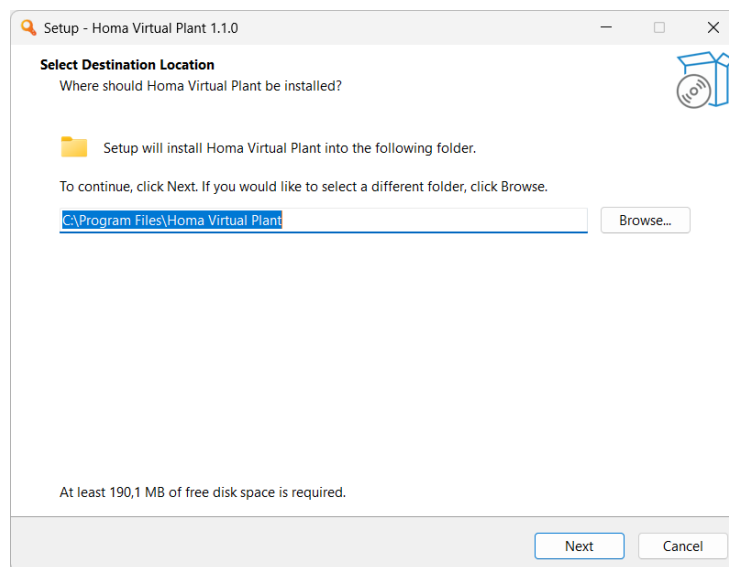


Figure 3: Installation wizard

IMPORTANT The application is currently not digitally signed, so Windows SmartScreen may display the warning „Windows protected your PC“ during download or launch. This does not mean the file is faulty. Select More info → Run anyway. You can verify the integrity of the downloaded file by comparing the SHA-256 checksum published with the download.

2.3. First launch

The application is launched from the Start menu under the name Homa Virtual Plant. On first launch, the interface is in Croatian, with no model loaded, and the status bar states that no FMU is loaded.

The interface language is changed via Help → Options (see chapter 10).

2.4. Where settings and logs are stored

HVP uses the standard Windows folder for application data. These locations are useful when troubleshooting:

Contents	Location
Application options	%AppData%\HomaVP\appsettings.json
Logs (log files)	%AppData%\HomaVP\logs\homavp-YYYYMMDD.log
Program files	C:\Program Files\Homa Virtual Plant
User projects	Wherever the user saves them (files with the .hvp extension)

TIP You can paste the %AppData% path directly into the Windows Explorer address bar and Windows will resolve it to the full path.

2.5. Upgrade and uninstallation

A newer version is installed over the existing one – there is no need to uninstall the old one first. Settings and projects are preserved because they are not located in the program folder.

Uninstallation is done via Windows Settings → Apps → Homa Virtual Plant → Uninstall. The settings in the %AppData%\HomaVP folder remain and can be deleted manually if needed.

3. Interface overview

3.1. Main window

The main window is divided into several sections that are always in the same place:

1. Menu and toolbar – at the top of the window.
2. Left panel – information about the loaded FMU, OPC settings and status, and live runtime data.
3. Central area – signal tables: Inputs on the left, Outputs on the right.
4. Bottom area – the Log with application messages.
5. Status bar – at the bottom, showing a short status and the state of the OPC connection.

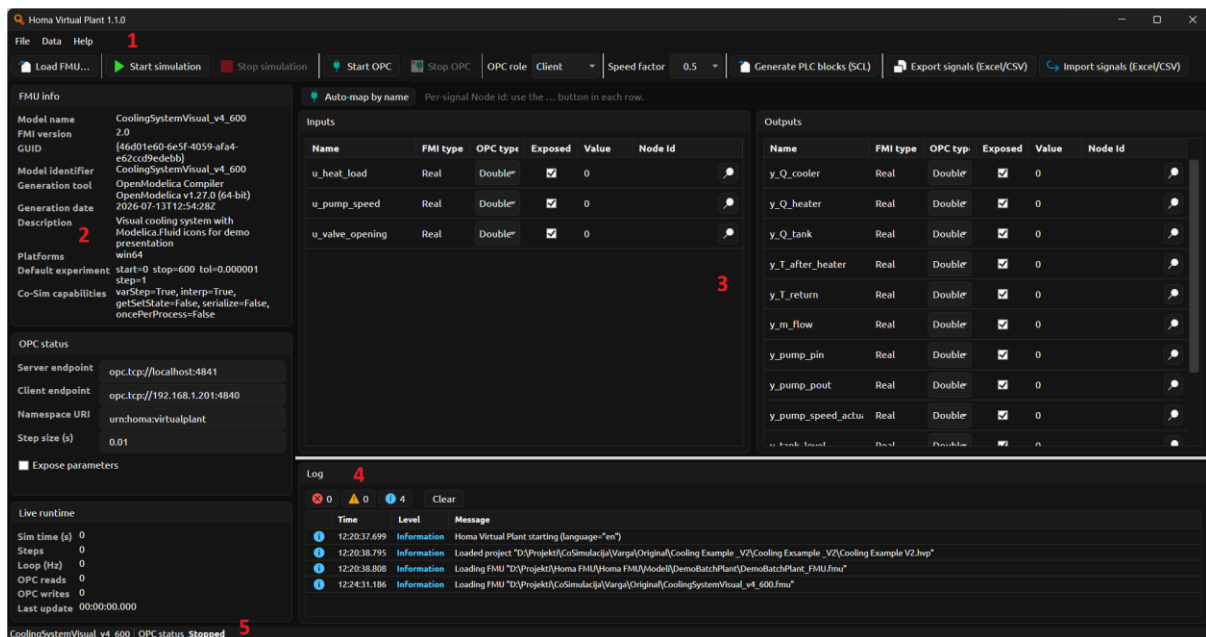


Figure 4: Main window with sections marked

3.2. Menu

Menu	Item	Meaning
File	New project...	Creates a new project and asks where to save it.
File	Open project...	Opens an existing project (.hvp).
File	Save (Ctrl+S)	Saves the current project.
File	Save as...	Saves the project under a new name.
File	Recent	A list of up to 10 recently used projects.
Data	Export signals (Excel/CSV)	Exports the signal table to .xlsx or .csv.
Data	Import signals (Excel/CSV)	Loads the modified signal table back into the application.
Data	Auto-map by name	Automatically fills the Node Id fields by comparing names (Client mode only).

Menu	Item	Meaning
Help	Options	Application options, currently the choice of interface language.
Help	About	Version, author, links and a list of used components.

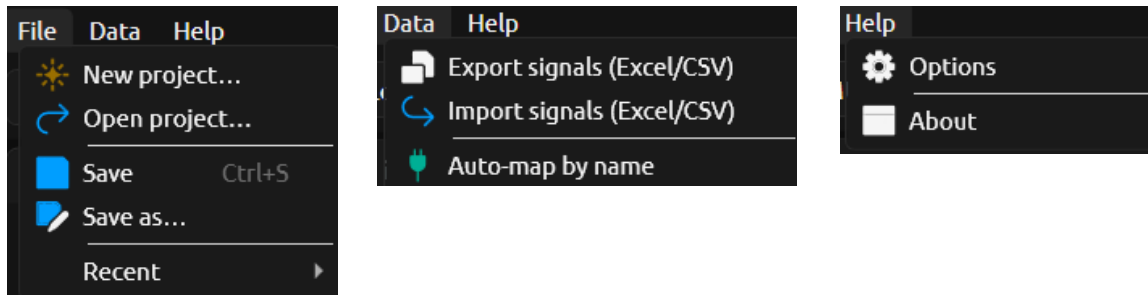


Figure 5: The File, Data and Help menus

3.3. Toolbar

The toolbar contains the most frequently used actions. Buttons that are not applicable at a given moment are grayed out.

Element	Meaning
Load FMU...	Opens a dialog for selecting a .fmu file and loads the model.
Start simulation	Starts the simulation from the beginning ($t = 0$). The same button changes its label depending on the state.
Pause simulation	The label of the same button while the simulation is running – it stops execution without losing state.
Resume simulation	The label of the same button after a pause – it resumes from the last time.
Stop simulation	Terminates the simulation and returns the model to the beginning ($t = 0$).
Start OPC / Stop OPC	Starts and stops OPC UA communication, independently of the simulation.
OPC role	Selection of the mode: Server or Client. It can be changed only while OPC is stopped.
Speed factor	The ratio of simulation speed to real time (1.0 = real time). The field is editable.
Generate PLC blocks (SCL)	Creates an SCL file with data blocks for a Siemens PLC (chapter 9).
Export signals / Import signals	A shortcut to the same actions from the Data menu.

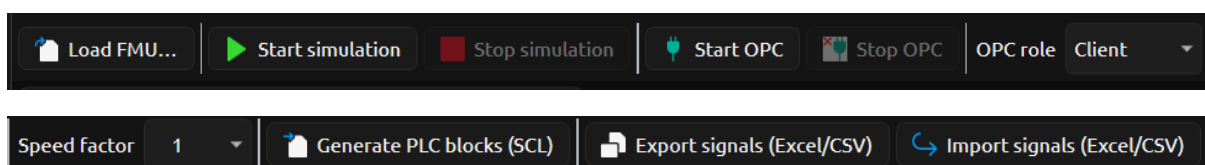


Figure 6: Toolbar

NOTE The simulation button is a single, dynamic button: its label and icon change between Start, Pause and Resume. Stop simulation is a separate button and always returns the simulation to the beginning.

3.4. Left panel

3.4.1. FMU info

Shows the data read from the loaded model. All values are read-only and serve to verify that the expected model has been loaded.

Field	Meaning
Model name	The model name from the FMU description.
FMI version	The version of the FMI standard (2.0 is expected).
GUID	A unique identifier of the model; it changes with every re-export.
Model identifier	The name of the binary file inside the FMU.
Generation tool	The program with which the FMU was exported (e.g. OpenModelica).
Generation date	The time the model was exported – useful for checking whether you are working with the latest export.
Description	A description of the model, if the author entered one.
Platforms	Supported platforms (win64 is required).
Default experiment	The suggested start time, step and tolerance from the model.
Co-Sim capabilities	The co-simulation capabilities that the model declares.

3.4.2. OPC status

This is where the communication parameters are set and diagnostics are shown. The fields can be changed only while OPC is stopped.

Field	Meaning	Default value
Server endpoint	The address at which HVP serves when the role is Server.	opc.tcp://localhost:4841
Client endpoint	The address of the OPC UA server that HVP connects to when the role is Client.	(empty)
Namespace URI	The identifier of the namespace in which HVP creates its nodes.	urn:homa:virtualplant
Step size (s)	The communication step of the simulation in seconds.	0.01
Expose parameters	If checked, the model parameters are exposed in addition to the inputs and outputs.	unchecked

NOTE The default port is 4841, not the standard 4840, because on many computers port 4840 is already used by the OPC UA Local Discovery Server. If the port is busy, HVP reports an error at startup.

3.4.3. Live runtime

Counters that show whether the simulation and communication are working properly:

Field	Meaning
Sim time (s)	The current time within the simulation.
Steps	The total number of steps executed since starting.
Loop (Hz)	The actual execution frequency of the loop. Compare it with the expected value $1 / \text{Step}$.
OPC reads	The number of reads over OPC.
OPC writes	The number of writes over OPC.
Last update	The time of the last data refresh.

FMU info

Model name	CoolingSystemVisual_v4_600
FMI version	2.0
GUID	{46d01e60-6e5f-4059-afa4-e62ccd9edebb}
Model identifier	CoolingSystemVisual_v4_600
Generation tool	OpenModelica Compiler OpenModelica v1.27.0 (64-bit)
Generation date	2026-07-13T12:54:28Z
Description	Visual cooling system with Modelica.Fluid icons for demo presentation
Platforms	win64
Default experiment	start=0 stop=600 tol=0.000001 step=1
Co-Sim capabilities	varStep=True, interp=True, getSetState=False, serialize=False, oncePerProcess=False

OPC status

Server endpoint	opc.tcp://localhost:4841
Client endpoint	opc.tcp://192.168.1.201:4840
Namespace URI	urn:homa:virtualplant
Step size (s)	0.01

☐ Expose parameters

State: Running
 Endpoint: opc.tcp://192.168.1.201:4840
 Server info: SIMATIC.S7-1200.OPC-
 UA.Application:PLC_1
 Session: ns=1;i=1230377605
 Monitored items: 3
 Reconnects: 0
 Last error:

Live runtime

Sim time (s)	13.71
Steps	1371
Loop (Hz)	100
OPC reads	95
OPC writes	95
Last update	14:06:11.057

Figure 7: Left panel

3.5. Signal tables

The central area of the window is divided into two tables. On the left are the Inputs (the values the simulation receives, i.e. what the PLC sends), and on the right the Outputs (the values the simulation produces, i.e. what the PLC reads). If the Expose parameters option is enabled, the model parameters are shown among the inputs.

Column	Meaning	Editable
Name	The variable name as defined in the model.	No
FMI type	The variable type in the model (Real, Integer, Boolean, String, Enumeration).	No
OPC type	The type under which the variable is exposed on OPC. It is selected from a drop-down list.	Yes
Exposed	Indicates whether the signal will be available over OPC and included in the generation of PLC blocks.	Yes
Value	The current value of the signal while the simulation is running.	No
Node Id	The address of the node on the remote server. Used only in Client mode.	Yes (client)

Inputs						Outputs					
Name	FMI type	OPC type	Exposed	Value	Node Id	Name	FMI type	OPC type	Exposed	Value	Node Id
u_heat_load	Real	Double*	☒	0	ns=4;i=2	y_Q_cooler	Real	Double*	☒	98.804431	ns=4;i=5
u_pump_speed	Real	Double*	☒	1500	ns=4;i=3	y_Q_heater	Real	Double*	☒	0	ns=4;i=6
u_valve_opening	Real	Double*	☒	100	ns=4;i=4	y_Q_tank	Real	Double*	☒	4.992174	ns=4;i=7
						y_T_after_heater	Real	Double*	☒	288.1728	ns=4;i=8
						y_T_return	Real	Double*	☒	288.19421	ns=4;i=9
						y_m_flow	Real	Double*	☒	0.99291	ns=4;i=10
						y_pump_pin	Real	Double*	☒	108018.31	ns=4;i=11
						y_pump_pout	Real	Double*	☒	242550.21	ns=4;i=12
						y_pump_speed_actual	Real	Double*	☒	1497.3	ns=4;i=13
						y_tank_level	Real	Double*	☒	1.001245	ns=4;i=14
						y_tank_temp	Real	Double*	☒	288.15781	ns=4;i=15
						y_valve_pos	Real	Double*	☒	1	ns=4;i=16

Figure 8: The Inputs and Outputs tables

NOTE The OPC type, Exposed and Node Id columns can be edited only while OPC is stopped. While communication is running, the configuration is locked so that the connection is not broken on the fly.

3.6. Log

The bottom area of the window shows application messages with a timestamp, level and text. Above the list there are three buttons with icons and counters:

- Red (errors) – problems that prevented an action from being carried out.
- Yellow (warnings) – something is not ideal, but operation continues.
- Blue (messages) – informational messages about normal operation.

Clicking an individual button hides or shows that category, just as in the error list in Visual Studio. The numbers next to the icons show how many messages of that type there are. The Clear button empties the log.

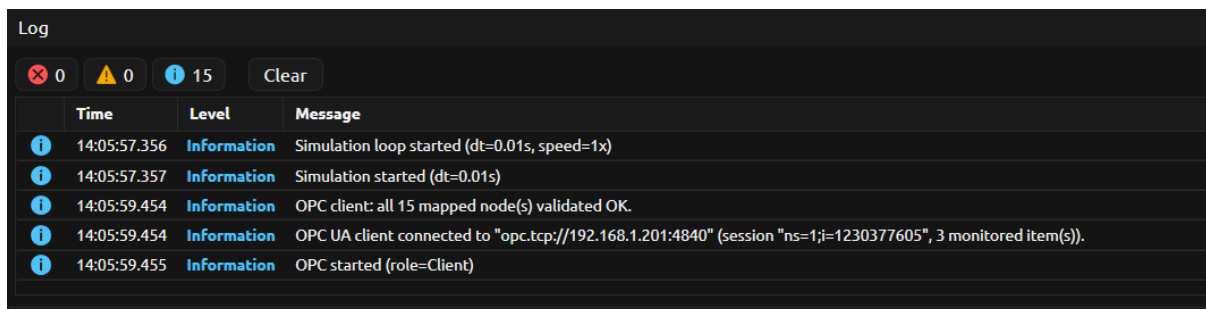


Figure 9: Log with filters

TIP The log shown in the application is simultaneously written to a file in the %AppData%\HomaVP\logs folder. If you report a problem, attach that file.

3.7. Status bar

At the bottom of the window, on the left is a short application status (e.g. Ready or No FMU loaded), and on the right the state of the OPC connection. The status bar is the fastest way to check whether communication is active.

4. Working with projects

4.1. What is a project

A project is a file with the .hvp extension that remembers the entire working configuration: which FMU is loaded, how the signals are configured and what the OPC settings are. This avoids having to reconfigure everything each time the application starts.

The project stores:

- The path to the .fmu file.
- The configuration of each signal: OPC type, the Exposed flag and Node Id.
- OPC settings: role, addresses, Namespace URI, the Expose parameters flag.
- Simulation parameters: Step size (s) and Speed factor.

NOTE The simulation model itself is not stored in the project, only the path to it. If you move or rename the .fmu file, the project will no longer be able to load it.

4.2. Basic actions

Action	Procedure
New project	File → New project... then choose a location and name.
Opening	File → Open project... or File → Recent.
Saving	File → Save or the keyboard shortcut Ctrl+S.
Saving under a new name	File → Save as...

HVP additionally saves changes to the signal configuration and settings automatically while you work, so under normal use little is lost. The last used project opens automatically the next time the application starts.

5. Basic workflow

This chapter walks through the entire procedure from an empty window to a running simulation. Connecting to a PLC is covered in chapters 6 and 7.

5.1. Step 1 – loading the model

1. Click Load FMU... in the toolbar.
2. Select a .fmu file and confirm.
3. Wait for the model to unpack and load – for larger models this can take a few seconds.

After successful loading, the FMU info in the left panel is filled in, and the Inputs and Outputs tables are populated with the model variables.

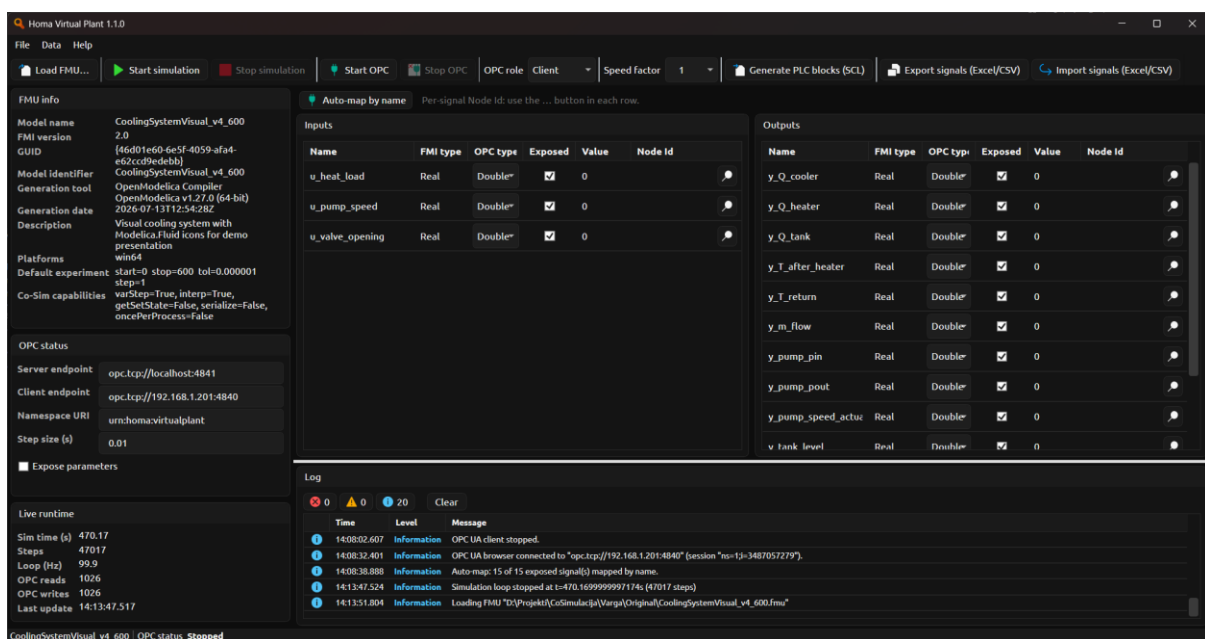


Figure 10: The application after loading an FMU

NOTE Check the Platforms field in the FMU info. If it does not say win64, the model will not work on this computer and needs to be re-exported with the correct setting.

5.2. Step 2 – selecting signals

Next, determine which signals will be available to the PLC. In the Exposed column, check the ones you need.

There is no need to expose all model variables. A smaller number of signals means clearer PLC blocks and less communication load. Typically exposed are:

- all inputs by which the PLC controls the process (commands to valves, pumps, motors),
- all outputs the PLC needs as measurements and feedback (levels, temperatures, positions, statuses).

TIP If the model has a great many variables, it is faster to export the configuration to Excel, mark everything needed there and import it back. See chapter 8.

5.3. Step 3 – checking OPC types

HVP automatically assigns each signal a reasonable OPC type according to its type in the model:

FMI type in the model	Default OPC type
Real	Double
Integer	Int32
Boolean	Boolean
String	String
Enumeration	Int32

The default choice is correct in most cases and does not need to be changed. A change makes sense when the type has to be adapted to what the PLC expects – for example, if the PLC program expects Int instead of DInt, select Int16 in the OPC type column.

The conversion is performed as a checked type conversion. If a value does not fit into the selected type, HVP clamps it to the nearest allowed value and writes a warning to the log.

5.4. Step 4 – simulation step and speed factor

Two parameters determine how the simulation runs in time:

Parameter	Meaning	Recommendation
Step size (s)	How much simulated time the model advances in one loop step. A smaller step means a more accurate simulation but a greater load on the computer.	0.01 for most processes; reduce to 0.001 if the model has fast dynamics or if it reports an error while executing a step
Speed factor	The ratio to real time. A value of 1.0 means that one second of simulation lasts one second.	1.0 when working with a PLC; higher values for quickly running through slow processes without a connected PLC

IMPORTANT When a PLC is connected, keep the speed factor at 1.0. The PLC runs in real time and cannot follow an accelerated simulation, so the results would be unusable.

5.5. Step 5 – starting the simulation

The simulation is controlled with two buttons in the toolbar:

Button	Action	State afterwards
Start simulation	Starts the simulation from the beginning.	The button changes its label to Pause simulation.

Button	Action	State afterwards
Pause simulation	Stops execution but retains the model state.	The button changes its label to Resume simulation.
Resume simulation	Resumes from the time at which it stopped.	The button again shows Pause simulation.
Stop simulation	Terminates the simulation and returns the model to $t = 0$.	The next start begins from the beginning.

That the simulation is actually running is confirmed by the increasing Sim time (s) and Steps values in the Live runtime section, and by the changing values in the Value column of the tables.

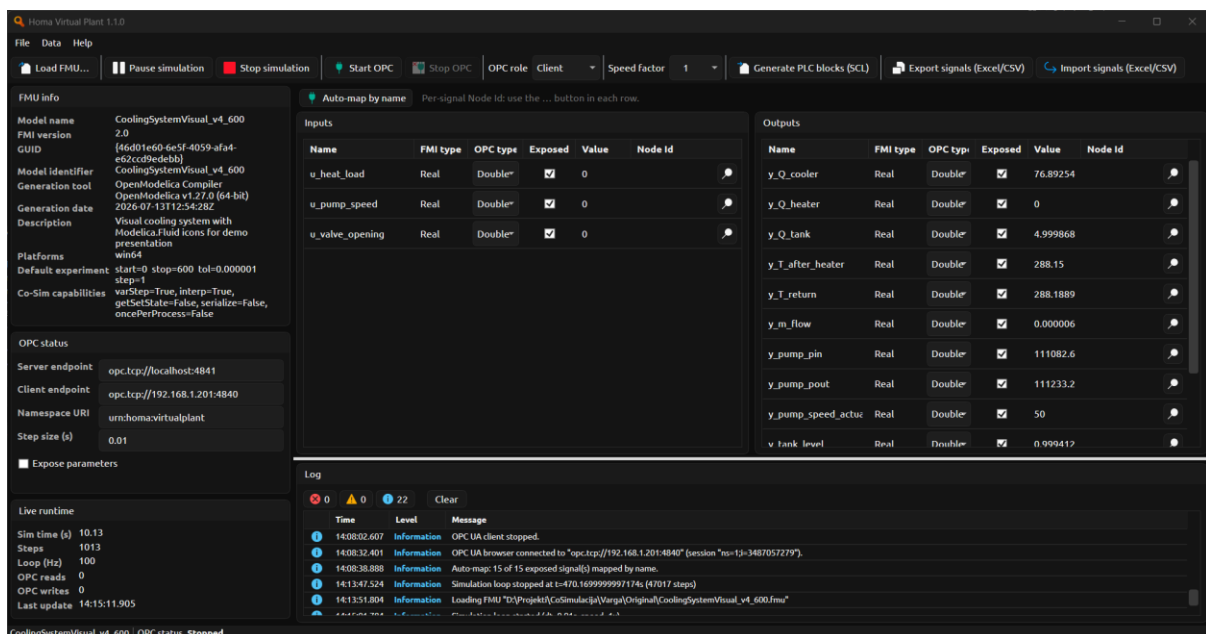


Figure 11: The simulation running

5.6. Checking whether the simulation runs in real time

Compare the displayed Loop (Hz) value with the expected one. The expected frequency is 1 divided by the step: with a step of 0.01 s, about 100 Hz is expected.

If the measured frequency is significantly lower, the computer cannot run the model fast enough. The solutions, in order, are:

- increase the simulation step (e.g. from 0.001 to 0.01),
- simplify the model in the simulation tool,
- reduce the number of exposed signals,
- close other demanding programs on the computer.

6. Server mode

In this mode, HVP is the OPC UA server. The simulation model exposes its signals, and a PLC, SCADA or any other OPC UA client connects to it. This is the most common way of use.

6.1. Setup

1. In the toolbar, under OPC role, select Server.
2. In the Server endpoint field, enter the address at which HVP will serve. The default value `opc.tcp://localhost:4841` is sufficient when the PLC simulator runs on the same computer.
3. If the PLC is on another device in the network, instead of localhost enter the IP address of this computer, e.g. `opc.tcp://192.168.0.10:4841`.
4. Change the Namespace URI if needed. The default value `urn:homa:virtualplant` is suitable for most cases.
5. Check Expose parameters if you want the model parameters to be available to the PLC as well.
6. Click Start OPC.

After a successful start, the status bar shows that OPC is active, and a message about the number of created nodes appears in the log.

FMU info

Model name

CoolingSystemVisual_v4_600

FMI version

2.0

GUID

{46d01e60-6e5f-4059-afa4-e62ccd9ede6b}

Model identifier

CoolingSystemVisual_v4_600

Generation tool

OpenModelica Compiler

Generation date

OpenModelica v1.27.0 (64-bit)

Generation date

2026-07-13T12:54:28Z

Description

Visual cooling system with Modelica.Fluid icons for demo presentation

Platforms

win64

Default experiment

start=0 stop=600 tol=0.000001 step=1

Co-Sim capabilities

varStep=True, interp=True, getSetState=False, serialize=False, oncePerProcess=False

OPC status

Server endpoint

opc.tcp://localhost:4841

Client endpoint

Namespace URI

urn:homa:virtualplant

Step size (s)

0.01

☐ Expose parameters

State: Running

Endpoint: opc.tcp://localhost:4841

Namespace URI: urn:homa:virtualplant

Exposed nodes: 15

Connected clients: 0

Uptime: 00:00:35

Last error:

Live runtime

Sim time (s)

82.15

Steps

8215

Loop (Hz)

100

OPC reads

350

OPC writes

350

Last update

14:16:23.925

CoolingSystemVisual_v4_600 | OPC status **Running**

Figure 12: OPC settings in Server mode

6.2. Address space structure

In the address space, HVP creates a HomaVirtualPlant folder inside the standard Objects folder. That folder contains one node for each exposed signal.

Node property	Value
BrowseName / DisplayName	The variable name from the model
NodeId	A numeric identifier equal to the variable reference in the model, in HVP's namespace
Data type	According to the OPC type column in the signal table

Node property	Value
Access rights – inputs	Read and write (the PLC writes values into the simulation)
Access rights – outputs	Read-only (the PLC reads the simulation results)

The screenshot shows the 'Unified Automation UaExpert - The OPC Unified Architecture Client - NewProject*' window. The 'PROJECT' tree on the left shows 'Homa Virtual Plant@localhost' under 'Servers'. The 'ADDRESS SPACE' tree shows a hierarchy from 'Root' to 'Objects' to 'Aliases' to 'HomaVirtualPlant', which contains various input and output nodes like 'u_heat_load', 'u_pump_speed', 'u_valve_opening', 'y_Q_cooler', 'y_Q_heater', 'y_Q_tank', and 'y_T_after_heater'. The 'DATA ACCESS VIEW' table lists 15 nodes with their server, node ID, node path, display name, value, datatype, source timestamp, server timestamp, and status code. The 'LOG' window at the bottom shows a series of successful connection and attribute reading messages.

#	SERVER	NODE ID	NODE PATH	DISPLAY NAME	VALUE	DATATYPE	SOURCE TIMESTAMP	SERVER TIMESTAMP	STATUSCODE
1	Homa Virtual Plant@localhost	NS2(Numeric)283	HomaVirtualPlant/u_heat_load	u_heat_load	0	Double	15:44:18.002	15:47:17.397	Good
2	Homa Virtual Plant@localhost	NS2(Numeric)284	HomaVirtualPlant/u_pump_speed	u_pump_speed	0	Double	15:44:18.002	15:47:29.793	Good
3	Homa Virtual Plant@localhost	NS2(Numeric)285	HomaVirtualPlant/u_valve_opening	u_valve_opening	0	Double	15:44:18.002	15:47:29.793	Good
4	Homa Virtual Plant@localhost	NS2(Numeric)291	HomaVirtualPlant/y_Q_cooler	y_Q_cooler	65.143484073303...	Double	15:48:50.312	15:48:50.312	Good
5	Homa Virtual Plant@localhost	NS2(Numeric)292	HomaVirtualPlant/y_Q_heater	y_Q_heater	0	Double	15:47:29.711	15:47:29.793	Good
6	Homa Virtual Plant@localhost	NS2(Numeric)293	HomaVirtualPlant/y_Q_tank	y_Q_tank	4.9970262144459...	Double	15:48:50.312	15:48:50.312	Good
7	Homa Virtual Plant@localhost	NS2(Numeric)294	HomaVirtualPlant/y_T_after_heater	y_T_after_heater	288.15000019933...	Double	15:48:50.312	15:48:50.312	Good
8	Homa Virtual Plant@localhost	NS2(Numeric)295	HomaVirtualPlant/y_T_return	y_T_return	288.94695858940...	Double	15:48:50.312	15:48:50.312	Good
9	Homa Virtual Plant@localhost	NS2(Numeric)296	HomaVirtualPlant/y_m_flow	y_m_flow	5.6212763509522...	Double	15:47:54.111	15:47:54.111	Good
10	Homa Virtual Plant@localhost	NS2(Numeric)297	HomaVirtualPlant/y_pump_pin	y_pump_pin	111092.41817624...	Double	15:48:50.312	15:48:50.312	Good
11	Homa Virtual Plant@localhost	NS2(Numeric)298	HomaVirtualPlant/y_pump_pout	y_pump_pout	111243.08725498...	Double	15:48:50.312	15:48:50.312	Good
12	Homa Virtual Plant@localhost	NS2(Numeric)299	HomaVirtualPlant/y_pump_speed_actua	y_pump_speed_actua	50	Double	15:47:29.711	15:47:29.793	Good
13	Homa Virtual Plant@localhost	NS2(Numeric)300	HomaVirtualPlant/y_tank_level	y_tank_level	1.0004153004717...	Double	15:48:50.312	15:48:50.312	Good
14	Homa Virtual Plant@localhost	NS2(Numeric)301	HomaVirtualPlant/y_tank_temp	y_tank_temp	288.15297378555...	Double	15:48:50.312	15:48:50.312	Good
15	Homa Virtual Plant@localhost	NS2(Numeric)302	HomaVirtualPlant/y_valve_pos	y_valve_pos	0.0050000000000...	Double	15:47:36.210	15:47:36.232	Good

Figure 13: Address space of the HVP server

NOTE The output signals are intentionally read-only. This prevents the PLC from accidentally overwriting a simulation result, which would bring the model into an inconsistent state.

6.3. Connecting the PLC

On the PLC side, you need to configure an OPC UA client connection to the address you entered in the Server endpoint field. On Siemens PLCs this is done through the OPC UA client interface in TIA Portal, and the details depend on the series and firmware version.

The security settings that HVP expects are:

Setting	Value
Security policy	None (no signing or encryption)
Login method	Anonymous
Certificates	HVP automatically accepts client certificates

IMPORTANT Because of these settings, HVP must not be exposed on a public or production network. It is intended for use on a development computer or in an isolated laboratory network.

6.4. Checking the connection

Before connecting the PLC, it is useful to check the server with an OPC UA client such as UaExpert:

1. In the client, add the server with the address you entered in HVP.
2. Select the None security policy and anonymous login.
3. Connect and find the HomaVirtualPlant folder in the tree.
4. Add a few variables to monitoring and start the simulation in HVP – the values must change.
5. To check writing, change the value of some input signal and watch the model's reaction.

7. Client mode

In this mode, HVP connects to an already existing OPC UA server – most often the one built into the PLC – and works with its existing variables. It is used when the PLC program and its variables already exist and you do not want to change them.

The key difference from Server mode is the Node Id column: for each exposed signal you need to enter the address of the corresponding variable on the remote server.

7.1. Setting up the connection

1. In the toolbar, under OPC role, select Client.
2. In the Client endpoint field, enter the address of the remote server, e.g.
`opc.tcp://192.168.0.1:4840`.
3. Make sure the remote server is running and reachable from this computer.

NOTE Enter the address before you try to browse the address space or use automatic mapping. Without it, HVP reports a message that the Client endpoint must be entered first.

7.2. Browsing the address space

To find out the exact Node Id of a variable, no external tool is needed – HVP has a built-in address space browser.

1. In the signal row, in the Node Id column, click the small button with a magnifier icon on the right side of the cell.
2. HVP connects to the server and opens a window with the address space tree.
3. Expand the tree down to the desired variable. Subnodes are loaded only when you expand a branch, so browsing is fast even with large servers.
4. Select the variable and confirm with OK.
5. The Node Id of the selected variable is automatically entered into the signal row.

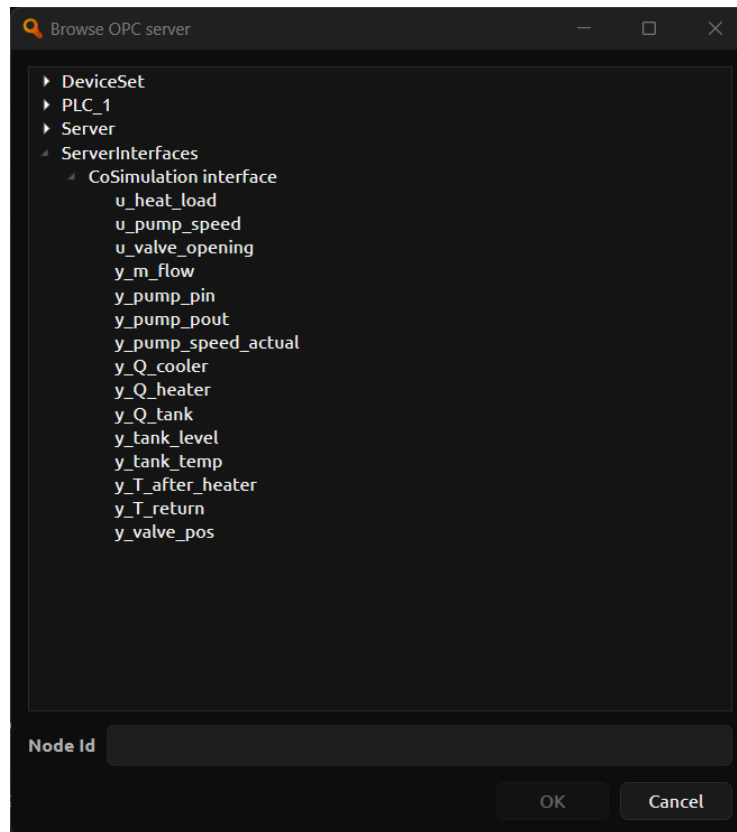


Figure 14: The address space browsing window

NOTE The magnifier button is visible only when the Client role is selected and OPC is stopped.

7.3. Automatic mapping by name

For models with many signals, manual browsing is slow. Automatic mapping connects to the server, reads all its variables and fills in the Node Id everywhere the variable name on the server matches the signal name in the model.

1. Make sure the Client endpoint is entered and OPC is stopped.
2. Click Auto-map by name – the button is on the bar above the signal tables, and the same action is also available in the Data menu.
3. If some signals already have a Node Id entered, HVP asks whether you want to overwrite them. Answering No fills in only the empty fields.
4. When finished, the number of successfully mapped signals is shown.

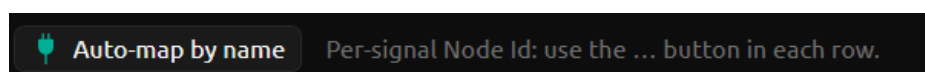


Figure 15: The bar with the automatic mapping button

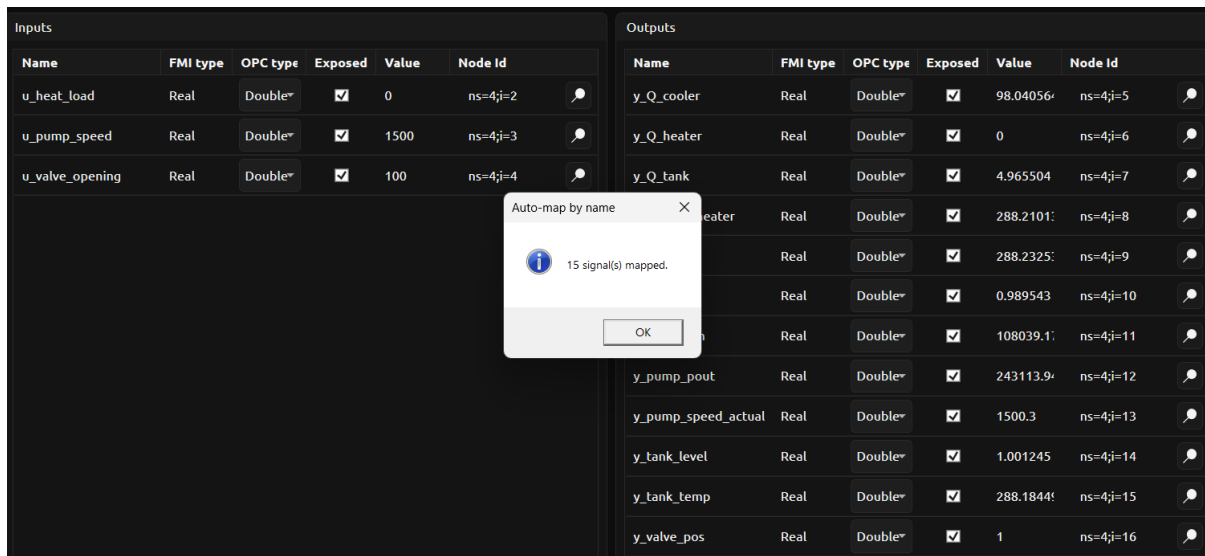


Figure 16: The result of automatic mapping

Name comparison is case-insensitive. In addition to the original name, a name in which special characters are replaced with underscores is also compared – the same rule used when generating PLC blocks. Thanks to this, a signal called tank1.level in the model will match the variable tank1_level on the PLC.

TIP If fewer signals were mapped after automatic mapping than you expected, check whether the signals are checked in the Exposed column – only exposed signals are mapped.

7.4. Starting communication

When the Node Id fields are filled in, click Start OPC. HVP connects to the server, subscribes to the variables and starts exchanging data. The OPC reads and OPC writes counters in the Live runtime section confirm that communication is flowing.

8. Signal table in Excel

The signal configuration can be exported to an Excel or CSV file, edited outside the application and imported back. For models with several hundred variables, this is considerably faster than editing row by row.

8.1. Export

1. Select Data → Export signals (Excel/CSV) or the same button in the toolbar.
2. Choose the location and file name and the format (.xlsx or .csv).
3. Confirm the save.

The resulting file contains one sheet named Signals with these columns:

Column	Meaning	Applied on import
ValueReference	A unique numeric identifier of the variable in the model. Used to match rows on import.	Key – do not change
Name	The variable name.	No (informational)
Causality	Causality: input, output or parameter.	No (informational)
FmiType	The variable type in the model.	No (informational)
OpcType	The OPC type under which the signal is exposed.	Yes
Exposed	TRUE or FALSE – whether the signal is exposed.	Yes
RemoteNodeId	The Node Id on the remote server (for Client mode).	Yes

	A	B	C	D	E	F	G	H
1	ValueReference	Name	Causality	FmiType	OpcType	Exposed	RemoteNodeId	
2	283	u_heat_load	Input	Real	Double	TRUE		
3	284	u_pump_speed	Input	Real	Double	TRUE		
4	285	u_valve_opening	Input	Real	Double	TRUE		
5	291	y_Q_cooler	Output	Real	Double	TRUE		
6	292	y_Q_heater	Output	Real	Double	TRUE		
7	293	y_Q_tank	Output	Real	Double	TRUE		
8	294	y_T_after_heater	Output	Real	Double	TRUE		
9	295	y_T_return	Output	Real	Double	TRUE		
10	296	y_m_flow	Output	Real	Double	TRUE		
11	297	y_pump_pin	Output	Real	Double	TRUE		
12	298	y_pump_pout	Output	Real	Double	TRUE		
13	299	y_pump_speed_actual	Output	Real	Double	TRUE		
14	300	y_tank_level	Output	Real	Double	TRUE		
15	301	y_tank_temp	Output	Real	Double	TRUE		
16	302	y_valve_pos	Output	Real	Double	TRUE		
17								
18								

Figure 17: The exported signal table in Excel

8.2. Editing

In Excel, the following changes are typically made:

- bulk checking or unchecking of signals in the Exposed column,
- changing OPC types for a larger number of signals at once,
- pasting a list of Node Id values obtained from the PLC project.

IMPORTANT Do not change or delete the ValueReference column and do not change the order of the columns. Import matches rows precisely by that value; without it the changes cannot be applied.

8.3. Import

1. Make sure OPC is stopped – import is not possible while communication is running.
2. Select Data → Import signals (Excel/CSV).
3. Select the previously edited file.
4. HVP shows the number of updated signals.

On import, only the OpcType, Exposed and RemoteNodeId columns are applied. Names, causality and FMI types are read from the model itself and cannot be changed through the file. Rows whose ValueReference value is not found in the loaded model are silently skipped.

TIP It is useful to keep the signal file together with the PLC project. When you re-export the model, you can restore the configuration in a few seconds instead of building it manually again.

9. Siemens: generating PLC blocks (SCL)

IMPORTANT This chapter applies exclusively to Siemens S7-1200 and S7-1500 series PLCs. The generated code is in SCL format and uses Siemens data block syntax, so it cannot be used on control systems from other manufacturers. All other HVP functions, including OPC UA communication, work with any PLC regardless of manufacturer – for those, see chapters 6 and 7.

9.1. What it is for

For the PLC to exchange data with the simulation, it needs a data structure that matches the model signals. Manually retyping dozens or hundreds of variables into TIA Portal is time-consuming and error-prone, especially with data types.

HVP can therefore automatically create an SCL file from which TIA Portal generates ready-made data blocks with correct names and types. The procedure requires neither TIA Openness nor additional licenses.

9.2. What is generated

HVP creates a single .scl file containing two data blocks:

Block	Contents	Data direction
HVP_FromPLC	All exposed model inputs	PLC → simulation (the PLC writes commands)
HVP_ToPLC	All exposed model outputs	Simulation → PLC (the PLC reads measurements)

The blocks are generated with optimized access (`S7_Optimized_Access := TRUE`), which is the recommended setting for S7-1200 and S7-1500.

NOTE Only signals checked in the Exposed column go into the blocks. Signals that are not exposed are skipped, just as in OPC communication.

9.3. Data type mapping

The type of each variable in the SCL is determined according to the OPC type column in the signal table:

OPC type in HVP	SCL type	Note
Boolean	Bool	Digital signals
SByte	SInt	8-bit signed integer
Byte	USInt	8-bit unsigned
Int16	Int	The classic Siemens Int
UInt16	UInt	

OPC type in HVP	SCL type	Note
Int32	DInt	The default for integer model variables
UInt32	UDInt	
Int64	LInt	S7-1500 only
UInt64	ULInt	S7-1500 only
Float	Real	32-bit real number
Double	LReal	The default for real model variables; 64-bit
String	String	Rarely used in this context

TIP If you work with an S7-1200 or want to save memory, switch the real signals in the OPC type column from Double to Float. The SCL will then produce Real instead of LReal, which is the more common type in Siemens programs.

9.4. Rules for variable names

Names from the model often contain characters that are not allowed in S7 identifiers, so HVP adapts them automatically:

- All characters except letters, digits and the underscore are replaced with an underscore. The name tank1.level becomes tank1_level.
- If a name begins with a digit, an underscore is added at the start.
- If after adaptation two signals get the same name, _1 is added to the second, _2 to the third, and so on.

A comment with the original causality and the variable reference in the model is written next to each variable, e.g.:

```
tank1_level          : LReal ; // FMU output vr=17
```

NOTE Remember how a name was adapted because the same procedure is used in automatic mapping in Client mode (chapter 7.3).

9.5. Generating the file

1. Load the FMU and check the signals you want in the Exposed column.
2. Check the OPC types against the table in section 10.3.
3. Click Generate PLC blocks (SCL) in the toolbar or select the same action in the menu.
4. Choose the location and file name. The suggested name is derived from the project name and ends with _DBs.scl.
5. Confirm the save. HVP shows the path to the created file.

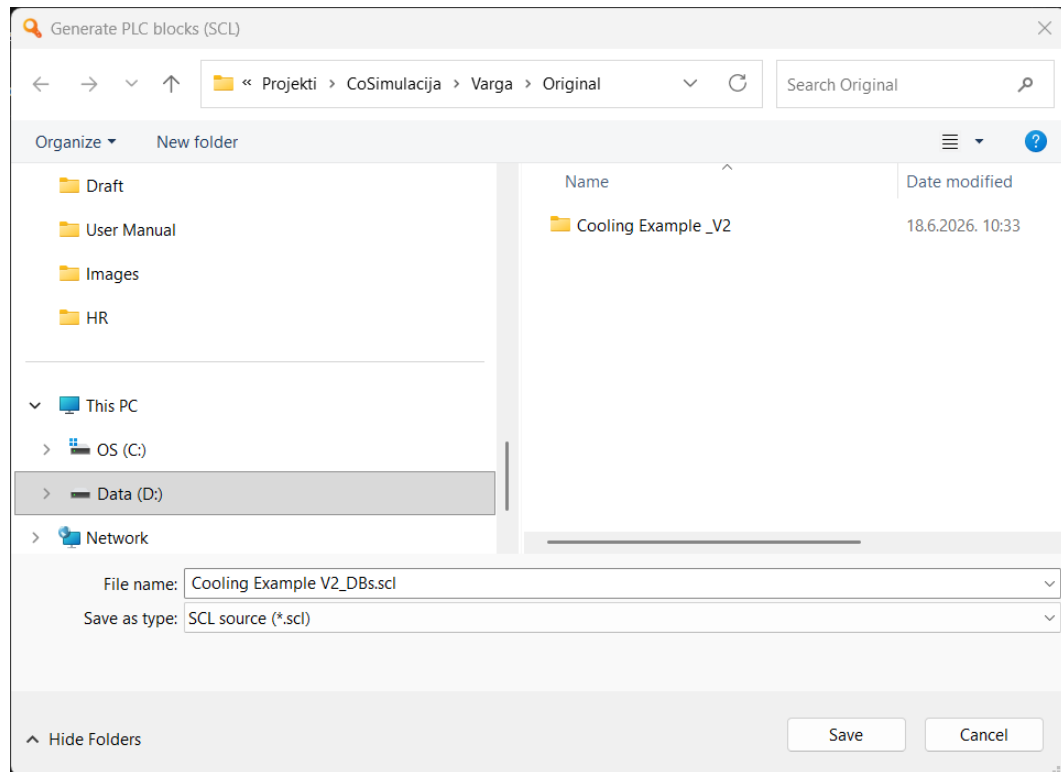


Figure 18: Generating the SCL file

9.6. Layout of the generated file

The file is a plain text file that you can open with any text editor. The structure is as follows (abbreviated example):

```
DATA_BLOCK "HVP_FromPLC"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
STRUCT
    pump_cmd           : Bool    ; // FMU input vr=3
    valve_setpoint     : LReal   ; // FMU input vr=5
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

```
DATA_BLOCK "HVP_ToPLC"
{ S7_Optimized_Access := 'TRUE' }
VERSION : 0.1
STRUCT
    tank1_level        : LReal   ; // FMU output vr=17
    pump_running       : Bool    ; // FMU output vr=18
END_STRUCT;
BEGIN
END_DATA_BLOCK
```

Both blocks are in the same file, so a single import in TIA Portal creates both data blocks.

9.7. Import into TIA Portal

The SCL file is imported as an external source file. The procedure is as follows:

1. In TIA Portal, open the project and find your PLC in the project tree.
2. Expand the External source files node.
3. Double-click Add new external file and select the generated .scl file.
4. The file appears in the list of external source files.
5. Right-click it → Generate blocks from source.
6. Confirm the warning that existing blocks will be overwritten, if it appears.
7. The blocks HVP_FromPLC and HVP_ToPLC are created in the Program blocks folder.

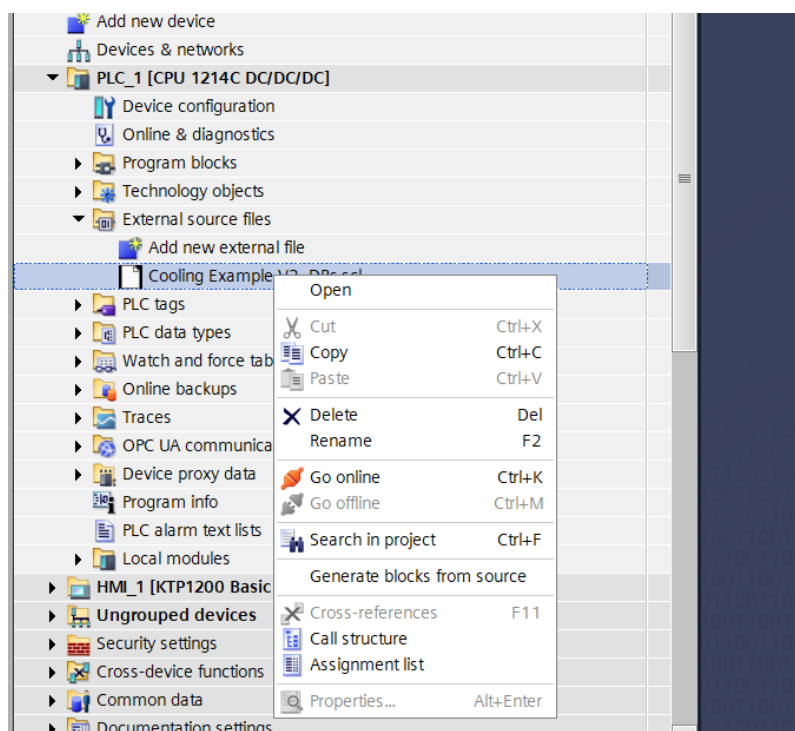


Figure 19: Importing the SCL file into TIA Portal

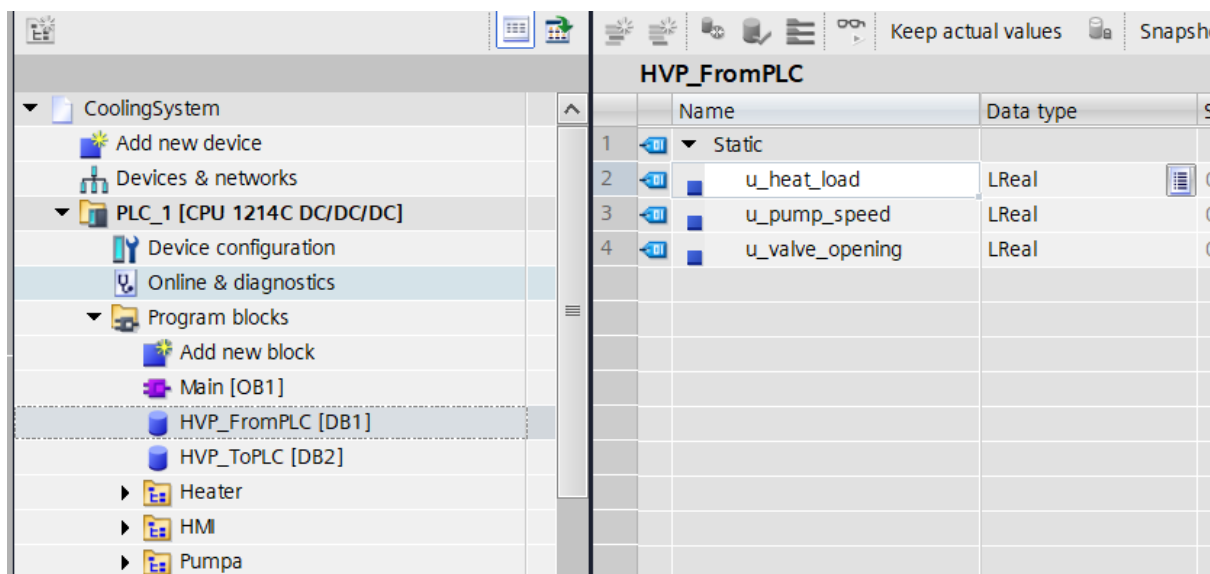


Figure 20: The generated blocks in TIA Portal

NOTE If an error is reported when generating the blocks, the most common cause is a variable name that coincides with a reserved word in SCL. In that case, rename the variable in the model or manually correct the name in the .scl file before importing.

9.8. Using the blocks in the program

After generation, the blocks are accessed like any other data block. The values are used in the program as follows:

```
// PLC sends a command to the simulation
"HVP_FromPLC".pump_cmd := "Start_Button";

// PLC reads a measurement from the simulation
IF "HVP_ToPLC".tank1_level > 80.0 THEN
    "Alarm_HighLevel" := TRUE;
END_IF;
```

It is recommended that the program does not read and write directly to these blocks throughout the entire logic, but instead copies the values into its own process variables in one place. This reduces the later transition from simulation to real equipment to changing that single place.

9.9. Connecting the blocks to the simulation

The generated blocks by themselves do not communicate with the simulation – they are only a data structure. The connection is established by OPC UA, in one of two ways:

HVP mode	How it connects
Server	The PLC, as an OPC UA client, connects to HVP and maps its variables from the blocks to the nodes in the HomaVirtualPlant folder. See chapter 6.

HVP mode	How it connects
Client	The PLC exposes its variables as an OPC UA server, and HVP connects to it. In the Node Id column, the addresses of the variables from the generated blocks are entered, most quickly using automatic mapping. See chapter 7.

TIP The combination that has proven fastest in practice: generate the blocks from HVP, expose them in the PLC as an OPC UA server, switch HVP to the Client role and run automatic mapping. The names then match because they originate from the same source, so the entire table is mapped at once.

9.10. When you change the signals

If you add or remove signals in the model, or change which ones are exposed, the generated blocks need to be refreshed:

1. In HVP, regenerate the .scl file.
2. In TIA Portal, replace the existing external source file with the new one.
3. Run Generate blocks from source again.
4. Compile the project and download it to the PLC.

IMPORTANT Regeneration overwrites the existing blocks. If you have manually added variables to them, those changes will be lost. Therefore keep your own variables in a separate block.

9.11. If you do not use a Siemens PLC

SCL generation is the only HVP function tied to a single manufacturer. Everything else works the same with any control system that supports OPC UA.

For PLCs from other manufacturers, the procedure is:

1. Skip SCL generation.
2. Use HVP in Server mode (chapter 6).
3. In your development tool, configure an OPC UA client to the address of the HVP server.
4. You will obtain the list of required variables and their types by exporting the signal table to Excel (chapter 8).

10. Application options

The options are opened via Help → Options.

10.1. Interface language

HVP supports Croatian, English and German. After selecting and confirming, the application offers to restart because the language is applied only on the next start. The choice is remembered and applies to all subsequent starts.

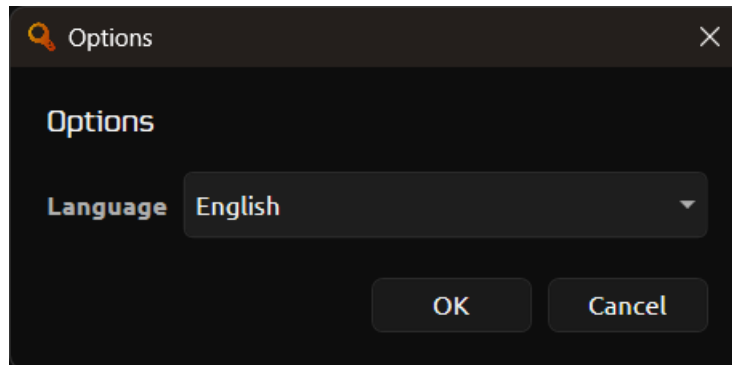


Figure 21: The Options window

10.2. About

Help → About shows the application version, information about the author, links to the website and email, and a list of the open-source components used. It is useful to state the version number with every problem report.

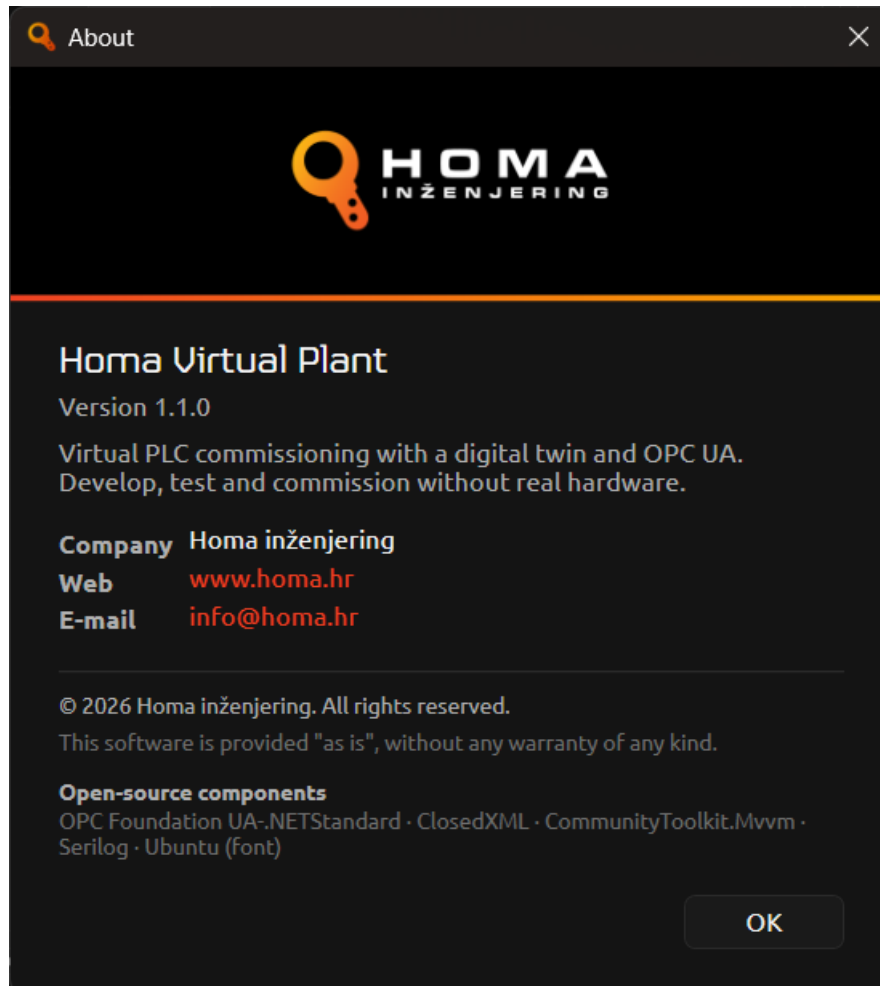


Figure 22: The About window

11. Troubleshooting

The tables list the most frequently occurring problems, their causes and solutions. For each problem, it is useful to also look at the log in the bottom area of the window.

11.1. Loading the model

Symptom	Cause	Solution
The FMU does not load, an error about loading a library	The FMU does not contain a 64-bit Windows binary file, or the accompanying libraries are missing.	Re-export the model with the win64 platform selected. Check the Platforms field in the FMU info.
A message about an unsupported FMI version	The model was exported as FMI 3.0 or as Model Exchange.	Re-export the model as FMI 2.0, Co-Simulation type.
The project opens, but the model does not load	The .fmu file was moved or renamed.	Reload the FMU via Load FMU... and save the project.

11.2. Starting the simulation

Symptom	Cause	Solution
An error when exiting initialization (ExitInitializationMode)	The model fails to solve its initial system of equations. The cause is in the model itself, not in HVP.	Check whether the model initializes in its source tool. A common cause is an input left at zero, making the system singular – set meaningful initial values.
An error when executing a step (DoStep)	The model's solver failed to compute a step – too large a step or a numerical problem in the model.	Click Stop simulation, reduce the Step size (s) (e.g. from 0.01 to 0.001) and try again.
After an error, restarting fails immediately	After an error, the model is in an invalid state.	Be sure to click Stop simulation (not Resume) before starting again – this returns the model to its initial state.
The simulation runs slower than real time	The computer cannot run the model fast enough.	Increase the Step size (s), simplify the model or reduce the number of exposed signals. Compare Loop (Hz) with the value $1 / \text{Step}$.
Memory usage keeps rising	A known bug in older versions of some FMU export tools.	Update the simulation tool to the current version and re-export the model.

11.3. OPC communication

Symptom	Cause	Solution
OPC does not start, an error about a busy port	The port is already used by another program – most often the OPC UA Local Discovery Server on port 4840 or another running instance of HVP.	In the Server endpoint field, enter a different port, e.g. <code>opc.tcp://localhost:4841</code> or <code>4842</code> .

Symptom	Cause	Solution
The client cannot connect to HVP	A firewall is blocking the port, or localhost was entered instead of the computer's IP address.	Allow the port in the firewall and enter the IP address of this computer instead of localhost in the Server endpoint field.
HVP does not connect to the remote server	An incorrect address, the server is not running, or it requires a security policy and login.	Check the address and the server's availability. HVP uses the None policy and anonymous login – this must be allowed on the server.
The values do not change even though OPC is running	The simulation is not running, or the signals are not marked as exposed.	Start the simulation and check the Exposed column. The OPC reads and OPC writes counters must increase.
The PLC cannot write to an output signal	The outputs are intentionally set to read-only.	The PLC may write only to the model inputs. Check whether you are writing to the correct signal.
The OPC settings are grayed out	Communication is running.	Click Stop OPC, change the settings and start again.

11.4. Mapping and browsing

Symptom	Cause	Solution
The magnifier button is not visible	The Server role is selected, or OPC is running.	Switch the role to Client and stop OPC.
A message that the Client endpoint is missing	The address of the remote server was not entered.	Enter the address in the Client endpoint field and try again.
Automatic mapping does not find signals	The names on the server do not match the names in the model, or the signals are not exposed.	Check the signals in the Exposed column. Check the names – the comparison allows special characters to be replaced with underscores, but not different names. Enter the remaining signals manually via the magnifier.

11.5. Other

Symptom	Cause	Solution
A Windows warning during installation or launch	The application is not digitally signed.	Select More info → Run anyway. If needed, compare the SHA-256 checksum of the downloaded file with the published one.
Changing the language has no effect	The language is applied only after a restart.	Confirm the offer to restart or manually close and reopen the application.
Where to find the log for reporting a problem	—	In the %AppData%\HomaVP\logs folder, the file homavp-YYYYMMDD.log for the corresponding date.

TIP When reporting a problem, attach: the HVP version (Help → About), the log file for that day, the name and the tool with which the FMU was exported, and a description of the steps that lead to the problem.

12. Glossary

Term	Explanation
Co-simulation	A coupling method in which each participant has its own solver and exchanges data with the others at defined time points.
Digital twin	A simulation model of a real process or machine that behaves faithfully enough that a control program can be tested against it.
Endpoint	The address of an OPC UA connection, of the form <code>opc.tcp://computer:port</code> .
FMI	Functional Mock-up Interface – an open standard for exchanging simulation models between tools.
FMU	Functional Mock-up Unit – a package (.fmu) that contains a model built according to the FMI standard.
Causality	The role of a variable in the model: input, output or parameter.
Communication step	The time interval at which HVP advances the model and exchanges data; the Step size (s) field in the interface.
Namespace URI	The identifier of a namespace in the OPC UA address space; it separates nodes from different sources.
Node Id	The unique address of a node on an OPC UA server.
OPC UA	A vendor-independent industrial communication standard, supported on most modern PLCs.
SCL	Structured Control Language – a textual programming language for Siemens PLCs, related to Structured Text.
SIL	Software-in-the-Loop – testing a control program against a simulation model instead of real equipment.
Solver	A numerical method that solves the model equations and advances it through time.
ValueReference	A numeric identifier of a variable within the FMU; HVP uses it as a key when importing the signal table.
Virtual commissioning	Commissioning a control program against a simulation model, before the equipment is physically available.

13. Appendices

13.1. Default values

Setting	Default value
Interface language	Croatian
OPC role	Server
Server endpoint	opc.tcp://localhost:4841
Client endpoint	(empty)
Namespace URI	urn:homa:virtualplant
Step size (s)	0.01
Speed factor	1.0
Expose parameters	unchecked

13.2. Summary table of type mappings

Through the entire chain – from the model, through OPC, to the Siemens PLC – the types are mapped as follows:

FMI type (model)	Default OPC type	SCL type (Siemens)
Real	Double	LReal
Real (manually switched to Float)	Float	Real
Integer	Int32	DInt
Integer (manually switched to Int16)	Int16	Int
Boolean	Boolean	Bool
String	String	String
Enumeration	Int32	DInt

13.3. Files and locations

What	Where
Application options	%AppData%\HomaVP\appsettings.json
Logs	%AppData%\HomaVP\logs\homavp-YYYYMMDD.log
Project	A file with the .hvp extension, at a location the user chooses
Signal export	An .xlsx or .csv file, at a location the user chooses
Generated PLC blocks	An .scl file, at a location the user chooses

13.4. Keyboard shortcuts

Shortcut	Action
Ctrl + S	Save project

13.5. Support

For questions, problem reports and suggestions:

Homa inženjering

Web: **www.homa.hr**

Email: **info@homa.hr**

NOTE With a problem report, attach the application version, the log file from the %AppData%\HomaVP\logs folder and, if possible, the FMU file on which the problem occurs.

© 2026 Homa inženjering. All rights reserved.
Homa Virtual Plant 1.1 – User Manual