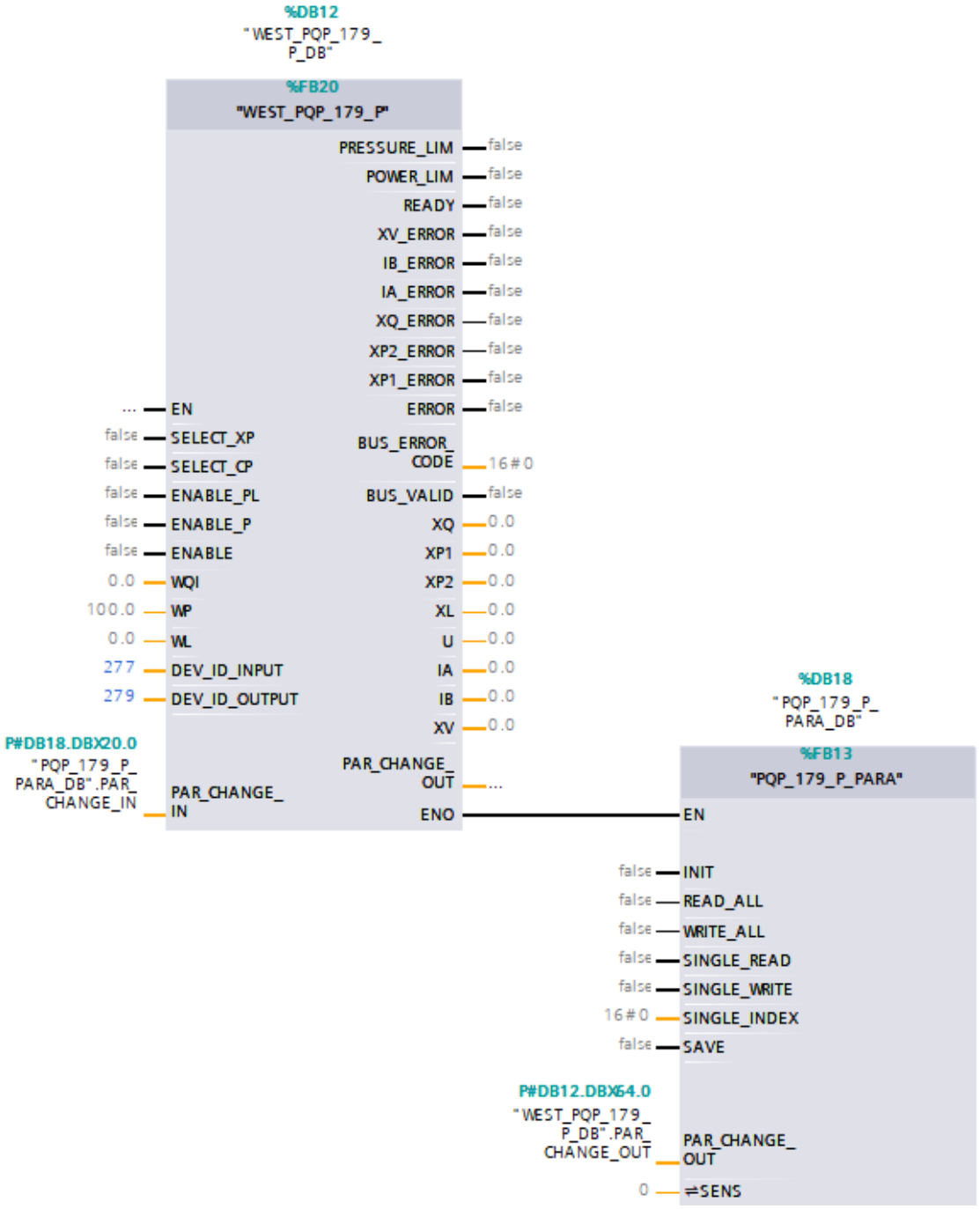
Documentation

PQP_179_P_PARA

Parameterisation module for the pump controller PQP-179-P-PFN
For use with Siemens PLCs (S7-1200 / 1500 / 300 / 400)



Contents

1 General information

1.1 Associated module

PQP-179-P-PFN - Pump control module for cascade control in open or closed circuit with analogue control output, integrated power stage, optionally activatable spool position controller for the control valve and Profinet interface

1.2 Required software and hardware

The programme module described here is intended for use in programmable logic controllers from Siemens. The programming environment "TIA - Portal", at least version 13, is required for engineering.

In addition, the general S7 driver module (WEST_PQP_179_P) is required, for integration see module instructions, as well as a Profinet-capable PLC.

1.3 Legal notice

W.E.St. Elektronik GmbH

Gewerbering 31
D-41372 Niederkrüchten

Tel.: +49 (0)2163 577355-0
Fax.: +49 (0)2163 577355-11

Home page: www.w-e-st.de
EMAIL: contact@w-e-st.de

Date: 26.11.2024

2 Properties

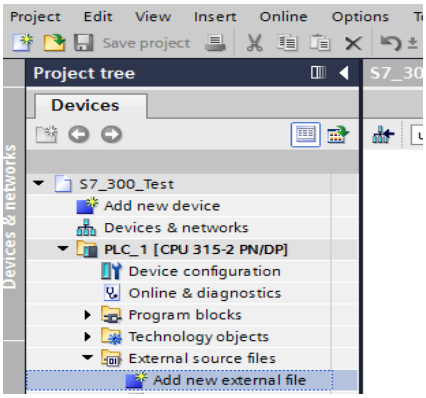
Supplementary TIA - Portal module for the pump controller driver
1. function for reading and transmitting entire parameter data sets (all readable and writable parameters)
2. function for changing individual parameters during operation
3. function for reading individual parameters

These functions are combined in one module.

3 Integration into the automation project

The installation of the basic module "WEST_PQP_179_P_PFN" is described in the documentation of the module.

1.) The parameterisation module is also provided as an SCL source. For installation in the project, this file must be added in the TIA portal as a "new external file":

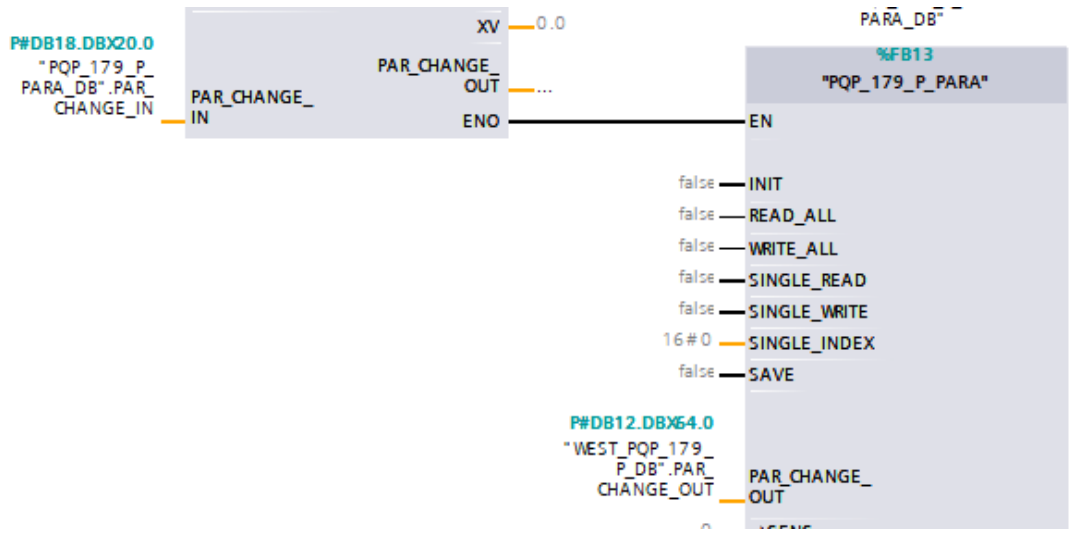


2.) Then click on the imported file with the right mouse button and select the option "Generate blocks from source". After the translation, the driver block is available in the block folder.

This FB can now be called up in the user program. This should be done in a cyclic OB with a cycle time >= 4 ms. We recommend processing as quickly as possible up to max. 50 ms so that the transmission, which takes place in several cycles (asynchronously), does not take too long.

Installation in the free cycle (OB1) is not possible!

Since the parameterisation module uses the access mechanisms of the driver module, the two modules must be interconnected. In addition, it makes sense to call them up in the same higher-level module.
The following illustration shows an example of a variant for linking under FBD. The necessary coupling is done by linking the corresponding structure output variable from the instance DB of the respective other module to the input parameter. In this way, the connections "PAR_CHANGE_IN" and "PAR_CHANGE_OUT" are to be linked. These names refer to the view of the driver module with regard to input and output. Thus, the output of the parameterisation module is called "PAR_CHANGE_IN", since it is connected to this input of the driver.



Due to the handling of the instance DB of the parameter module as an image of the setting values, it is not recommended to create it in the form of a multi-instance.

A variable must be connected to the "INIT" input, which becomes "TRUE" at least during one module call when the controller starts up. For this purpose, a flag can be used, for example, which is set in OB100 and re-set after the block has been processed.

The parameters of the position controller are linked as I/O parameters of the function block in the instance DB.

There are two alternative usage options:

- Setting and observing the values in the online view of the instance DB in the TIA portal.
- Linking the values with I/O fields of a WinCC, WinCC flexible or other HMI applications.

4 Mode of Operation

The following inputs and outputs are used to control the function:

PQP_179_P_PARA		
Input / **O**utput	**Name**	**Function**
I	INIT	Initialise the module: The current parameters and their limits are read from the pump controller and stored in an internal memory area of the module.
I	READ_ALL	A rising edge at this input causes all currently set parameters of the module to be read out and transferred to the parameters of the module interface (IO - range, see below).
I	WRITE_ALL	A rising edge at this input causes the parameters of the module interface (IO area, see below) to be transferred to the controller module.
I	SINGLE_READ	The "SINGLE_..." inputs are intended for access to individual values. A rising edge at this input causes the parameter whose index is specified at the "SINGLE_INDEX" input according to the parameter list (controller documentation) to be read out of the controller. The value is output under "SINGLEACCESS" (see below).
I	SINGLE_WRITE	A rising edge at this input causes the parameter whose index is specified at the "SINGLE_INDEX" input according to the parameter list (controller doc) to be written to the controller. The value is to be entered under "SINGLEACCESS" (see below).
I	SAVE	A rising edge at this input causes the parameters to be permanently stored in the controller. Caution: Do not activate automatically, as otherwise the number of write cycles for the EEPROM could become too large.
I	SINGLE_INDEX	Index for single access
IO	*„Parameter name“*	Image of the values on the controller or parameter set to be written. The comments on the individual values provide more detailed information. Due to language restrictions, special characters are not possible in the names. Therefore, some of the names are slightly modified. Selection parameters are given as numerical values. The selection of an invalid value leads to an error message when writing.
IO	SINGLEACCESS	Parameter to be written or read by means of the "SINGLE" commands.
O	STATUS	The status of the module is displayed here: 0 = inactive, 1 = ready and initialised, 2 = busy, 3 = error. During command processing, the module displays "2" = busy for a number of cycles. Afterwards, the status changes back to "1" = ready if execution is successful. If an error occurs during processing or access is not permitted (e.g. ENABLE + WRITE_ALL), the status changes to "3". This message remains until the next command execution.

O	ERROR_LINE	In the event of parameter errors (range violation or timeout during writing), the number of the affected parameter is output here.
O	DISC_LINE	If there is a discrepancy between the offline parameters (at the I/Os of this module) and the values in the controller, the number of the first deviating value is displayed here. This function allows minor deviations that may result from rounding errors when saving the values in the module.
O	DISC_COUNT	Number of discrepancies. Set to "-1" if monitoring is not taking place (e.g. bus inactive).

5 Practical operation

5.1 Variant 1: Parameterisation via the PLC only

After the module has been connected and the Profinet connection is established, a rising edge should be applied to the input parameter ‘INIT’. The FB now reads information about the module's parameters and their limits into an internal memory area.

Since there are still no meaningful values at the I/O parameters, all parameters are initially regarded as deviating:

10		Output										
11		STATUS	Int	12.0	0	1					0 = no active, 1 = ready and intitialize
12		ERROR_LINE	Int	14.0	0	0					number of the faulty parameter (rang
13		DISC_LINE	Int	16.0	0	1					line of 1st deviation / Zeile mit der er
14		DISC_COUNT	Int	18.0	0	32					number of deriving parameters / Zah
15		PAR_CHANGE_IN	Struct	20.0							structure input for parameter change

The default parameters can now be read out (rising edge "READ_ALL").

After waiting a few seconds, the values are adjusted and can be displayed:

	Output				
	STATUS	Int	12.0	0	1
	ERROR_LINE	Int	14.0	0	0
	DISC_LINE	Int	16.0	0	0
	DISC_COUNT	Int	18.0	0	0
	PAR_CHANGE_IN	Struct	20.0		
	InOut				
	SENS	Int	30.0	0	1
	CIRCUIT	Int	32.0	0	2
	CTRLOUT	Int	34.0	0	4
	LIM_XQ	Int	36.0	0	0
	FUNCT	Int	38.0	0	1
	SYS_RANGE	Int	40.0	0	100
	SIGNAL_XP1	Int	42.0	0	3
	N_RANGE_XP1	Int	44.0	0	100
	SIGNAL_XP2	Int	46.0	0	3
	N_RANGE_XP2	Int	48.0	0	123
	SIGNAL_XQ	Int	50.0	0	1
	ZERO_XQ	Int	52.0	0	5000
	FULL_XQ_PLUS	Int	54.0	0	10000
	FULL_XQ_MINUS	Int	56.0	0	2000

Now the values are set that are known from the design or from data sheets, e.g. SYS_RANGE, N_RANGE.., CURRENT....

These values are then transferred to the module with the command "WRITE_ALL".

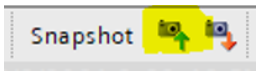
Other values must be adjusted during operation, such as PID – parameters, MIN:A / MIN:B, OFFSET, etc.

During optimisation, individual parameters are usually accessed for this purpose; this can be done via SINGLE_WRITE, via WPC or alternatively by transferring the entire parameter set via "WRITE_ALL".

It should be noted, however, that in the case of individual access, no synchronisation of the internal online image takes place.

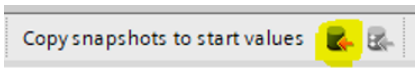
Therefore, in order to update it, the complete parameter set must be read out via the "READ_ALL" command at the end.

As soon as all parameters are set as desired, the TIA / S7 project status should be saved offline. To do this, go to the online view of the module and take a snapshot of the monitoring values:



The read-out values are now displayed in another column.

Via the button "Copy snapshots to start values" the parameters are taken over into the column "Start value":



Then save the TIA portal project and reload the changed module from the module folder to the controller.

In this way, the set values are saved both on the controller and in the offline file folder, but not in the EEPROM of the module. This should be done after the setting work has been completed, so that the pump controller uses the current parameters from the controller after a power recovery without having to write to them again.

Saving in the controller is triggered by a rising edge of the input signal at "SAVE". This must not be done cyclically from the programme in order not to exceed the maximum write cycles of the unit.

5.2 Variant 2: Saving a parameter set on the controller in the PLC

It is also possible to carry out the settings on the controller completely with the WPC programme and then to create a backup of the parameter set on the controller and in the offline programme folder.

If the controller has to be replaced, it is easy to transfer this data set to the new controller using the command sequence "WRITE_ALL" followed by "SAVE".

To create the backup:

- Perform READ_ALL
- Snapshot the observed values (see above)
- Take over the snapshot values as start values
- Save the TIA project

5.3 Integrity of the parameterisation

Correct parameterisation is of fundamental importance for the proper functioning of the pump control module. Incorrect or incomplete parameterisation can lead to serious malfunctions.

It is therefore important to monitor the integrity of the parameters. For example, after a "WRITE_ALL" write operation, check whether it was carried out successfully (STATUS = 1) and there are no more deviations (DISC_COUNT = 0). If you want to carry out the transfer of the entire parameter set automatically from the PLC, a corresponding check must be carried out after writing in the user programme of the PLC before the control function is released.