

Short documentation

Integration of i550 into Siemens Step 7 and Siemens TIA Portal on Profibus and Profinet

Software versions used:

Lenze Easy Starter V1.10.2

i550, Profibus Standard IO with FW 03.00.02, Profinet Standard IO with FW 03.00.02

Siemens TIA Portal V14 Update 1

Siemens Step 7 V5.5 SP4

Version	Notes	Author / Date
V1.0	First version	Dittmann / 23.11.2015
V1.1	Update Profinet	Dittmann / 22.02.2016
V2	Update TIA-Portal V14	Dittmann / 06.12.2016

Contents:

1	i550 on Profibus	2
1.1	Parameter setting with the EASY Starter.....	2
1.1.1	Setting of the Profibus address	2
1.1.2	Establish a connection to the i550 via USB diagnostic interface.....	2
1.1.3	Setting of the required basic parameters	3
1.2	Integration under Step - 7.....	4
1.2.1	Hardware configuration	4
1.2.2	Implementing the "LCB-ActuatorSpeed_V2_1" function block into the control program of the S7.....	8
1.3	Integration under the Siemens TIA portal	10
1.3.1	Hardware configuration in the TIA portal with S7-300 control	10
1.3.2	Implementing the "LCB-ActuatorSpeed_V2_1" function block into the control program of the S7.....	16
1.3.3	Special features of S7-1200 and S7-1500 control	20
2	i550 on Profinet	22
2.1	Parameter setting with the EASY Starter.....	22
2.1.1	Setting of the Profinet address and stationname	22
3	Parameter data communication via DP-V1	23
3.1	Example: Reading / writing a parameter with an S7-1500 CPU	23
3.1.1	Determining the diagnostics address in case of an S7-300	23
3.1.2	Determining the HW identification for the parameter transfer for S7-1200 and S71500	23
3.1.3	Implementing the modules into the PLC program.....	24

1 i550 on Profibus

This documentation describes the integration of the i550 on Profibus with the Lenze S7-AppSample libraries and the speed actuating drive application.

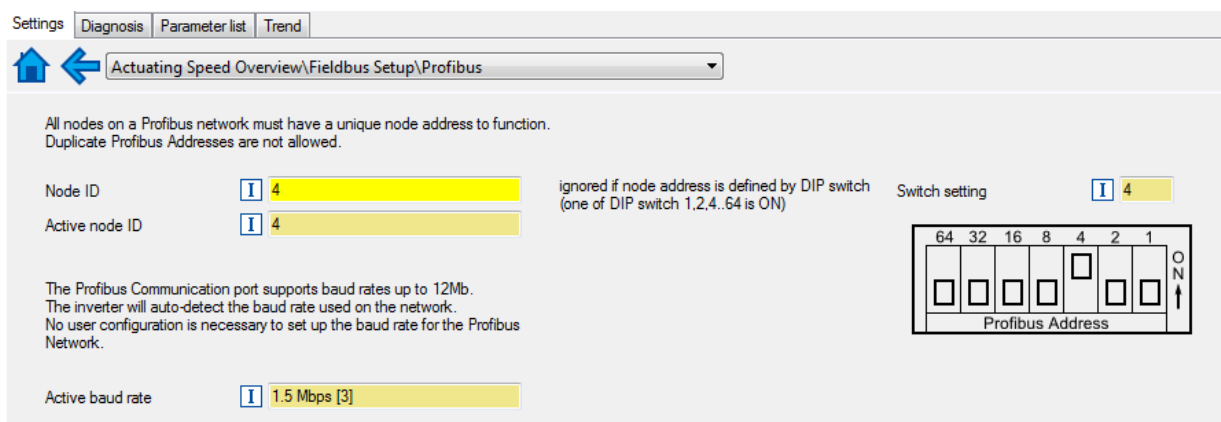
1.1 Parameter setting with the EASY Starter

For the general control via Profibus, only a few codes have to be set at the i550 via the keypad or the EASY Starter. The basic drive parameters such as base frequency or maximum current are also parameterised via the EASY Starter.

1.1.1 Setting of the Profibus address

The Profibus address can be set via the dip switches at the front of the i550.

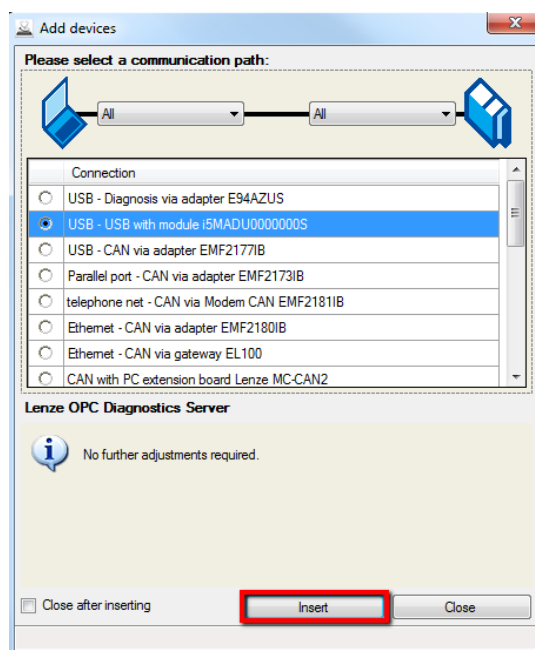
If the address is not set via the dip switches, it can be set via the ".\Fieldbus Setup\Profibus" (index 2341:001).



The screenshot shows the 'Settings' tab in the EASY Starter software. The breadcrumb path is 'Actuating Speed Overview\Fieldbus Setup\Profibus'. A warning message states: 'All nodes on a Profibus network must have a unique node address to function. Duplicate Profibus Addresses are not allowed.' The 'Node ID' and 'Active node ID' are both set to 4. A note indicates that the address is ignored if defined by DIP switches (one of 1, 2, 4, 64 is ON). The 'Switch setting' is also 4. A diagram of the DIP switches shows the 4th switch (labeled 4) is turned ON. The 'Active baud rate' is set to 1.5 Mbps [3]. A note mentions that the Profibus Communication port supports baud rates up to 12Mb and that the inverter will auto-detect the baud rate.

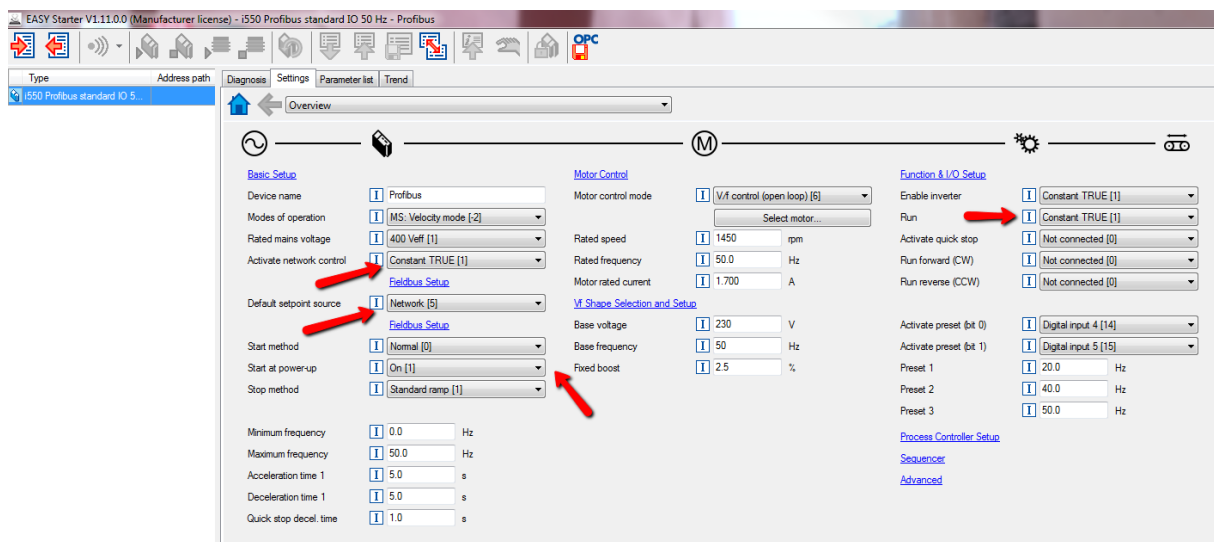
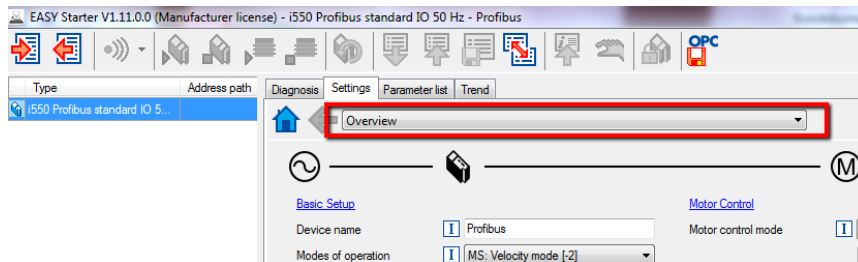
The example uses the address 4.

1.1.2 Establish a connection to the i550 via USB diagnostic interface



1.1.3 Setting of the required basic parameters

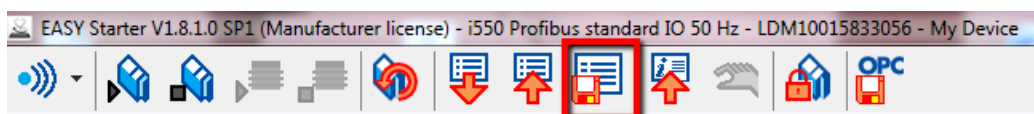
Select the "Overview" point via the fast access drop-down menu:



Set the Network enable (index 0x2631:037) to "Constant True", set Default setpoint source (index 0x2860:001) to "Network [5]".

Set the "Starting-Stopping Behavior" depending on the requirement, set "RUN enable" (index 2631:002) to "Constant True[1]". This means that no hardware controller enable is required. The starting performance can be defined as well at start-at powerup (index 0x2838:002).

Finally, save the parameter set in the i550.



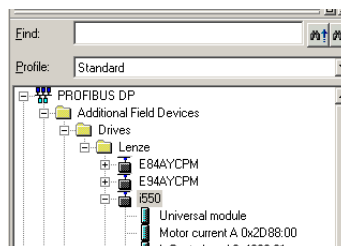
1.2 Integration under Step - 7

The integration under Step-7 requires the GSE file of the i550. This file has to be integrated into the hardware catalog of Step-7 via the hardware manager.

In order to use the "LCB_ActuatorSpeed_V2_1" module from the Lenze sample library, it needs to be integrated.

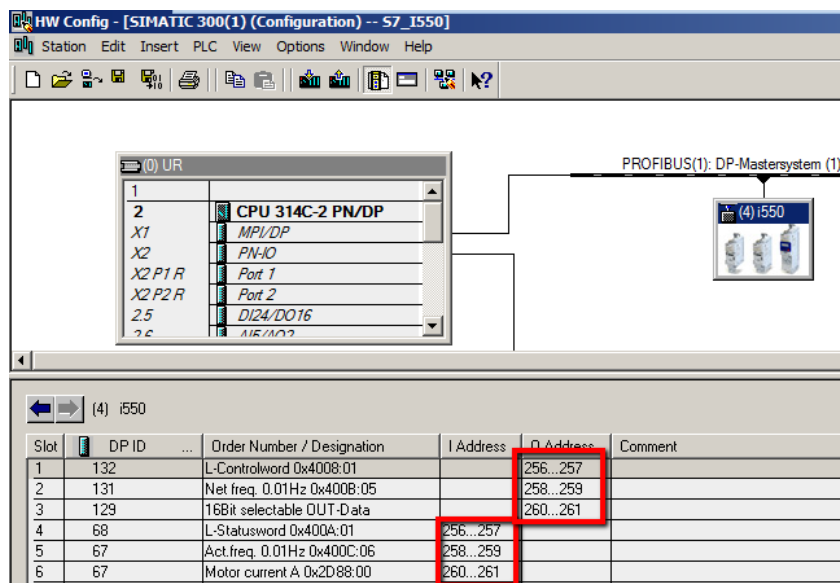
1.2.1 Hardware configuration

Select the i550 under "Profibus-DP\Additional Field Devices\Drives\Lenze".



Afterwards, set the Profibus address (in the example: 4)

If the i550 is moved into the hardware configuration per "Drag and Drop", the standard I/O configuration of the process data has already been created. It does not need to be created manually but can be edited or extended.



Slot	DP ID	...	Order Number / Designation	I Address	Q Address	Comment
1	132		L-Controlword 0x4008:01		256...257	
2	131		Net freq. 0.01Hz 0x4008:05		258...259	
3	129		16Bit selectable QUT-Data		260...261	
4	68		L-Statusword 0x400A:01	256...257		
5	67		Act.freq. 0.01Hz 0x400C:06	258...259		
6	67		Motor current A 0x2D88:00	260...261		

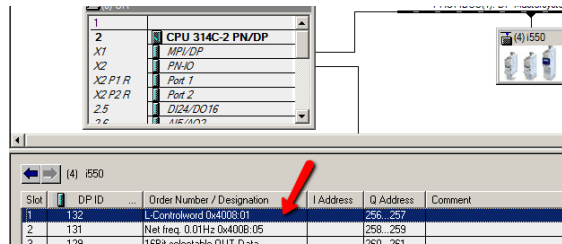
The "LCB_ActuatorSpeed_V2_1" requires a data length of 3 words each of input data and output data. **In order that the module operates correctly, you have to make sure that the addresses follow each other directly without any jumps and the start addresses of the I/O ranges have to be identical (256).** The basic configuration for the basic functionality does not need to be adapted anymore. Individual adaptations, however, are possible.

1.2.1.1 Adjusting the process data mapping

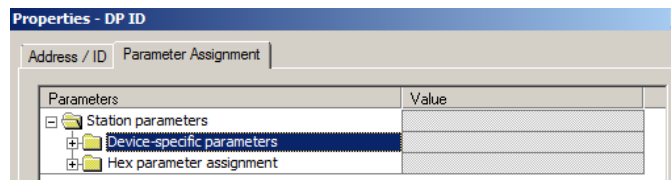
The process data mapping can only be changed on the control side in the hardware manager. At every restart, the Siemens control writes this down onto the i550 via Profibus.

The assignment of the free input bits (xFreeCtrl_1-4) of the Lenze FB can be directly changed in the hardware manager in the control word.

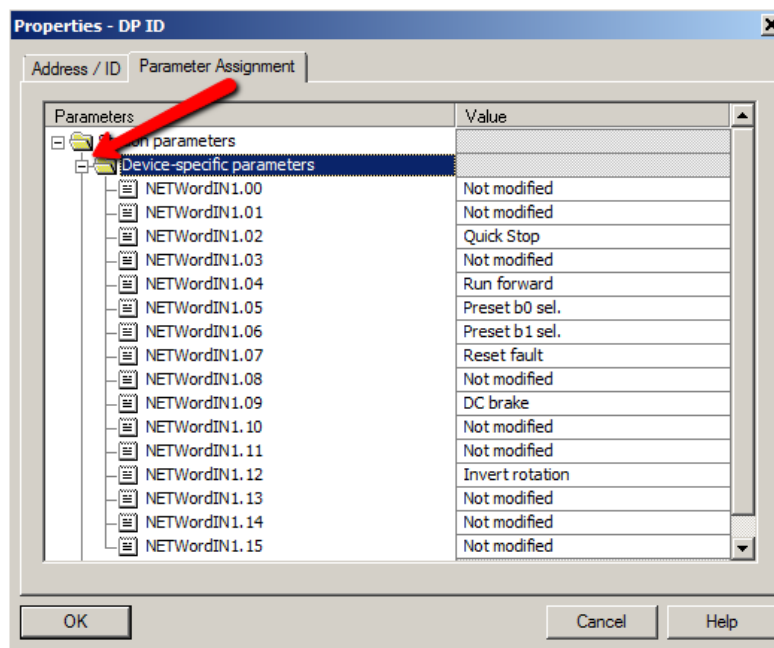
Double-clicking the control word opens the parameterisation dialog:



Selecting Parameter Assignment:



Expanding device-specific parameters:



The free control bits are assigned as follows:

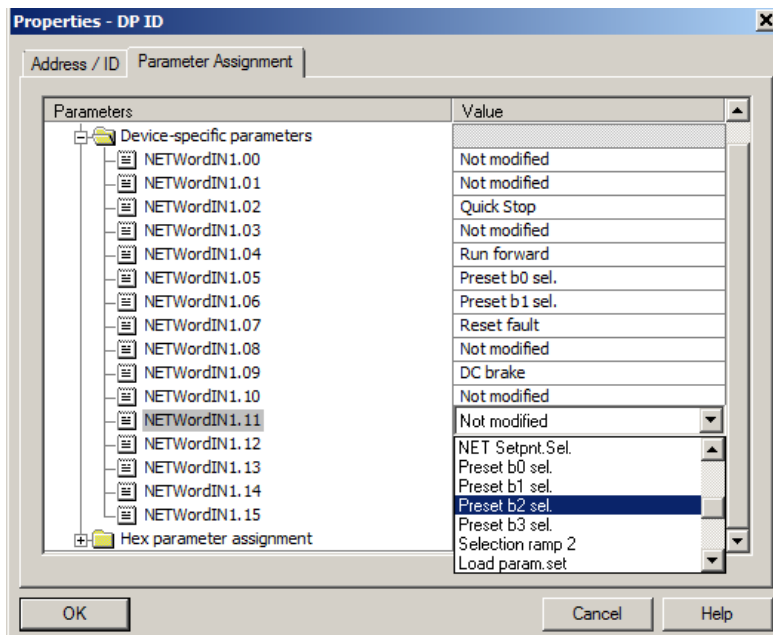
xFreeCtrl_1 <-> NetWordIn1.11

xFreeCtrl_2 <-> NetWordIn1.13

xFreeCtrl_3 <-> NetWordIn1.14

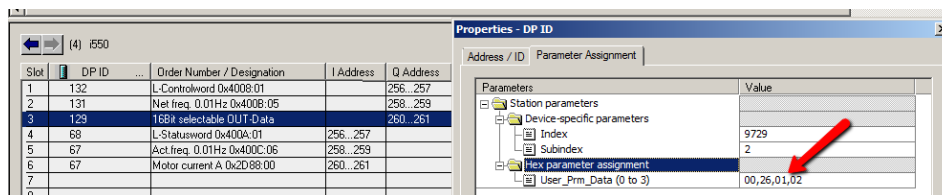
xFreeCtrl_4 <-> NetWordIn1.15

For every single bit, a new function can be assigned via the drop-down menu:



The free input word wFreeCtrl_1 of the module is linked to the 3 output word of the process image (16Bit selectable Out-Data) and can also only be changed in the hardware configuration on the S7 side:

When you open the parameterisation dialog by double-clicking, you can select which index is to be written in the controller:



"Device-specific parameters" include the index and subindex in decimal form. Hex parameter setting directly shows the HEX index: (0x2601:02 default assignment):

Address	Display code	Name	Value	Unit
0x2601:000	PAR0202:000	highest sub-index supported (generated)	3	
0x2601:001	PAR0202:001	Keypad setpoint commands: Frequenc...	20.0	Hz
0x2601:002	PAR0202:002	Keypad setpoint commands: Process c...	0.00	PUnit
0x2601:003	PAR0202:003	Keypad setpoint commands: Torque s...	100.0	%

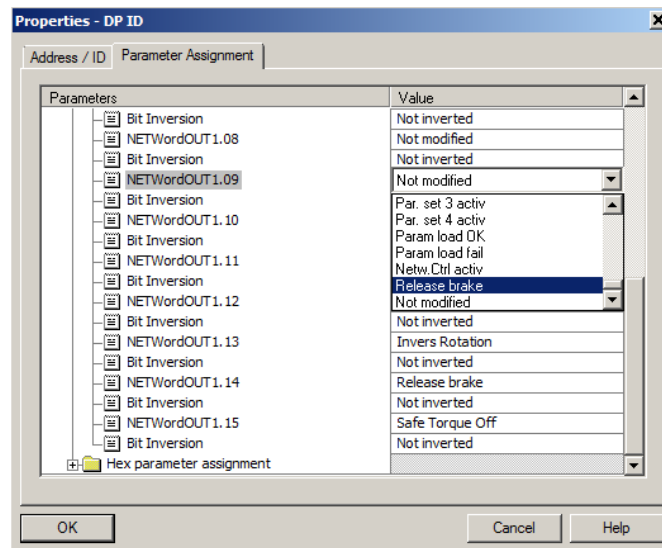
If a different setpoint is to be transmitted, this index must have the "mappable" property. This can be obtained from the parameter description in the documentation.

0x2601	Keypad-Sollwerte			
	001: Frequenz-Sollwert	20.0 Hz	0.0 ... 599.0 Hz	U16 / 10 / P / r
	002: Prozessregler-Sollwert	0.00 PUnit	-300.00 ... 300.00 PUnit	116 / 100 / P / r

The small "r" stands for Receive Mapping permitted.

The status data of the i550 can also be changed in an analog manner in the process image:

Status word:



In addition, each bit of the status word can be inverted individually here. (Shall be implemented for the control word as well).

The bits are assigned to the free outputs (xFreeState_1-4) of the module as follows:

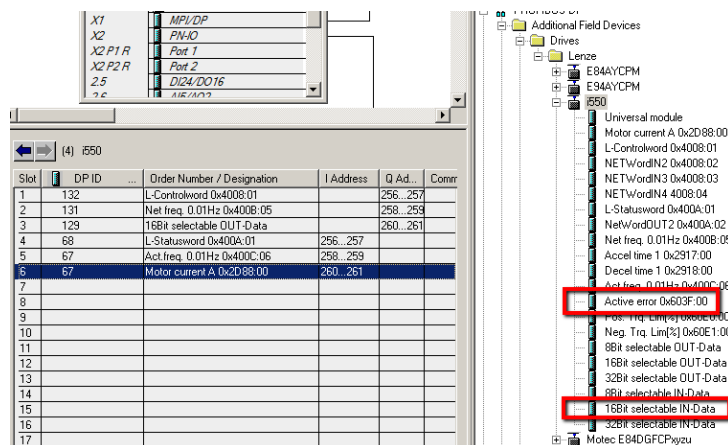
xFreeState_1 <-> NetWordOut1.01

xFreeState_2 <-> NetWordOut1.04

xFreeState_3 <-> NetWordOut1.08

xFreeState_4 <-> NetWordOut1.09

The free status word of the module is assigned to the 3 input word and, by default, assigned to the current motor current in amperes (index 0x2D88:000). Here, a different value could be as well taken from the selection list or a different index can be used via the "16Bit selectable IN-Data" module.



1.2.2 Implementing the "LCB-ActuatorSpeed_V2_1" function block into the control program of the S7

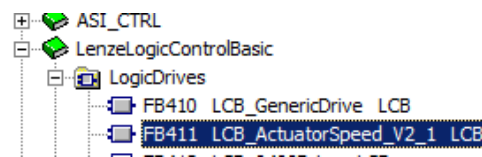
A more detailed functionality description of the module and the input/output data can be obtained from the "S7_Lenze_Application_Sample_V3-0 / V4.0_EN.PDF" documentation; these short instructions only describe the standard procedure.

As the i550 also sends diagnostic messages (e.g. DC-bus undervoltage) to the S7 control, we recommend the implementation of the following organisation blocks into the S7 project. Without these blocks, the control would stop as soon as a diagnostic message appears.

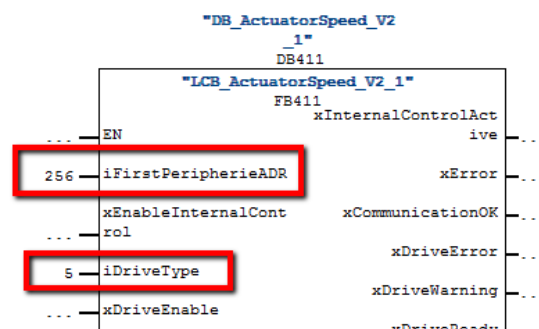
OB80	CYCL_FLT	AWL
OB82	I/O_FLT1	AWL
OB83	I/O_FLT2	AWL
OB86	RACK_FLT	AWL
OB121	PROG_ERR	FUP
OB122	MOD_ERR	FUP

The modules do not need to contain any further program code.

In the example, the required module is inserted from the library into the OB1:



For a first test via the variable table contained in the library, only the following inputs shall be connected:



The start address (iFirstPeripherieADR) can be obtained from the hardware configuration:

Slot	DP ID	...	Order Number / Designation	I Address	Q Address	Comment
1	132		L-Controlword 0x4008:01	256...257		
2	131		Net freq. 0.01Hz 0x400B:05	258...259		
3	129		16Bit selectable OUT-Data	260...261		
4	68		L-Statusword 0x400A:01	256...257		
5	67		Act.freq. 0.01Hz 0x400C:06	258...259		
6	67		Motor current A 0x2D88:00	260...261		
7						

When iDriveType = 5, it is reported to the module that an i550 is triggered.

The other inputs and outputs have to be interconnected in the user program according to the machine function.

Rebuild all and load the modules into the control.

Currently, after initial parameter setting, a "mains switching" has to be executed once for PLC and i550 in order that the parameter setting will be accepted.

Afterwards, the drive can be triggered and tested via the "Variable Table – LCB_ActuatorSpeedIntV2_1" variable table:

Address	Symbol	Display format	Status value	Modify value
DB411 DBX 2.9	'DB_ActuatorSpeed_V2_1'xEnableInternalControl	BOOL	True	
DB411 DBW 0	'DB_ActuatorSpeed_V2_1'xFirstPerphenADR	DEC	256	
DB411 DBW 4	'DB_ActuatorSpeed_V2_1'xDriveType	DEC	5	
DB411 DBX 30.0	'DB_ActuatorSpeed_V2_1'xDriveEnable_IC	BOOL	false	
DB411 DBX 30.1	'DB_ActuatorSpeed_V2_1'xDriveSetOp_IC	BOOL	false	
DB411 DBX 30.2	'DB_ActuatorSpeed_V2_1'xResetError_IC	BOOL	false	
DB411 DBX 30.3	'DB_ActuatorSpeed_V2_1'xEnableSpeedSetpoint_IC	BOOL	false	
DB411 DBX 30.4	'DB_ActuatorSpeed_V2_1'xInvtSet_IC	BOOL	false	
DB411 DBX 30.5	'DB_ActuatorSpeed_V2_1'xInvtSet_IC	BOOL	false	
DB411 DBX 30.6	'DB_ActuatorSpeed_V2_1'xInvtDirectionSet_IC	BOOL	false	
DB411 DBD 32	'DB_ActuatorSpeed_V2_1'xSpeedSetpoint_IC	FLOATING_P...	15.0	15.0
DB411 DBX 36.0	'DB_ActuatorSpeed_V2_1'xFreeCtrl_1_IC	BOOL	false	
DB411 DBX 36.1	'DB_ActuatorSpeed_V2_1'xFreeCtrl_2_IC	BOOL	false	
DB411 DBX 36.2	'DB_ActuatorSpeed_V2_1'xFreeCtrl_3_IC	BOOL	false	
DB411 DBX 36.3	'DB_ActuatorSpeed_V2_1'xFreeCtrl_4_IC	BOOL	false	
DB411 DBW 38	'DB_ActuatorSpeed_V2_1'xFreeCtrl_1_IC	DEC	0	0
DB411 DBX 16.0	'DB_ActuatorSpeed_V2_1'xInternalControlActive	BOOL	True	
DB411 DBX 16.1	'DB_ActuatorSpeed_V2_1'xError	BOOL	false	
DB411 DBX 16.2	'DB_ActuatorSpeed_V2_1'xCommunicationOK	BOOL	True	
DB411 DBX 16.3	'DB_ActuatorSpeed_V2_1'xDriveError	BOOL	false	
DB411 DBX 16.4	'DB_ActuatorSpeed_V2_1'xDriveWarning	BOOL	false	
DB411 DBX 16.5	'DB_ActuatorSpeed_V2_1'xDriveReady	BOOL	True	
DB411 DBX 16.6	'DB_ActuatorSpeed_V2_1'xDriveEnabled	BOOL	false	
DB411 DBX 16.7	'DB_ActuatorSpeed_V2_1'xDriveOpActive	BOOL	false	
DB411 DBX 17.0	'DB_ActuatorSpeed_V2_1'xSpeedEqZero	BOOL	True	
DB411 DBX 17.1	'DB_ActuatorSpeed_V2_1'xDirectionCCW	BOOL	false	
DB411 DBX 26.0	'DB_ActuatorSpeed_V2_1'xFreeState_1	BOOL	false	
DB411 DBX 26.1	'DB_ActuatorSpeed_V2_1'xFreeState_2	BOOL	false	
DB411 DBX 26.2	'DB_ActuatorSpeed_V2_1'xFreeState_3	BOOL	false	
DB411 DBX 26.3	'DB_ActuatorSpeed_V2_1'xFreeState_4	BOOL	false	
DB411 DBD 18	'DB_ActuatorSpeed_V2_1'xSpeedActual	FLOATING_P...	0.0	0.0
DB411 DBW 28	'DB_ActuatorSpeed_V2_1'xFreeState_1	DEC	0	
DB411 DBW 22	'DB_ActuatorSpeed_V2_1'xReadError	HEX	VW16#0000	
DB411 DBW 24	'DB_ActuatorSpeed_V2_1'xWriteError	HEX	VW16#0000	

Setting the "xEnableInternalControl" bit decouples all inputs of the module from the control program and only the setpoint selections of the variable table are effective.

The following control bits have to be set in the variable table so that the drive rotates:

xDriveEnable_IC = Drive enable

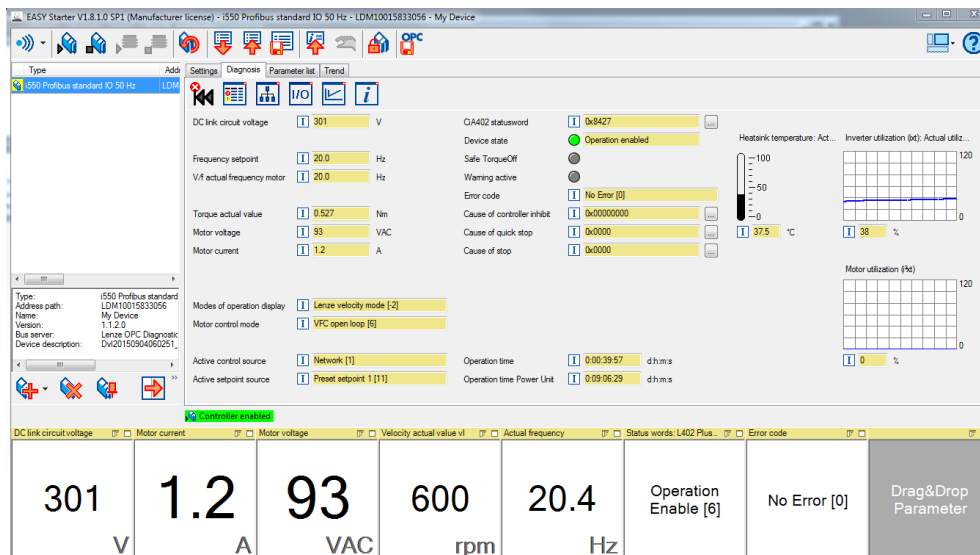
xEnableSpeedSetpoint_IC = setpoint enable, the drive rotates with the setpoint specified in "rSpeedSetpoint_IC". Only positive values are permitted here, inversion of the direction via the xInvertDirectionSet_IC control bit.

LCB_ActuatorSpeedIntV2_1 -- @S7_I550\SIMATIC 300(1)\CPU 314C-2 PN/DP\S7-Programm(1) ONLINE					
	Address	Symbol	Display format	Status value	Modify valu
1	DB411.DBX 2.0	"DB_ActuatorSpeed_V2_1".xEnableInternalControl	BOOL	true	
2	DB411.DBW 0	"DB_ActuatorSpeed_V2_1".iFirstPeripherieADR	DEC	256	
3	DB411.DBW 4	"DB_ActuatorSpeed_V2_1".iDriveType	DEC	5	
4	DB411.DBX 30.0	"DB_ActuatorSpeed_V2_1".xDriveEnable_IC	BOOL	true	
5	DB411.DBX 30.1	"DB_ActuatorSpeed_V2_1".xDriveSetQsp_IC	BOOL	false	
6	DB411.DBX 30.2	"DB_ActuatorSpeed_V2_1".xResetError_IC	BOOL	false	
7	DB411.DBX 30.3	"DB_ActuatorSpeed_V2_1".xEnableSpeedSetpoint_IC	BOOL	true	
8	DB411.DBX 30.4	"DB_ActuatorSpeed_V2_1".xJog1Set_IC	BOOL	true	
9	DB411.DBX 30.5	"DB_ActuatorSpeed_V2_1".xJog2Set_IC	BOOL	false	
10	DB411.DBX 30.6	"DB_ActuatorSpeed_V2_1".xInvertDirectionSet_IC	BOOL	false	
11	DB411.DBD 32	"DB_ActuatorSpeed_V2_1".rSpeedSetpoint_IC	FLOATING_P...	15.0	15.0

Status values:

18	DB411.DBX 16.0	"DB_ActuatorSpeed_V2_1".xInternalControlActive	BOOL	true	
19	DB411.DBX 16.1	"DB_ActuatorSpeed_V2_1".xError	BOOL	false	
20	DB411.DBX 16.2	"DB_ActuatorSpeed_V2_1".xCommunicationOK	BOOL	true	
21	DB411.DBX 16.3	"DB_ActuatorSpeed_V2_1".xDriveError	BOOL	false	
22	DB411.DBX 16.4	"DB_ActuatorSpeed_V2_1".xDriveWarning	BOOL	false	
23	DB411.DBX 16.5	"DB_ActuatorSpeed_V2_1".xDriveReady	BOOL	false	
24	DB411.DBX 16.6	"DB_ActuatorSpeed_V2_1".xDriveEnabled	BOOL	true	
25	DB411.DBX 16.7	"DB_ActuatorSpeed_V2_1".xDriveQspActive	BOOL	false	
26	DB411.DBX 17.0	"DB_ActuatorSpeed_V2_1".xSpeedEqZero	BOOL	false	
27	DB411.DBX 17.1	"DB_ActuatorSpeed_V2_1".xDirectionCCW	BOOL	false	
28	DB411.DBX 26.0	"DB_ActuatorSpeed_V2_1".xFreeState_1	BOOL	false	
29	DB411.DBX 26.1	"DB_ActuatorSpeed_V2_1".xFreeState_2	BOOL	false	
30	DB411.DBX 26.2	"DB_ActuatorSpeed_V2_1".xFreeState_3	BOOL	false	
31	DB411.DBX 26.3	"DB_ActuatorSpeed_V2_1".xFreeState_4	BOOL	false	
32	DB411.DBD 18	"DB_ActuatorSpeed_V2_1".rSpeedActual	FLOATING_P...	20.0	
33	DB411.DBW 28	"DB_ActuatorSpeed_V2_1".wFreeState_1	DEC	12	
34	DB411.DBW 22	"DB_ActuatorSpeed_V2_1".ReadError	HEX	W#16#0000	
35	DB411.DBW 24	"DB_ActuatorSpeed_V2_1".WriteError	HEX	W#16#0000	

Status in the EASY Starter:



1.3 Integration under the Siemens TIA portal

The integration under TIA requires the GSE file of the i550. This file has to be integrated into the hardware catalog of the TIA portal via the hardware manager.

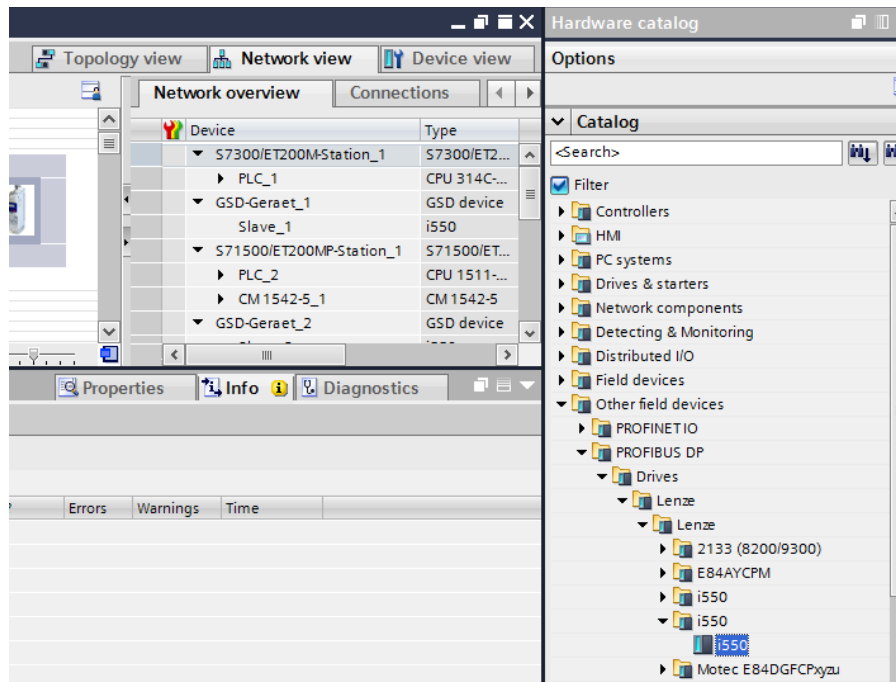
In order to use the "LCB_ActuatorSpeed_V2_1" module from the Lenze sample library, it still needs to be integrated.

1.3.1 Hardware configuration in the TIA portal with S7-300 control

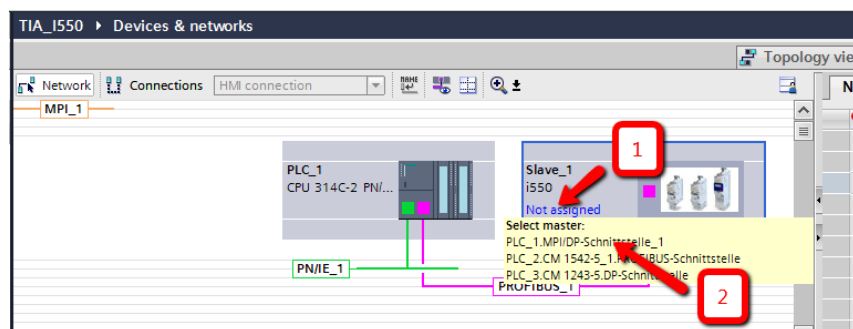
i550 Siemens Integration V2.0

Select the network view in the device configuration of the TIA portal

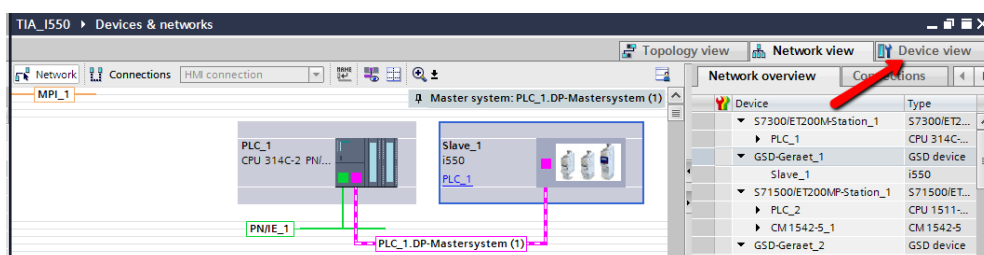
Selection of the i550 in the hardware catalog under "Other Field Devices\Drives\Lenze".



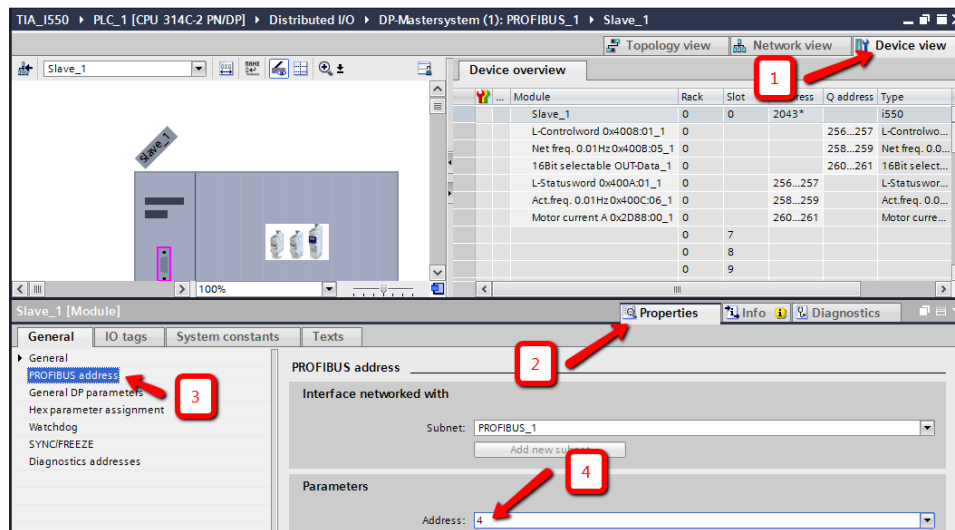
If the i550 is moved into the net view on the Profibus per "Drag and Drop", the Profibus connection to the PLC is shown but the i550 is not assigned to the PLC. This is done by mouse-clicking the "Not assigned" text in the symbol for the i550. Then select the wanted PLC in the window that opens.



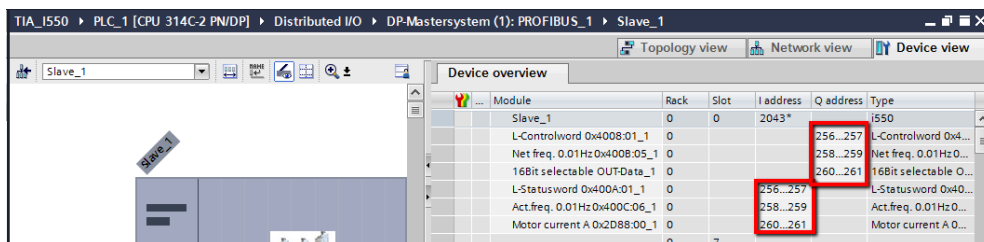
Only mark the i550 and change to the device view:



First set the Profibus address in the device features, which is address 4 in the sample project:



As a standard assignment is saved in the GSE file of the i550, the standard IO configuration of the process data has already been created. It does not need to be created manually anymore but can be edited or extended.



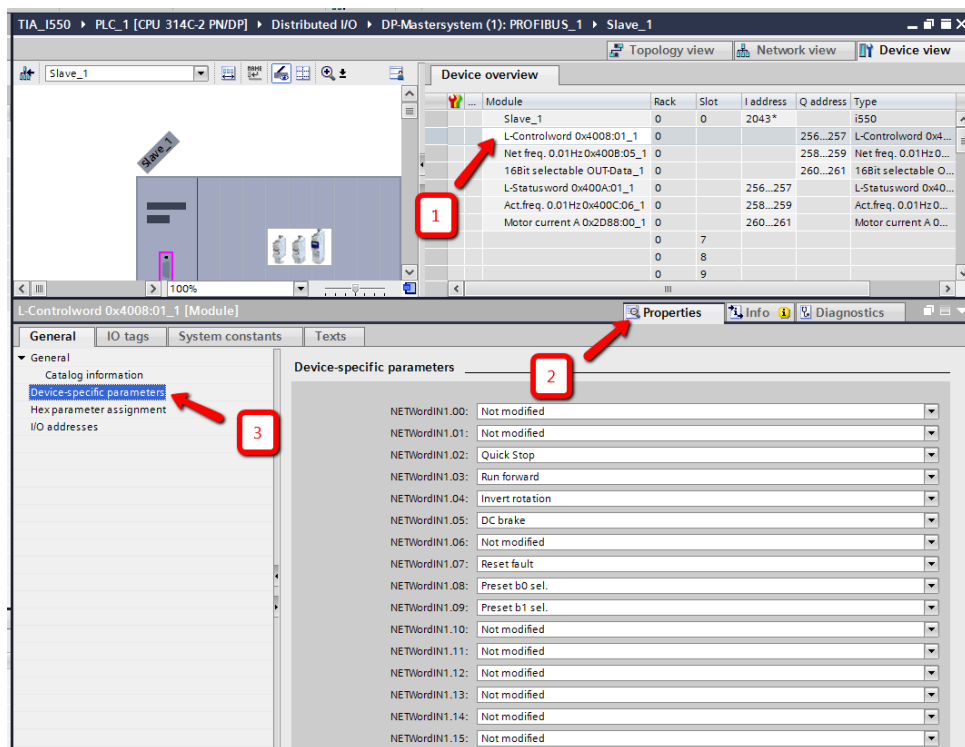
The "LCB_ActuatorSpeed_V2_1" requires a data length of 3 words each of input data and output data. In order that the module operates correctly, you have to make sure that the addresses follow each other directly without any jumps and the start addresses of the I/O ranges have to be identical (256). The basic configuration for the basic functionality does not need to be adapted anymore. Individual adaptations, however, are possible.

1.3.1.1 Adjusting the process data mapping

The process data mapping can only be changed on the control side in the hardware manager. At every restart, the Siemens control writes this down to the i550 via Profibus.

The assignment of the free input bits (xFreeCtrl_1-4) of the Lenze FB can be directly changed in the properties dialog of the device view in the control word.

Mark the control word and select Device-specific parameters in Properties:



The free control bits are assigned as follows:

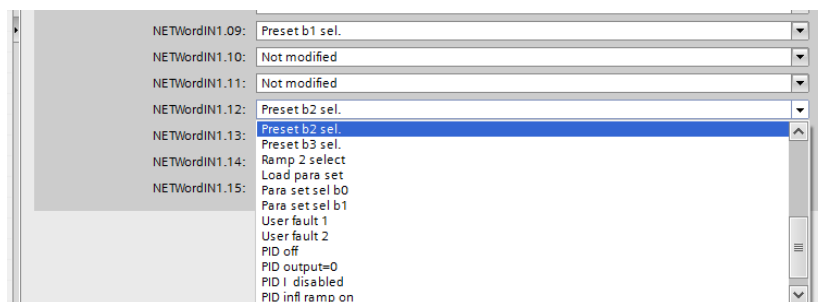
xFreeCtrl_1 <-> NetWordIn1.11

xFreeCtrl_2 <-> NetWordIn1.13

xFreeCtrl_3 <-> NetWordIn1.14

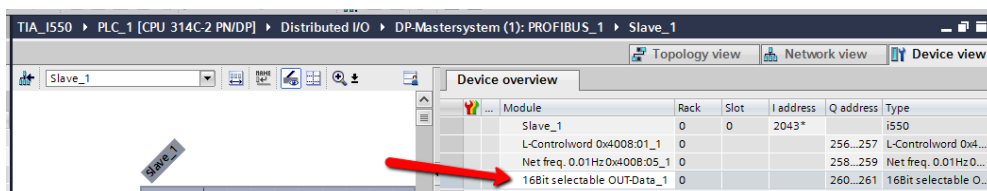
xFreeCtrl_4 <-> NetWordIn1.15

For every single bit, a new function can be assigned via the drop-down menu:

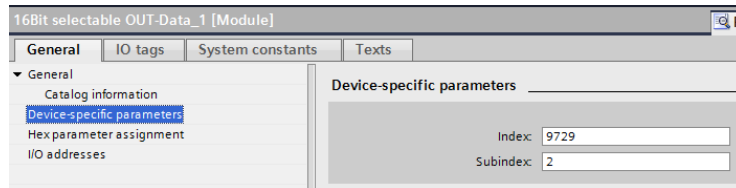


The free input word wFreeCtrl_1 of the module is linked to the 3 output word of the process image (16Bit selectable Out-Data) and can also only be changed in the properties dialog of the device view on the S7 side:

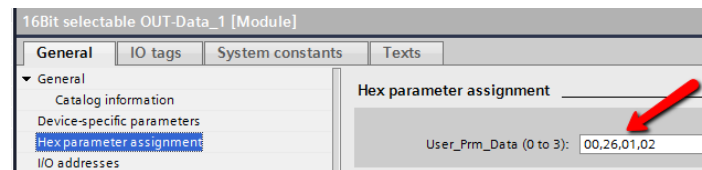
In the properties dialog you can select which index is to be written into the controller:



Under Device-specific Parameters, the index and subindex are shown in decimal form,



Hex parameter assignment directly shows the HEX index: (0x2601:02 default assignment):



Address	Display code	Name	Value	Unit
0x2601:000	PAR0202:000	highest sub-index supported (generated)	3	
0x2601:001	PAR0202:001	Keypad setpoint commands: Frequenc...	20.0	Hz
0x2601:002	PAR0202:002	Keypad setpoint commands: Process c...	0.00	PUnit
0x2601:003	PAR0202:003	Keypad setpoint commands: Torque s...	100.0	%

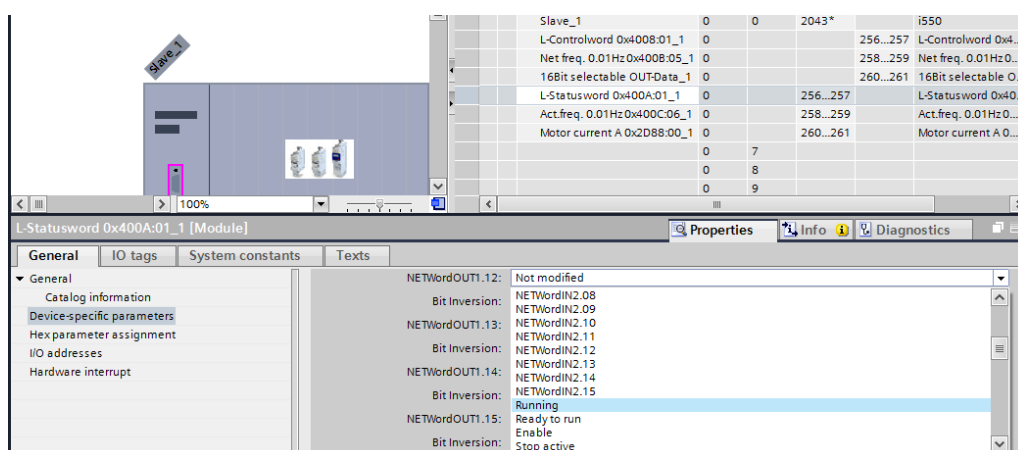
If a different setpoint is to be transmitted, this index must have the "mappable" property. This can be obtained from the parameter description in the documentation.

0x2601	Keypad-Sollwerte			
001:	Frequenz-Sollwert	20.0 Hz	0.0 ... 599.0 Hz	U16 / 10 / P / r
002:	Prozessregler-Sollwert	0.00 PUnit	-300.00 ... 300.00 PUnit	I16 / 100 / P / r

The small "r" stands for Receive Mapping permitted.

The status data of the i550 can also be changed in an analog manner in the process image:

Status word:



In addition, each bit of the status word can be inverted individually here. (Shall be implemented for the control word as well).

The bits are assigned to the free outputs (xFreeState_1-4) of the module as follows:

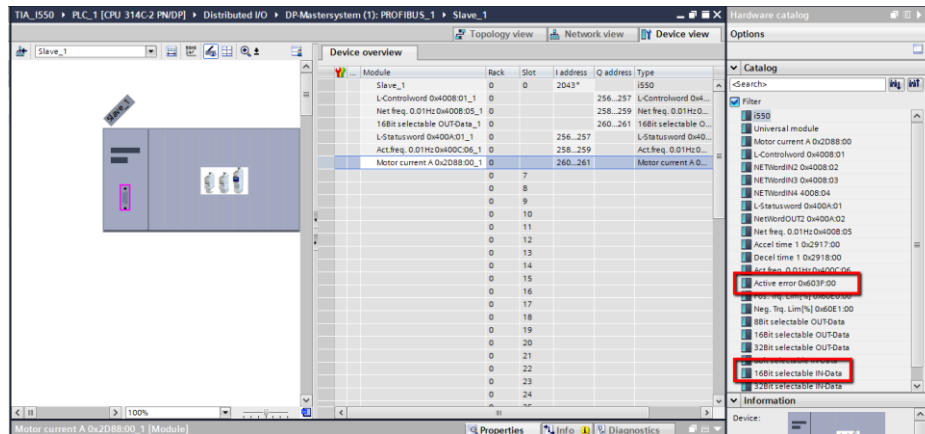
xFreeState_1 <-> NetWordOut1.01

xFreeState_2 <-> NetWordOut1.04

xFreeState_3 <-> NetWordOut1.08

xFreeState_4 <-> NetWordOut1.09

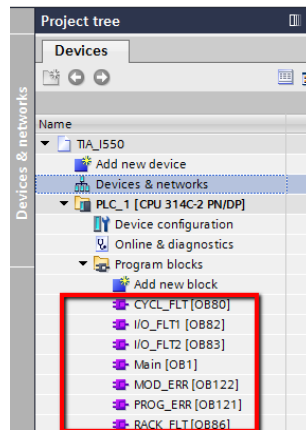
The free status word of the module is assigned to the 3 input word and, by default, assigned to the current motor current in amperes (index 0x2D88:000). Here, a different value could be as well taken from the selection list or a different index can be used via the "16Bit selectable IN-Data" module.



1.3.2 Implementing the "LCB-ActuatorSpeed_V2_1" function block into the control program of the S7

A more detailed functionality description of the module and the input/output data can be obtained from the "S7_Lenze_Application_Sample_V3-0_DE.PDF" documentation; this short instructions only describe the standard procedure.

As the i550 also sends diagnostic messages (e.g. DC-bus undervoltage) to the S7 control, we recommend the implementation of the following organisation blocks into the S7 project. Without these blocks, the control would stop as soon as a diagnostic message appears.



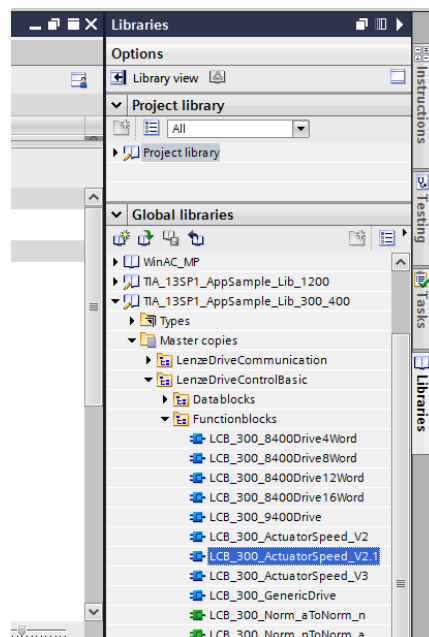
The modules do not need to contain any further program code.

Copy the following module copy templates into the project from the PLC type corresponding library:

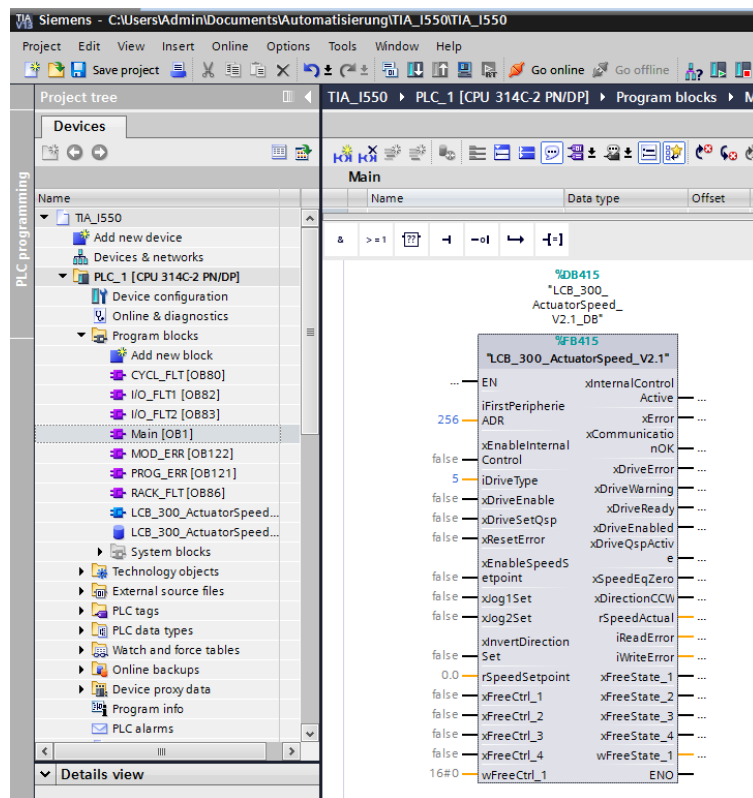
„LCB_300_ActuatorSpeed_V2.1“

„LCB_300_ActuatorSpeed_V2.1_DB“

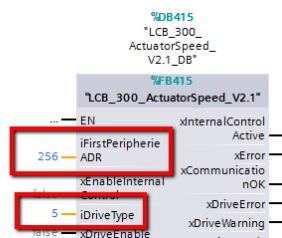
and the "LCB_300_Actuator_Speed_V2.1" monitoring table for controlling the drive.



In the example, the required module is inserted from the library into the OB1:



For a first test via the watch table contained in the library, only the following inputs shall be connected:



When iDriveType = 5, it is reported to the module that an i550 is triggered.

The start address (iFirstPeripherieADR) can be obtained from the hardware configuration:

Device overview						
	Module	Rack	Slot	I address	Q address	Type
	Slave_1	0	0	2043*		i550
	L-Controlword 0x4008:01_1	0			256...257	L-Controlword 0x4...
	Net freq. 0.01Hz 0x400B:05_1	0			258...259	Net freq. 0.01Hz 0x...
	16Bit selectable OUT-Data_1	0			260...261	16Bit selectable OU...
	L-Statusword 0x400A:01_1	0		256...257		L-Statusword 0x40...
	Act.freq. 0.01Hz 0x400C:06_1	0		258...259		Act.freq. 0.01Hz 0x...
	Motor current A 0x2D88:00_1	0		260...261		Motor current A 0x...
		0	7			

The other inputs and outputs have to be interconnected in the user program according to the machine function.

Rebuild all and load the modules into the control.

Currently, after initial parameter setting, a "mains switching" has to be executed once for PLC and i550 in order that the parameter setting will be accepted.

Afterwards, the drive can be triggered and tested via the "VAR – LCB_ActuatorSpeedIntV2_1" variable table:

TIA_I550 > PLC_1 [CPU 314C-2 PN/DP] > Watch and force tables > LCB_300_Actuator_Speed_V2.1					
	Name	Address	Display format	Monitor value	Modify value
1	"LCB_300_ActuatorSpeed_V2.1_DB".xEnableInternalControl	%DB415.DBX2.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
2	"LCB_300_ActuatorSpeed_V2.1_DB".iFirstPeripherieADR	%DB415.DBW0	DEC+/-	256	
3	"LCB_300_ActuatorSpeed_V2.1_DB".iDriveType	%DB415.DBW4	DEC+/-	5	
4	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveEnable_IC	%DB415.DBX32.0	Bool	<input type="checkbox"/> FALSE	FALSE
5	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveSetQsp_IC	%DB415.DBX32.1	Bool	<input type="checkbox"/> FALSE	FALSE
6	"LCB_300_ActuatorSpeed_V2.1_DB".xResetError_IC	%DB415.DBX32.2	Bool	<input type="checkbox"/> FALSE	FALSE
7	"LCB_300_ActuatorSpeed_V2.1_DB".xEnableSpeedSetpoint_IC	%DB415.DBX32.3	Bool	<input type="checkbox"/> FALSE	FALSE
8	"LCB_300_ActuatorSpeed_V2.1_DB".xlog1Set_IC	%DB415.DBX32.4	Bool	<input type="checkbox"/> FALSE	FALSE
9	"LCB_300_ActuatorSpeed_V2.1_DB".xlog2Set_IC	%DB415.DBX32.5	Bool	<input type="checkbox"/> FALSE	FALSE
10	"LCB_300_ActuatorSpeed_V2.1_DB".xInvertDirectionSet_IC	%DB415.DBX32.6	Bool	<input type="checkbox"/> FALSE	FALSE
11	"LCB_300_ActuatorSpeed_V2.1_DB".rSpeedSetpoint_IC	%DB415.DBX34	Floating-point nu...	10.0	10.0
12	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_1_IC	%DB415.DBX38.0	Bool	<input type="checkbox"/> FALSE	FALSE
13	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_2_IC	%DB415.DBX38.1	Bool	<input type="checkbox"/> FALSE	FALSE
14	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_3_IC	%DB415.DBX38.2	Bool	<input type="checkbox"/> FALSE	FALSE
15	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_4_IC	%DB415.DBX38.3	Bool	<input type="checkbox"/> FALSE	FALSE
16	"LCB_300_ActuatorSpeed_V2.1_DB".wFreeCtrl_1_IC	%DB415.DBW40	DEC	0	
17	"LCB_300_ActuatorSpeed_V2.1_DB".wOUT	%DB415.DBW56	Hex	16#0000	
18	"LCB_300_ActuatorSpeed_V2.1_DB".xInternalControlActive	%DB415.DBX16.0	Bool	<input checked="" type="checkbox"/> TRUE	
19	"LCB_300_ActuatorSpeed_V2.1_DB".xError	%DB415.DBX16.1	Bool	<input type="checkbox"/> FALSE	
20	"LCB_300_ActuatorSpeed_V2.1_DB".xCommunicationOK	%DB415.DBX16.2	Bool	<input checked="" type="checkbox"/> TRUE	
21	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveError	%DB415.DBX16.3	Bool	<input type="checkbox"/> FALSE	
22	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveWarning	%DB415.DBX16.4	Bool	<input type="checkbox"/> FALSE	
23	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveReady	%DB415.DBX16.5	Bool	<input checked="" type="checkbox"/> TRUE	
24	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveEnabled	%DB415.DBX16.6	Bool	<input type="checkbox"/> FALSE	
25	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveQspActive	%DB415.DBX16.7	Bool	<input type="checkbox"/> FALSE	
26	"LCB_300_ActuatorSpeed_V2.1_DB".xSpeedEqZero	%DB415.DBX17.0	Bool	<input checked="" type="checkbox"/> TRUE	
27	"LCB_300_ActuatorSpeed_V2.1_DB".xDirectionCCW	%DB415.DBX17.1	Bool	<input type="checkbox"/> FALSE	
28	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeState_1	%DB415.DBX26.0	Bool	<input type="checkbox"/> FALSE	TRUE
29	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeState_2	%DB415.DBX26.1	Bool	<input type="checkbox"/> FALSE	TRUE
30	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeState_3	%DB415.DBX26.2	Bool	<input type="checkbox"/> FALSE	
31	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeState_4	%DB415.DBX26.3	Bool	<input type="checkbox"/> FALSE	
32	"LCB_300_ActuatorSpeed_V2.1_DB".rSpeedActual	%DB415.DBX18	Floating-point nu...	0.0	
33	"LCB_300_ActuatorSpeed_V2.1_DB".wFreeState_1	%DB415.DBW28	DEC	0	
34	"LCB_300_ActuatorSpeed_V2.1_DB".iReadError	%DB415.DBW22	DEC+/-	0	
35	"LCB_300_ActuatorSpeed_V2.1_DB".iWriteError	%DB415.DBW24	DEC+/-	0	

Setting the "xEnableInternalControl" bit decouples all inputs of the module from the control program and only the setpoint selections of the variable table are effective.















The following control bits have to be set in the variable table so that the drive rotates:

xDriveEnable_IC = Drive enable

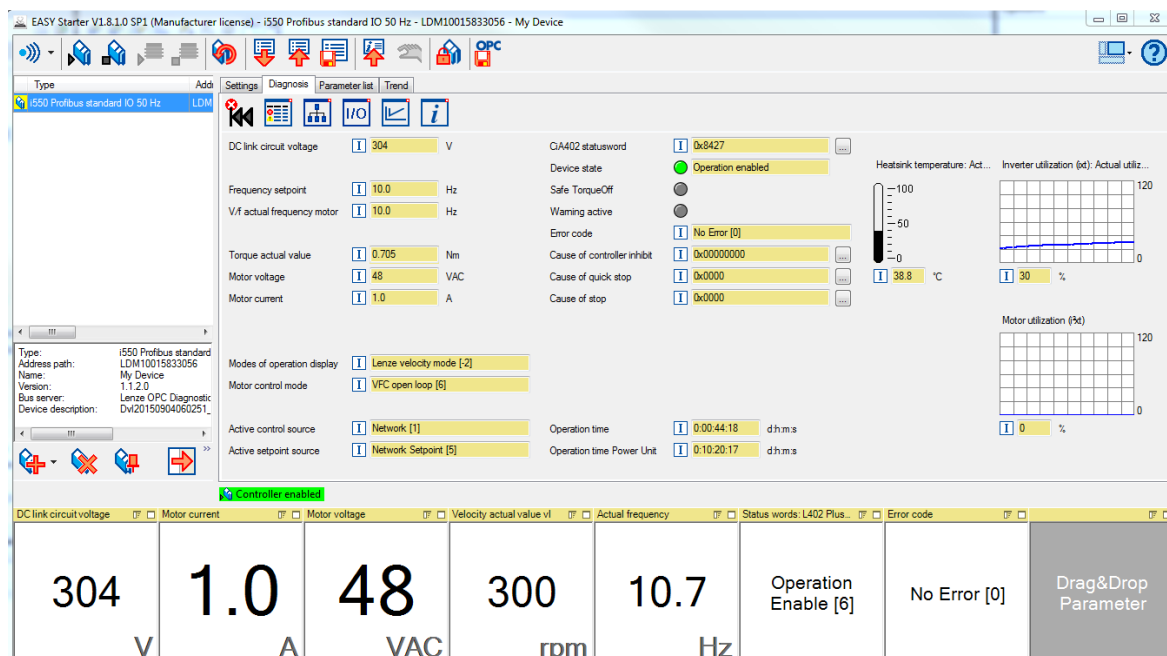
xEnableSpeedSetPoint_IC = setpoint enable, the drive rotates with the setpoint specified in "rSpeedSetpoint_IC". Only positive values are permitted here, inversion of the direction via the xInvertDirectionSet_IC control bit.

TIA_I550 > PLC_1 [CPU 314C-2 PN/DP] > Watch and force tables > LCB_300_Actuator_Speed_V2.1					
	Name	Address	Display format	Monitor value	Modify value
1	"LCB_300_ActuatorSpeed_V2.1_DB".xEnableInternalControl	%DB415.DBX2.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
2	"LCB_300_ActuatorSpeed_V2.1_DB".iFirstPeripherieADR	%DB415.DBW0	DEC+/-	256	
3	"LCB_300_ActuatorSpeed_V2.1_DB".iDriveType	%DB415.DBW4	DEC+/-	5	
4	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveEnable_IC	%DB415.DBX32.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
5	"LCB_300_ActuatorSpeed_V2.1_DB".xDriveSetQsp_IC	%DB415.DBX32.1	Bool	<input type="checkbox"/> FALSE	FALSE
6	"LCB_300_ActuatorSpeed_V2.1_DB".xResetError_IC	%DB415.DBX32.2	Bool	<input type="checkbox"/> FALSE	FALSE
7	"LCB_300_ActuatorSpeed_V2.1_DB".xEnableSpeedSetpoint_IC	%DB415.DBX32.3	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
8	"LCB_300_ActuatorSpeed_V2.1_DB".xlog1Set_IC	%DB415.DBX32.4	Bool	<input type="checkbox"/> FALSE	FALSE
9	"LCB_300_ActuatorSpeed_V2.1_DB".xlog2Set_IC	%DB415.DBX32.5	Bool	<input type="checkbox"/> FALSE	FALSE
10	"LCB_300_ActuatorSpeed_V2.1_DB".xInvertDirectionSet_IC	%DB415.DBX32.6	Bool	<input type="checkbox"/> FALSE	FALSE
11	"LCB_300_ActuatorSpeed_V2.1_DB".rSpeedSetpoint_IC	%DB415.DBX34	Floating-point nu...	10.0	10.0
12	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_1_IC	%DB415.DBX38.0	Bool	<input type="checkbox"/> FALSE	FALSE
13	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_2_IC	%DB415.DBX38.1	Bool	<input type="checkbox"/> FALSE	FALSE
14	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_3_IC	%DB415.DBX38.2	Bool	<input type="checkbox"/> FALSE	FALSE
15	"LCB_300_ActuatorSpeed_V2.1_DB".xFreeCtrl_4_IC	%DB415.DBX38.3	Bool	<input type="checkbox"/> FALSE	FALSE
16	"LCB_300_ActuatorSpeed_V2.1_DB".wFreeCtrl_1_IC	%DB415.DBW40	DEC	0	

Status values:

18	*LCB_300_ActuatorSpeed_V2.1_DB*.xInternalControlActive	%DB415.DBX16.0	Bool		TRUE	
19	*LCB_300_ActuatorSpeed_V2.1_DB*.xError	%DB415.DBX16.1	Bool		FALSE	
20	*LCB_300_ActuatorSpeed_V2.1_DB*.xCommunicationOK	%DB415.DBX16.2	Bool		TRUE	
21	*LCB_300_ActuatorSpeed_V2.1_DB*.xDriveError	%DB415.DBX16.3	Bool		FALSE	
22	*LCB_300_ActuatorSpeed_V2.1_DB*.xDriveWarning	%DB415.DBX16.4	Bool		FALSE	
23	*LCB_300_ActuatorSpeed_V2.1_DB*.xDriveReady	%DB415.DBX16.5	Bool		FALSE	
24	*LCB_300_ActuatorSpeed_V2.1_DB*.xDriveEnabled	%DB415.DBX16.6	Bool		TRUE	
25	*LCB_300_ActuatorSpeed_V2.1_DB*.xDriveQspActive	%DB415.DBX16.7	Bool		FALSE	
26	*LCB_300_ActuatorSpeed_V2.1_DB*.xSpeedEqZero	%DB415.DBX17.0	Bool		FALSE	
27	*LCB_300_ActuatorSpeed_V2.1_DB*.xDirectionCCW	%DB415.DBX17.1	Bool		FALSE	
28	*LCB_300_ActuatorSpeed_V2.1_DB*.xFreeState_1	%DB415.DBX26.0	Bool		FALSE	TRUE
29	*LCB_300_ActuatorSpeed_V2.1_DB*.xFreeState_2	%DB415.DBX26.1	Bool		FALSE	TRUE
30	*LCB_300_ActuatorSpeed_V2.1_DB*.xFreeState_3	%DB415.DBX26.2	Bool		FALSE	
31	*LCB_300_ActuatorSpeed_V2.1_DB*.xFreeState_4	%DB415.DBX26.3	Bool		FALSE	
32	*LCB_300_ActuatorSpeed_V2.1_DB*.rSpeedActual	%DB415.DBX18	Floating-point nu...		10.0	
33	*LCB_300_ActuatorSpeed_V2.1_DB*.wFreeState_1	%DB415.DBW28	DEC		10	
34	*LCB_300_ActuatorSpeed_V2.1_DB*.iReadError	%DB415.DBW22	DEC+/-		0	
35	*LCB_300_ActuatorSpeed_V2.1_DB*.iWriteError	%DB415.DBW24	DEC+/-		0	

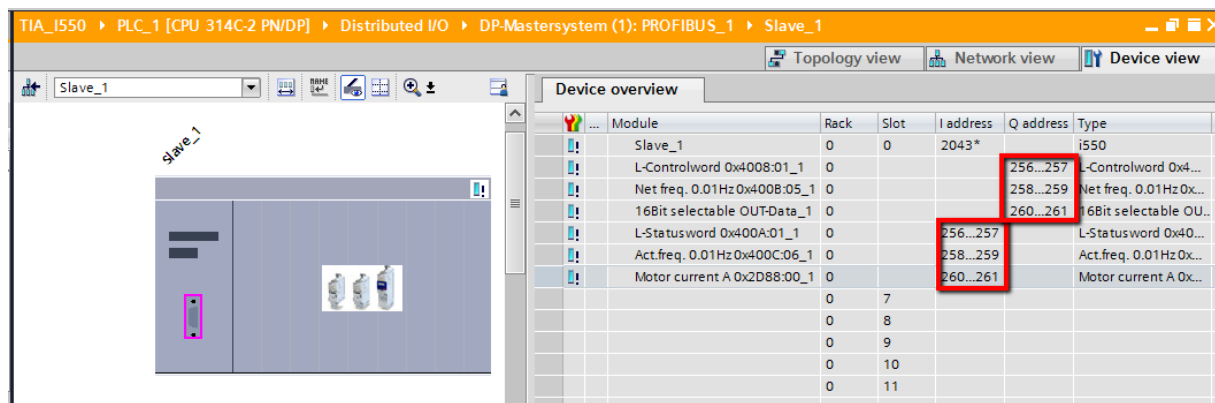
Status in the EASY Starter:



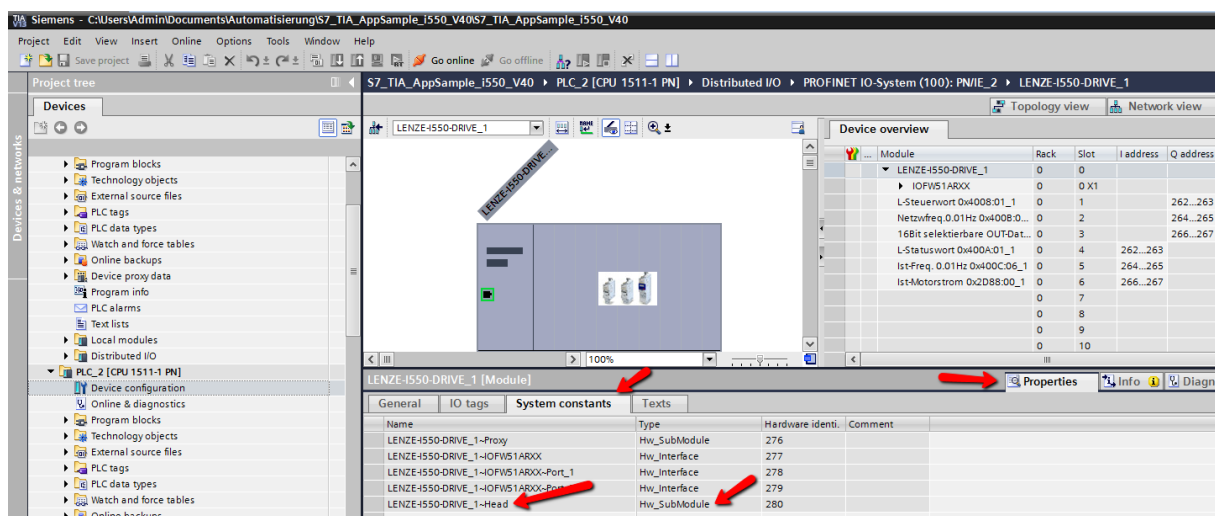
1.3.3 Special features of S7-1200 and S7-1500 control

The basic configuration is identical to the one of a control of type S7-300/400. However, the process data are normally not addressed directly via addresses anymore but via the "HW identification". As the system assigns HW identifications arbitrarily, the i550 cannot operate with them.

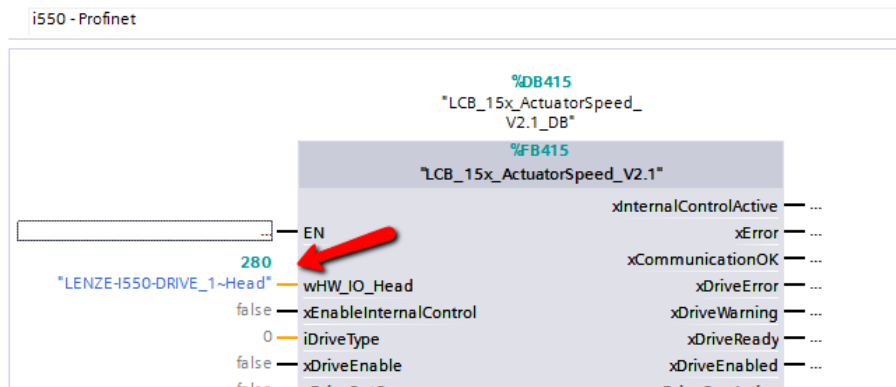
Therefore, you have to make sure in case of the i550 that the process data addresses have the same start address (256) and directly follow each other without any gaps or jumps:



All Functionblocks in the Lenze Library's for S7-1200 and S7-1500 use the „HEAD Hardware-Identifier“ to address the drives:



Then the start address "HW_IO_HEAD" (280) is given at the LCB_15x_ActuatorSpeed_V2.1 FB.



As a consistent access with the Siemens functions SFC14 and SFC15 is not possible via 3 slots, the 3 slots are individually evaluated in the Lenze FB. If an error occurs, it is entered in the diagnostic messages of the CPU as peripheral access error. The "xCommunicationOK" output bit of the Lenze FB has no function in combination with i550.

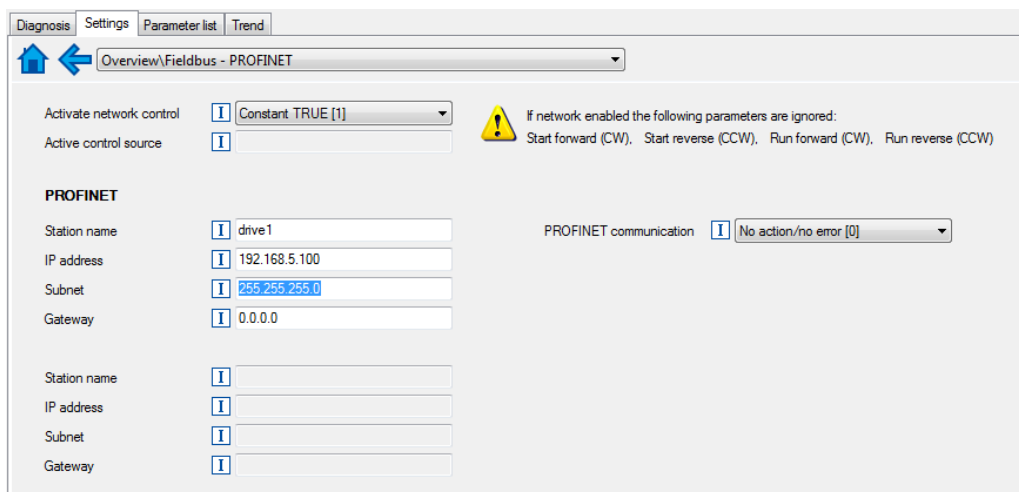
2 i550 on Profinet

2.1 Parameter setting with the EASY Starter

For the general control via Profinet, only a few codes have to be set at the i550 via the EASY Starter. The basic drive parameters such as base frequency or maximum current are also parameterised via the EASY Starter.

2.1.1 Setting of the Profinet address and stationname

The Profinet station name and IP address should be set at the setting menu "Actuating Speed Overview\Fieldbus Setup\PROFINET(Index 2381:001, 2381:002, 2381:004).



The screenshot shows the 'Settings' tab in the EASY Starter software. The main menu is 'Overview\Fieldbus - PROFINET'. The 'Activate network control' is set to 'Constant TRUE [1]'. The 'Active control source' is set to 'I'. A warning icon indicates that if network is enabled, the following parameters are ignored: Start forward (CW), Start reverse (CCW), Run forward (CW), Run reverse (CCW). The 'PROFINET' section contains the following fields:

Field	Value
Station name	drive1
IP address	192.168.5.100
Subnet	255.255.255.0
Gateway	0.0.0.0
Station name	
IP address	
Subnet	
Gateway	

The 'PROFINET communication' is set to 'No action/no error [0]'.

Further necessary settings in the Easy Starter (see chapter 1.1.3) and on the Siemens PLC side are the same like Profibus.

3 Parameter data communication via DP-V1

The parameter data communication works the same way as in case of 8400 and 9400. Both via Profibus and Profinet. In case of i550, however, the internal parameter structure has never been changed. There are no array parameters anymore, which is a code with x subcodes of the same type (e.g.: Jog speed, ramps). The i550 still has subcodes, but these have not been created as an array, thus each subcode has to be read out/written separately.

The following function blocks from the Lenze library can be used with the i550:

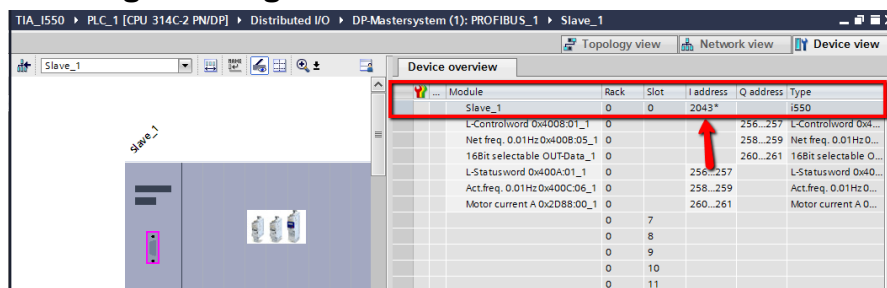
- ReadDriveParameter – reading of a single parameter
- WriteDriveParameter – writing of a single parameter
- ReadDriveParamString – reading a string parameter
- R_W_n_DrivePar – reading / writing of up to 32 parameters

All modules for arrays cannot be used.

3.1 Example: Reading / writing a parameter with an S7-1500 CPU

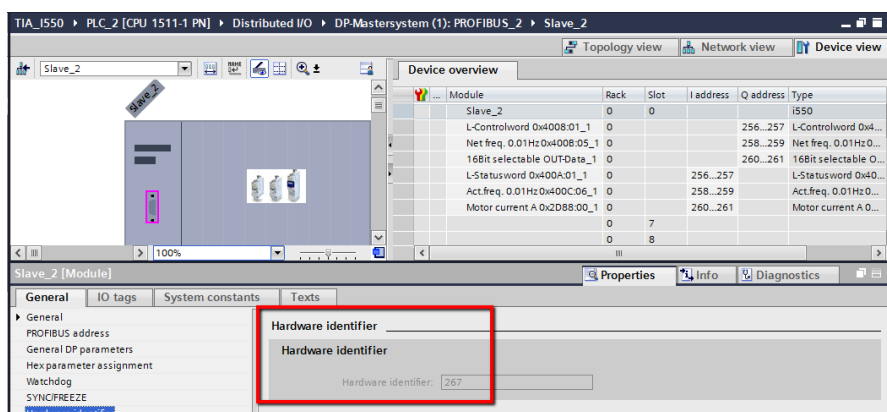
Furthermore, the code names for the i550 have been changed, the old Lenze numbering is not valid anymore. The HEX indices are indicated directly, thus the "xUseIndexAsParam" input at the modules has to be set to FALSE and the index has to be directly transferred in the hex format.

3.1.1 Determining the diagnostics address in case of an S7-300



The diagnostics address of the i550 slave is 2043.


3.1.2 Determining the HW identification for the parameter transfer for S7-1200 and S71500



HW identification (diagnostic address) of the i550 slave is 267.

In the example, the index 0x2916 subindex 0 (maximum frequency) is read and written,

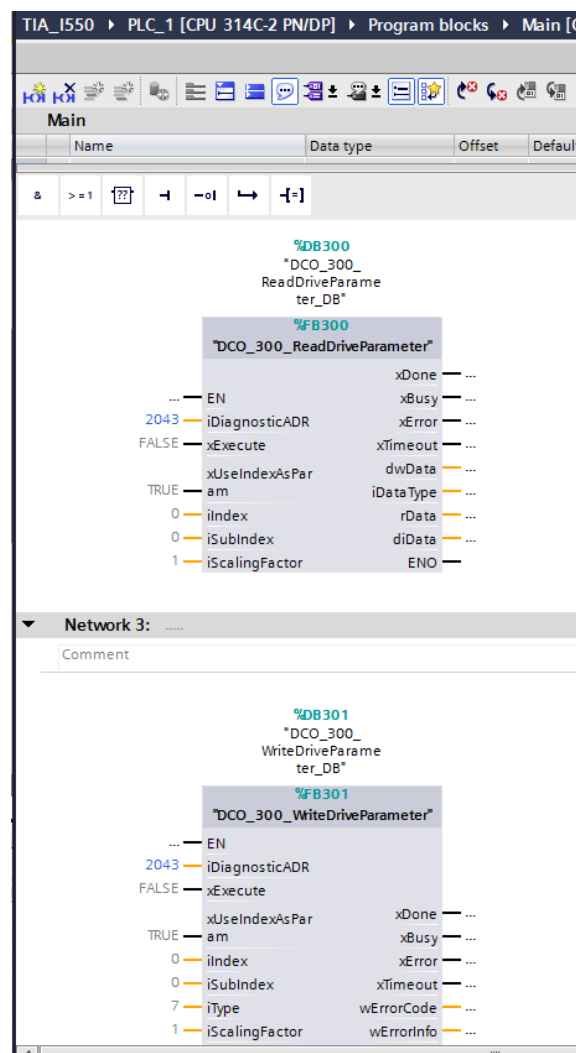
data type (6 = Unsigned Integer) and scaling factor (10) can be obtained from the parameter attribute list from the "reference manual i550 Cabinet":



Anhang
Parameter-Attributliste

Adresse	Name / Subindex: Name	Voreinstellung	Einstellbereich	D / F / A / M
0x2916	Maximalfrequenz	50.0 Hz	0.0 ... 599.0 Hz	U16 / 10 / P / -

3.1.3 Implementing the modules into the PLC program



Reading the parameter with the monitoring table:

TIA_I550 > PLC_1 [CPU 314C-2 PN/DP] > Watch and force tables > DCO_300_Read/WriteDrivePara

	Name	Address	Display format	Monitor value	Modify value
1	"DCO_300_ReadDriveParameter_DB".iDiagnosticADR	%DB300.DBW0	DEC+/-	2043	2043
2	"DCO_300_ReadDriveParameter_DB".xExecute	%DB300.DBX2.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
3	"DCO_300_ReadDriveParameter_DB".iIndex	%DB300.DBW4	Hex	16#2916	16#2916
4	"DCO_300_ReadDriveParameter_DB".iSubIndex	%DB300.DBW6	DEC+/-	0	0
5	"DCO_300_ReadDriveParameter_DB".iUseIndexAsParam	%DB300.DBX2.1	Bool	<input type="checkbox"/> FALSE	FALSE
6	"DCO_300_ReadDriveParameter_DB".iScalingFactor	%DB300.DBW8	DEC+/-	10	10
7					
8	"DCO_300_ReadDriveParameter_DB".xDone	%DB300.DBX10.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
9	"DCO_300_ReadDriveParameter_DB".xBusy	%DB300.DBX10.1	Bool	<input type="checkbox"/> FALSE	FALSE
10	"DCO_300_ReadDriveParameter_DB".xError	%DB300.DBX10.2	Bool	<input type="checkbox"/> FALSE	FALSE
11	"DCO_300_ReadDriveParameter_DB".dwData	%DB300.DBD12	Hex	16#0000_01F4	
12	"DCO_300_ReadDriveParameter_DB".iDataType	%DB300.DBW16	DEC+/-	6	
13	"DCO_300_ReadDriveParameter_DB".rData	%DB300.DBD18	Floating-poi...	50.0	
14	"DCO_300_ReadDriveParameter_DB".dData	%DB300.DBD22	DEC+/-	500	

Settings | Diagnosis | Parameter list | Trend

Actuating Speed Overview

Basic Setup

Rated mains voltage: Minimum frequency: Hz

Rotation mode: Maximum frequency: Hz

Writing the parameter with the monitoring table:

17	"DCO_300_WriteDriveParameter_DB".iDiagnosticADR	%DB301.DBW0	DEC+/-	2043	2043
18	"DCO_300_WriteDriveParameter_DB".xExecute	%DB301.DBX2.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
19	"DCO_300_WriteDriveParameter_DB".iUseIndexAsParam	%DB301.DBX2.1	Bool	<input type="checkbox"/> FALSE	FALSE
20	"DCO_300_WriteDriveParameter_DB".iIndex	%DB301.DBW4	Hex	16#2916	16#2916
21	"DCO_300_WriteDriveParameter_DB".iSubIndex	%DB301.DBW6	DEC+/-	0	0
22	"DCO_300_WriteDriveParameter_DB".iType	%DB301.DBW8	DEC+/-	6	6
23	"DCO_300_WriteDriveParameter_DB".iScalingFactor	%DB301.DBW10	DEC+/-	10	10
24	"DCO_300_WriteDriveParameter_DB".rValue	%DB301.DBD12	Floating-poi...	60.0	60.0
25	"DCO_300_WriteDriveParameter_DB".scWriteRecord.Paramet...	%DB301.DBD70	Hex	16#0258_0000	
26					
27	"DCO_300_WriteDriveParameter_DB".xDone	%DB301.DBX16.0	Bool	<input checked="" type="checkbox"/> TRUE	TRUE
28	"DCO_300_WriteDriveParameter_DB".xBusy	%DB301.DBX16.1	Bool	<input type="checkbox"/> FALSE	FALSE
29	"DCO_300_WriteDriveParameter_DB".xError	%DB301.DBX16.2	Bool	<input type="checkbox"/> FALSE	FALSE
30	"DCO_300_WriteDriveParameter_DB".scReadRecord.Response	%DB301.DBB109	Hex	16#02	
31	"DCO_300_WriteDriveParameter_DB".wErrorCode	%DB301.DBW18	Hex	16#0000	
32	"DCO_300_WriteDriveParameter_DB".wErrorInfo	%DB301.DBW20	Hex	16#0000	
33	"DCO_300_WriteDriveParameter_DB".scReadRecord.Format01	%DB301.DBB112	Hex	16#00	

Settings | Diagnosis | Parameter list | Trend

Actuating Speed Overview

Basic Setup

Rated mains voltage: Minimum frequency: Hz

Rotation mode: Maximum frequency: Hz

Further information on the function blocks can be obtained from the documentation of the S7-App samples.