TIA Portal V17 Application Example with SIMATIC Process Function Library V2.0

SIMATIC Process Function Library (PFL) V2.0 SP1

https://support.industry.siemens.com/cs/ww/en/view/109805499

Siemens Industry Online Support

Legal information

Use of application examples

Application examples illustrate the solution of automation tasks through an interaction of several components in the form of text, graphics and/or software modules. The application examples are a free service by Siemens AG and/or a subsidiary of Siemens AG ("Siemens"). They are non-binding and make no claim to completeness or functionality regarding configuration and equipment. The application examples merely offer help with typical tasks; they do not constitute customer-specific solutions. You yourself are responsible for the proper and safe operation of the products in accordance with applicable regulations and must also check the function of the respective application example and customize it for your system.
Siemens grants you the non-exclusive, non-sublicensable and non-transferable right to have the application examples used by technically trained personnel. Any change to the application examples is your responsibility. Sharing the application examples with third parties or copying the application examples or excerpts thereof is permitted only in combination with your own products. The application examples are not required to undergo the customary tests and quality inspections of a chargeable product; they may have functional and performance defects as well as errors. It is your responsibility to use them in such a manner that any malfunctions that may occur do not result in property damage or injury to persons.

Disclaimer of liability

Siemens shall not assume any liability, for any legal reason whatsoever, including, without limitation, liability for the usability, availability, completeness and freedom from defects of the application examples as well as for related information, configuration and performance data and any damage caused thereby. This shall not apply in cases of mandatory liability, for example under the German Product Liability Act, or in cases of intent, gross negligence, or culpable loss of life, bodily injury or damage to health, non-compliance with a guarantee, fraudulent non-disclosure of a defect, or culpable breach of material contractual obligations. Claims for damages arising from a breach of material contractual obligations shall however be limited to the foreseeable damage typical of the type of agreement, unless liability arises from intent or gross negligence or is based on loss of life, bodily injury or damage to health. The foregoing provisions do not imply any change in the burden of proof to your detriment. You shall indemnify Siemens against existing or future claims of third parties in this connection except where Siemens is mandatorily liable.
By using the application examples, you acknowledge that Siemens cannot be held liable for any damage beyond the liability provisions described.

Other information

Siemens reserves the right to make changes to the application examples at any time without notice. In case of discrepancies between the suggestions in the application examples and other Siemens publications such as catalogs, the content of the other documentation shall have precedence.
The Siemens terms of use (https://support.industry.siemens.com) shall also apply.

Security information

Siemens provides products and solutions with Industrial Security functions that support the secure operation of plants, systems, machines and networks.
In order to protect plants, systems, machines and networks against cyber threats, it is necessary to implement – and continuously maintain – a holistic, state-of-the-art industrial security concept. Siemens' products and solutions constitute one element of such a concept.
Customers are responsible for preventing unauthorized access to their plants, systems, machines and networks. Such systems, machines and components should only be connected to an enterprise network or the Internet if and to the extent such a connection is necessary and only when appropriate security measures (e.g. firewalls and/or network segmentation) are in place.
For additional information on industrial security measures that may be implemented, please visit https://www.siemens.com/industrialsecurity.
Siemens' products and solutions undergo continuous development to make them more secure. Siemens strongly recommends that product updates are applied as soon as they are available and that the latest product versions are used. Use of product versions that are no longer supported, and failure to apply the latest updates may increase customer's exposure to cyber threats.
To stay informed about product updates, subscribe to the Siemens Industrial Security RSS Feed at: https://www.siemens.com/industrialsecurity.

Table of contents

1 Introduction

1.1 Overview

This documentation gives a small practical overview over the Process Function Library V2.0 SP1 and its components. It also gives an overview and an explanation about the application example for the MTP TIA Portal project and how it works.

With this project the SIMATIC Process Function Library is shown, how they work and how you can use them. You see how it needs to be configured in the Service Engineering Tool and what is necessary in the TIA Portal for their usage.

1.2 Mode of operation

The hole application example can be operated with MTP services. You only need to start these services via the HMI runtime and the configured services will do the process of the small plant. You can start the process of a cleaner Station, which is the content of the project, in three different ways, which will be explained later in Chapter “Services in the HMIServices in the ”. For the implementation of all components the blocks from the SIMATIC Process Function Library are used.

1.3 Components used

This application example has been created with the following hardware and software components:

Table 1-1

Component	Note
Windows 10 Enterprise LTSC 2019	
TIA V17 Upd2	
WinCC Unified V17	
S7-PLCSIM Advanced V4.0	
SIMATIC Process Function Library V2.0 SP1	

(The update 2 or update 1 of TIA V17 is mandatory, otherwise the project can´t be opened)

2 How to use the SIMATIC Process Function Library V2.0 SP1

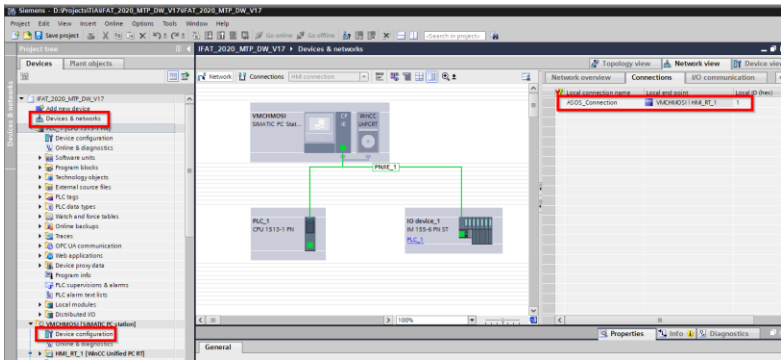
Please note:

The software controller CPU 1507 is not supported. Please use hardware controller e.g. S7-1512C-1PN or S7-1510SP-1PN.

2.1 Library

In the following chapter the general handling of the library is shown. Especially how the user can use the included objects and bring them to visualization in runtime.

The description starts with a TIA project which includes a PLC [CPU 1513-1 PN], an IO device_1 [ET200SP] and a PC-System [SIMATIC PC station]. All devices are connected via PN/IE and one HMI connection.



The following IP addresses are used:

Table 2-1

	Station	IP address
1.	IE general (HMI)	192.168.0.2
2.	PLC (AS)	192.168.0.5
3.	IO device	192.168.0.10
4.	Host	192.168.0.2

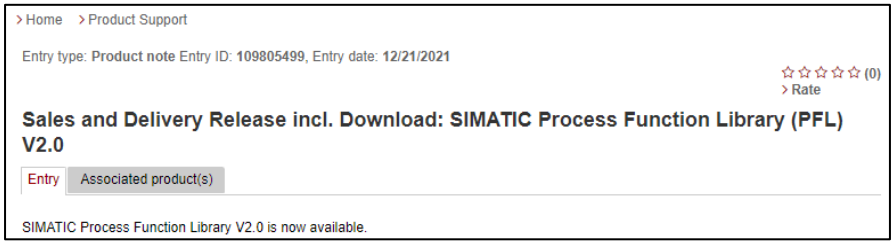
2.1.1 Library download and import

The following chapter describes how to include the SIMATIC Process Function Library into the project and how you can use it.

Download and install the SIMATIC Process Function Library on your target system which should fulfill the Software and Hardware prerequisite.

You can find the software with the following link:

https://support.industry.siemens.com/cs/ww/en/view/109805499



6. Download and Readme

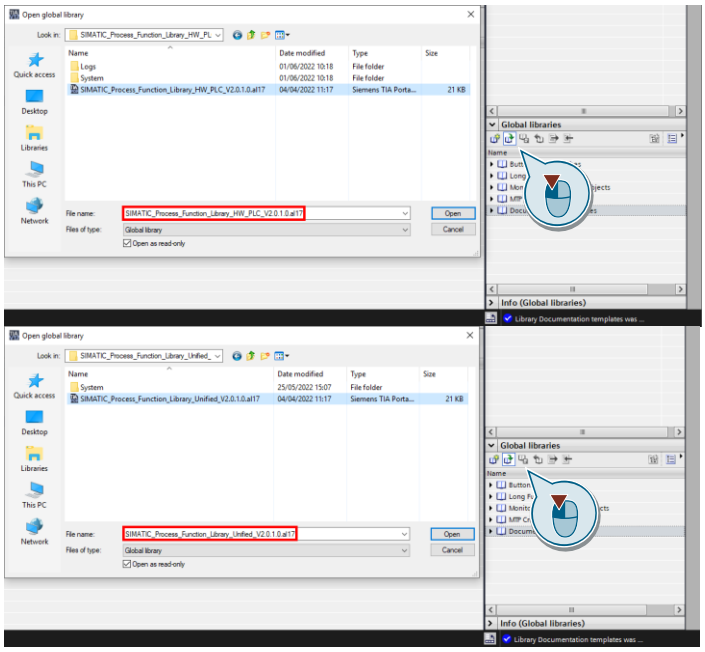
Before installing, refer to the notes and compatibility statements in the readme.

Product	Software version	Download	Readme
SIMATIC Process Function Library	V2.0	Download (77 MB) (SHA-256)	Readme

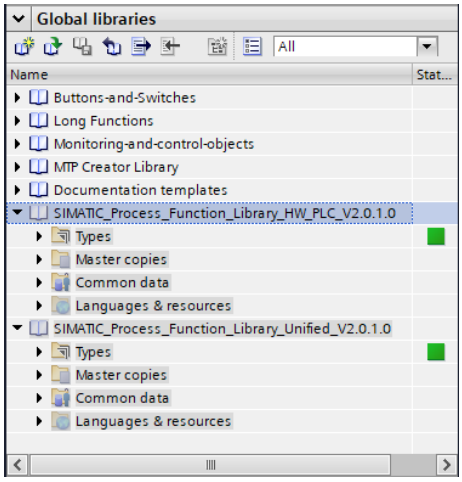
After the installation the SIMATIC Process Function Library is in the following folder:

C:\Program Files\Siemens\Automation\SIMATIC Process Function Library\V2.0.1.0

Open the two libraries in your TIA Portal environment. For that open the menu “Global libraries” on the right side. Open the explorer window with the “Open global library” button. Choose the global library in the field “Files of type”. Select the library and press “Open”. Store the library in a separate folder. Repeat this for the second library.



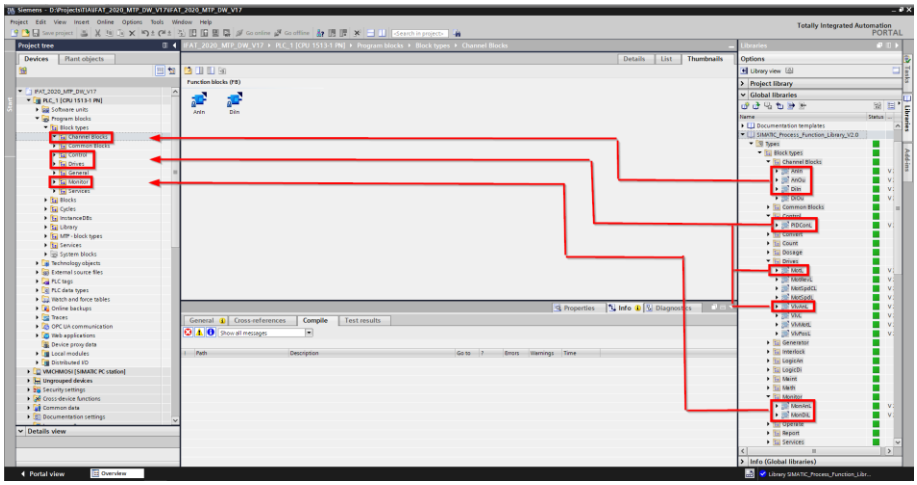
Afterwards it looks like this in the menu “Global libraries”.



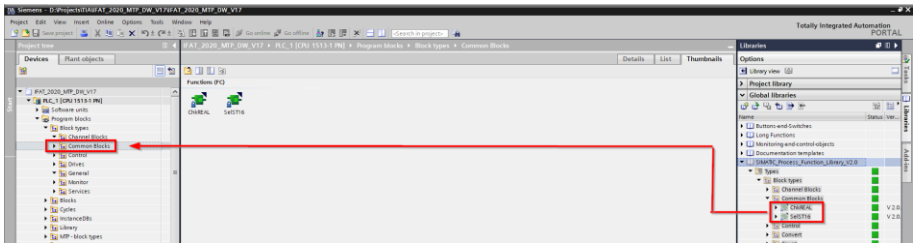
2.1.2 Usage of SIMATIC Process Function Library block/data types

The following chapter describes how the library objects can be used in the PLC [CPU 1516-3 PN/DP].

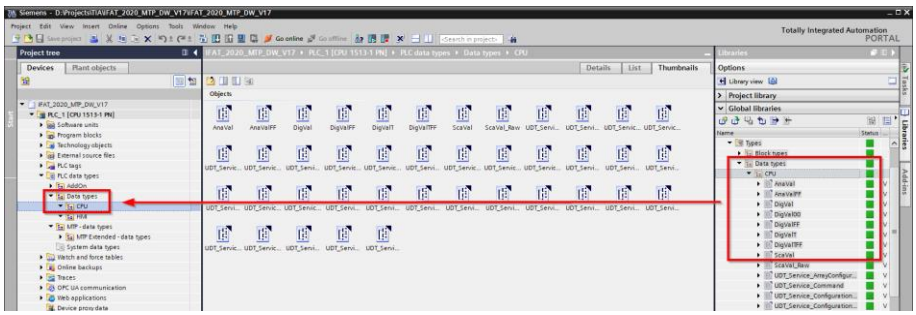
1. Open the folders for the block types in the library on the right side. Select the PFL-block types “PIDConL”, “MotL”, “MotSpdCL”, “VlvL”, “AccuS”, “MonAnL” and “MonDiL” and shift them via drag & drop to the program blocks of the PLC [CPU 1516-3 PN/DP].



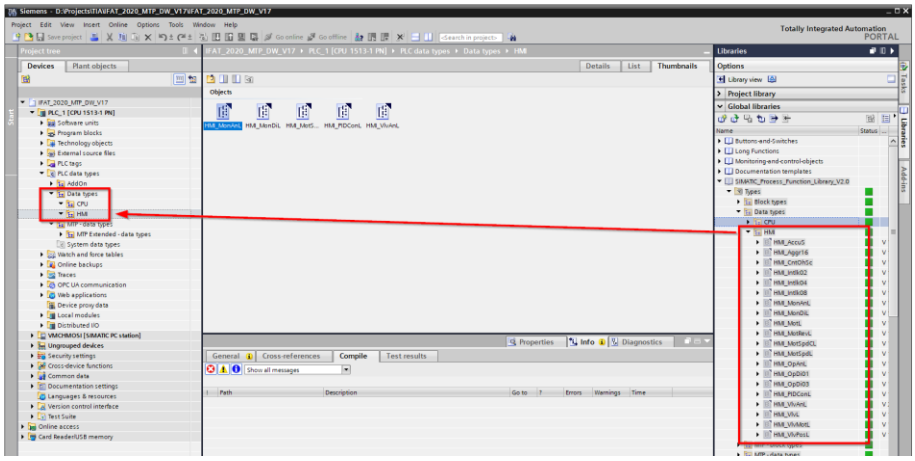
2. After the PFL-block types are copied to the program blocks, the functions “ChkREAL” and “SelST16” are copied automatically to the program blocks because these functions are used into the PFL-block types.



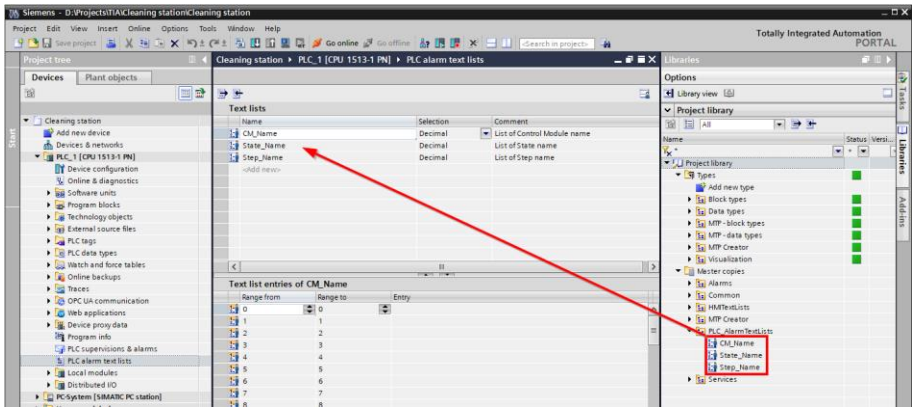
3. After the PFL-block types are copied to the program blocks the CPU data types “AnaVal”, “AnaValFF”, “DigVal”, “DigValFF”, “DigValT”, “DigValTFF” and “ScaVal” are copied automatically to the PLC data types because these data types are also used into the PFL-block types.



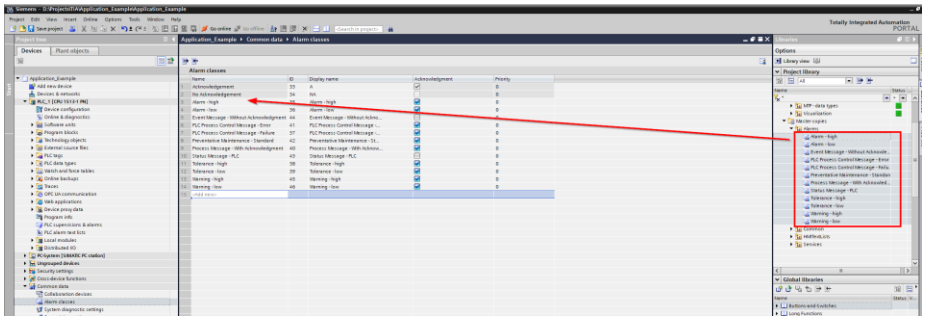
4. After the PFL-block types are copied to the program blocks the HMI data types “HMI_AccuS”, “HMI_MonAnL”, “HMI_MonDiL”, “HMI_MotL”, “HMI_MotSpdCL”, “HMI_PIDConL” and “HMI_VlvL” are copied automatically to the PLC data types because these data types are also used into the PFL-block types.



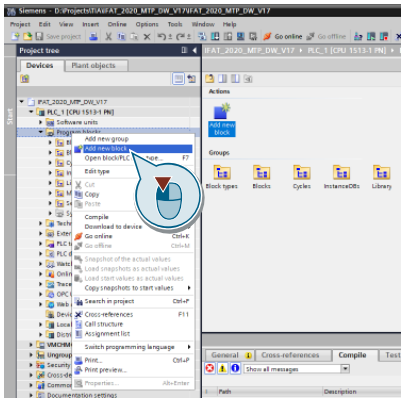
5. Add the PLC alarm text lists via drag & drop from the PFL to your project. It is located unter “PLC_1 > PLC alarm text lests”



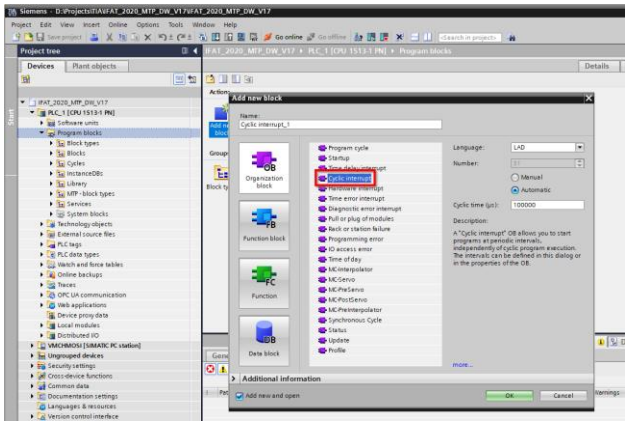
6. Add all alarm classes via copy & paste from the PFL to your project. It is located into “Common data” > “Alarm classes”



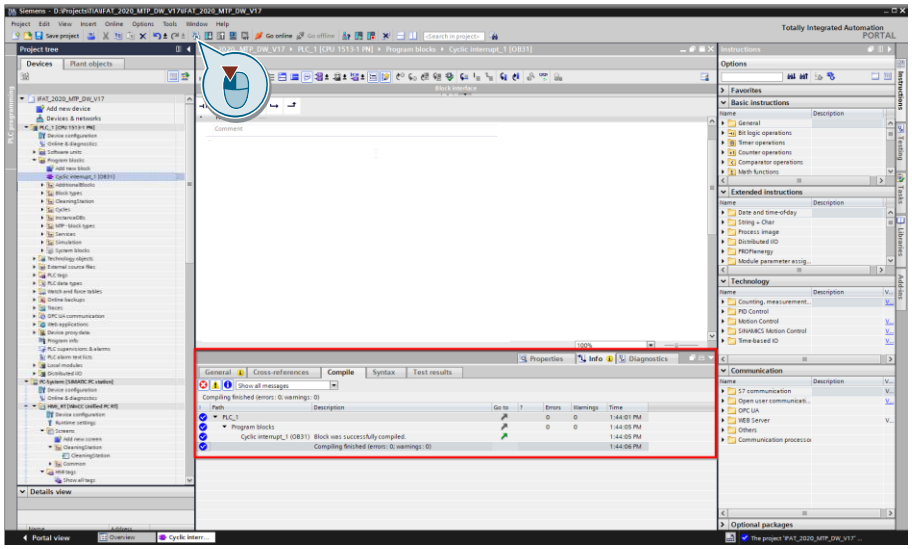
7. Go to the folder “Program blocks” and click on the right mouse button to add a new block.



8. Select “Cyclic interrupt” from the organization blocks, choose “FBD” as language and insert this block via the Ok button.



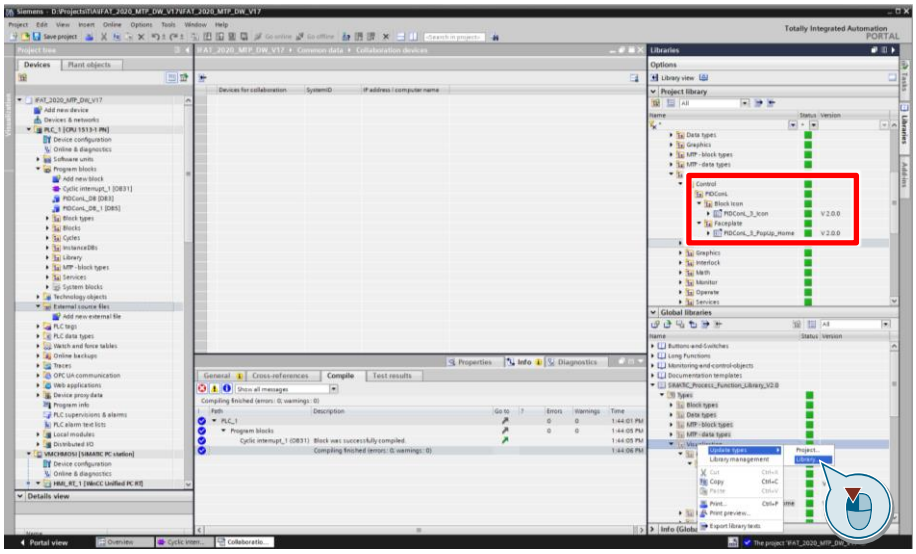
9. As the last step compile the program.



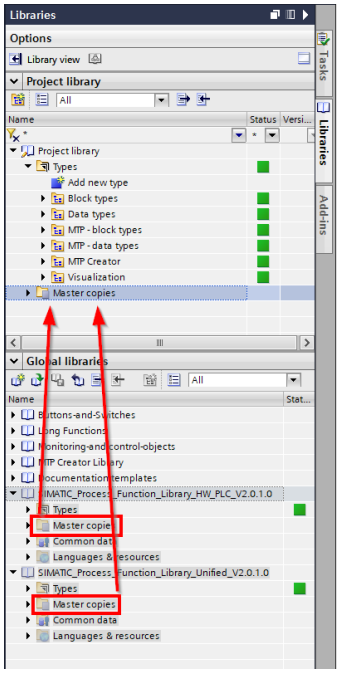
2.1.3 Usage of SIMATIC Process Function Library visualization

In the following chapter it is described how the library objects are used in the PC-System [SIMATIC PC station].

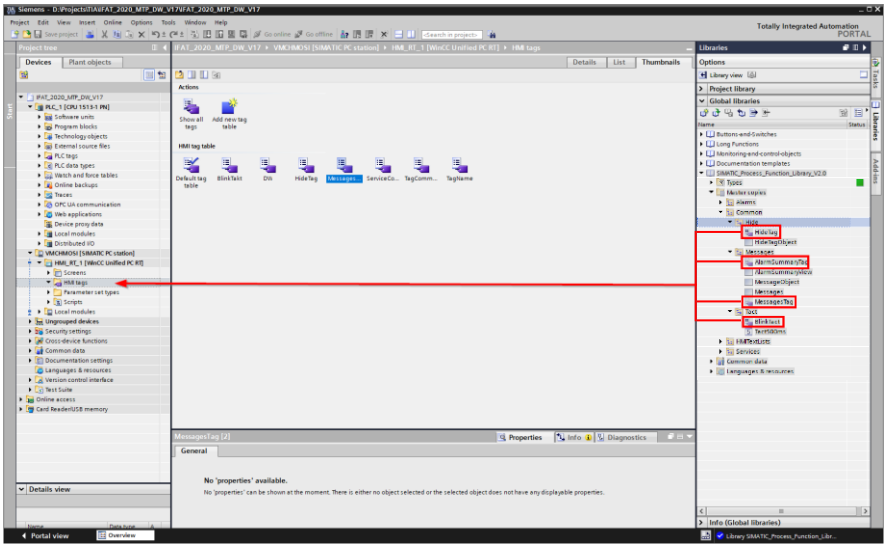
1. All faceplates from the library folder “PFL–Visualization” should be added to the project library over the context menu of the “Visualization” folder in the global library.



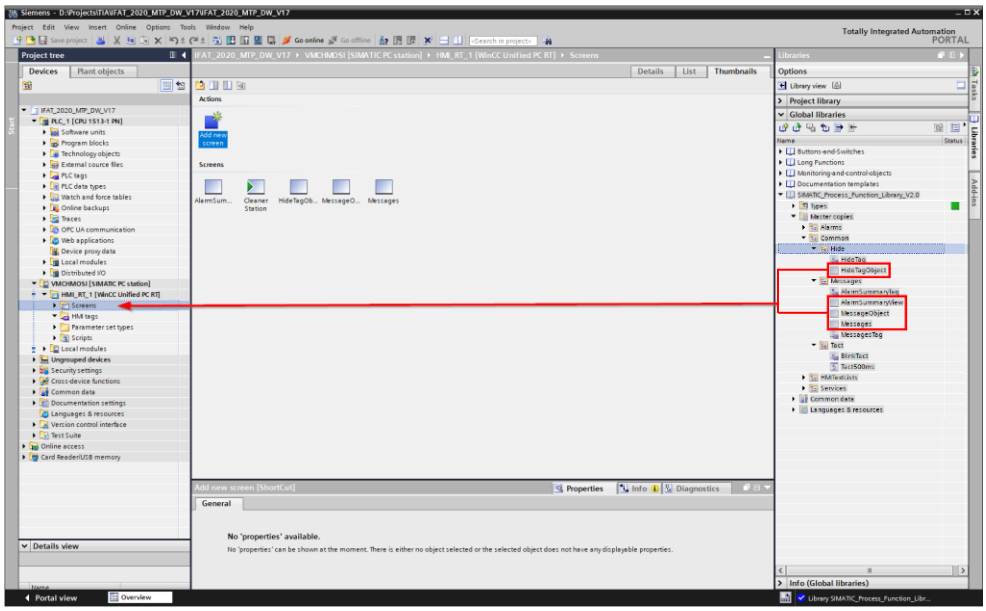
2. Add the master copies from the SIMATIC Process Function Library via copy & paste to your project library



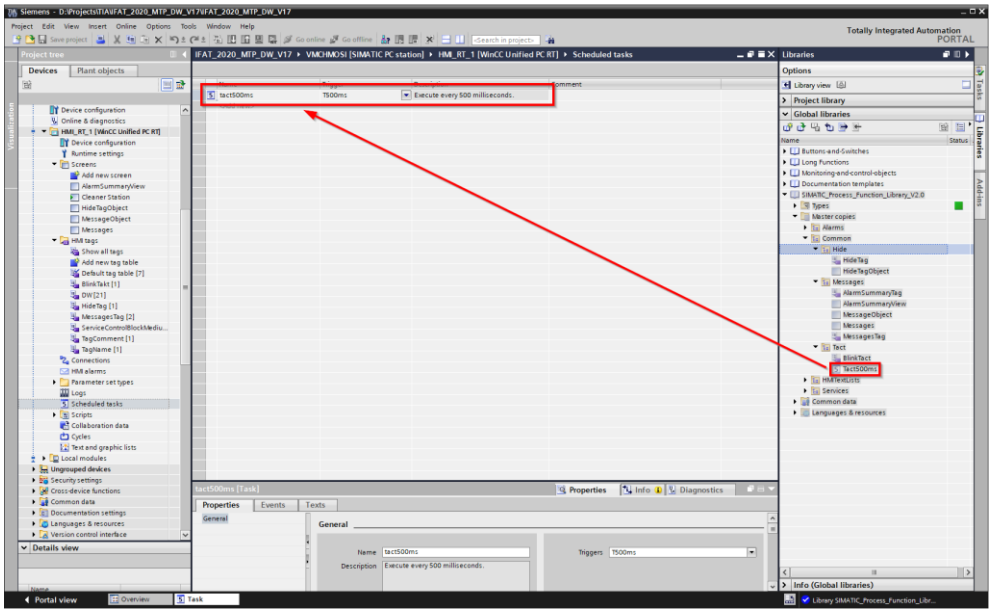
3. All common HMI tables from the library folder “Common” should be copied via drag & drop to the folder “HMI tags” under the HMI_RT of the PC station.



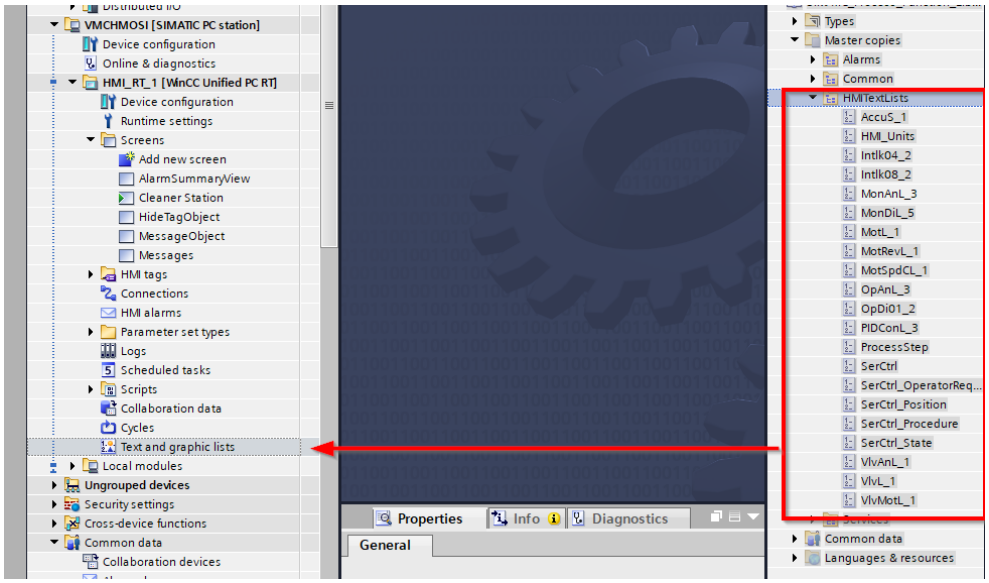
4. All screens from the library folder “Common” should be copied via drag & drop to the folder “Screens” under the HMI_RT of the PC station.



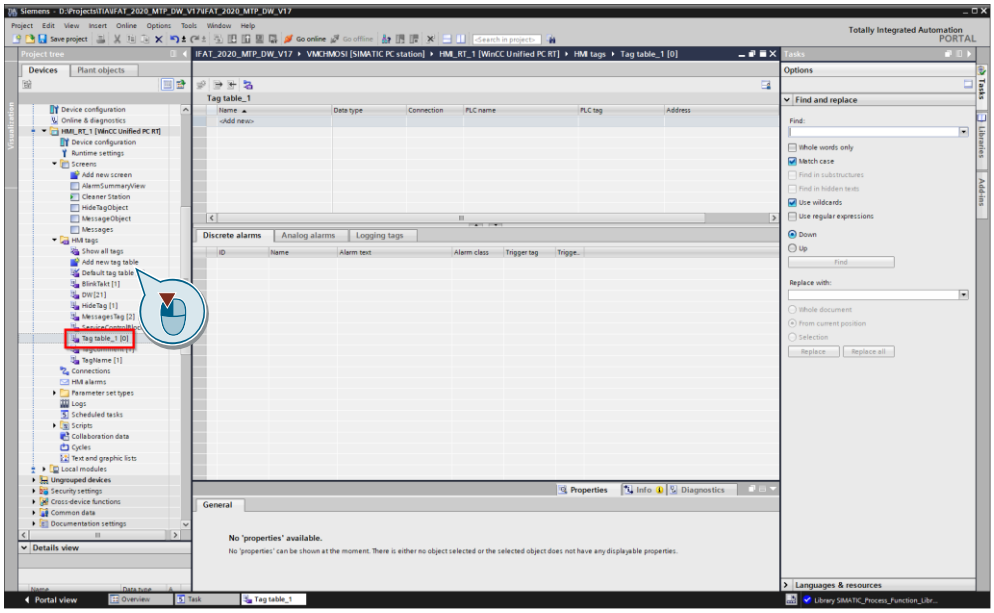
5. The scheduled tasks “Tact500ms” from the library folder “Tact” should be copied via drag & drop to the Scheduled tasks under the HMI_RT of the PC station. This task is used to toggle the internal HMI-Tag “tact500ms”.



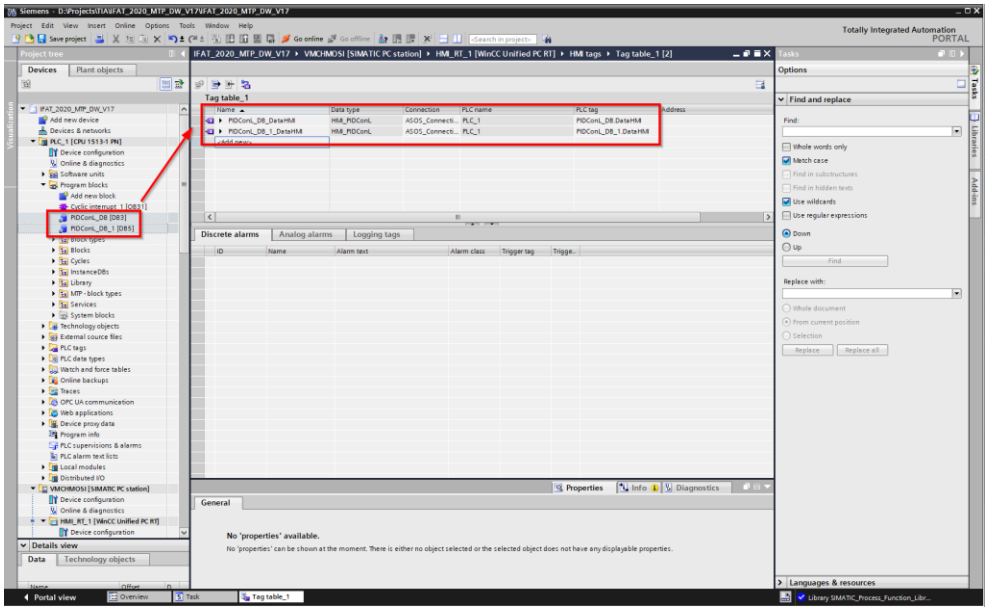
6. HMITextLists to Text and graphic lists



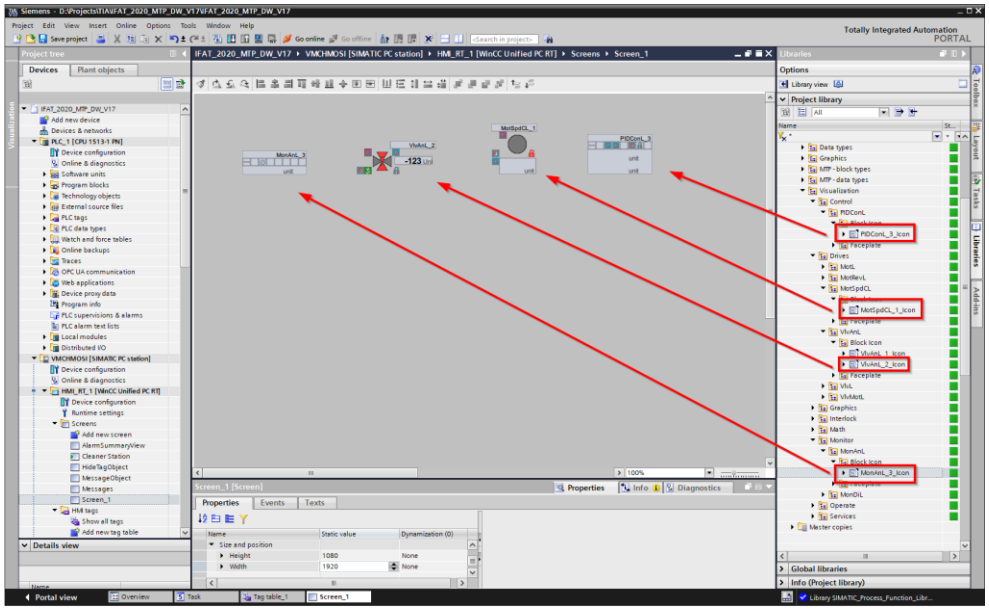
7. Add a new tag table “PLCTags” with a click on the left mouse button.



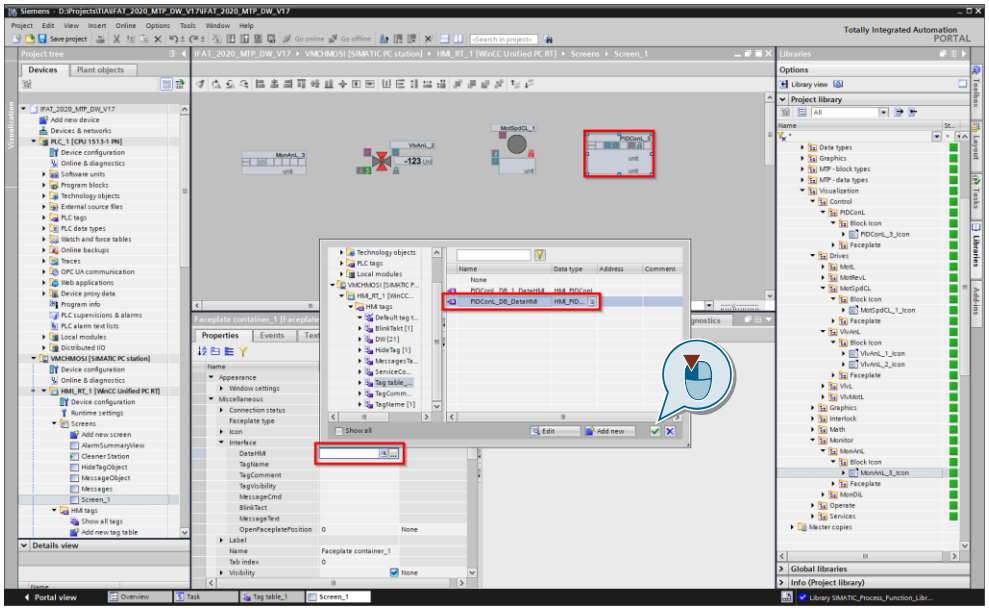
8. Move the instance data blocks from the program blocks folder via drag & drop to the tag table. With that step all tags from the PLC are configured because all tags are stored in a separate data structure.



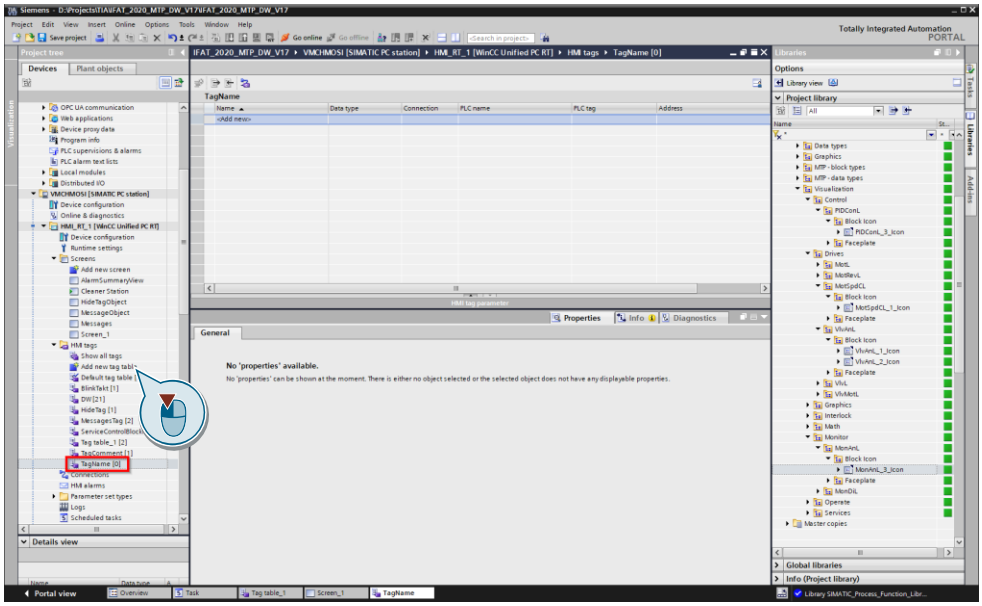
9. Place the faceplate types with "_Icon" into the "Screen" window.



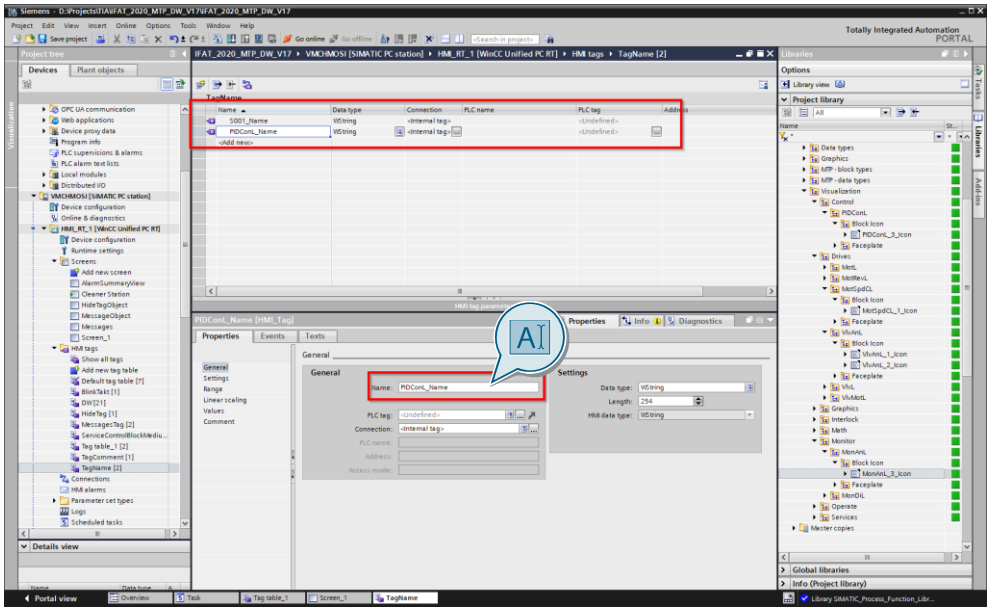
10. Select the faceplate icon instance and go to the properties. Under the interface the first connection needs to be set to the DataHMI-Tag of the cooresponding instance. In this example the faceplate icon instance is the "PIDConL". So it must be connected with "PIDConL_DB_DataHMI". Confirm that step with the green hook.



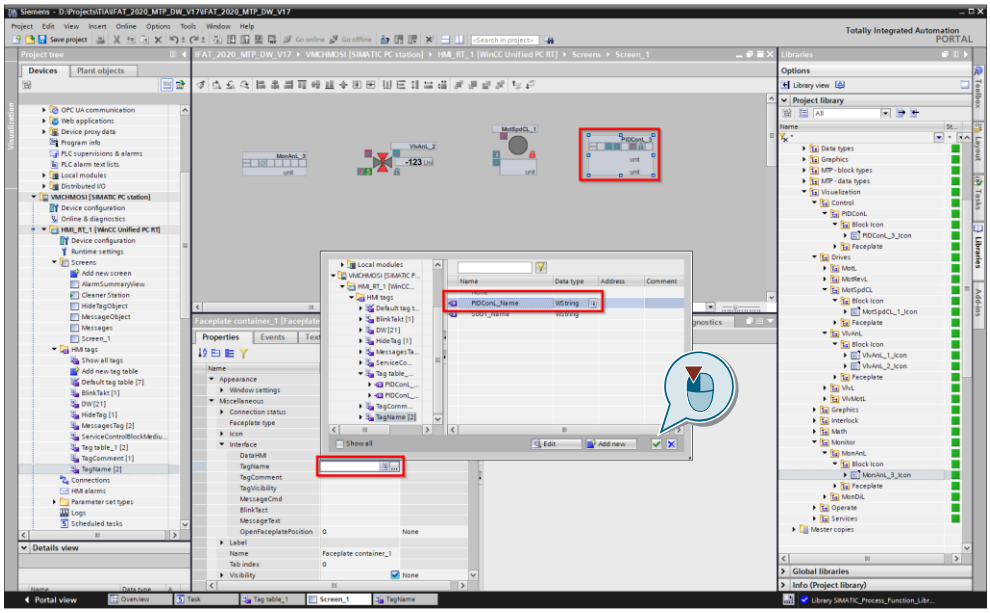
11. For the next interface tag an internal tag is required. Add a new tag table “TagNames” with a click on the left mouse button.



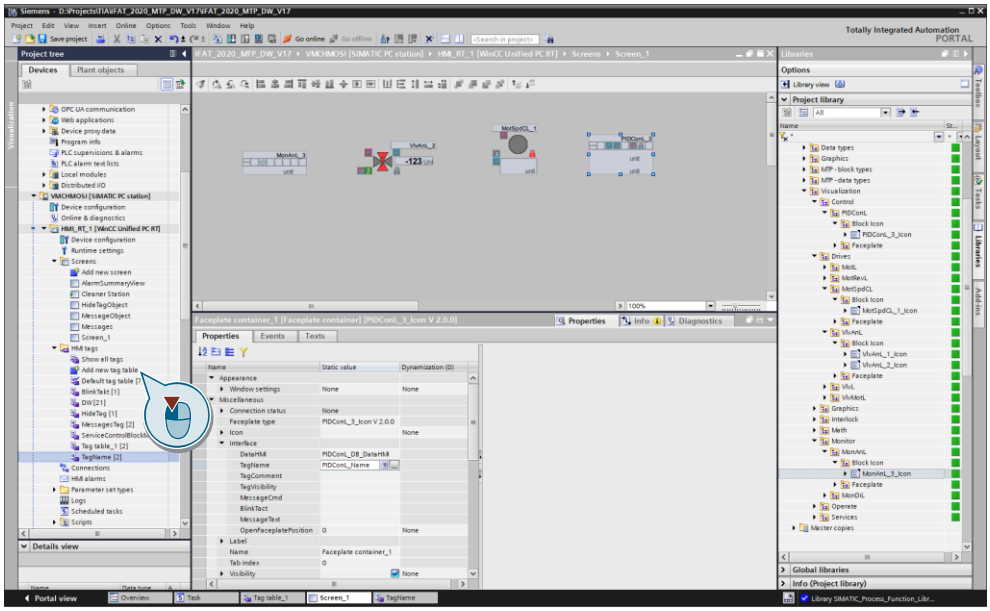
12. Create an internal tag for every instance and set a unique tag name into the start value. It is important that there is a “/” in the text string. Before the “/” there should be the PLC name where the instance is located. After the “/” it should be the same name as the instance data block in the PLC. This is very important because this name is also used for the message windows and to setup the corresponding message filter.



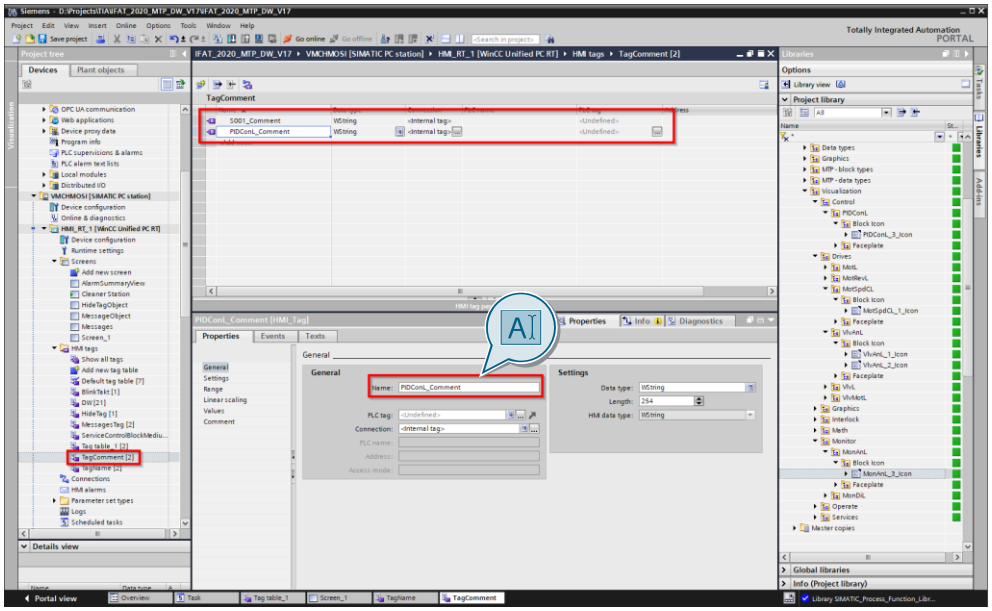
13. The “TagName” of the interface can now be connected to the internal tag. In this example it relates to “PIDConL_Name”. Confirm that step with the green hook.



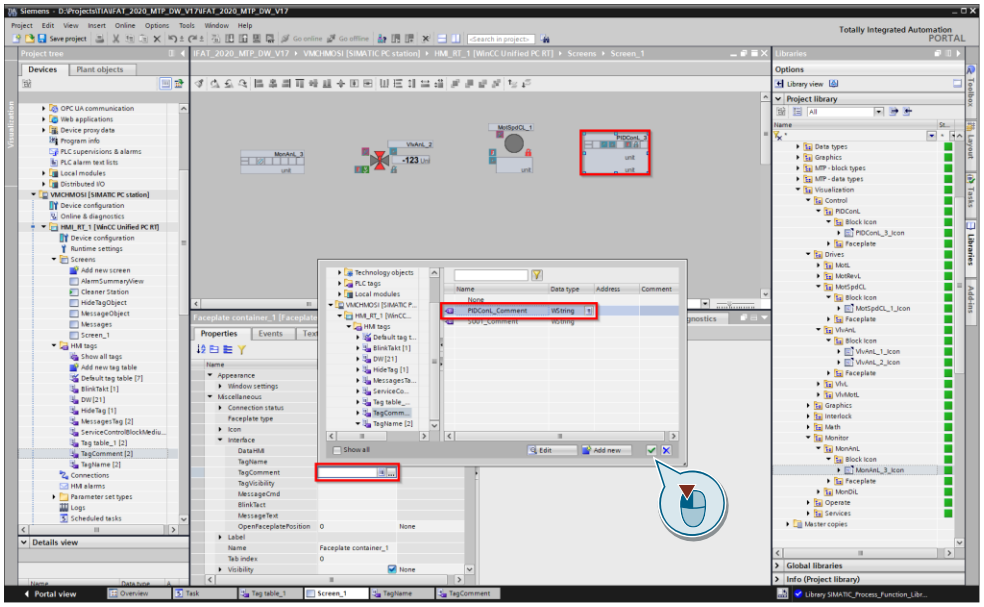
14. For the next interface tag an internal tag is required. Add a new tag table “TagComments” with a click on the left mouse button. This tag interface can also be set empty. In this case the default comment will be shown in the faceplate.



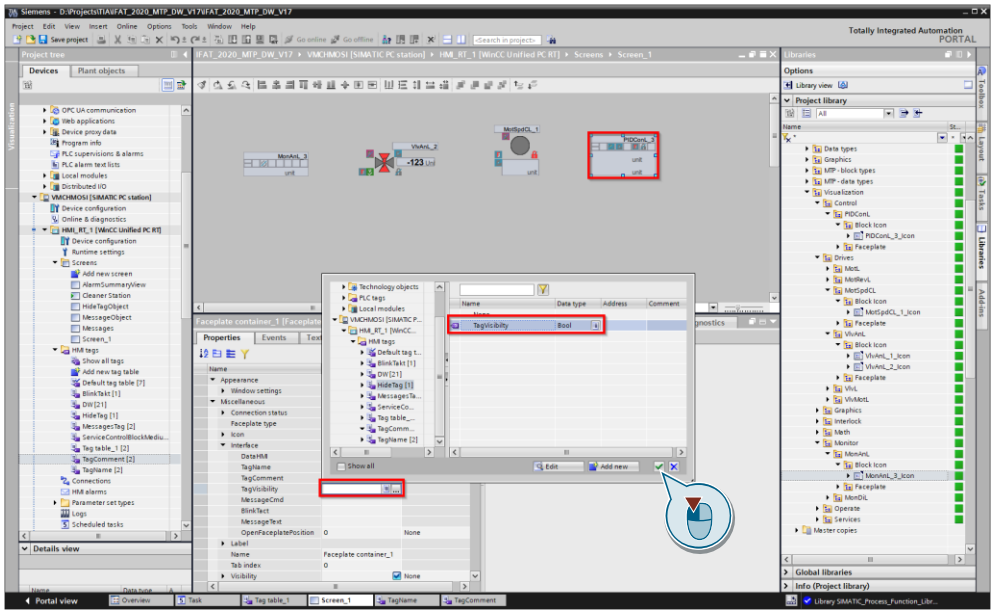
15. Create an internal tag for every instance and set a comment.



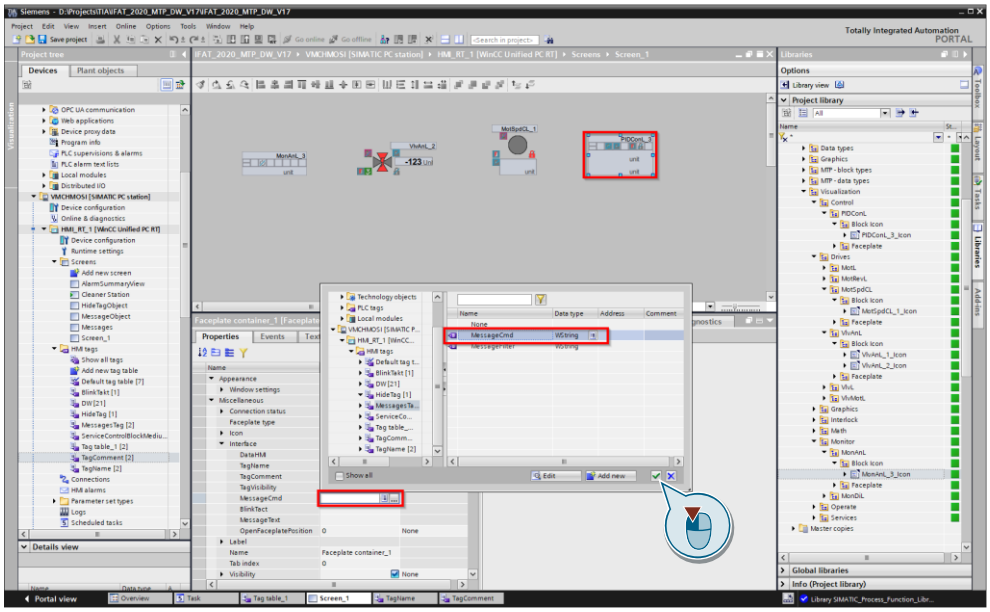
16. The “TagComment” of the interface can now be connected to the internal tag. In this example it relates to “PIDConL_Comment”. Confirm that step with the green hook.



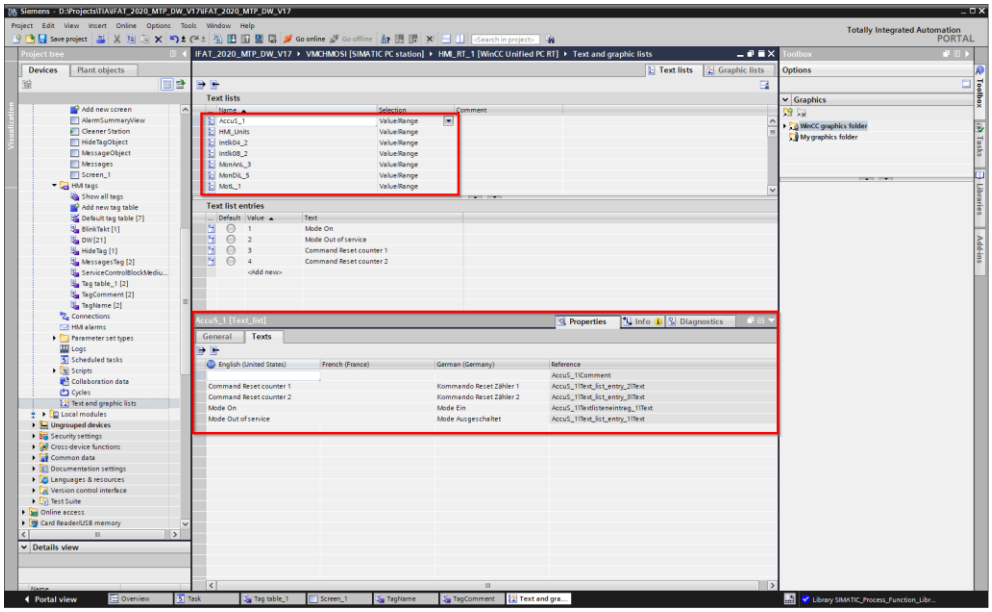
17. The interface tag “TagVisibilty” relates to the internal tag “TagVisibilty”. This internal tag should be used for all faceplate instances. This tag is used to hide & unhide the tag name of the block icon. Confirm that step with the green hook.



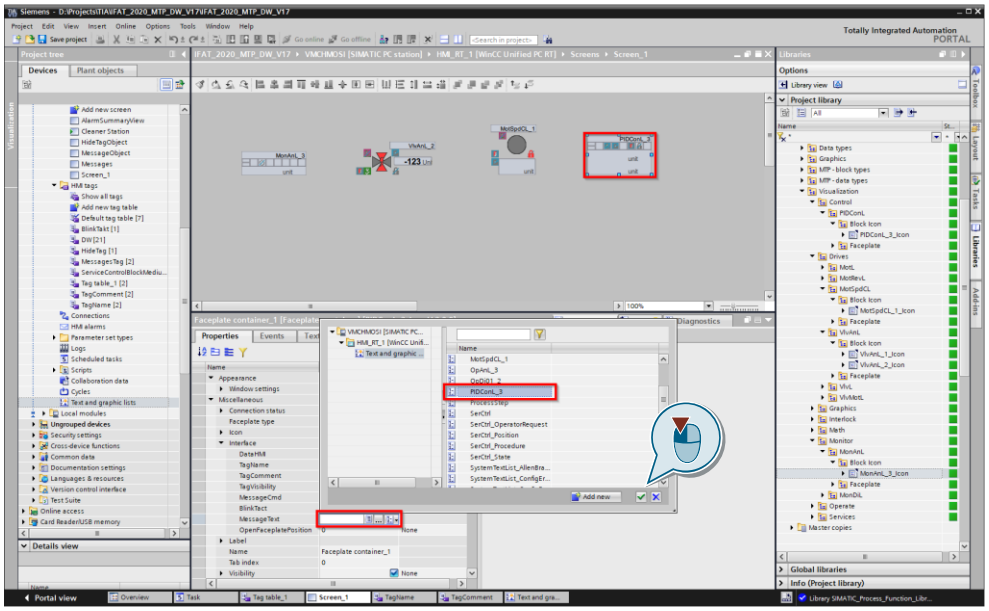
18. The interface tag “MessageCmd” relates to the internal tag “MessageCmd”. This internal tag should be used for all faceplate instances. This tag is used to handle the call of the message window and to set the necessary message filter. Confirm that step with the green hook.



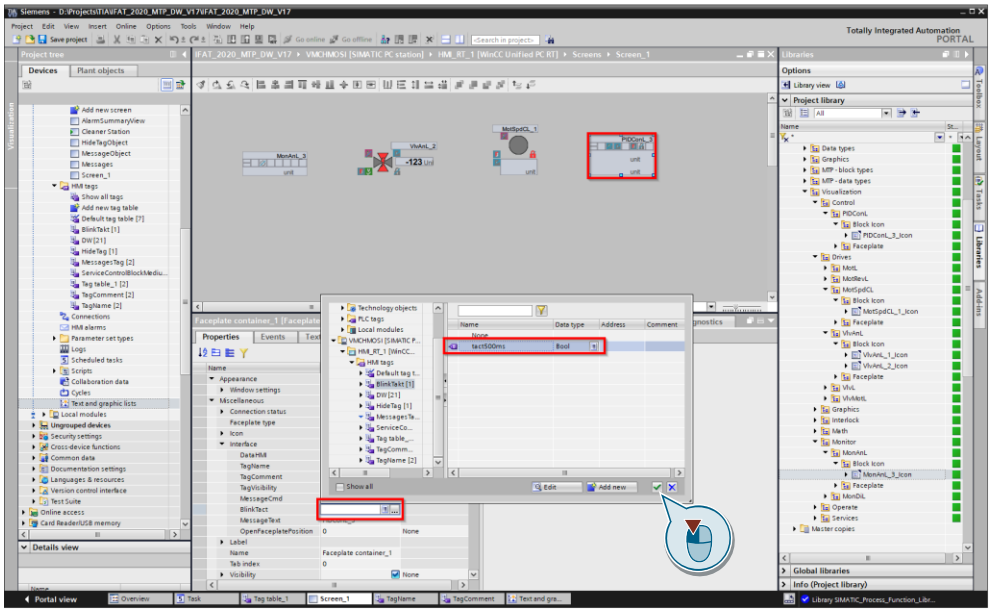
19. For every faceplate type an own list is imported. The corresponding operator messages for the faceplate type are listed. The texts are multi-language capable.



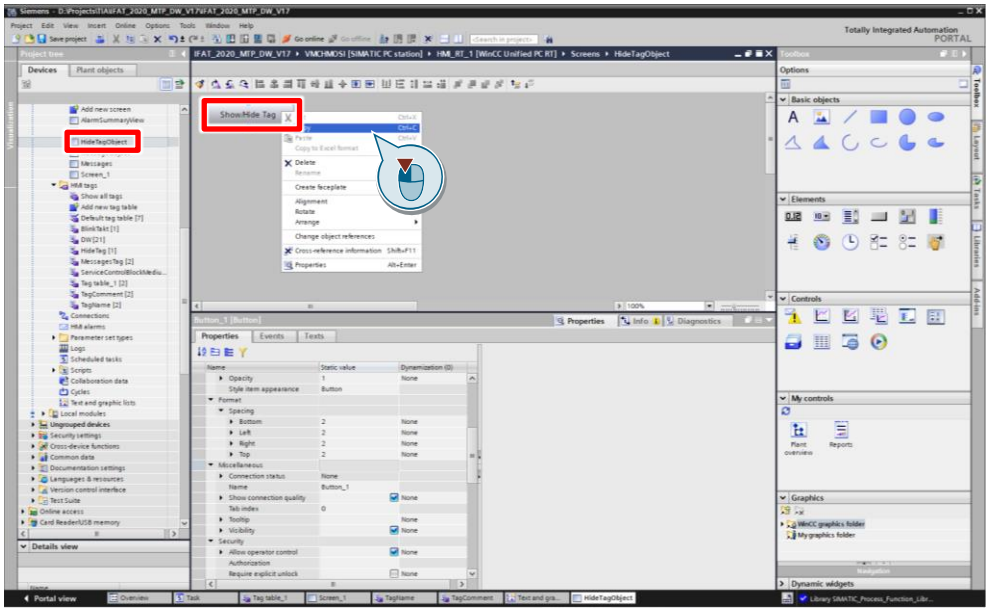
20. The interface object “MessageText” relates to the corresponding text lists. This object is used to handle the operator message from the faceplate window. In this example it relates to “PIDConL_3”. Confirm that step with the green hook.



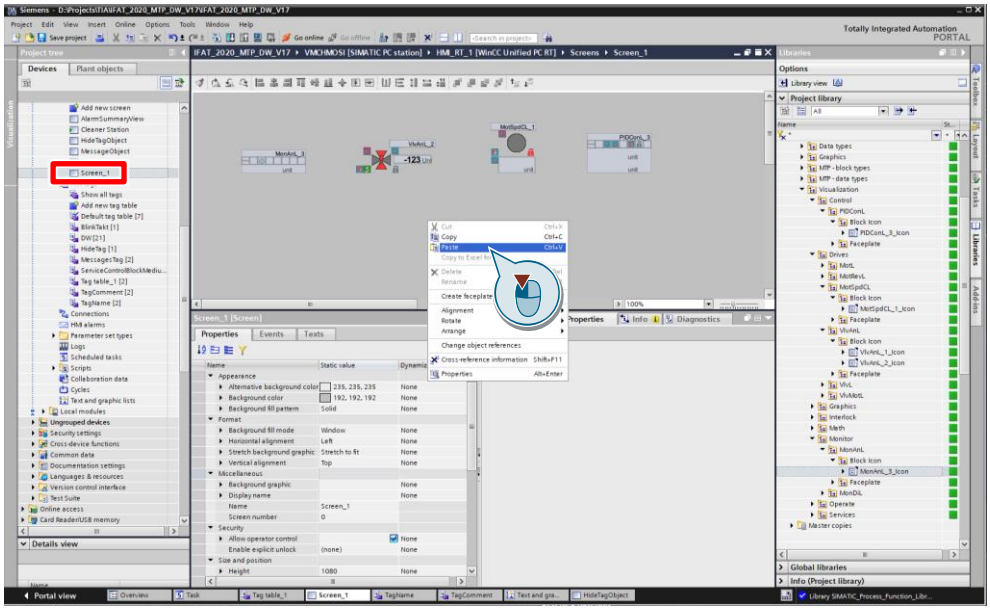
21. Some of the faceplate types have got the interface tag “BlinkTact” in addition and it relates to the internal tag “tact500ms”. This internal tag should be used for all instances. This tag is used for blinking several objects in the faceplate. Confirm that step with the green hook.



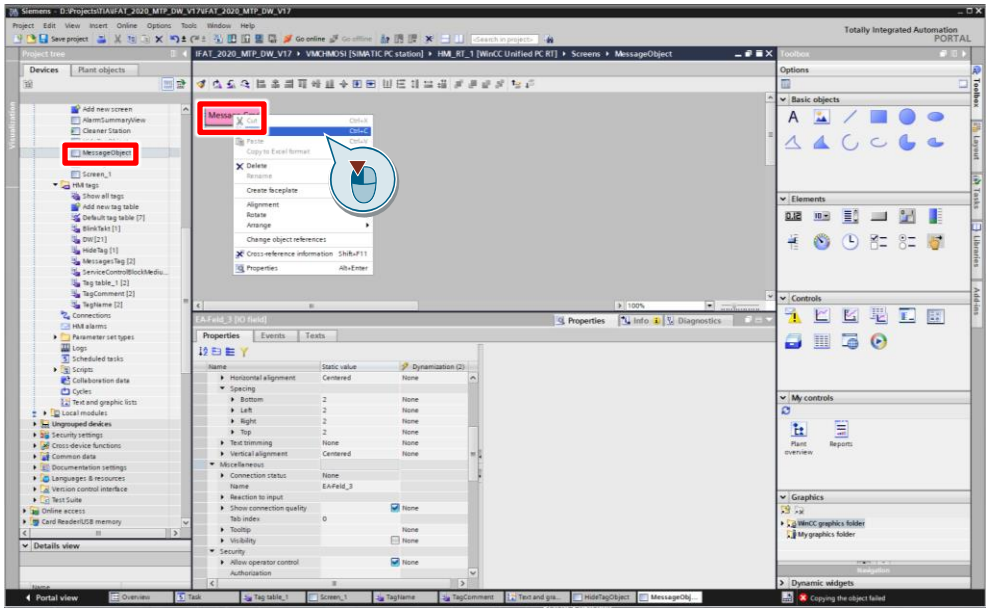
22. Open the screen window “HideTagObject” and copy the button with the text “Show/Hide Tag”.



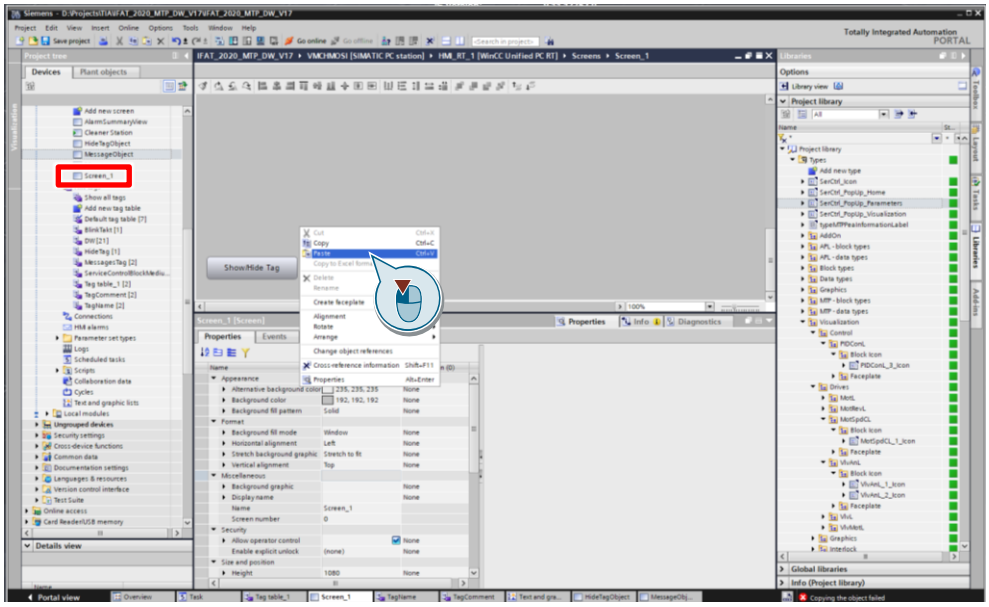
23. Paste the button with the text “Show/Hide Tag” into all screen windows. With that button the tag names of all faceplate block icon can be shown hide or unhide.



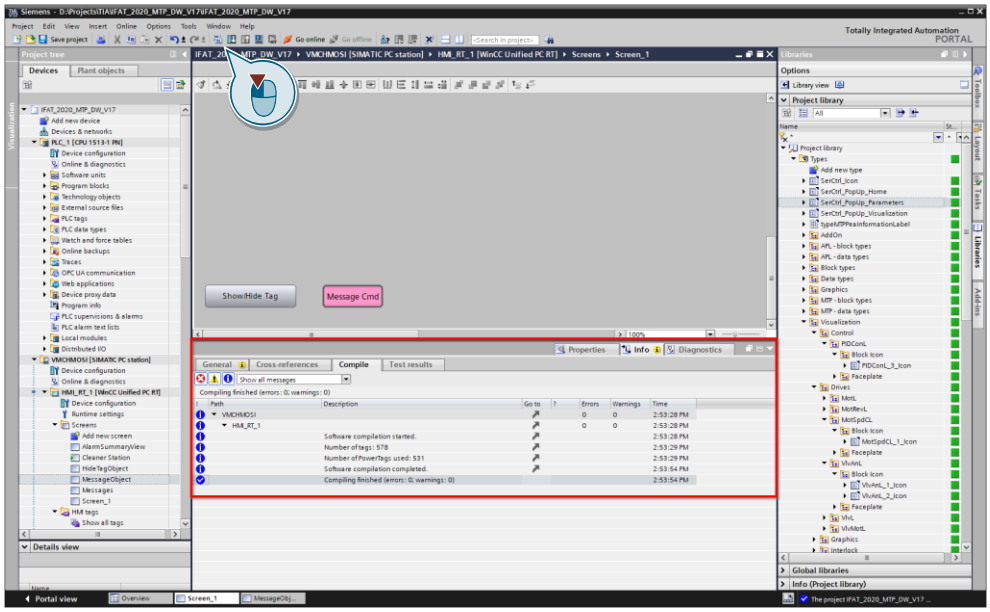
24. Open the screen window “MessagesObject” and copy the text field with the text “MessageCmd”.



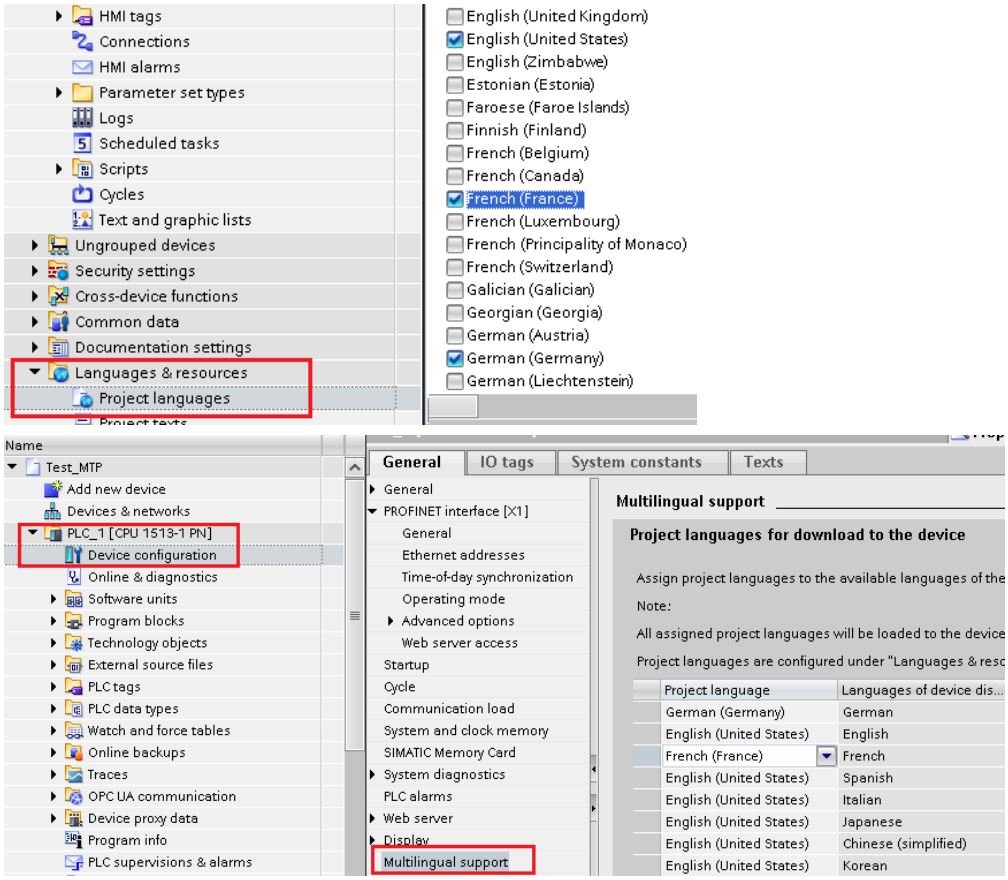
25. Paste the text field with the text “MessageCmd” into all screen windows. With that text field the message window is called, and the setup of the corresponding message filter will be done.



26. As the last step the HMI_RT of PC-System [SIMATIC PC station] needs to be compiled.



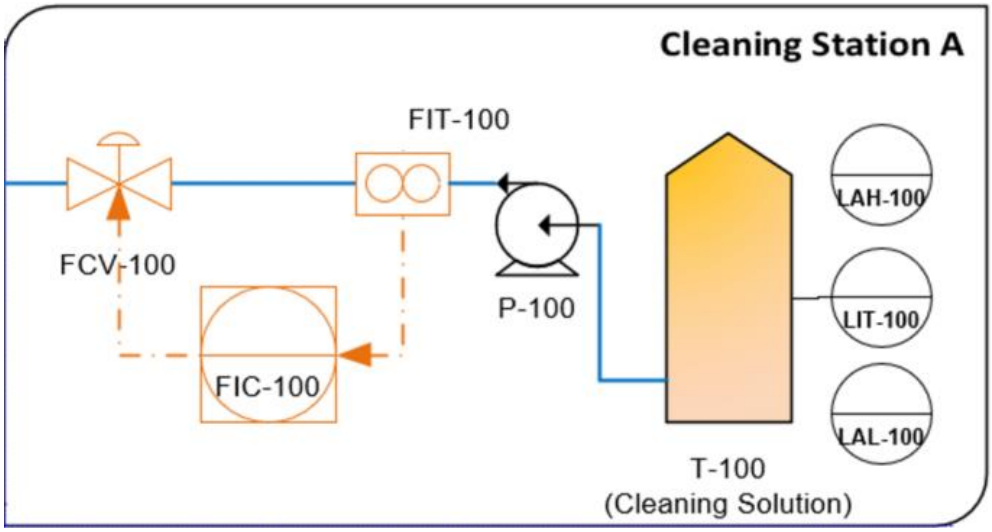
27. Setup Project languages



3 TIA Portal V17 Application Example with SIMATIC Process Function Library

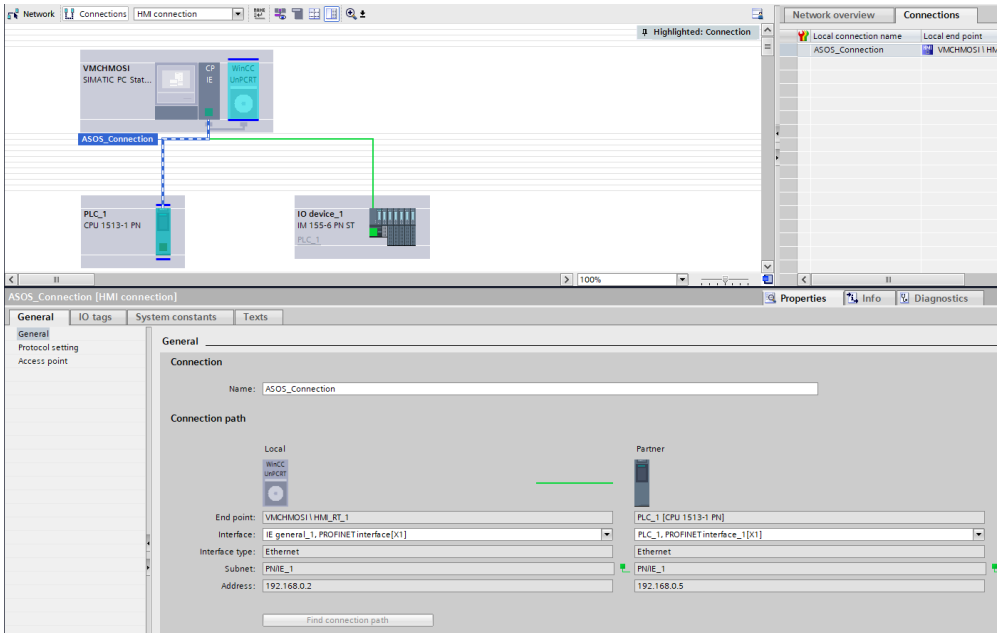
3.1 Project Setup

In this project a Cleaner Station for a plant is configured. The process is simulated in the PLC, so SIMIT is not needed. The process is controlled by MTP services, and it pump Water from a tank to a reactor for cleaning purpose. It is also possible to refill the tank. Which procedures are available and how this works is described in chapter 4 Services in the HMI. For the process there are different components in use. There is a tank with level measurements, a digital pump, a flow indicator, which is connected to an PID controller, which controls the valve at the end of the process.



3.1.1 Hardware Configuration

The hardware configuration is built up with an 1513-1 CPU and a ET200-SP. For the HMI runtime WinCC Unified V17 is in use.



3.1.2 Software

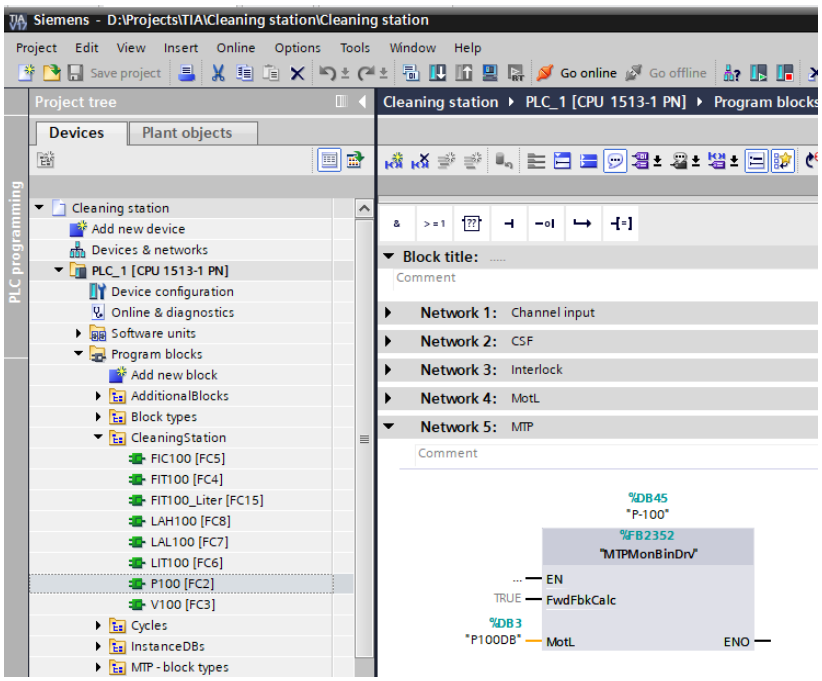
3.1.2.1 Program blocks

In total there are eight main program blocks for the functionality. They are located in the folder “CleanerStation” For each process object is one function block. All of them are programmed in FBD.

In the functions are always different networks for the different usages. The first network is always used for interlocks or other functions for the main block. In the second one you can find the main block like the “MotL” for the pump. And in the last network is always the MTP block, which is connected to the SIMATIC Process Function Library block. This MTP block is used as the interface out of the project to another DCS. All networks are named after their usage.

Table 3-1

	Blockname	Usage
1.	FIC100	PID control
2.	FIT100	Flow measure
3.	FIT100_Liters	Pumped liters
4.	LAH100	High Level
5.	LAL100	Low Level
6.	LIT100	Level measure
7.	P100	Pump
8.	V100	Valve

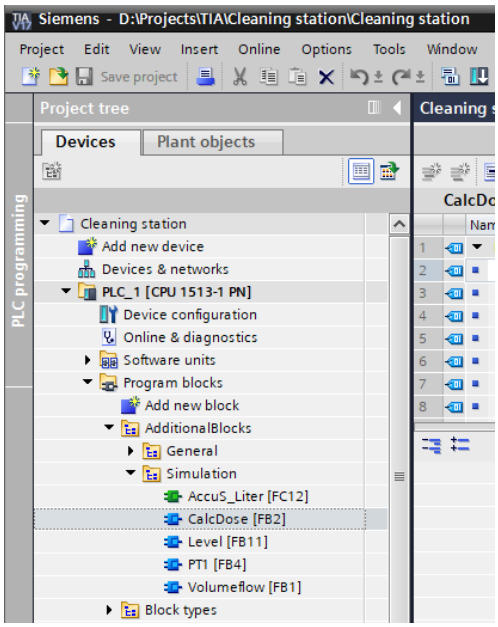


3.1.2.2 Simulation

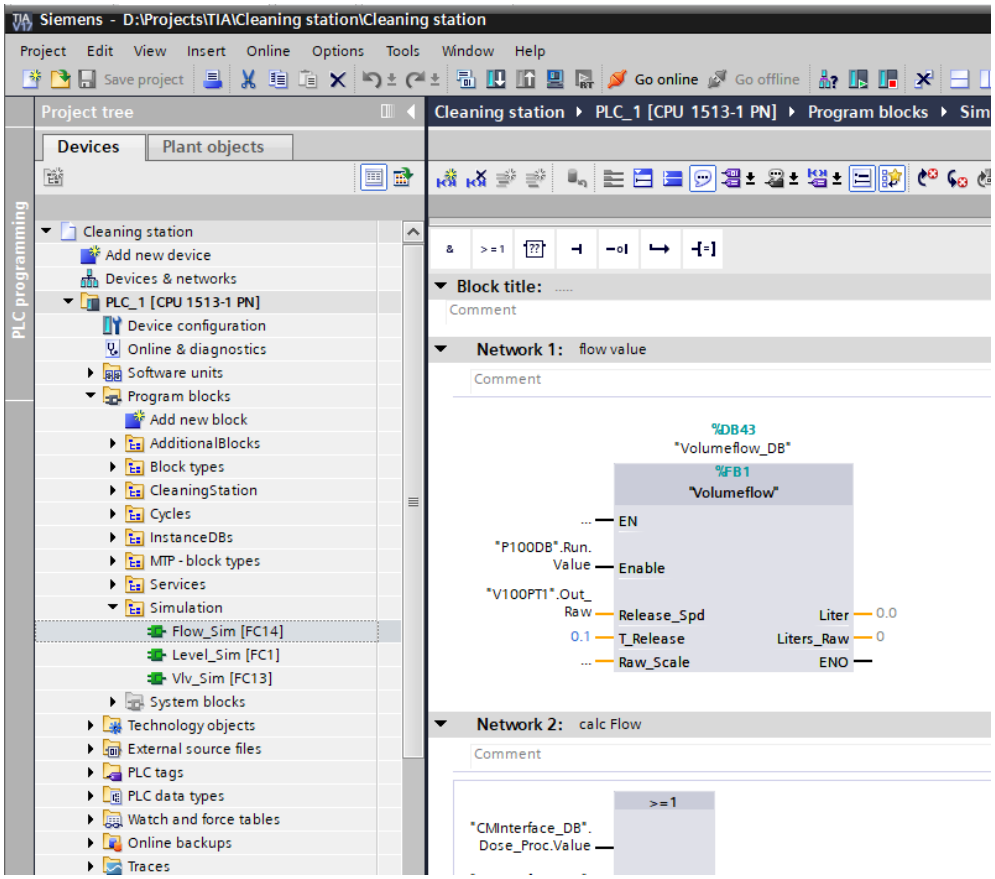
For the simulation there are four function blocks. These blocks are written in SCL.

Table 3-2

	Function block	Usage
1.	CalcDose	Calculates the pumped water in liter
2.	Level	Calculates the fill level of the tank
3.	PT1	Opens the valve with PT1 behavior
4.	Volumeflow	Calculates the volume flow in l/s



These blocks have their call in the corresponding simulation function in the folder “Simulation”. In the first function “Flow_Sim” everything for the simulating the flow is located. For example, the function blocks “Volumeflow” and “CalcDose” have their call there. The same is for the function “Level_Sim” and “Vlv_Sim”. In the “Level_Sim” the function block “Level” and the refill function are located. In the function “Vlv_Sim” is the call of the PT1 controller for the opening and closing behavior of the valve “V100”.



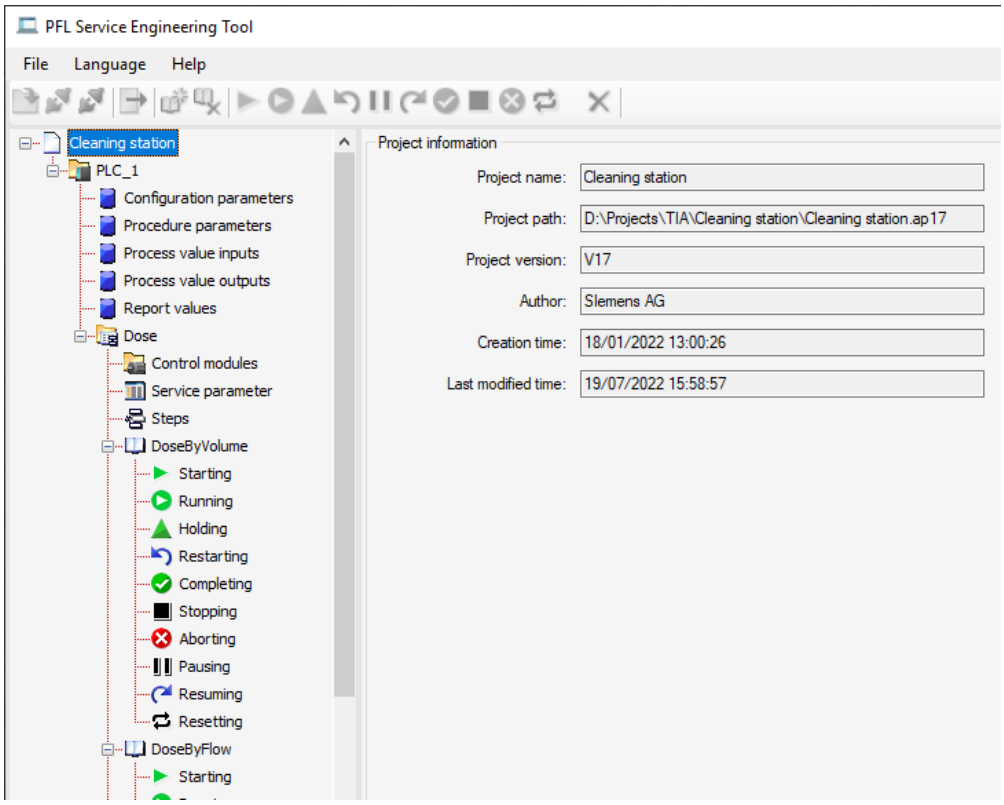
3.1.2.3 MTP Services

The Services are configured in two different tools, in the TIA Portal and in the Service Engineering Tool. For general information about the Service Engineering Tool a manual is delivered with the installation package of the SIMATIC Process Function Library. It is in the following folder:

C:\Program Files\Siemens\Automation\SIMATIC Process Function Library\V2.0.1.0\Manuals\en-US

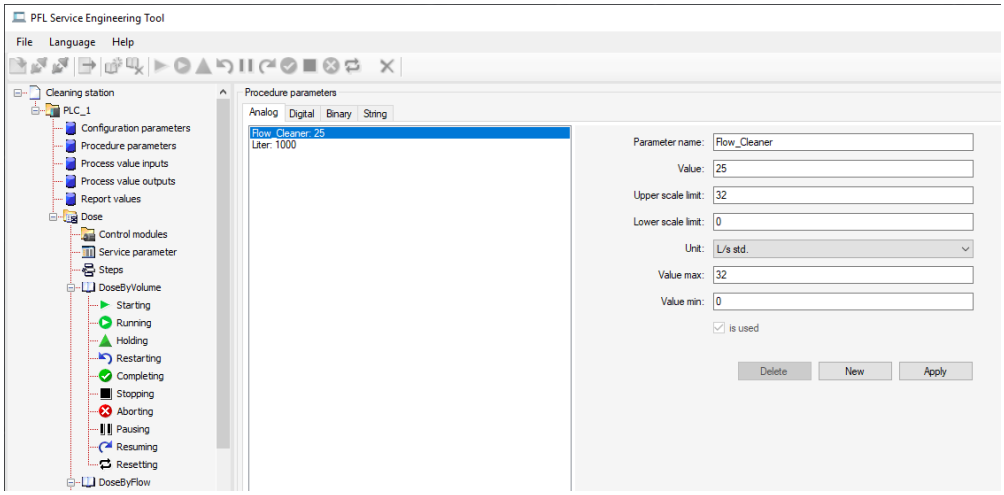
In the chapter 12 Services all parameters and settings/configurations of the tool are explained. In this document is only a general overview over the tool and the explanation for the project specific configuration.

In the Service Engineering Tool, the general structure of the services is configured. There you can define the global service parameters, the control modules, the connection of the global parameter to the procedures, the steps, and the procedures itself.

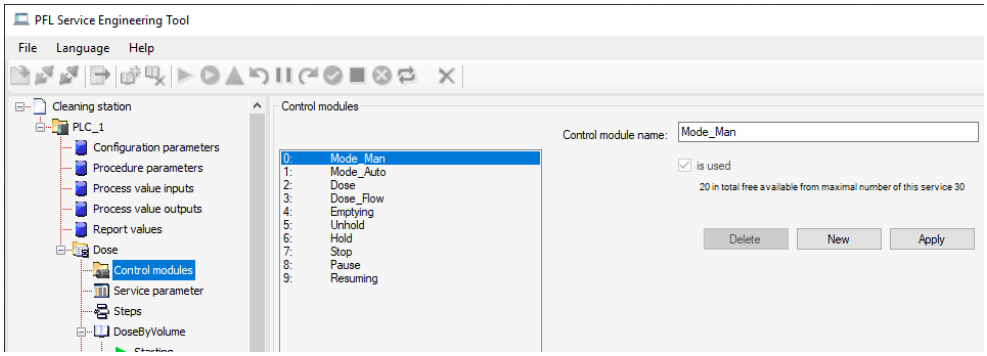


For this project there are only two global parameters needed. Both are configured in the “Procedure parameter” folder. The first one is “Flow_Cleaner”. This parameter defines the volume flow for the service (Used in the DoseByFlow service). The second parameter is “Liter”, this one defines the amount of liter, which should get pumped out of the tank.

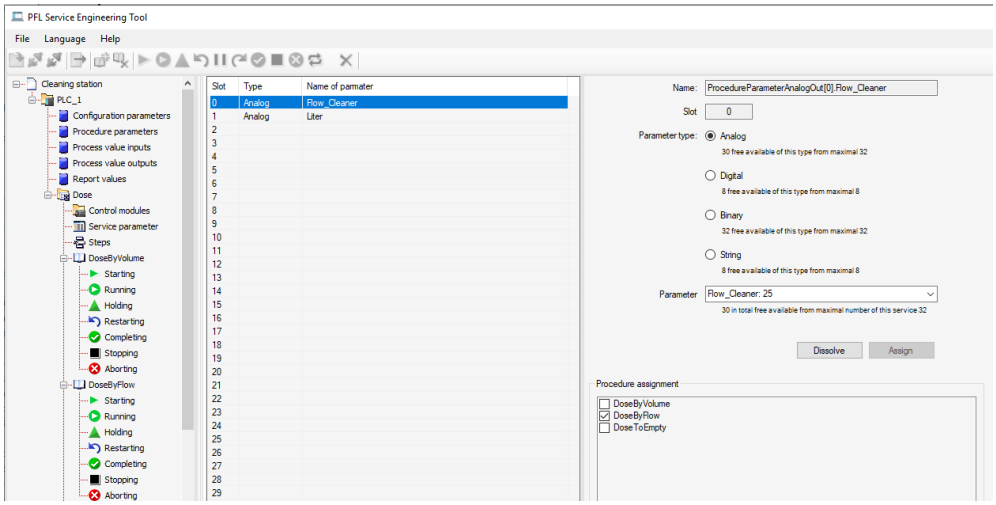
The configuration values for these parameters are also defined here. After the export to the TIA Portal, they get stored in the corresponding DB for the global service parameters. They can only be changed in the Service Engineering Tool and needs to be exported to the TIA Portal afterwards. For example, the default value or the max and min values/limits. For the “Flow_Cleaner” there is the default value “25 l/s”, the high limit “32 l/s” and the low limit “0 l/s”. For “liters” the default value is “1000 l”, the limits are “3200 l” (high) and “0 l” (low). The high limit/max value is defined by the maximum fill level of the tank, and this is 3200l.



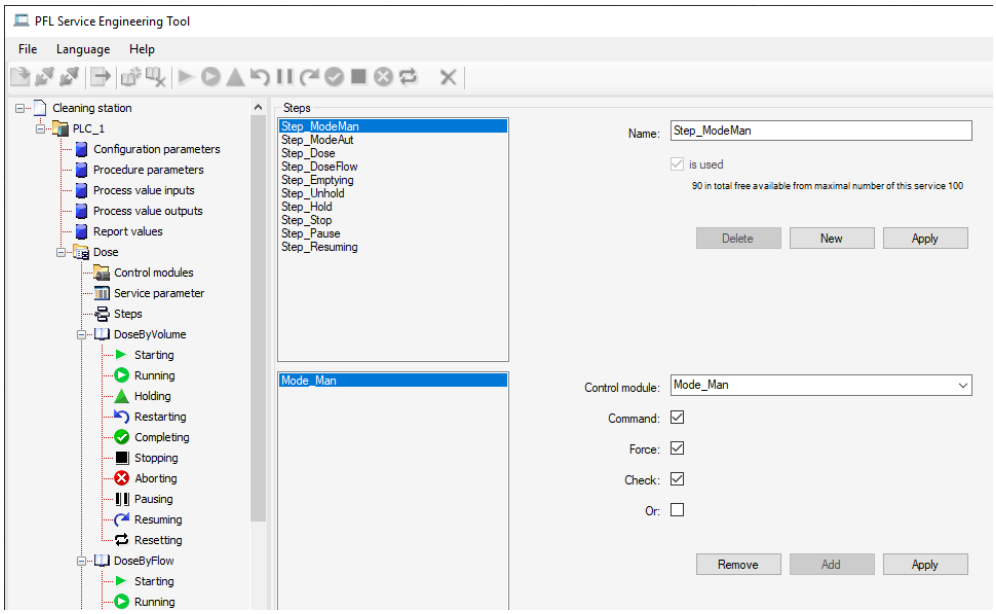
For the process are seven different control modules available. These control modules are bits in the ServiceControlBlockMedium, which are stored in the “CmCommand” array. With these bits it is possible to activate the different states and conditions of the objects and to define the process. The first one is responsible to set all components in the manual mode. This is needed when the service is done. The second one sets all components in the automatic mode. Before each service can be started, the components need to be in automatic mode. The third, the fourth and the fifths control module are for the procedures, so the main function of each service. The fifths control module “Unhold” is used to define the operation during the state “Unholding” of the service. The same is for the control module “Hold”, “Stop”, “Pause” and “Resuming”.



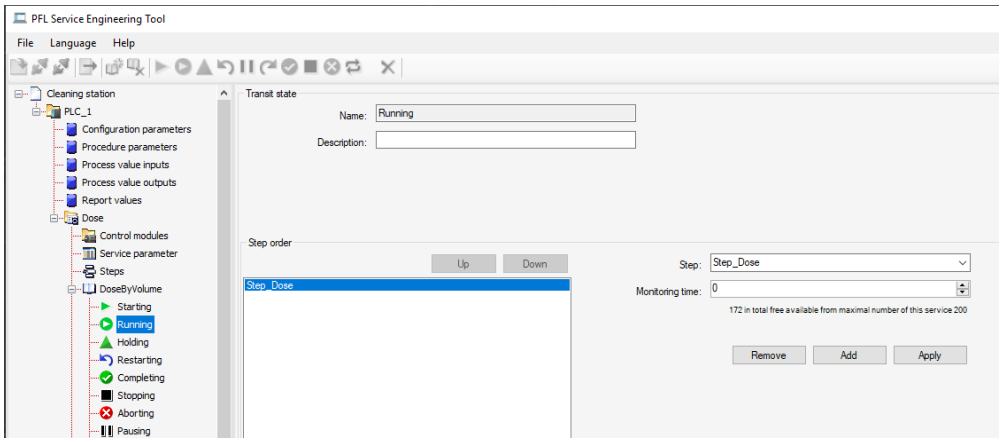
In the folder Service parameter, the global parameters are connected to the procedures of the project. The “Flow_Cleaner” is connected to the “DoseByFlow” procedure and the “Liter” parameter is connected to “DoseByVolume” and to “DoseByFlow”.



In the next folder “steps” is the configuration of the steps, which gets executed in the different states of the procedure. In the project is one step for each control module. So, there are seven steps. The step “Step_ModeMan”, “Step_ModeAut”, “Step_Dose”, “Step_DoseFlow”, “Step_Emptying”, “Step_Unhold”, “Step_Hold”, “Step_Stop”, “Step_Pause” and “Step_Resuming”. Each step is connected to its corresponding control module and the options for “Command”, “Force” and “Check” are activated for all.



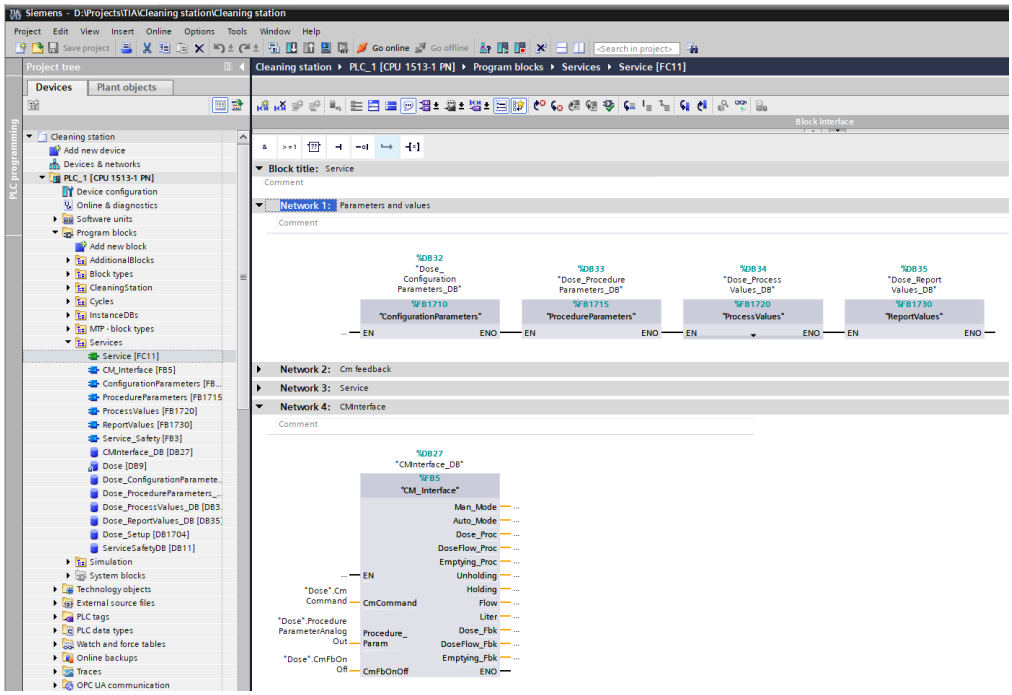
At the end is the configuration of the procedures. In this project are three of them. The “DoseByVolume”, “DoseByFlow” and “DoseToEmpty” procedure. In each procedure are the states “Starting”, “Running”, “Completing”, “Stopping” and “Aborting” configured. The inserted steps in these states are almost the same for all procedures. In the “Starting” state is always the step “Step_ModeAut”, in the three states “Completing”, “Aborting” and “Resetting” is the step “Step_ModeMan” configured. In the “Restarting” state is the step “Step_Unholding”, in the state “Holding” is the step “Step_Holding”, in “Stopping” is the step “Step_Stop”, in “Pausing” is the step “Step_Pause” and in the state “Resuming is the step “Step_Resume” configured. In the “Running” state is in all three procedure the specific step for the procedure inserted. For example, the step “Step_Dose” is in the “Running” state of the “DoseByVolume” procedure.



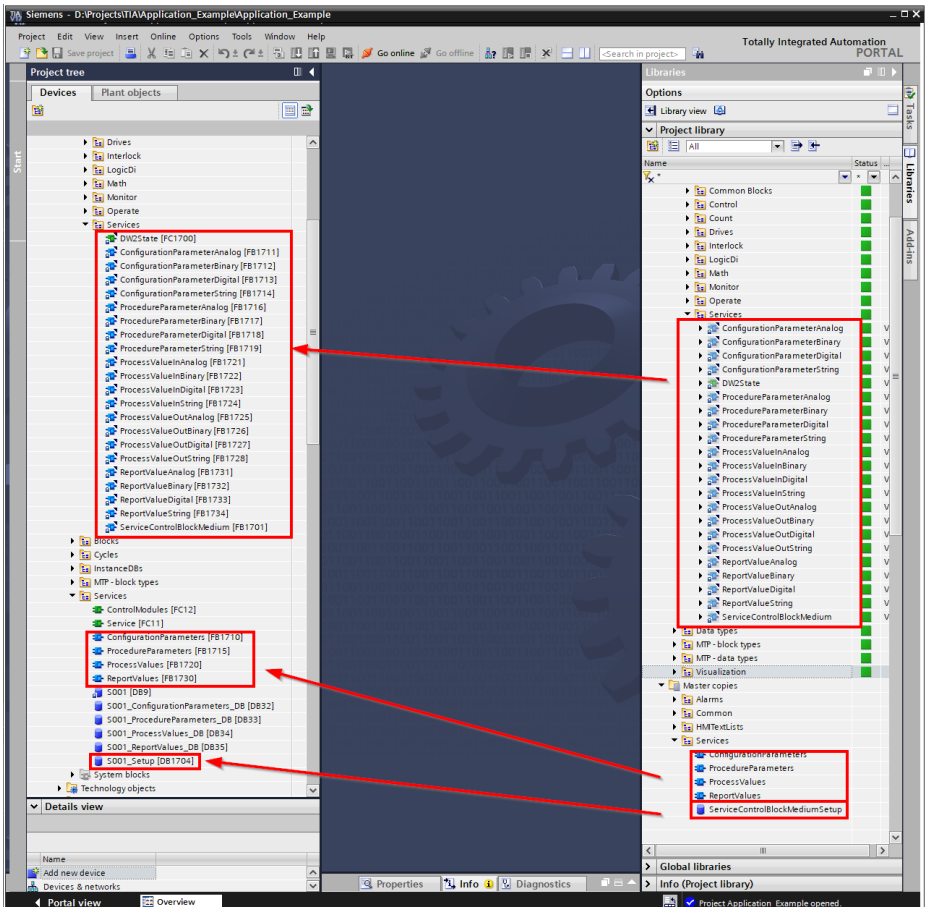
In the TIA Portal you define, what is to do, when the different control modules are activated and you need to define all DBs, where the information for the configuration in the Service Engineering Tool is stored.

In the project are two main functions, in which almost the whole implementation for the MTP services is located. It is the “CM_Interface” function block and the “Service” function. In the “Service” function all needed blocks and functions have their call, like the function blocks for the parameters and values, the data block for the services, the control module interface function, or the safety function block for

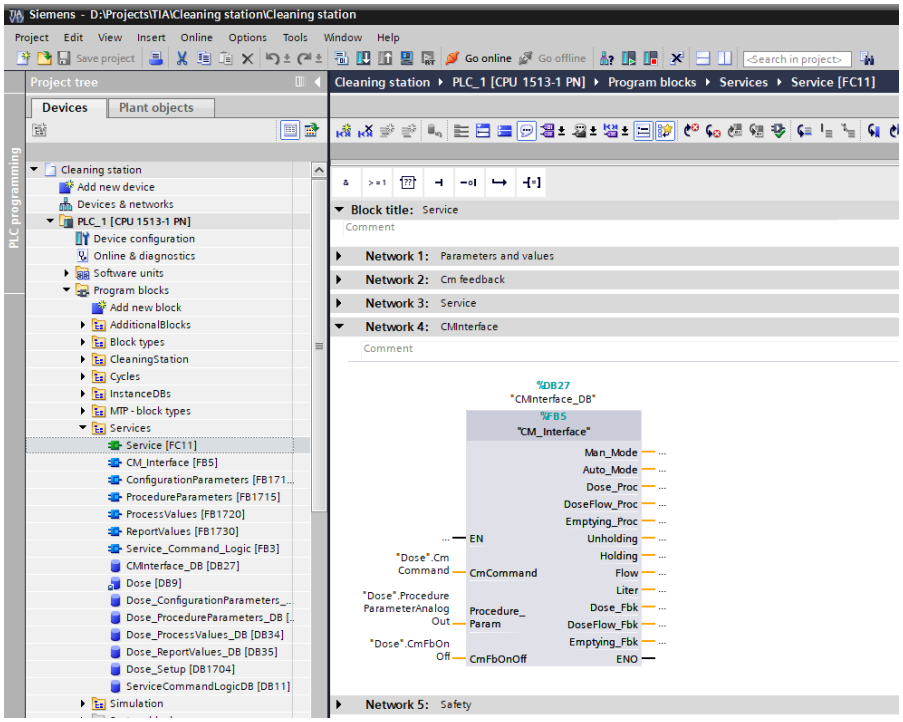
the services. The blocks for the parameters and values are used to store the information of the global and local parameters, which are configured in the Service Engineering Tool.



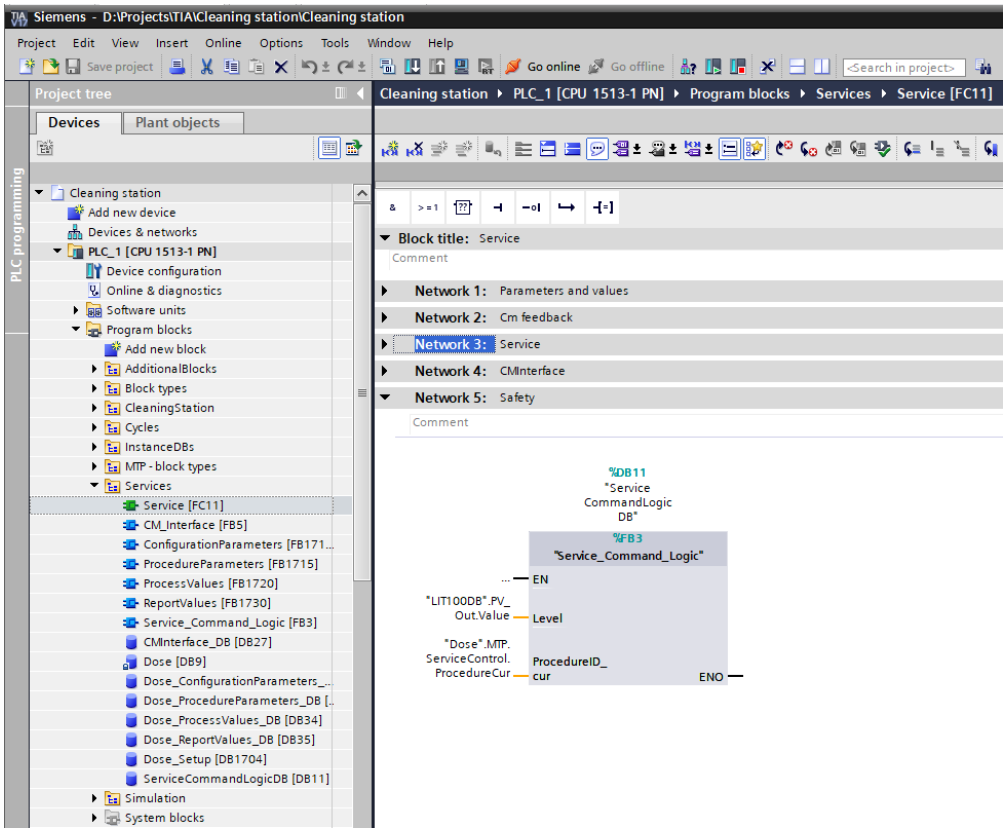
These blocks, like “ConfigurationParameters” or “ProcedureParameters” are copied from the SIMATIC Process Function Library. It is the same for the “S001_Setup”, this one is also copied from the SIMATIC Process Function Library and renamed afterwards. What other blocks are copied from the SIMATIC Process Function Library can be seen in the following picture.



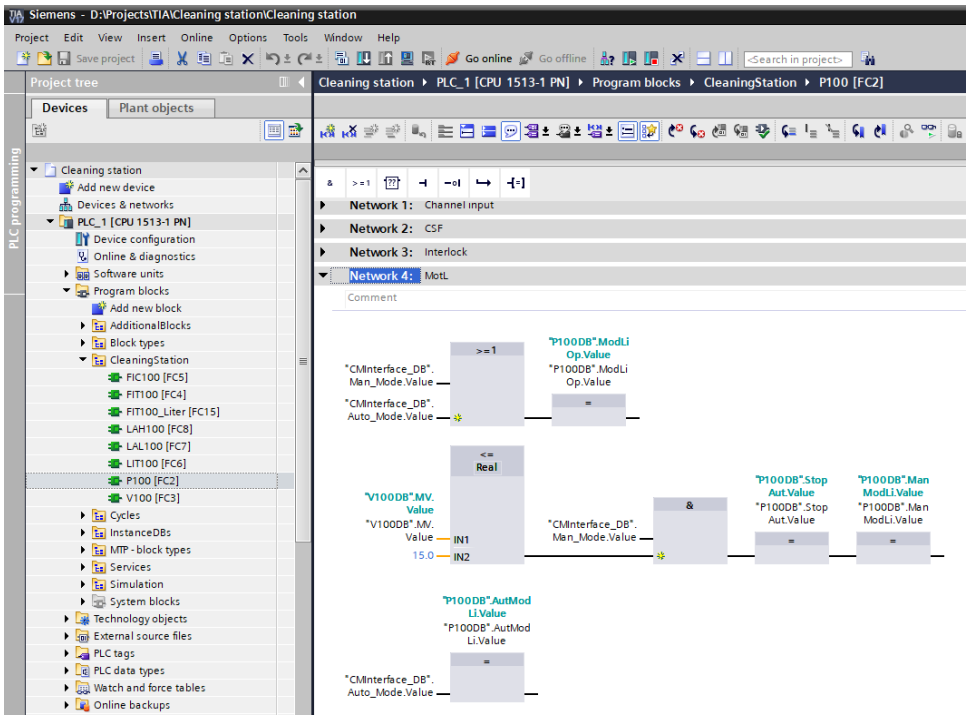
In the “CM_Interface” function block is the conversion of the array bits to its intended purpose. The block converts the bool and real values to the datatypes “AnaVal” and “DigVal”. So, the outputs are named like their purpose, e.g. the bit, which is intended to switch to manual mode, is called “Man_Mode”.



The ” Service_Command_Logic” function block is responsible for the behavior of the service if the level of the tanks falls under 0.5 meter. Then the service goes to the state “hold”. Only if the level of the tank is over the limit of 1.0 meter it is possible to set the service via “unhold” in the execution state. This mechanism is activated for the service “DoseByVolume” and “DoseByFlow”. It is not needed for “DoseToEmpty”, because there we want to empty the tank up to 0.5m. There is also the implementation for the reaction of the failure of the pump implemented. If the pump failure gets activated in the runtime, then the service goes to abort.



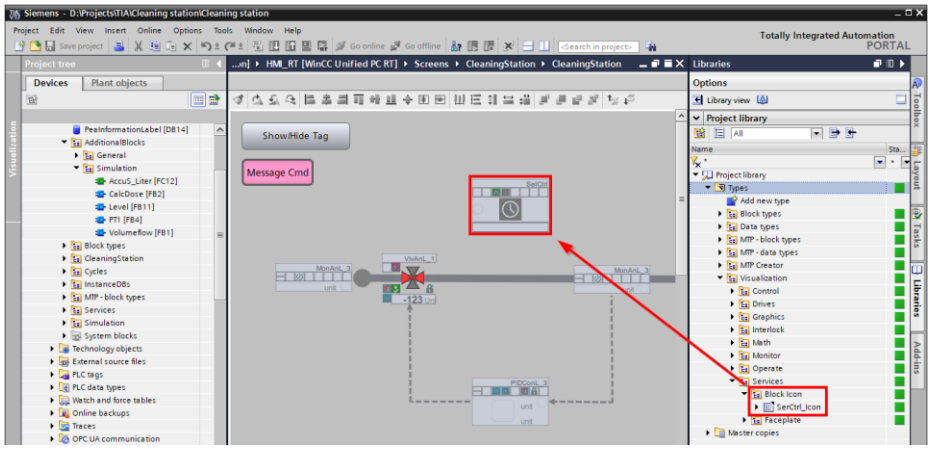
The implementation of this mode switching, and the real implementation is to find in the specific function for the different component. In the picture below you can see the code for the pump. If the step “Step_ModeAut” is activated, the pump should go in automatic mode. For the other components, like the PID controller, the Valve, etc. it looks similar. For each step from the Service Engineering Tool is an implementation for the components.



3.1.2.4 HMI Runtime

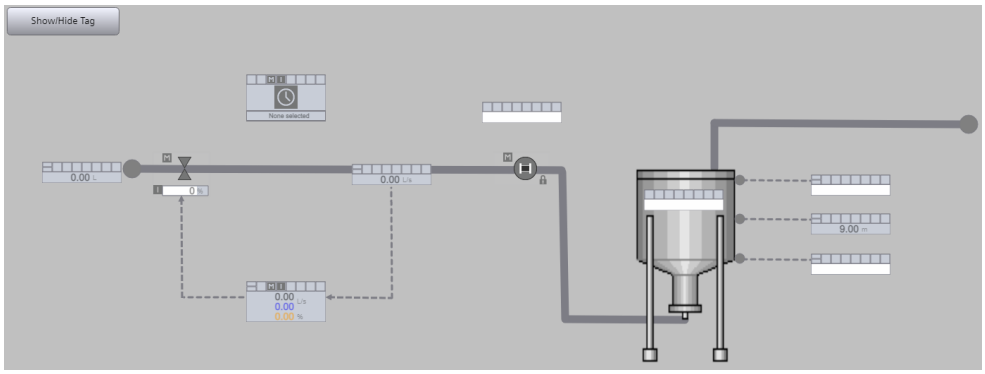
In the HMI Runtime you have following structure. On the right side is the tank with its level indicators and on the left are the valve, the pump, the PID controller and the flow indicator. In the tank is the operating faceplate for refilling the tank. And on the left side on top, with the orange area with the text “idle”, is the service control faceplate.

This faceplate is also copied from the SIMATIC Process Function Library like the blocks in chapter 3.1.2.3 MTP Services.

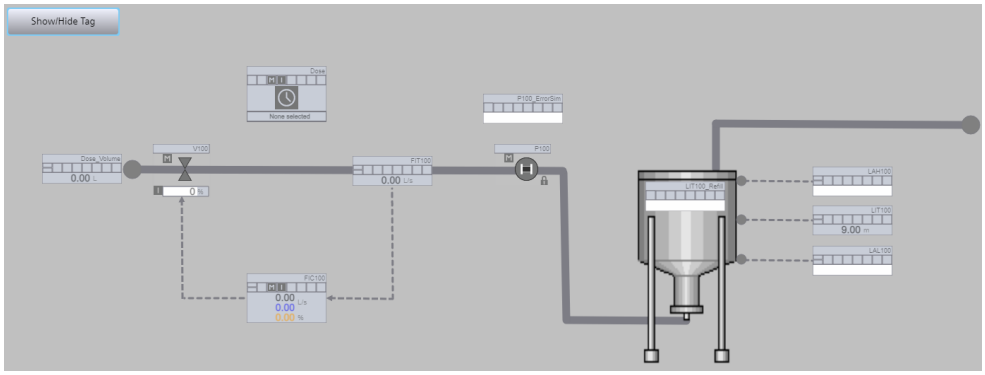


With the button “Show/Hide Tag” it is possible to show or to hide the tags for all components.

“Hide Tags”

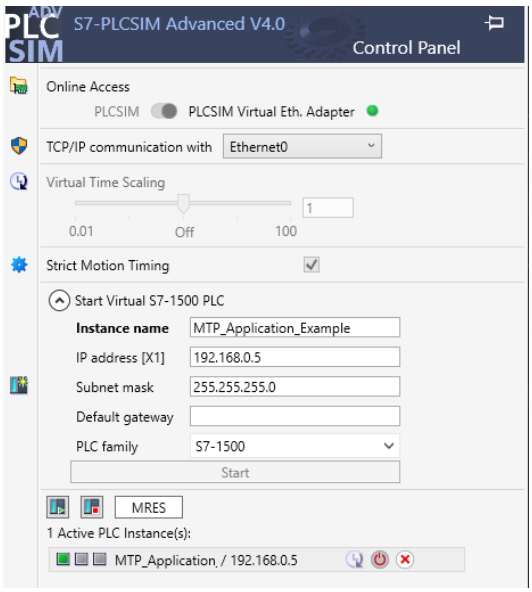


“Show Tags”

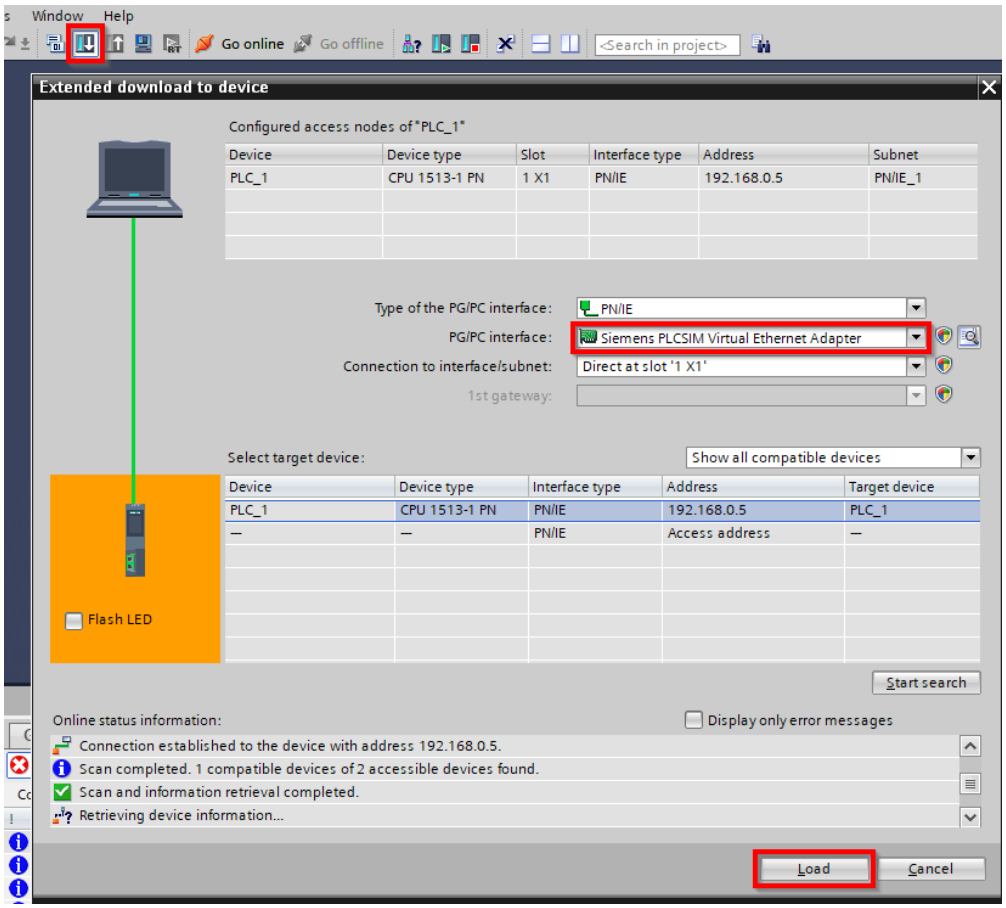


3.1.3 Start PLCSIM ADV

Start PLCSIM Adv with PLCSIM Virtual Eth. Adapter and add there a PLC S7-1500 with corresponding IP Address of the controller. Then start the controller.

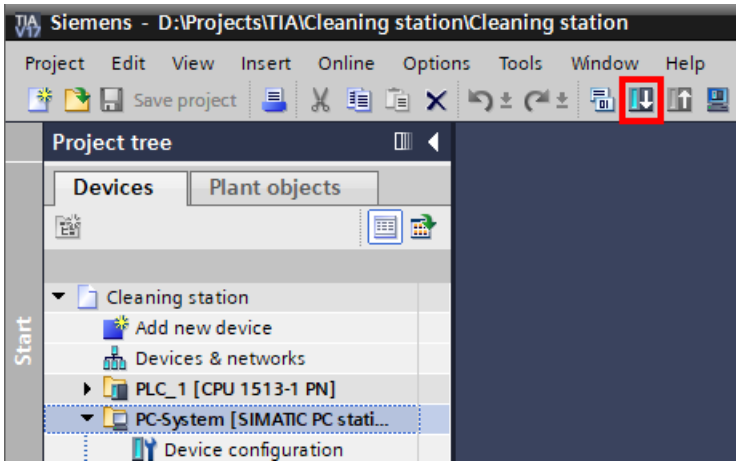


Afterwards download all data via virtual Ethernet Adapter from AS.

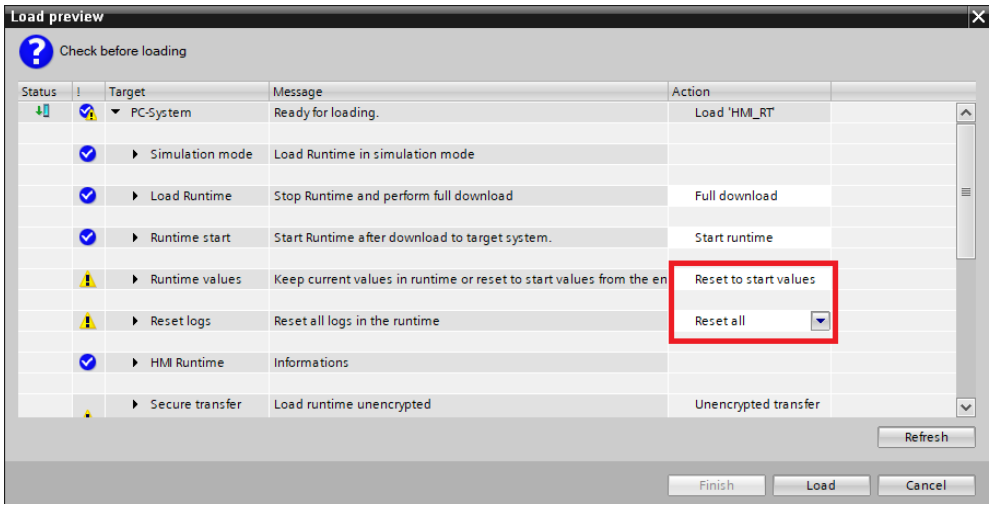


3.2 Panel Simulation

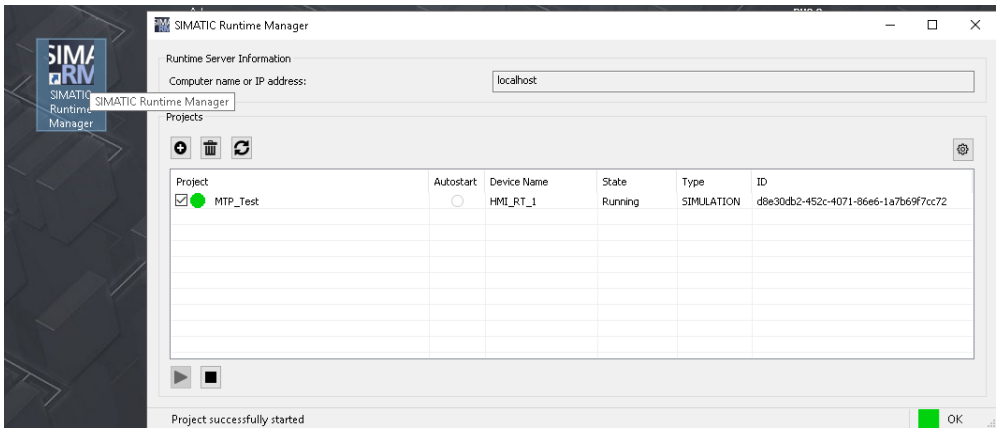
3.2.1 Download Panel to the Runtime Server



Select during the download following settings to delete possible legacy settings from previous projects.

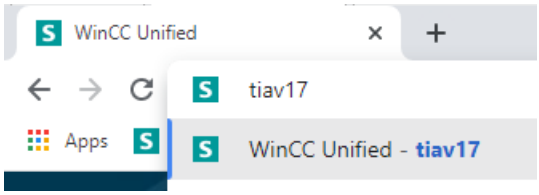


3.2.2 Start your downloaded Project on Runtime Server

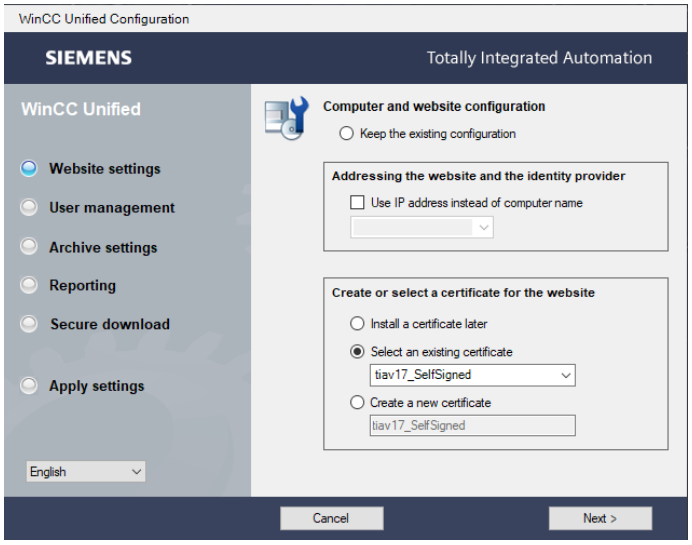


3.2.3 Open google Chrome and connect to the runtime Server

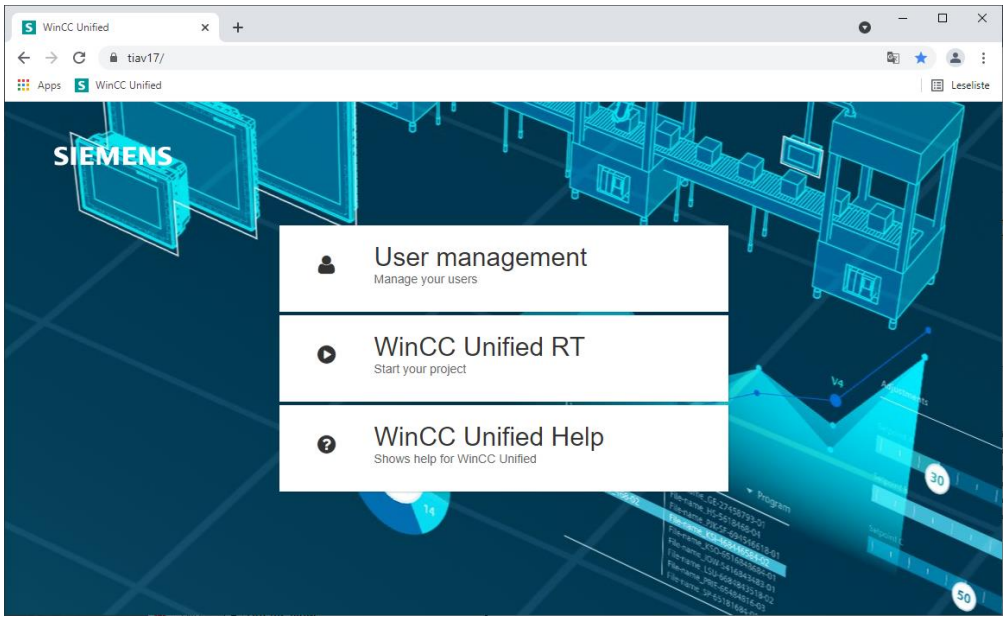
To connect to the runtime Server, you need to enter the computer name of your machine as URL. In this project it is “tiav17”.



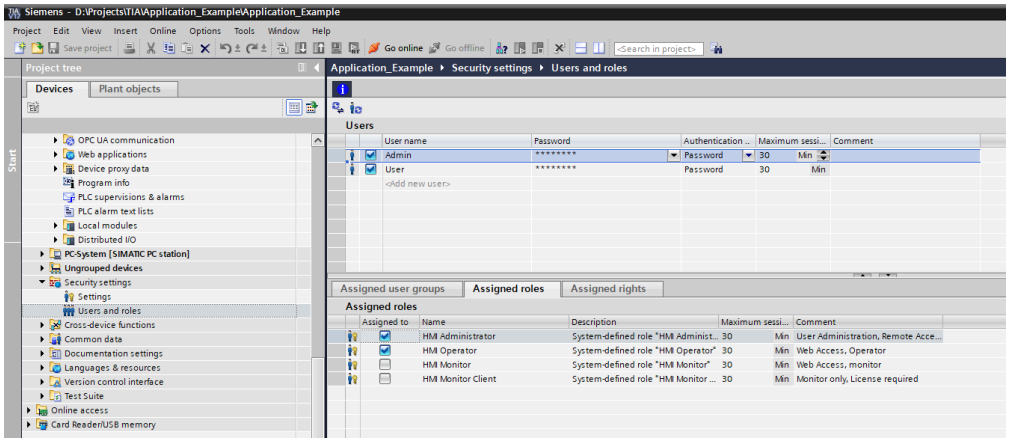
If you want to change this, you just need to execute the “WinCC Unified configuration” tool. There on the first page, you can configure these settings.



Afterwards you can login into “WinCC Unified RT”.



Login with the in the TIA Portal created accounts.

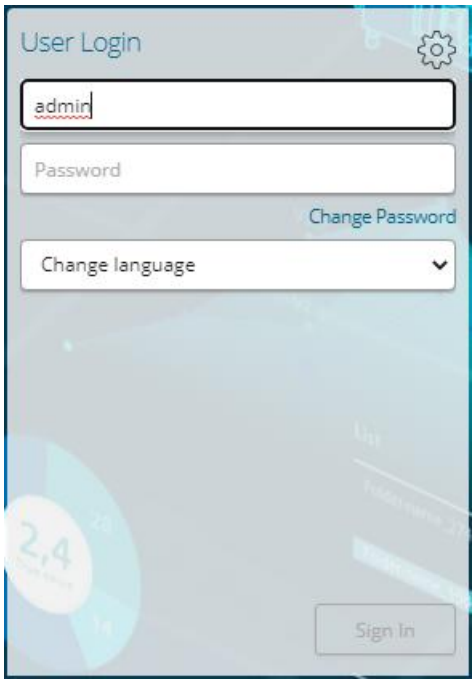


For this project are two accounts available:

- Username: Admin; Password: Admin!!1234
- Username: User; Password: Admin!!1234

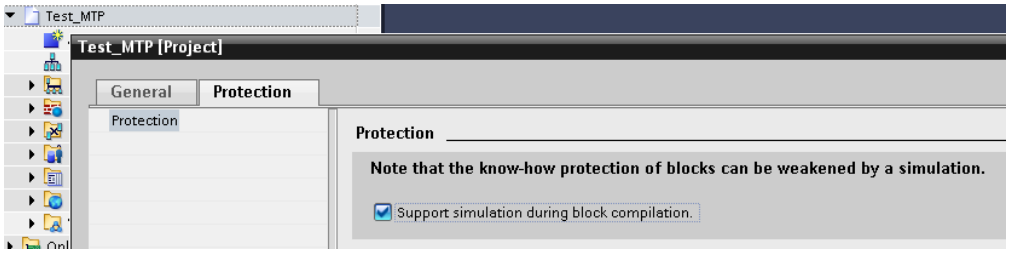
Table 3-3

	Username	Password	Rights
1.	Admin	Admin!!1234	HMI Administrator
2.	User	Admin!!1234	HMI Administrator HMI Operator



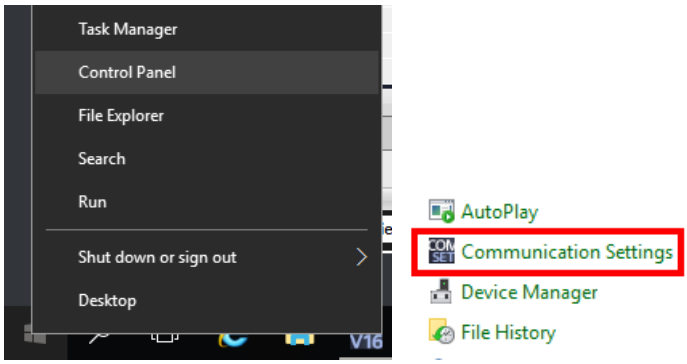
3.3 Remarks:

3.3.1 Setup Project Properties

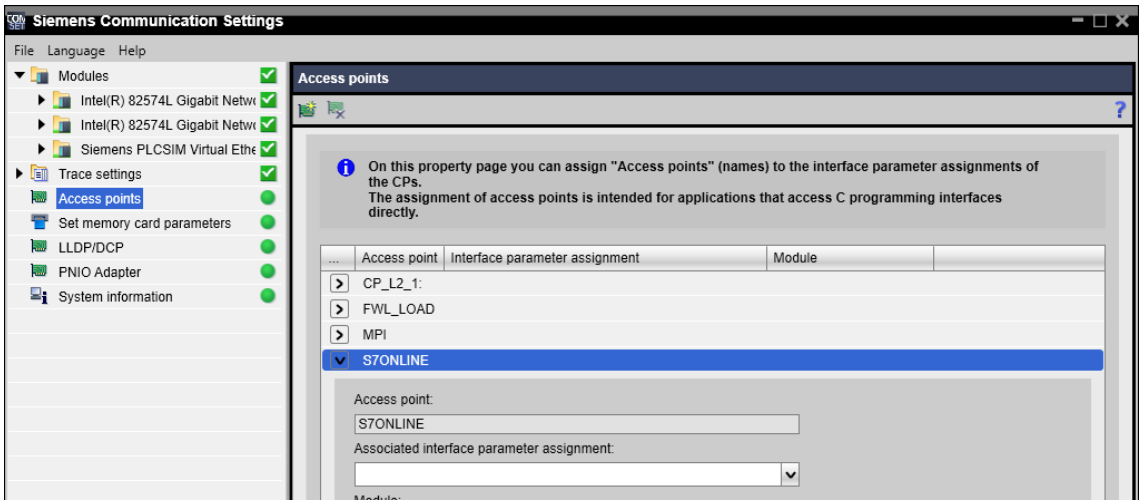


3.3.2 Setup PG/PC Interface or Communication Settings

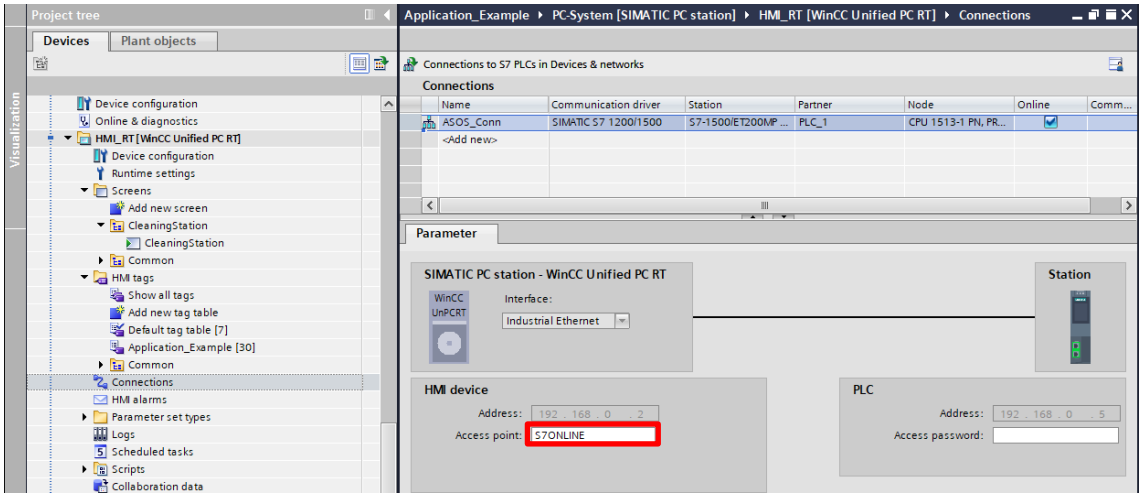
Right mouse click on windows button and open control panel and open the communication settings.



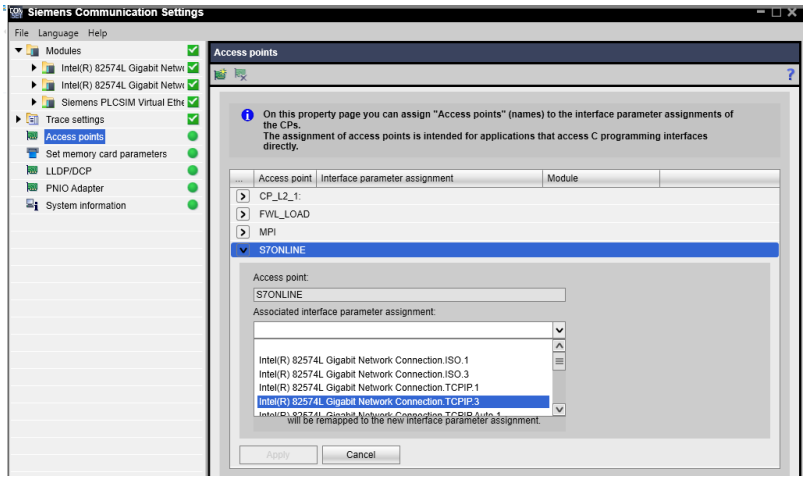
After the installation of the TIA Portal it will look like this:



Which Access point you need/use in the TIA Portal can be checked in the “Connections” Folder.



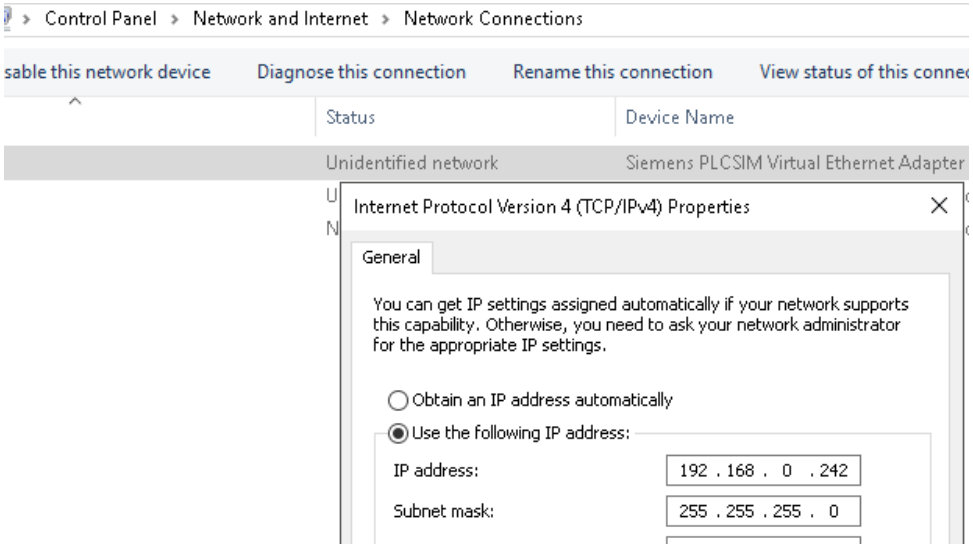
To configure the PG/PC interface, select the interface “Intel® 82574L Gigabit Network Connection.TCPIP.3” for the S7Online Access point.



3.3.3 PLCSIM Virtual Ethernet adapter

Usually, the PLCSIM Virtual Ethernet adapter the ip address is not setup.

The IP Address of the PLCSIM Virtual Ethernet adapter must be in the same subnet as the plant bus.



4 Services in the HMI

4.1 Explanation of the services

1. DoseByVolume:

This service pumps the configured amount of water with 100% opened Valve through the system (1000l). The configuration is done in the Service Engineering Tool.

2. DoseByFlow:

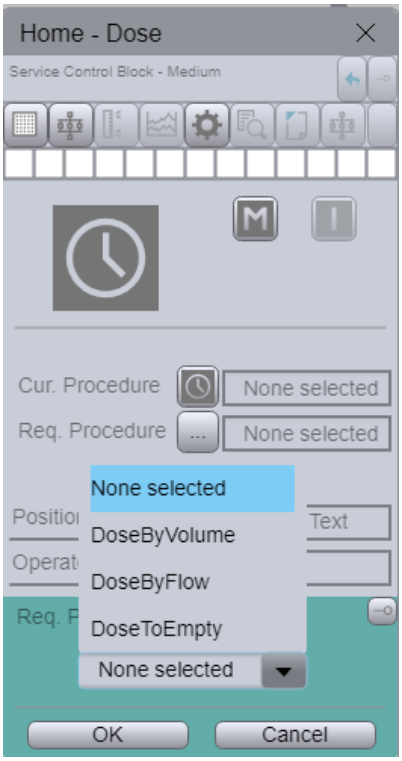
This service pumps the configured amount of water through the system (1000l) with a configured flow (25 l/s). The configuration is done in the Service Engineering Tool.

3. DoseToEmpty:

This service empties the tank up to 0,5m fill level with 100% opened Valve.

4.2 Start services

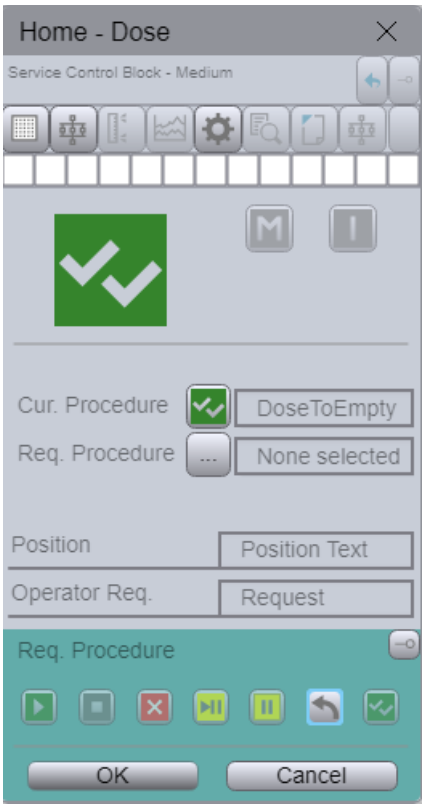
1. Open Service Control Block faceplate
2. Click on the  icon next to “Req. Procedure”
3. Select a Service (e.g. “DoseToEmpty”) and confirm with “OK”



4. Click the  icon next to “Cur. Procedure”
5. Select “Start” and confirm with “OK”

4.3 Reset services

1. Open Service Control Block faceplate
2. Click the  icon next to “Cur. Procedure”
3. Select “Reset” and confirm with “OK”

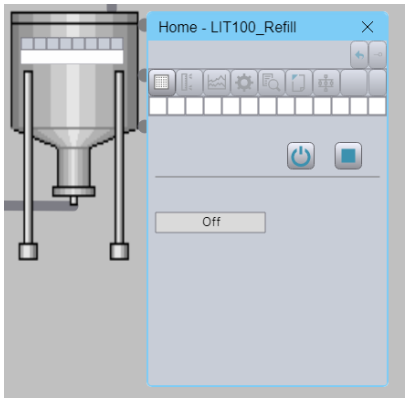


4. Start another service is possible

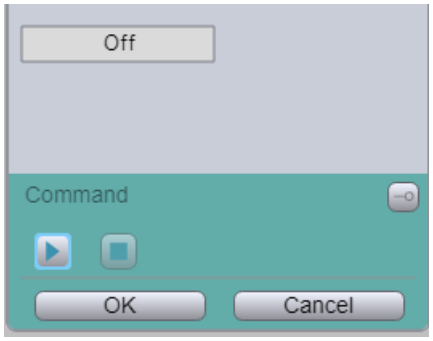
4.4 Start Refill process

It is possible to refill the tank. The tank gets refilled much faster than drained.

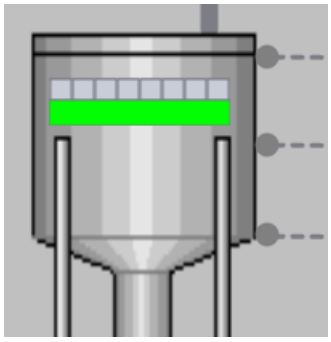
1. Open faceplate of the icon on the tank



2. Click on the “Command Off”  icon
3. And switch it to “Command On”  and confirm with “OK”



4. Refill process starts (Icons turns green)



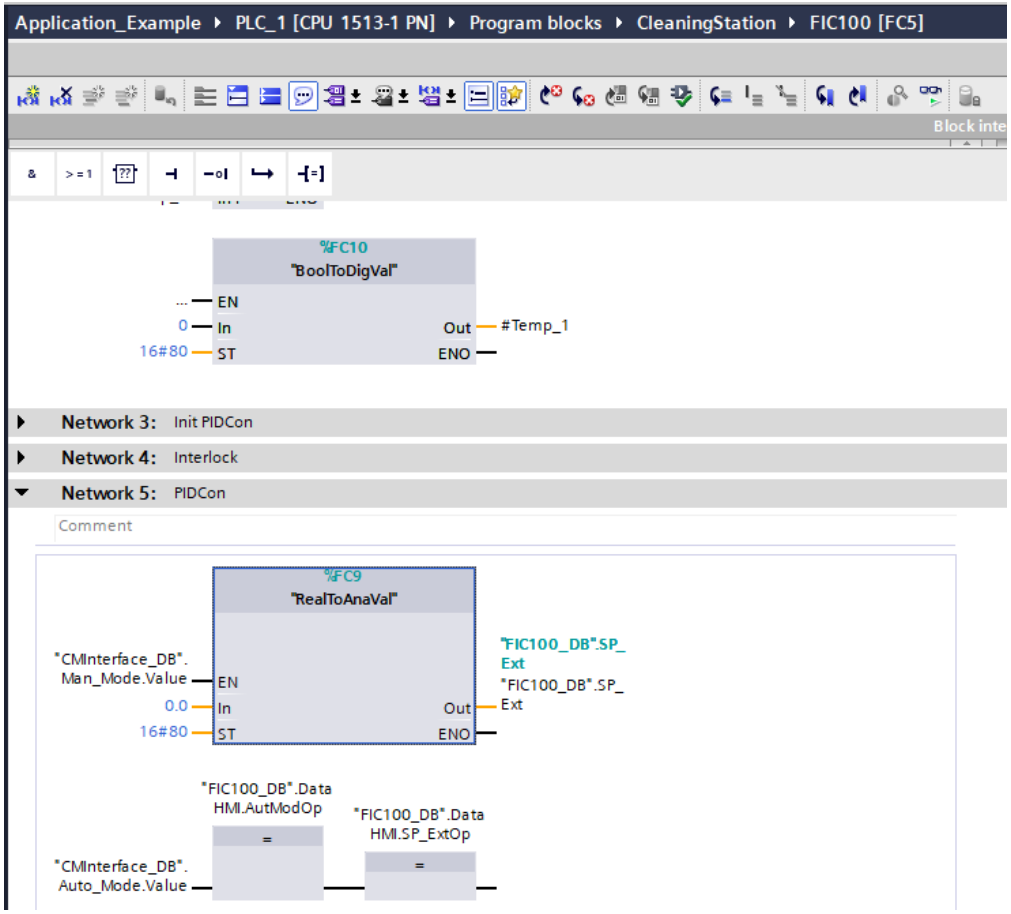
5. Refill process stops automatically if the tank is refilled. It is not possible to stop the process

5 Additional information

5.1 Convert blocks “BoolToDigVal” and “RealToAnaVal”

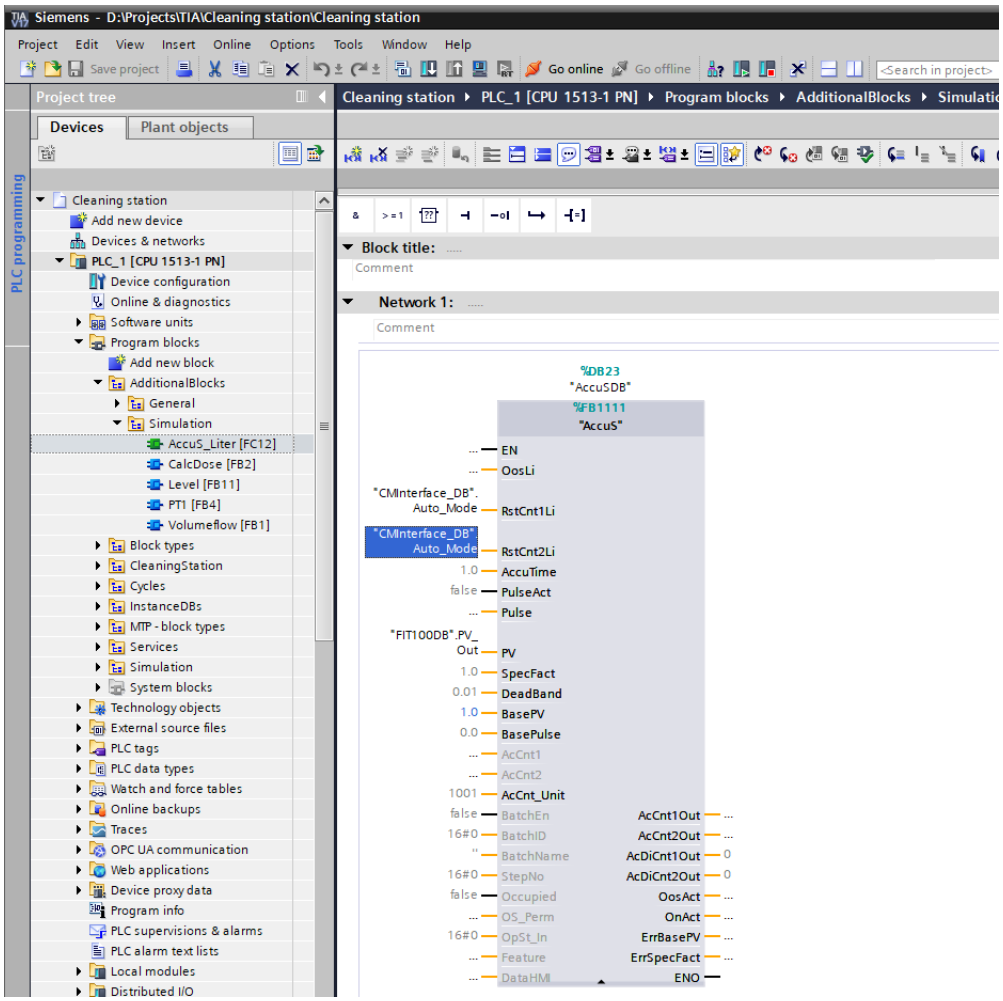
With the SIMATIC Process Function Library and the TIA Portal it is not possible to write integer or boolean directly to the block and input. The reason for this is, that in the TIA Portal each parameter/input/output has its own status. This status must also be set to “16#80” that it can be written. Therefore, are two self-written functions in the project. The first one is the “BoolToDigVal” (FC10). These blocks convert the input of datatype “Bool” with the given status to the output with datatype “DigVal”. This output can then be connected to the blocks in the project.

The second function “RealToAnaVal” (FC9) converts the input value with datatype “Real” and the given status to the datatype “AnaVal”.



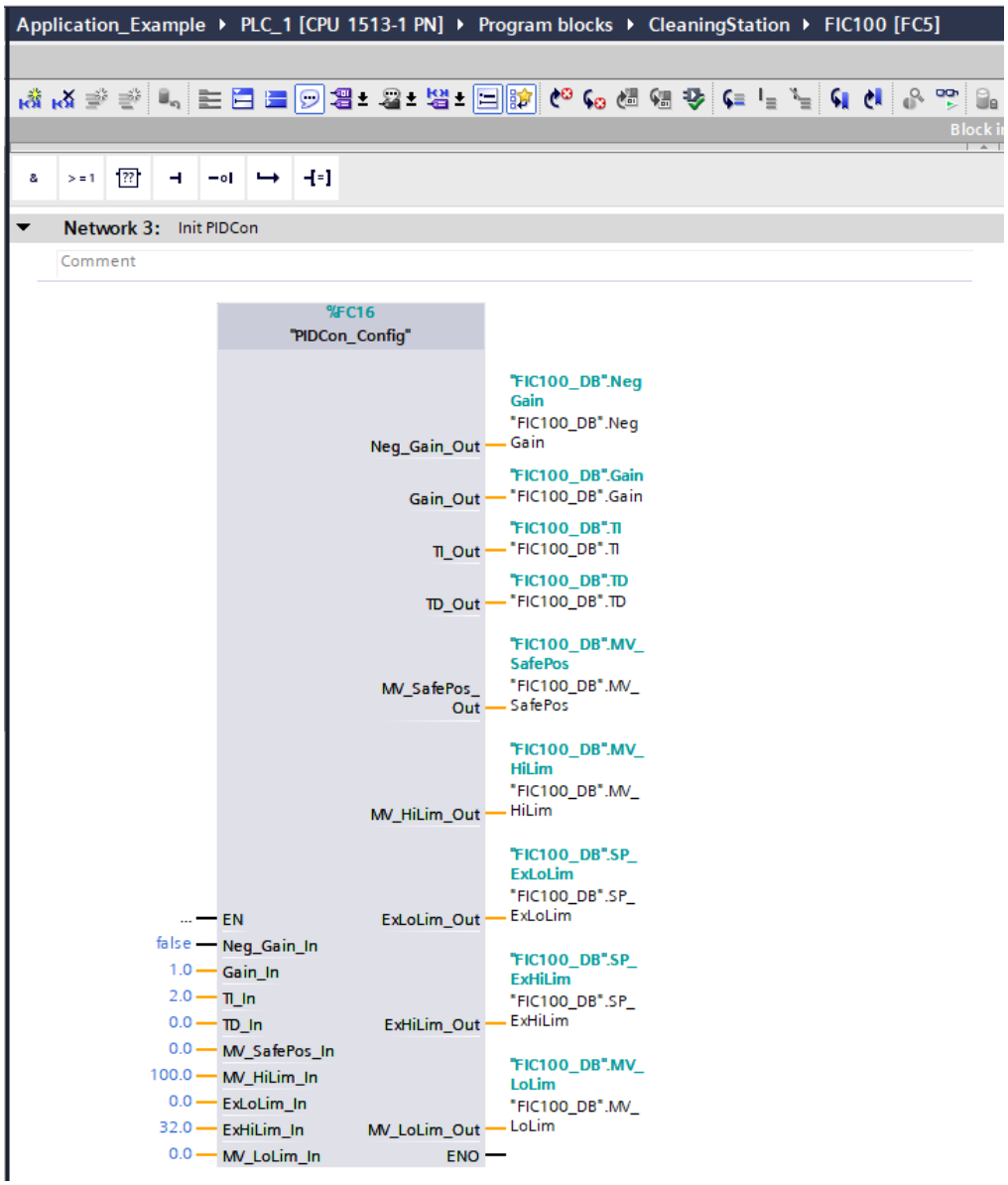
5.2 AccuS_Liter

This block is responsible for the calculation of the pumped liters. It is a block out of the SIMATIC Process Function Library and is called in the “CalcDose” function block. The calculation of the pumped liters gets always restarted when the services set the components of the process to the automatic mode.

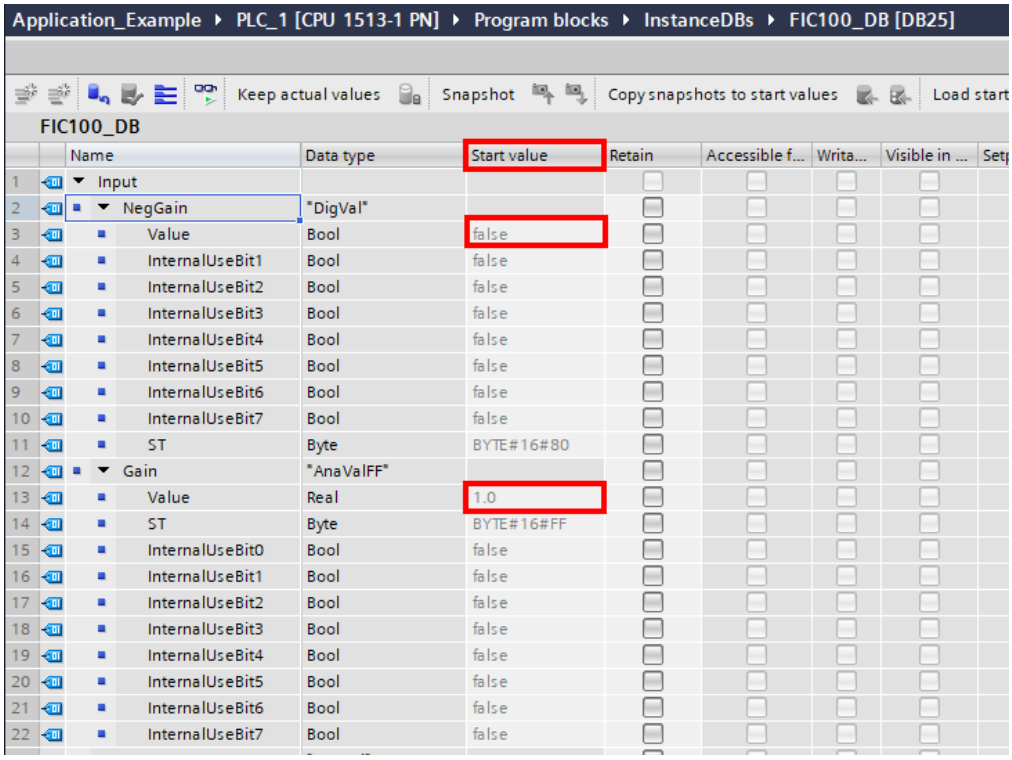


5.3 PIDConL parameter

The same problem like in chapter 5.1 Convert blocks “BoolToDigVal” and “RealToAnaVal” is the reason for the function “PIDCon_Config” (FC16). This block writes the config parameter for the “PIDConL” block. With this method it is not possible to change the parameter during the runtime.



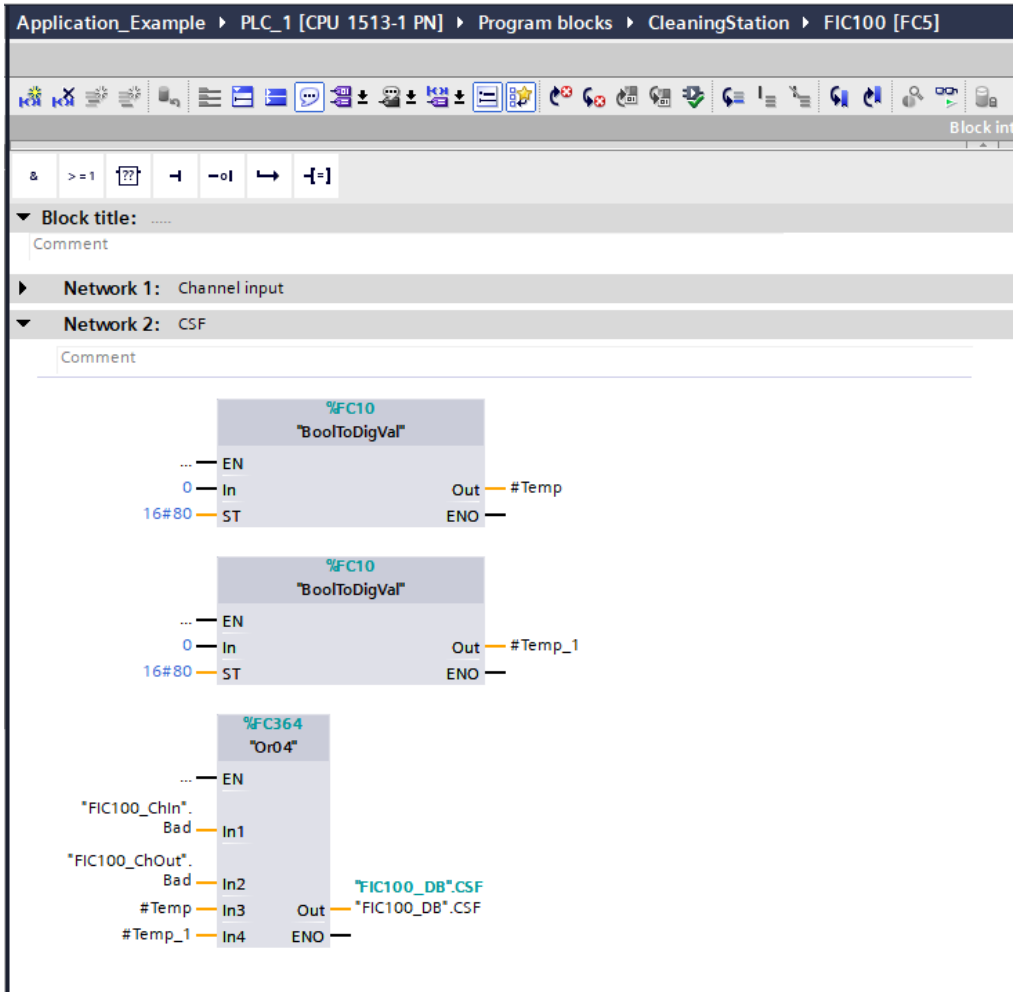
If you want to make it possible for the operator to change them during the runtime, you need to write the parameter directly into the instance DB as start value.

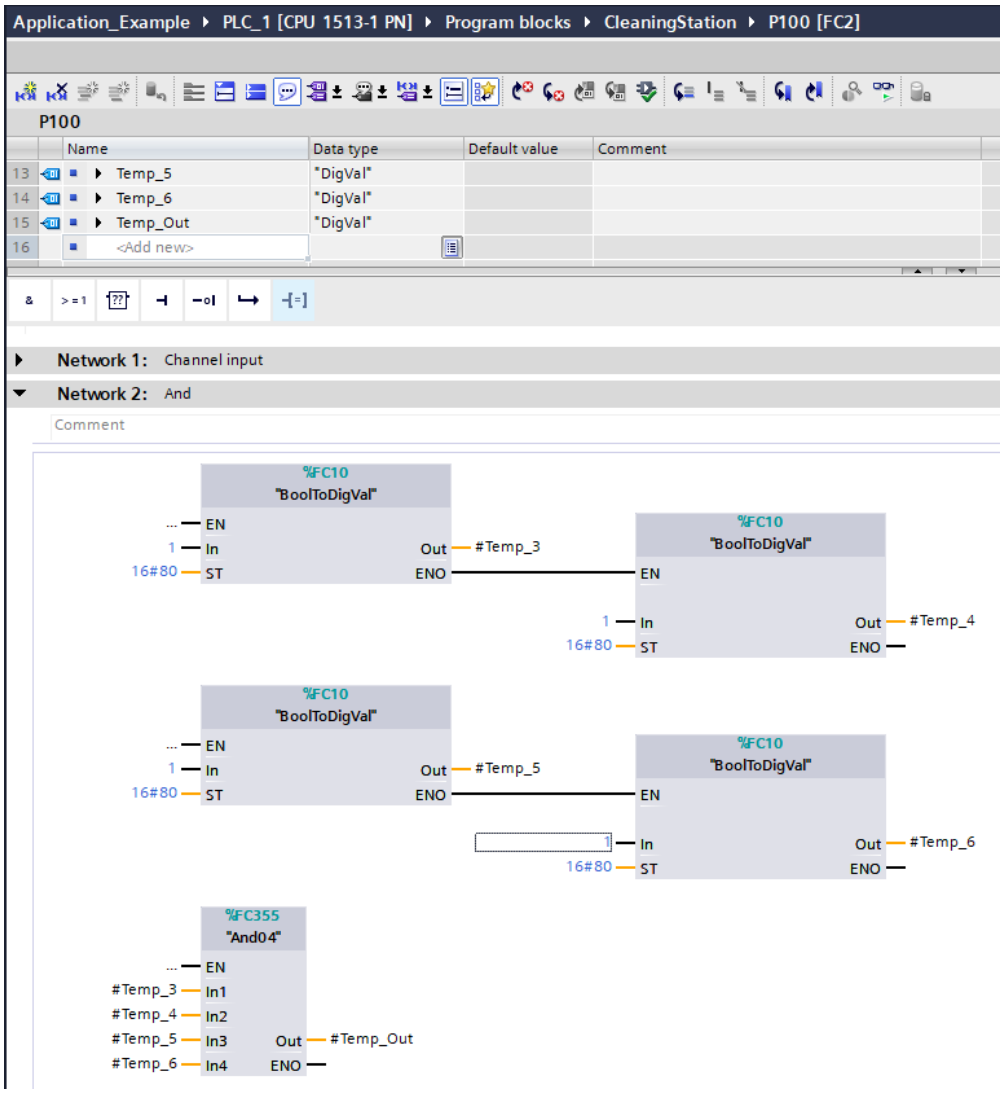


	Name	Data type	Start value	Retain	Accessible f...	Writa...	Visible in ...	Set
1	Input							
2	NegGain	"DigVal"						
3	Value	Bool	false					
4	InternalUseBit1	Bool	false					
5	InternalUseBit2	Bool	false					
6	InternalUseBit3	Bool	false					
7	InternalUseBit4	Bool	false					
8	InternalUseBit5	Bool	false					
9	InternalUseBit6	Bool	false					
10	InternalUseBit7	Bool	false					
11	ST	Byte	BYTE#16#80					
12	Gain	"AnaValFF"						
13	Value	Real	1.0					
14	ST	Byte	BYTE#16#FF					
15	InternalUseBit0	Bool	false					
16	InternalUseBit1	Bool	false					
17	InternalUseBit2	Bool	false					
18	InternalUseBit3	Bool	false					
19	InternalUseBit4	Bool	false					
20	InternalUseBit5	Bool	false					
21	InternalUseBit6	Bool	false					
22	InternalUseBit7	Bool	false					

5.4 “And04” and “Or04” from SIMATIC Process Function Library

The SIMATIC Process Function Library also provides an “And04” and an “Or04” function. The reason for this is the same like in the last two chapters because of the status of the parameter. These two blocks are already capable to handle the status. But for the using you need to know, that it is mandatory to connect all inputs of this block. A function does not have an instance DB, so all inputs need to be configured, otherwise it will not work. If you do not have enough inputs you need to connect some “temp” variables to the input. The “temp” variables for the “Or04” block need to be set to “0” and the ones for the “And04” block need to be set to “1” for the correct function of them.

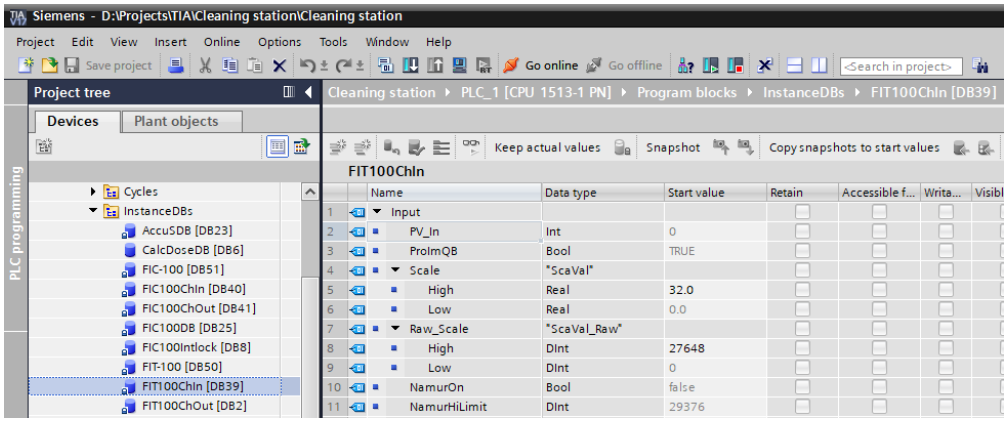




5.5 Instance DB “Raw_Scale” and “Scale”

For the correct function of the simulation, it is mandatory to configure the “Scale” and the “Raw_Scale” parameter in the instance DB for the channel input. In this project the “Raw_Scale” needs to have his “Low” parameter at “0” and the “High” parameter at “27648”. The “Scale” parameter needs to have his “Low” parameter also at “0” and the “High” parameter must be the high limit of its corresponding value. So, in the picture we can see the instance DB from “FIT100” channel input. The maximum of the flow can be 32 L/s, so the “High” value of “Scale” is 32.

This configuration is needed because for example the “AnIn” block reads a raw analog value from the process image (partition) and converts it to its physical value or calculates a percentage value based on this raw value.



6 Appendix

6.1 Service and support

Industry Online Support

Do you have any questions or need assistance?

Siemens Industry Online Support offers round the clock access to our entire service and support know-how and portfolio.

The Industry Online Support is the central address for information about our products, solutions and services.

Product information, manuals, downloads, FAQs, application examples and videos – all information is accessible with just a few mouse clicks:

support.industry.siemens.com

Technical Support

The Technical Support of Siemens Industry provides you fast and competent support regarding all technical queries with numerous tailor-made offers – ranging from basic support to individual support contracts.

Please send queries to Technical Support via Web form:

siemens.com/SupportRequest

SITRAIN – Digital Industry Academy

We support you with our globally available training courses for industry with practical experience, innovative learning methods and a concept that’s tailored to the customer’s specific needs.

For more information on our offered trainings and courses, as well as their locations and dates, refer to our web page:

siemens.com/sitrain

Service offer

Our range of services includes the following:

- Plant data services
- Spare parts services
- Repair services
- On-site and maintenance services
- Retrofitting and modernization services
- Service programs and contracts

You can find detailed information on our range of services in the service catalog web page:

support.industry.siemens.com/cs/sc

Industry Online Support app

You will receive optimum support wherever you are with the "Siemens Industry Online Support" app. The app is available for iOS and Android:

support.industry.siemens.com/cs/ww/en/sc/2067

6.2 Industry Mall

The Siemens Industry Mall is the platform on which the entire siemens Industry product portfolio is accessible. From the selection of products to the order and the delivery tracking, the Industry Mall enables the complete purchasing processing – directly and independently of time and location:
mall.industry.siemens.com

6.3 Links and literature

Table 6-1

No.	Topic
\1\	Siemens Industry Online Support https://support.industry.siemens.com
\2\	Link to this entry page of this application example https://support.industry.siemens.com/cs/ww/en/view/109810699
\3\	

6.4 Change documentation

Table 6-2

Version	Date	Modifications
V1.0	05/2022	First version
V2.0	08/2022	Update of chapters 3 and 4.