

Programming Guide

Function Blocks with SIEMENS TIA Portal

VLT_FC_BASIC			
BOOL	EN	ENO	BOOL
HW_IO	ADR	READY	BOOL
1 BOOL	DRV_EN	FAULT	BOOL
BOOL	RUN	WARNING	BOOL
BOOL	REVERSE	RUNNING	BOOL
BOOL	RESET	RUN_ON_REF	BOOL
INT	REF_VALUE	MAV	INT
	1 0 1 1 1	MOTORCURRENT	REAL
	1 1 0 0 1	COMM_ERR	INT

VLT_FC_PARAM_ACCESS			
BOOL	EN	ENO	BOOL
HW_IO	ADR	BUSY	BOOL
BOOL	EXECUTE	DONE	BOOL
BOOL	RD_WR	FAULT	BOOL
WORD	PAR_NO	RD_VALUE	DWORD
BYTE	INDEX	FAULT_CODE	DWORD
DWORD	WR_VALUE		

Contents

1	Introduction	4
1.1	Disclaimer	4
1.2	Purpose of the User Guide	4
1.3	Abbreviations	4
1.4	What are Function Blocks?	5
1.4.1	Advantages of Using Function Blocks	5
1.5	Overview of the Danfoss Library (VLT_PROFI_LIB_V1_12)	5
1.6	Basic Operation Function Block (VLT_PROFI_FC_BASIC [FC300])	5
1.7	Parameter Access Function Block (VLT_PROFI_FC_PARAM_ACCESS [FB300])	8
1.8	Diagnostics Function Block (VLT_PROFI_FC_DIAGNOSTICS [FB302])	9
1.9	Flexible Function Block (VLT_PROFI_FC_FLEXIBLE [FC301])	10
1.10	Flexible Control Function Block (VLT_PROFI_FC_FLEXIBLE_CTRL [FB301])	12
2	Using FBs in TIA Portal	17
2.1	Importing Danfoss Library into a Project	17
2.2	Creating PLC Tags for Function Blocks	22
2.3	Hardware Identifier in S7-1500 PLC	25
2.3.1	Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_BASIC [FC300]	25
2.3.2	Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_PARAM_ACCESS [FB300] and VLT_PROFI_FC_DIAGNOSTICS [FB302]	26
2.3.3	Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_FLEXIBLE [FC301]	27
2.3.4	Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_BASIC [FC300]	28
2.3.5	Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_PARAM_ACCESS [FB300]	29
2.3.6	Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_DIAGNOSTICS [FB302]	30
2.3.7	Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_FLEXIBLE [FC301]	31
2.4	Mapping of PCDREAD & PCDWRITE with VLT_PCD_DATA [DB100]	32
2.5	Setting/Verifying the Value to Pin Using Watch Table	34
3	Examples	36
3.1	General Configuration of the Drive	36
3.2	Basic Operation Function Block	37
3.3	Parameter Access Function Block	41
3.4	Diagnostics Function Block	44
3.5	Flexible Function Block	47
3.6	Flexible Control Function Block	56

1 Introduction

1.1 Disclaimer

The software is provided "as is", without warranty of any kind, expressed or implied, including, but not limited to, the warranties of merchantability, fitness for a particular purpose, and noninfringement. In no event shall the authors or any legal entity part of Danfoss group be liable for any claim, damages, or other liability, whether in an action of contract, tort, or otherwise, arising from, out of, or in connection with the software, or the use, or other dealings in the software.

1.2 Purpose of the User Guide

The function blocks show examples on how it is possible to integrate Danfoss VLT® drives in a SIEMENS TIA Portal V15 system. The function blocks are not protected and can be altered to serve the specific needs for the application. Danfoss takes no responsibility to losses due to code faults in these function blocks or wrong use.

This manual provides:

- Step-by-step approach on how to integrate Danfoss VLT® drives into a SIEMENS TIA Portal V15 system.
- Procedure on using the library to communicate with Danfoss VLT® drives in SIEMENS TIA Portal V15, including examples.

Function blocks for SIEMENS TIA Portal supporting Danfoss VLT® drives.

Danfoss VLT® drives with the following options:

- VLT® PROFIBUS DP MCA 101
- VLT® PROFINET MCA 120

The manual is intended for use by qualified personnel.

1.3 Abbreviations

Abbreviation	Description
FB	Function block
RPM	Revolution per minute
STW	Status word
CTW	Control word
Amp	Ampere
FC	Frequency converter
DDT	Derived data types
PDO	Process data object
SDO	Service data object
DUT	Device under test
LIB	Library
PROFI	PROFIBUS or PROFINET

1.4 What are Function Blocks?

Function blocks are predefined programs or functions contained within a single program element that can be used in the PLC program.

1.4.1 Advantages of Using Function Blocks

- **Basic skeleton:**
 - FB provides the basic infrastructure towards the user.
 - Frees up time to focus on complex and application-specific implementation of the external device.
 - Reuse of an FB several times in a program without rewriting the FB.
 - Easy to use - knowledge of the internal operation of the drive or complex algorithms is not required.
- **Pretested function:**
 - The FB is pretested for working and functionality.
- **Extensibility:**
 - FBs can be extended in future by Danfoss. It is possible to incorporate the FBs with minimal modification in the existing program.

1.5 Overview of the Danfoss Library (VLT_PROFI_LIB_V1_12)

The *VLT_PROFI_LIB_V1_12* library is a collection of predefined function blocks provided by Danfoss. Use these blocks as an aid to simplify programs, containing standard functionality for programming PROFIBUS & PROFINET, SIEMENS TIA Portal systems, and Danfoss drives.

The library contains the following FBs:

- **Basic operation function block (VLT_PROFI_FC_BASIC [FC300]):** Dedicated to handling the basic operation of the drive and connected motor operations.
- **Parameter access function block (VLT_PROFI_FC_PARAM_ACCESS [FB300]):** Dedicated to parameter read/write through an acyclic channel.
- **Diagnostics function block (VLT_PROFI_FC_DIAGNOSTICS [FB302]):** Dedicated to read the diagnostic information.
- **Flexible function block (VLT_PROFI_FC_FLEXIBLE [FC301]):** Dedicated to handle flexible operation of drive and connected motor operations.
- **Flexible control function block (VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]):** Dedicated to handle flexible control operation of drive and connected motor operations.

The function blocks do not support the following features:

- PROFIdrive profile.
- S7 300 & S7 400 PLC.

1.6 Basic Operation Function Block (VLT_PROFI_FC_BASIC [FC300])

The function block provides the following functionalities:

- **Control and monitoring:** monitor the drive and control the command or setpoint from the controller to/from the drive.
- **Reverse:** forward or reverse the direction of the motor.
- **Speed regulation:** allows the speed reference of the drive.
- **Failure management:** the *FAULT* output pin is set to *TRUE* if there is a drive fault. This drive fault must be reset by the input pin *RESET* to close the fault. The fault only disappears if the actual root cause of the fault has disappeared.

VLT_PROFI_FC_BASIC			
BOOL	EN	ENO	BOOL
HW_IO	ADR	READY	BOOL
BOOL	DRV_EN	FAULT	BOOL
BOOL	RUN	WARNING	BOOL
BOOL	REVERSE	RUNNING	BOOL
BOOL	RESET	RUN_ON_REF	BOOL
INT	REF_VALUE	MAV	INT
		MOTORCURRENT	REAL
		COMM_ERR	INT

e30bu515.10

Illustration 1: Basic Operation Function Block Layout
Table 1: Input Parameter

Parameter	Type	Description
EN	BOOL	Default function block enable input pin.
ADR	HW_IO	S7-1200/1500 requires HW identifier. This parameter can be found in HW configuration for the specific PPO type.
DRV_EN	BOOL	Enable the drive. Set the drive in <i>Ready to start</i> mode.
RUN	BOOL	Run/stop drive.
REVERSE	BOOL	Drive <i>Reverse</i> mode. Set <i>parameter 4-10 Motor Speed Direction</i> to [2] <i>Both directions</i> to allow this function.
REF_VALUE	INT	Reference speed 0–10000=0–100.00%.
RESET	BOOL	Reset alarms. Only for trip, not for trip lock.

Table 2: Output Parameter

Parameter	Type	Description
ENO	BOOL	Default function block enable output pin.
READY	BOOL	Drive is ready to start.
FAULT	BOOL	Fault is present. Trip or trip lock.
WARNING	BOOL	Warning is present.
RUNNING	BOOL	Drive is started but not necessarily running on reference speed.
RUN_ON_REF	BOOL	Running on reference.
MAV	INT	Main actual value read from drive 0–10000=0–100.00%.
MOTORCURRENT	INT	Actual motor current.
COMM_ERR	REAL	If the communication between PLC and the device is healthy, result of DPRD_DAT and 0=no error.

e30bu431.1.0

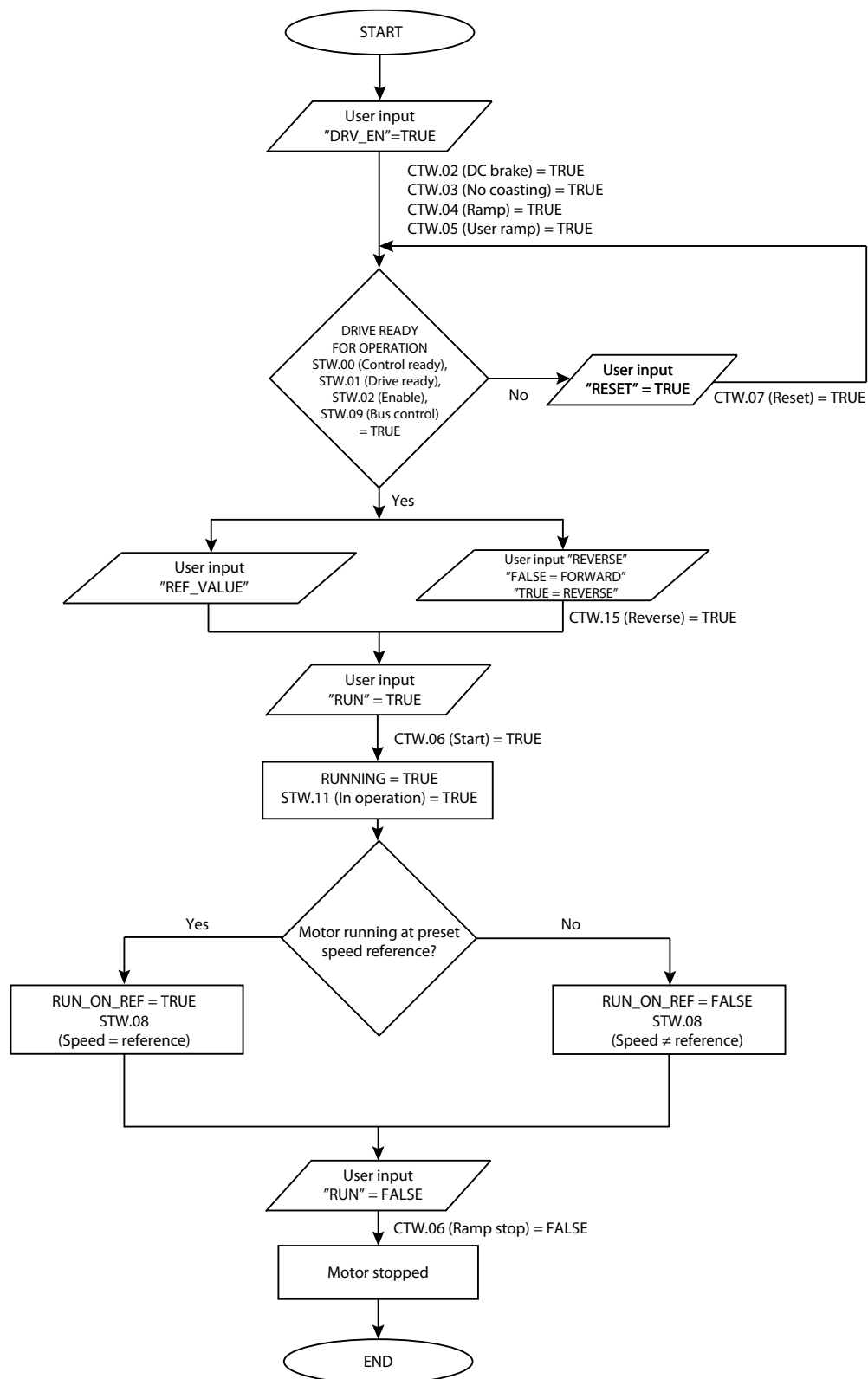


Illustration 2: Flow Chart for Basic Operation Of Drive Control

1.7 Parameter Access Function Block (VLT_PROFI_FC_PARAM_ACCESS [FB300])

The function block provides functionality to read and write VLT FC series parameters access through acyclic service.

- Read non-array type parameters.
- Write non-array type parameters.
- Read array type parameters.
- Write array type parameters.

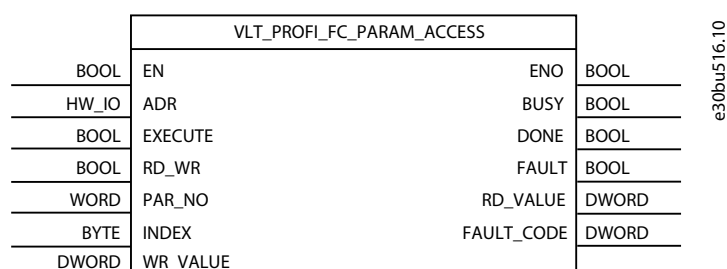


Illustration 3: Parameter Access Function Block Layout

The following elementary function blocks are embedded inside *VLT_PROFI_FC_PARAM_ACCESS [FB300]*, see *SIEMENS TIA Portal Instruction⇒Extended Instruction⇒Distributed IO⇒DP & PROFINET⇒“RDREC & WRREC”*.

- RDREC: a function block which allows to read the data on PROFIBUS slave & PROFINET I/O through acyclic communication.
- WRREC: a function block which allows to write the data on PROFIBUS slave & PROFINET I/O slave through acyclic communication.

Table 3: Input Parameter

Parameter	Type	Description
EN	BOOL	Default function block enable input pin.
ADR	HW_IO	S7-1200/1500 requires HW identifier. This parameter can be found in HW configuration for the specific PPO type.
EXECUTE	BOOL	1=the rising edge of this signal starts the requested operation, 0=triggers an ACK of the end-of-operation notification and the client.
RD_WR	BOOL	Type of operation. 0=read operation, 1=write operation.
PAR_NO	WORD	Enter the drive parameter number in the Hex format. (Example: <i>parameter 3-41 Ramp 1 Ramp Up Time=155 Hex</i>).
INDEX	BYTE	Enter the index number (Example: <i>parameter 310 [2] Preset Reference</i>) 3–10 is the parameter number and 2 is the index number.
WR_VALUE	DWORD	Write value to the drive.

Table 4: Output Parameter

Parameter	Type	Description
ENO	BOOL	Default function block enable input pin.
BUSY	BOOL	0=execution is not started, or execution is completed, 1=execution is processing.
DONE	BOOL	0=value exchange is not completed, 1=value exchange is completed.
FAULT	BOOL	0=value exchange is completed, 1=value exchange is unsuccessful.

Parameter	Type	Description
RD_VALUE	DWORD	Read value from the drive.
FAULT_CODE	WORD	Indicates the fault code associated with the most recently executed command.

1.8 Diagnostics Function Block (VLT_PROFI_FC_DIAGNOSTICS [FB302])

The function block provides functionality to read VLT FC series diagnostics information.

If the drive *parameter 8-07 Diagnosis Trigger* is set to:

- **Disable:** do not send extended diagnosis/emergency data even if they appear in the drive.
- **Trigger on alarms:** send extended diagnosis data when 1 or more alarms appear in *parameter 16-90 Alarm Word*.
- **Trigger alarm/warn.:** send extended diagnosis data if 1 or more alarms or warnings appear in *parameter 16-90 Alarm Word* and *parameter 16-92 Warning Word*.
- EXEMPTION! Warning word 3 and alarm word 3 are not reflected on fieldbus.

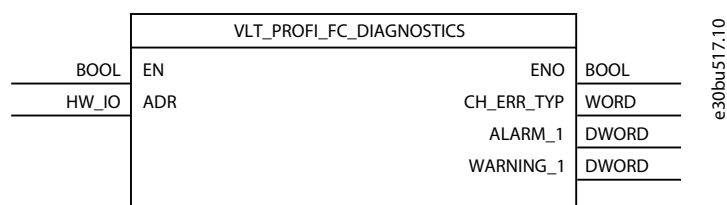


Illustration 4: Diagnostics Function Block Layout

The following elementary function blocks are embedded inside *VLT_PROFI_FC_DIAGNOSTICS* function block, see *SIEMENS TIA Portal Instruction* ⇒ *Extended Instruction* ⇒ *Distributed IO* ⇒ *DP & PROFINET* ⇒ "RALRM".

- RALRM: a function block which allows to read the diagnostics information (channel error type, alarm word & warning word) from PROFIBUS slave & PROFINET I/O.

Table 5: Input Parameter

Parameter	Type	Description
EN	BOOL	Default function block enable input pin.
ADR	HW_IO	S7-1200/1500 requires HW identifier. This parameter can be found in HW configuration for the specific PPO type.

Table 6: Output Parameter

Parameter	Type	Description
ENO	BOOL	Default function block enable output pin.
CH_ERR_TYP	WORD	Channel number error type (only for PROFINET).
ALARM_1	DWORD	Alarm is present (PROFIBUS or PROFINET).
WARNING_1	DWORD	Warning is present (PROFIBUS or PROFINET).

1.9 Flexible Function Block (VLT_PROFI_FC_FLEXIBLE [FC301])

This function block provides the functionality to read and write cyclic process data (*parameter 9-15 PCD Write Configuration and parameter 9-16 PCD Read Configuration*).

The function block supports the following functionalities:

- Read process data from the drive in user accessible format. That is, PCD Read [0] (STW expressed in data bits), PCD Read [1] (main actual value), and PCD Read [2] to PCD Read [9] depends on the drive configuration.
- Write process data to the drive in user accessible format. That is, PCD Write [0] (control word expressed in data bits), PCD Write [1] (reference value), and PCD Write [2] to PCD Write [9] depends on the drive configuration.
- Drive communication health status is continuously monitored using *COMM_ERR* output.

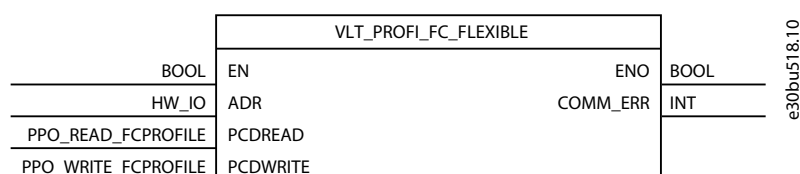


Illustration 5: Flexible Function Block Layout

Table 7: Input Parameter

Parameter	Type	Description
EN	BOOL	Default function block enable input pin.
ADR	HW_IO	S7-1200/1500 requires HW identifier. This parameter can be found in HW configuration for the specific PPO type.

Table 8: Output Parameter

Parameter	Type	Description
ENO	BOOL	Default function block enable output pin.
COMM_ERR	INT	If communication between PLC and the device is healthy, result of DPRD_DAT and 0=no error.

Table 9: Input/output Parameter

Parameter	Type	Description	
PCDREAD	PPO_READ_FCPROFILE	Pointer to data structure for drive data.	
	Parameter	Type	Description
	STW	Structure	Status word
	BIT_00_CTRL_RDY	BOOL	0=control not ready, 1=control ready.
	BIT_01_DRV_RDY	BOOL	0=drive, 1=drive ready.
	BIT_02_COST_RDY	BOOL	0=coasting, 1=enable.
	BIT_03_TRIP	BOOL	0=no error, 1=trip.
	BIT_04_ERROR	BOOL	0=no error, 1=error (no trip).
	BIT_05_RESERVED	BOOL	0=reserved, 1=no action.
	BIT_06_TRIPLOCK	BOOL	0=no error, 1=trip lock.
	BIT_07_WARNING	BOOL	0=no warning, 1=warning.
	BIT_08_RUNNING_REF	BOOL	0=speed<>reference, 1=speed=reference.
	BIT_09_CTRL_SOURCE	BOOL	0=local operation, 1=bus control.
	BIT_10_F_LIMIT_OK	BOOL	0=out of frequency limit, 1=frequency limit ok.
	BIT_11_IN_OPR	BOOL	0=no operation, 1=in operation.
	BIT_12_IN_AUTOSTART	BOOL	0=drive OK, 1=stopped, auto start.
	BIT_13_VOLT_MAX	BOOL	0=voltage OK, 1=voltage exceeded.
	BIT_14_TORQ_MAX	BOOL	0=torque OK, 1=torque exceeded.
	BIT_15_THERMAL_MAX	BOOL	0=timer OK, 1=timer exceeded.
	MAV	INT	Measured actual value (speed) in raw value (-16384) to (+16384).
	PCD_02	INT	Depends on the drive configuration.
	PCD_03	INT	Depends on the drive configuration.
	PCD_04	INT	Depends on the drive configuration.
	PCD_05	INT	Depends on the drive configuration.
	PCD_06	INT	Depends on the drive configuration.
	PCD_07	INT	Depends on the drive configuration.
	PCD_08	INT	Depends on the drive configuration.
	PCD_09	INT	Depends on the drive configuration.

Parameter	Type	Description	
PCDWRITE	PPO_WRITE_FCPROFILE	Pointer to data structure for drive data.	
	Parameter	Type	Description
	CTW	Structure	Control word
	BIT_00_PRESET_REF_1	BOOL	0=reference value, 1=selection lsb.
	BIT_01_PRESET_REF_2	BOOL	0=reference value, 1=selection msb.
	BIT_02_DC_BRAKE_RAMP	BOOL	0=DC brake, 1=ramp.
	BIT_03_COASTING	BOOL	0=coasting, 1=no coasting.
	BIT_04_QSTP_RMP	BOOL	0=quick stop, 1=ramp.
	BIT_05_HLD_RMP	BOOL	0=hold output frequency, 1=use ramp.
	BIT_06_START	BOOL	0=ramp stop, 1=start.
	BIT_07_RESET	BOOL	0=no function, 1=reset.
	BIT_08_JOG	BOOL	0=no function, 1=jog.
	BIT_09_RAMP_1_2	BOOL	0=ramp 1, 1=ramp 2.
	BIT_10_DATA_VALID	BOOL	0=do not use CTW, 1=use CTW.
	BIT_11_RLY_1_ON	BOOL	0=relay 01 deactivated, 1=relay 01 active.
	BIT_12_RLY_2_ON	BOOL	0=relay 02 deactivated, 1=relay 02 active.
	BIT_13_SETUP_SEL_1	BOOL	0=parameter set-up 1, 1=selection msb.
	BIT_14_SETUP_SEL_2	BOOL	0=parameter set-up 2, 1=selection msb.
	BIT_15_REVERSE	BOOL	0=no reverse, 1=reverse.
	REF_VALUE	INT	Reference speed is a raw value (-16384) to (+16384).
	PCD_02	INT	Depends on the drive configuration.
	PCD_03	INT	Depends on the drive configuration.
	PCD_04	INT	Depends on the drive configuration.
	PCD_05	INT	Depends on the drive configuration.
	PCD_06	INT	Depends on the drive configuration.
	PCD_07	INT	Depends on the drive configuration.
	PCD_08	INT	Depends on the drive configuration.
	PCD_09	INT	Depends on the drive configuration.

1.10 Flexible Control Function Block (VLT_PROFI_FC_FLEXIBLE_CTRL [FB301])

This function block provides the functionality to read and write cyclic process data (*parameter 9-15 PCD Write Configuration and parameter 9-16 PCD Read Configuration*).

The function block supports the following functionalities:

- CTW as expressed in the input of the function block and drive operation status as expressed in the output of the function block.
- Read process data from the drive in user accessible format. That is, PCD Read [0] (STW expressed in data bits), PCD Read [1] (main actual value), and PCD Read [2] to PCD Read [9] depend on the drive configuration.
- Write process data to the drive in user accessible format. That is, PCD Write [0] (CTW expressed in data bits), PCD Write [1] (reference value), and PCD Write [2] to PCD Write [9] depend on the drive configuration.
- *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block should be used along with *VLT_PROFI_FC_FLEXIBLE* [FC301] function block.

VLT_PROFI_FC_FLEXIBLE_CTRL			
BOOL	EN	ENO	BOOL
BOOL	PRESET_REF_1	READY	BOOL
BOOL	PRESET_REF_2	FAULT	BOOL
BOOL	DC_BRAKE_RAMP	WARNING	BOOL
BOOL	COAST_INV	RUNNING	BOOL
BOOL	QSTP_RMP	RUN_ON_REF	BOOL
BOOL	HLD_RMP	MAV	INT
BOOL	START		
BOOL	RESET		
BOOL	JOG		
BOOL	RAMP_1_2		
BOOL	DATA_VALID		
BOOL	RLY_1_ON		
BOOL	RLY_2_ON		
BOOL	SETUP_SEL_1		
BOOL	SETUP_SEL_2		
BOOL	REVERSE		
INT	REF_VALUE		
PPO_READ_FCPROFILE	PCDREAD		
PPO_WRITE_FCPROFILE	PCDWRITE		
INT	DPRDWR_ERR		

e30bu519,10

Illustration 6: Flexible Control Function Block Layout
Table 10: Input Parameter

Parameter	Type	Description
EN	BOOL	Default function block enable input pin.
PRESET_REF_1	BOOL	Bit 0.0=reference value, 1=selection lsb.
PRESET_REF_2	BOOL	Bit 1.0=reference value, 1=selection msb.
DC_BRAKE_RAMP	BOOL	Bit 2.0=DC brake, 1=ramp.
COAST_INV	BOOL	Bit 3.0=coasting, 1=no coasting.
QSTP_RMP	BOOL	Bit 4.0=quick stop, 1=ramp.
HLD_RMP	BOOL	Bit 5.0=hold output frequency, 1=use ramp.
START	BOOL	Bit 6.0=ramp stop, 1=start.
RESET	BOOL	Bit 7.0=no function, 1=reset.
JOG	BOOL	Bit 8.0=no function, 1=Jog.
RAMP_1_2	BOOL	Bit 9.0=ramp 1, 1=ramp 2.

Parameter	Type	Description
DATA_VALID	BOOL	Bit 10.0=do not use CTW, 1=use CTW.
RLY_1_ON	BOOL	Bit 11.0=relay 01 deactivated, 1=relay 01 active.
RLY_2_ON	BOOL	Bit 12.0=relay 02 deactivated, 1=relay 02 active.
SETUP_SEL_1	BOOL	Bit 13.0=parameter set-up, 1=selection lsb. Multi setup in <i>parameter 0-10 Active Set-up</i> must be selected.
SETUP_SEL_2	BOOL	Bit 14.0=parameter set-up, 1=selection msb. Multi setup in <i>parameter 0-10 Active Set-up</i> must be selected.
REVERSE	BOOL	Bit 15.0=no reverse, 1=reverse.
REF_VALUE	INT	Reference speed is a raw value (-16384) to (+16384).

Table 11: Output Parameter

Parameter	Type	Description
READY	BOOL	Drive is ready to start.
FAULT	BOOL	Fault is present. Trip or trip lock.
WARNING	BOOL	Warning is present.
RUNNING	BOOL	Drive is started, but not necessarily running on reference speed.
RUN_ON_REF	BOOL	Running on reference.
MAV	INT	Measured actual value (speed) in raw value (-16384) to (+16384).

Table 12: Input/output Parameter

Parameter	Type	Description	
PCDREAD	PPO_READ_FCPROFILE	Pointer to data structure for drive data.	
	Parameter	Type	Description
	STW	Structure	Status word
	BIT_00_CTRL_RDY	BOOL	0=control not ready, 1=control ready.
	BIT_01_DRV_RDY	BOOL	0=drive, 1=drive ready.
	BIT_02_COST_RDY	BOOL	0=coasting, 1=enable.
	BIT_03_TRIP	BOOL	0=no error, 1=trip.
	BIT_04_ERROR	BOOL	0=no error, 1=error (no trip).
	BIT_05_RESERVED	BOOL	0=reserved, 1=no action.
	BIT_06_TRIPLOCK	BOOL	0=no error, 1=trip lock.
	BIT_07_WARNING	BOOL	0=no warning, 1=warning.
	BIT_08_RUNNING_REF	BOOL	0=speed<>reference, 1=speed=reference.
	BIT_09_CTRL_SOURCE	BOOL	0=local operation, 1=bus control.
	BIT_10_F_LIMIT_OK	BOOL	0=out of frequency limit, 1=frequency limit ok.
	BIT_11_IN_OPR	BOOL	0=no operation, 1=in operation.
	BIT_12_IN_AUTOSTART	BOOL	0=drive OK, 1=stopped, auto start.
	BIT_13_VOLT_MAX	BOOL	0=voltage OK, 1=voltage exceeded.
	BIT_14_TORQ_MAX	BOOL	0=torque OK, 1=torque exceeded.
	BIT_15_THERMAL_MAX	BOOL	0=timer OK, 1=timer exceeded.
	MAV	INT	Measured actual value (speed) in raw value (-16384) to (+16384).
	PCD_02	INT	Depends on the drive configuration.
	PCD_03	INT	Depends on the drive configuration.
	PCD_04	INT	Depends on the drive configuration.
	PCD_05	INT	Depends on the drive configuration.
	PCD_06	INT	Depends on the drive configuration.
	PCD_07	INT	Depends on the drive configuration.
	PCD_08	INT	Depends on the drive configuration.
	PCD_09	INT	Depends on the drive configuration.

Parameter	Type	Description	
PCDWRITE	PPO_WRITE_FCPROFILE	Pointer to data structure for drive data.	
	Parameter	Type	Description
	CTW	Structure	Control word
	BIT_00_PRESET_REF_1	BOOL	0=reference value, 1=selection lsb.
	BIT_01_PRESET_REF_2	BOOL	0=reference value, 1=selection msb.
	BIT_02_DC_BRAKE_RAMP	BOOL	0=DC brake, 1=ramp.
	BIT_03_COASTING	BOOL	0=coasting, 1=no coasting.
	BIT_04_RAMP	BOOL	0=quick stop, 1=ramp.
	BIT_05_USE_RAMP	BOOL	0=hold output frequency, 1=use ramp.
	BIT_06_START	BOOL	0=ramp stop, 1=start.
	BIT_07_RESET	BOOL	0=no function, 1=reset.
	BIT_08_JOG	BOOL	0=no function, 1=jog.
	BIT_09_RAMP_1_2	BOOL	0=ramp 1, 1=ramp 2.
	BIT_10_DATA_VALID	BOOL	0=do not use CTW, 1=use CTW.
	BIT_11_RLY_1_ON	BOOL	0=relay 01 deactivated, 1=relay 01 active.
	BIT_12_RLY_2_ON	BOOL	0=relay 02 deactivated, 1=relay 02 active.
	BIT_13_SETUP_SEL_1	BOOL	0=parameter set-up 1, 1=selection msb.
	BIT_14_SETUP_SEL_2	BOOL	0=parameter set-up 2, 1=selection msb.
	BIT_15_REVERSE	BOOL	0=no reverse, 1=reverse.
	REF_VALUE	INT	Main reference speed is a raw value (-16384) to (+16384).
	PCD_02	INT	Depends on the drive configuration.
	PCD_03	INT	Depends on the drive configuration.
	PCD_04	INT	Depends on the drive configuration.
	PCD_05	INT	Depends on the drive configuration.
	PCD_06	INT	Depends on the drive configuration.
	PCD_07	INT	Depends on the drive configuration.
	PCD_08	INT	Depends on the drive configuration.
	PCD_09	INT	Depends on the drive configuration.
DPRDWR_ERR		INT	Map the COMM_ERR tag from VLT_PROFI_FC_FLEXIBLE function block.

2 Using FBs in TIA Portal

2.1 Importing Danfoss Library into a Project

Procedure

1. Download the VLT_PROFI_LIB_V1_12.ZIP file from the Danfoss website.
2. Unzip the VLT_PROFI_LIB_V1_12.ZIP file.

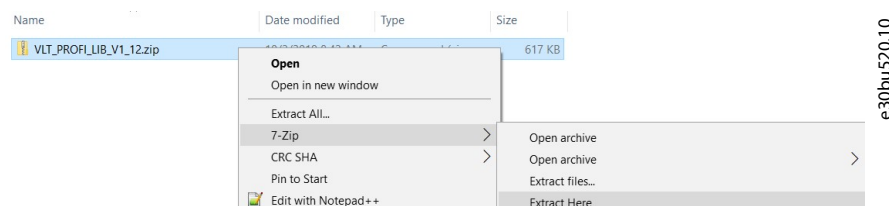


Illustration 7: Extract VLT_PROFI_LIB_V1_12 Library

3. Open the *Project* view in the SIEMENS TIA Portal and click *Options* in the menu bar.
4. Select *Global libraries* and click *Retrieve library...*

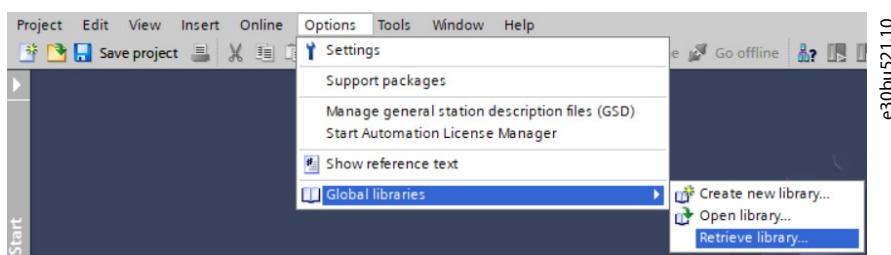


Illustration 8: TIA Portal Retrieve Library

5. In *Retrieve archived global library* pop-up window, navigate to unzipped VLT_PROFI_LIB_V1_12 library folder location.
6. Select the VLT_PROFI_LIB_V1_12.zal15 library file, and uncheck the *Open as read only* checkbox.

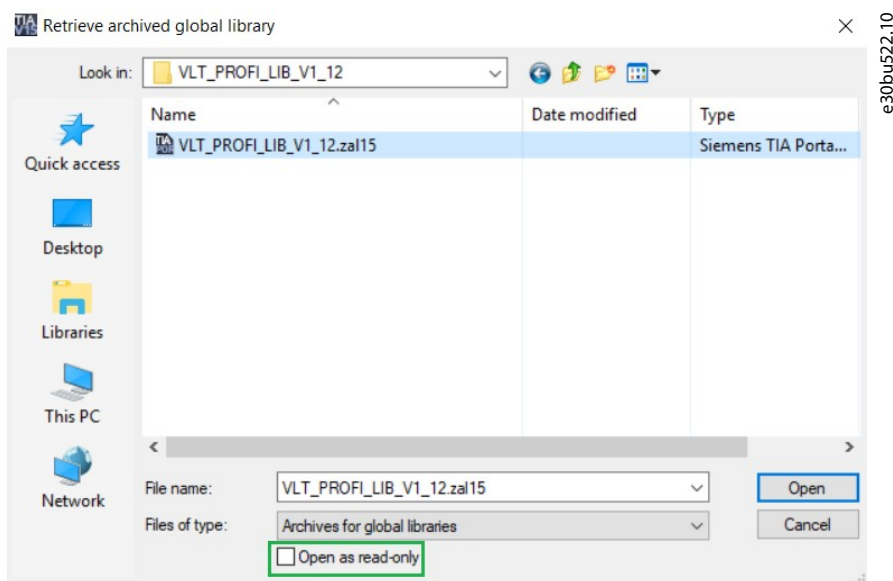


Illustration 9: Retrieve Archived Global Library

7. Press the *Open* button.
8. Select the target directory *Automation* folder (default location) and press *OK*.

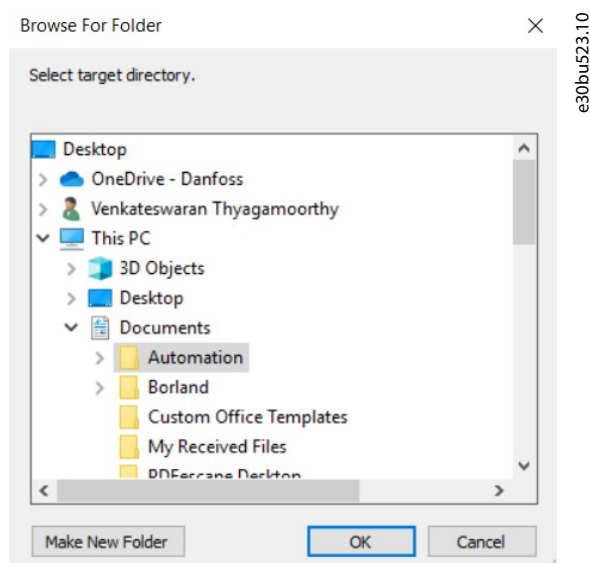


Illustration 10: Select Target Directory

9. Open the *Libraries* task card and click *Global libraries*. Verify whether the library *VLT_PROFI_LIB_V1_12* has installed successfully.

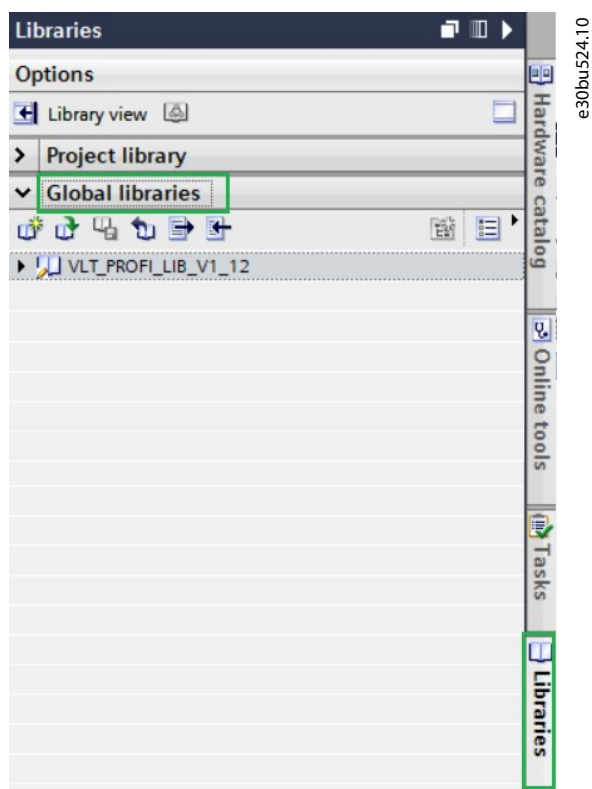


Illustration 11: Global Libraries

10. Expand the `VLT_PROFI_LIB_V1_12` library folder and check whether the DB's and DUT's function blocks are retrieved as shown in the following illustration.

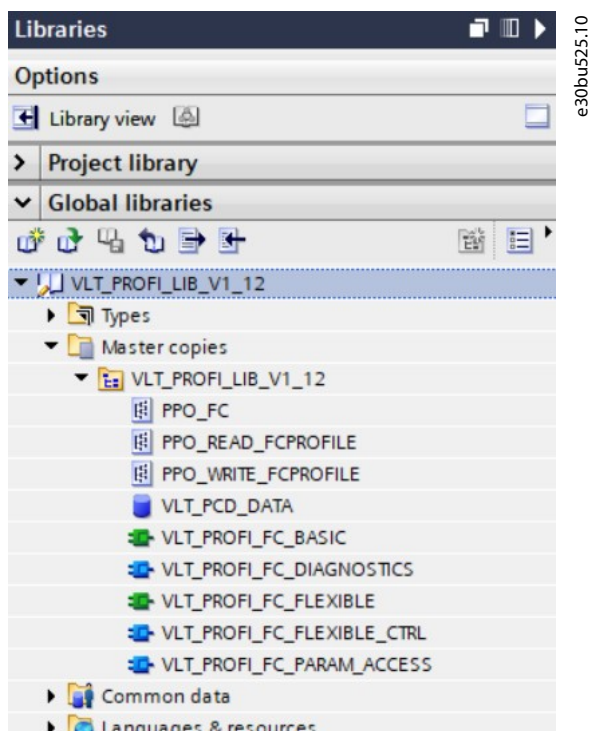


Illustration 12: VLT_PROFI_LIB_V1_12 Libraries

11. From *Global libraries*, select *VLT_PROFI_LIB_V1_12* library folder.
12. Drag and drop the *VLT_PROFI_LIB_V1_12* library folder to *Program blocks* folder as shown in the following illustration.

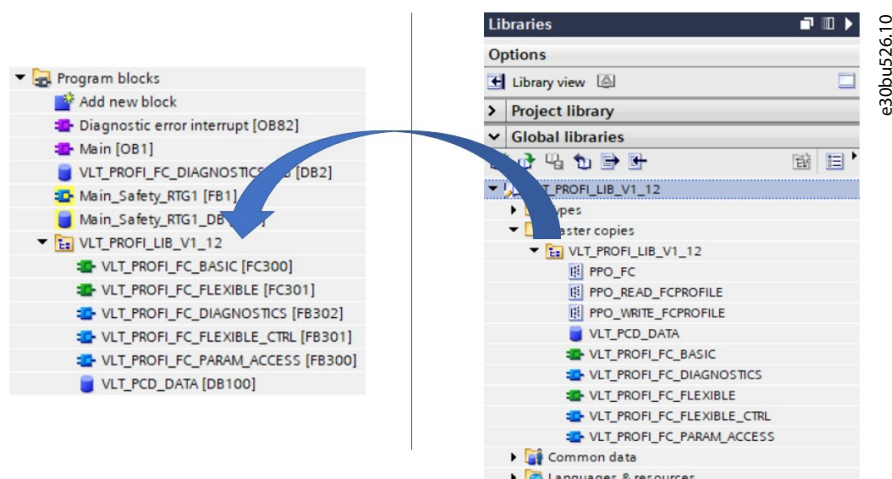


Illustration 13: VLT_PROFI_LIB_V1_12 Libraries in Program Blocks

13. From *Global libraries*, select *VLT_PROFI_LIB_V1_12* library folder.
14. Drag and drop the *PPO_FC*, *PPO_READ_FCPROFILE*, and *PPO_WRITE_FCPROFILE* DUTs to *PLC data types* folder as shown in the following illustration.

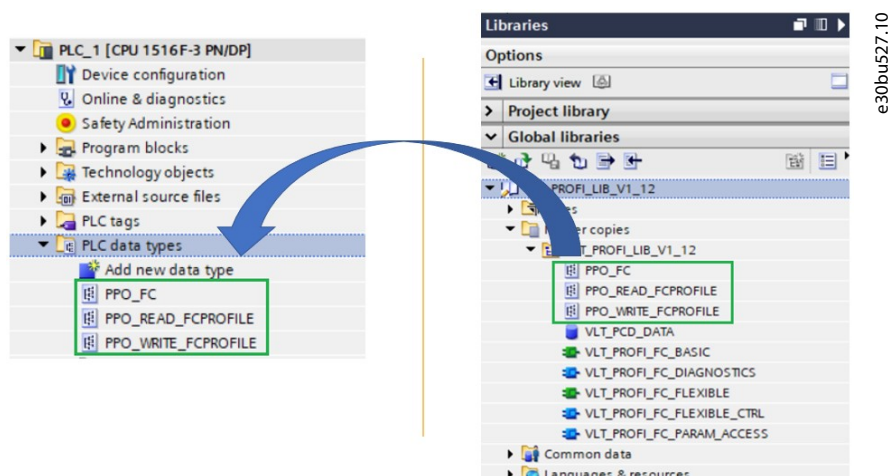


Illustration 14: DUTs in PLC Data Types

2.2 Creating PLC Tags for Function Blocks

Procedure

1. From *Devices & Network*⇒*Project tree*⇒*Devices*⇒*VLT_PROFI_LIB_SIEMENS*⇒*PLC_1 [CPU 1516F-3 PN/DP]*, expand the *Program blocks* folder, and double-click *Main [OB1]* organization block.
2. Open *VLT_PROFI_LIB_V1_12* library folder.
3. Drag and drop the *VLT_PROFI_FC_BASIC [FC300]* function block in *Main [OB1]* as shown in the following illustration.

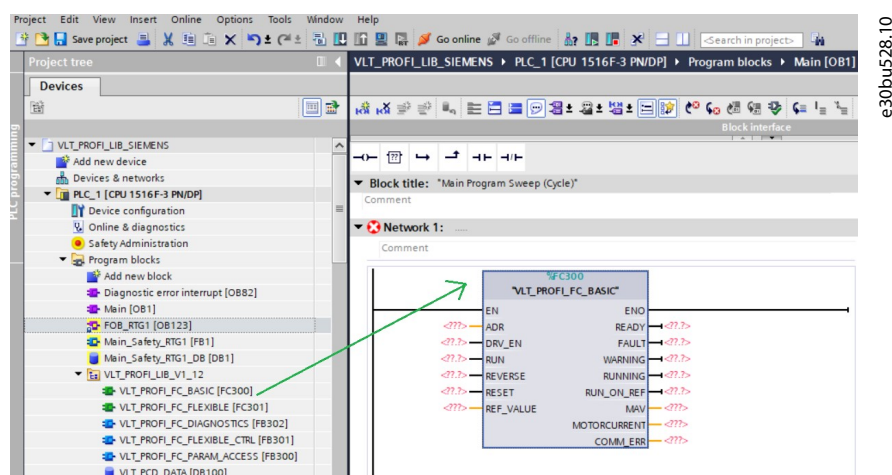


Illustration 15: VLT_PROFI_FC_BASIC [FC300] Function Block

4. Double-click the *DRV_EN* input pin question marks.

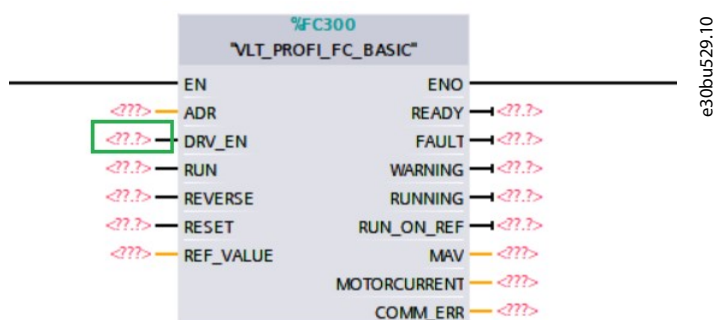


Illustration 16: VLT_PROFI_FC_BASIC [FC300] Function Block

5. Enter the tag name *DRV_EN_DUT1*.

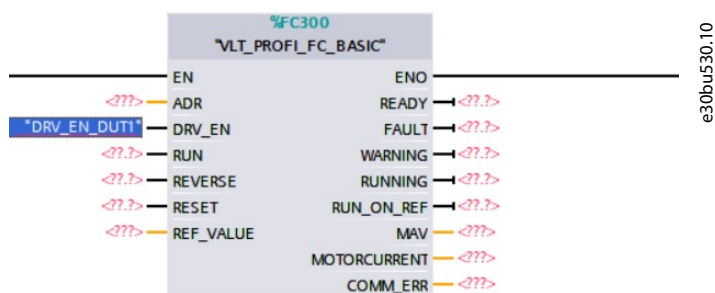


Illustration 17: VLT_PROFI_FC_BASIC [FC300] Function Block Tag Creation

6. Press the enter button on keyboard.
7. Select and right-click *DRV_EN_DUT1*, then select *Define tag...*

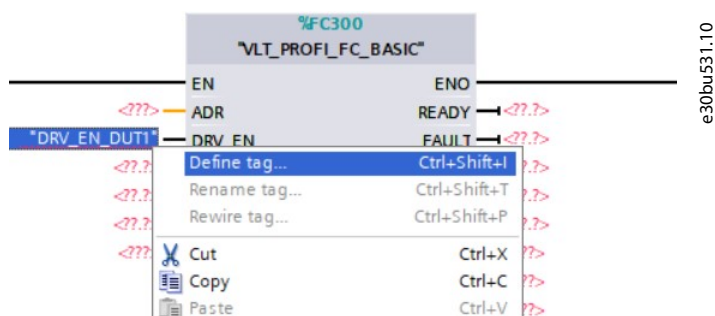


Illustration 18: VLT_PROFI_FC_BASIC [FC300] Function Block with Define Tag

8. In *Define tag* tab, change *Section* from *Local Temp* to *Global Memory*.

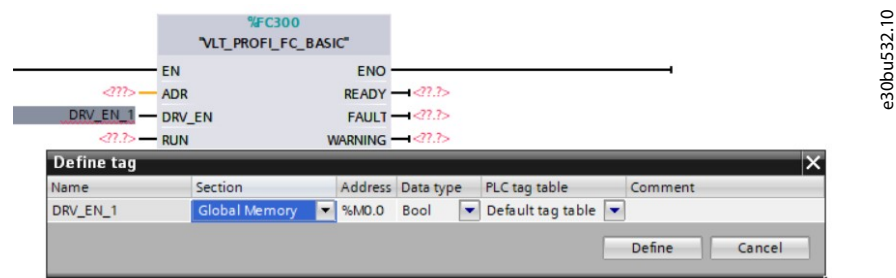


Illustration 19: VLT_PROFI_FC_BASIC [FC300] Function Block Define Tag

9. Press *Define* button.
10. *DRV_EN_DUT1* tag is created and mapped with *DRV_EN* input pin as shown in the following illustration.

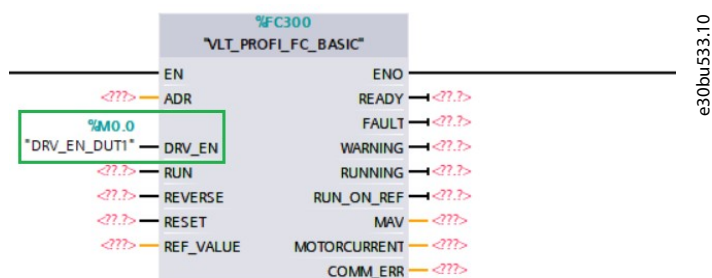


Illustration 20: VLT_PROFI_FC_BASIC [FC300] Function Block with Tag

11. To create the tags for remaining I/O pins, repeat the same steps as shown in the following illustration.

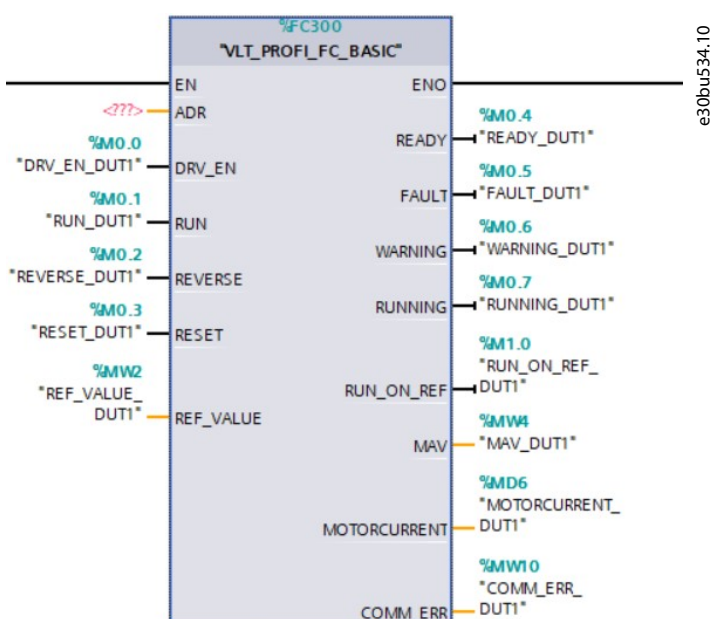


Illustration 21: VLT_PROFI_FC_BASIC [FC300] Function Block with Tags

12. Follow the same procedure to create tags for VLT_PROFI_FC_PARAM_ACCESS [FB300], VLT_PROFI_FC_DIAGNOSTICS [FC302], VLT_PROFI_FC_FLEXIBLE [FC301], and VLT_PROFI_FC_FLEXIBLE_CTRL [FB301] function blocks.

2.3 Hardware Identifier in S7-1500 PLC

2.3.1 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_BASIC [FC300]

Procedure

1. From *Devices & Network*⇒*Project tree*⇒*Devices*⇒*VLT_PROFI_LIB_SIEMENS*⇒*PLC_1 [CPU 1516F-3 PN/DP]*, double-click *Device Configuration*.
2. Select *Network view* tab, and double-click *profinetdut1* PROFINET I/O device.

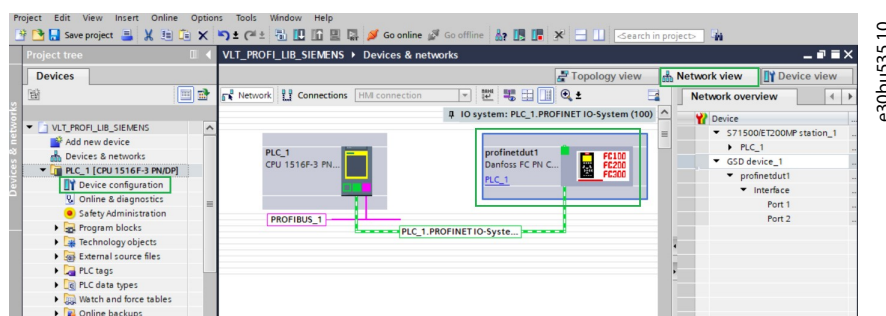


Illustration 22: Network View for PROFINET I/O Device

3. From *Device view*⇒*Device overview*, select *PPO 6 - 6/6 Words, Danfoss Telegram 106* on slot 12.
4. From *Inspector window* view, select *Properties* tab.
5. Select *System Constants* tab and note down the hardware identifier of the drive for *VLT_PROFI_FC_BASIC [FC300]* function block as shown in the following illustration.

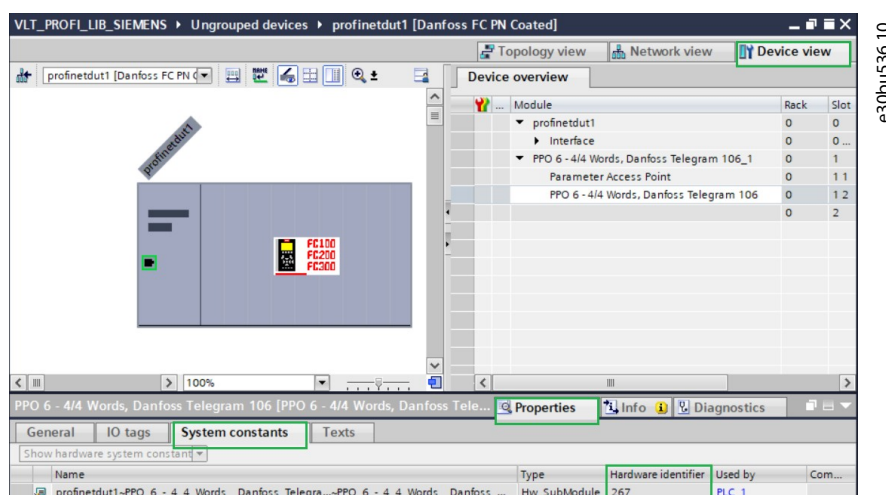


Illustration 23: Hardware ID for VLT_PROFI_FC_BASIC [FC300] Function Block

2.3.2 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_PARAM_ACCESS [FB300] and VLT_PROFI_FC_DIAGNOSTICS [FB302]

Procedure

1. From **Devices & Network**⇒**Project tree**⇒**Devices**⇒**VLT_PROFI_LIB_SIEMENS**⇒**PLC_1 [CPU 1516F-3 PN/DPI]**, double-click **Device Configuration**.
2. Select **Network view** tab, and double-click **profinetdut1** PROFINET I/O device.

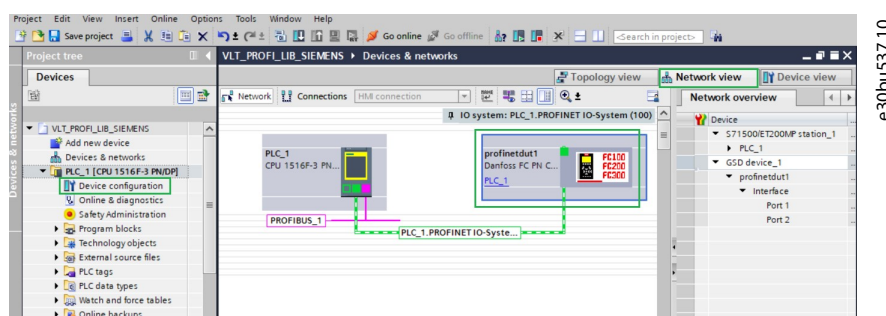


Illustration 24: Network View for PROFINET I/O Device

3. From **Device view**⇒**Device overview**, select **Parameter Access Point** on slot 11.
4. From **Inspector window** view, select **Properties** tab.
5. Select **System Constants** tab and note down the hardware identifier of the drive for **VLT_PROFI_FC_PARAM_ACCESS [FB302]** function block as shown in the following illustration.

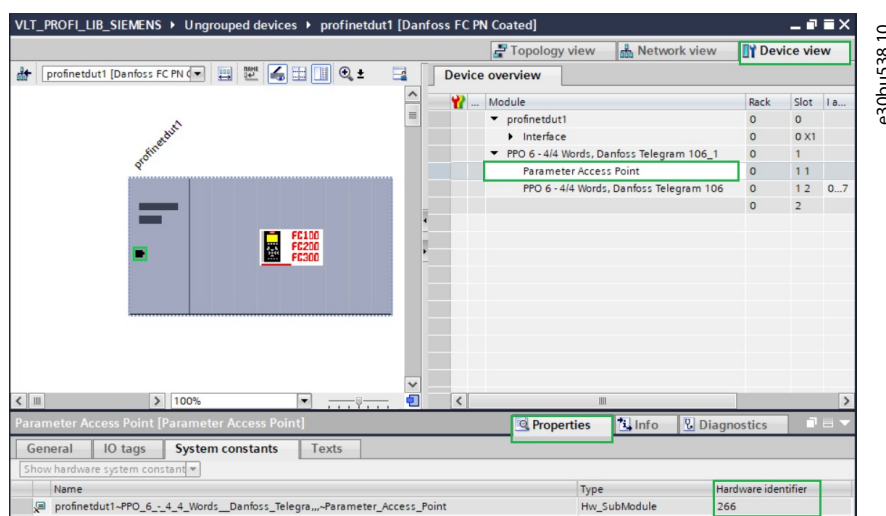


Illustration 25: Hardware ID for VLT_PROFI_FC_PARAM_ACCESS [FB300] and VLT_PROFI_FC_DIAGNOSTICS [FB302] Function Blocks

The same hardware identifier can be used for **VLT_PROFI_FC_DIAGNOSTICS [FB302]** function block.

2.3.3 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_FLEXIBLE [FC301]

Procedure

1. From **Devices & Network**⇒**Project tree**⇒**Devices**⇒**VLT_PROFI_LIB_SIEMENS**⇒**PLC_1 [CPU 1516F-3 PN/DP]**, double-click **Device Configuration**.
2. Select **Network view** tab, and double-click **profinetdut1** PROFINET I/O device.

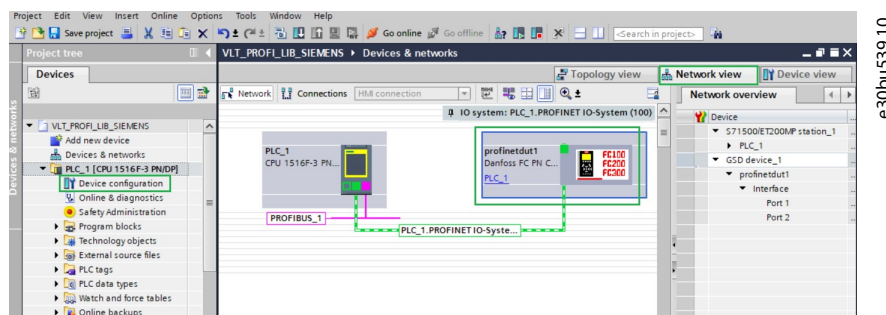


Illustration 26: Network View for PROFINET I/O Device

3. From **Device view**⇒**Device overview**, select **PPO 8-10/10 Words, Danfoss Telegram 108** on slot 12.
4. From **Inspector window** view, select **Properties** tab.
5. Select **System Constants** tab and note down the hardware identifier of the drive for **VLT_PROFI_FC_FLEXIBLE [FC301]** function block as shown in the following illustration.

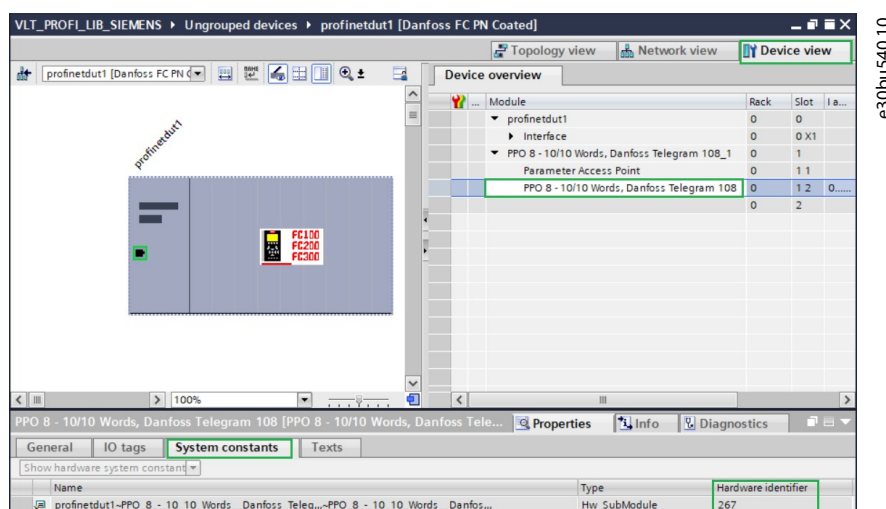


Illustration 27: Hardware ID for VLT_PROFI_FC_FLEXIBLE [FC301] Function Block

2.3.4 Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_BASIC [FC300]

Procedure

1. From *Devices & Network*⇒*Project tree*⇒*Devices*⇒*VLT_PROFI_LIB_SIEMENS*⇒*PLC_1 [CPU 1516F-3 PN/DP]*, double-click *Device Configuration*.
2. Select *Network view* tab, and double-click *profibusdut1* PROFIBUS slave device.

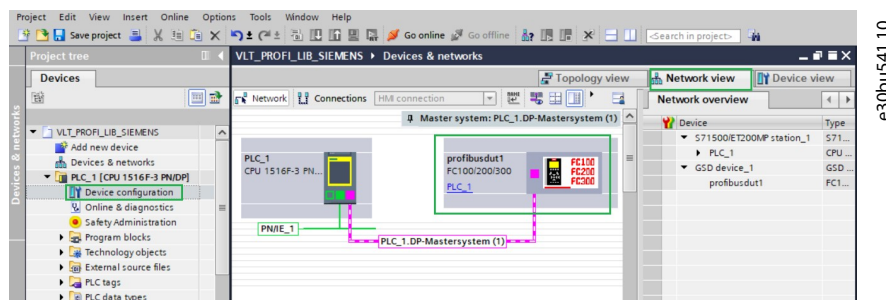


Illustration 28: Network View for PROFIBUS Slave Device

3. From *Device view*⇒*Device overview*, select *PPO Type 2 Module consistent PCD_2_2* on slot 2.
4. From *Inspector window* view, select *Properties* tab.
5. Select *System Constants* tab and note down the hardware identifier of the drive for *VLT_PROFI_FC_BASIC [FC300]* function block as shown in the following illustration.

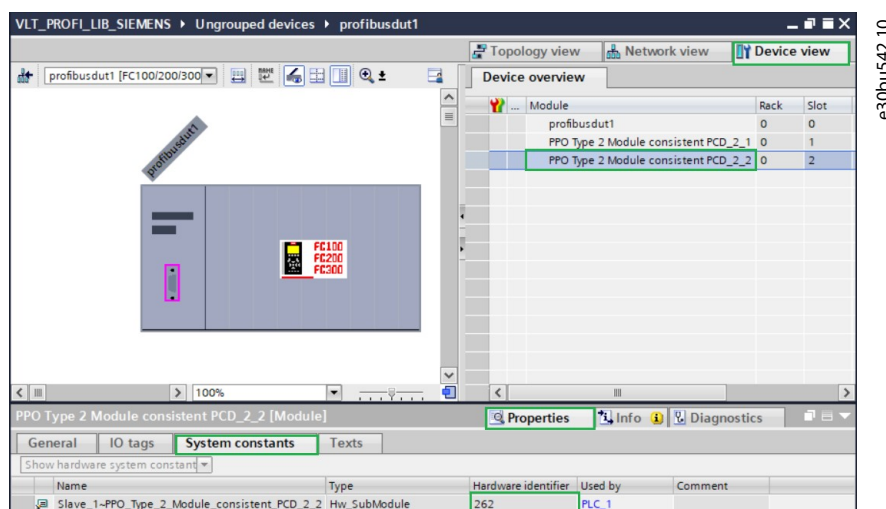


Illustration 29: Hardware ID for VLT_PROFI_FC_BASIC [FC300] Function Block

2.3.5 Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_PARAM_ACCESS [FB300]

Procedure

1. From *Devices & Network*⇒*Project tree*⇒*Devices*⇒*VLT_PROFI_LIB_SIEMENS*⇒*PLC_1 [CPU 1516F-3 PN/DP]*, double-click *Device Configuration*.
2. Select *Network view* tab, and double-click *profibusdut1* PROFIBUS slave device.

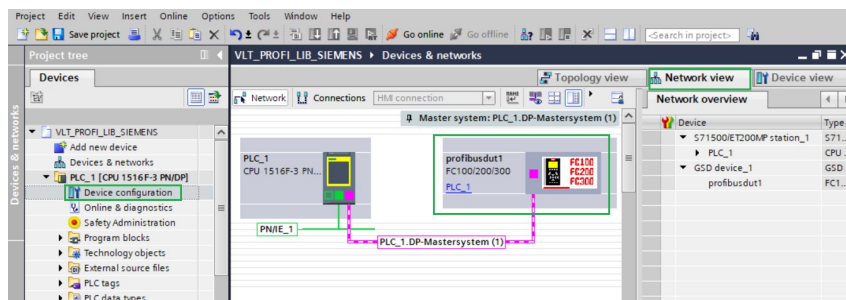


Illustration 30: Network View for PROFIBUS Slave Device

3. From *Device view*⇒*Device overview*, select *PPO Type 2 Module consistent PCD_2_1* on slot 1.
4. From *Inspector window* view, select *Properties* tab.
5. Select *System Constants* tab and note down the hardware identifier of the drive for *VLT_PROFI_FC_PARAM_ACCESS [FB300]* function block as shown in the following illustration.

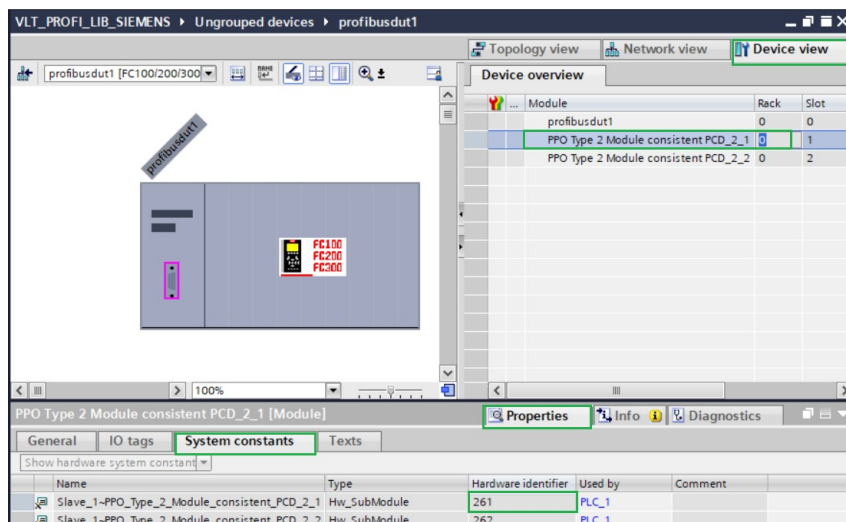


Illustration 31: Hardware ID for VLT_PROFI_FC_PARAM_ACCESS [FB300] Function Block

2.3.6 Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_DIAGNOSTICS [FB302]

Procedure

1. From *Devices & Network*⇒*Project tree*⇒*Devices*⇒*VLT_PROFI_LIB_SIEMENS*⇒*PLC_1 [CPU 1516F-3 PN/DP]*, double-click *Device Configuration*.
2. Select *Network view* tab⇒*Network overview*⇒*GSD device_1*.
3. From *Inspector window* view, select *Properties* tab
4. Select *System Constants* tab and note down the hardware identifier of the drive for *VLT_PROFI_FC_DIAGNOSTICS [FB302]* function block as shown in the following illustration.

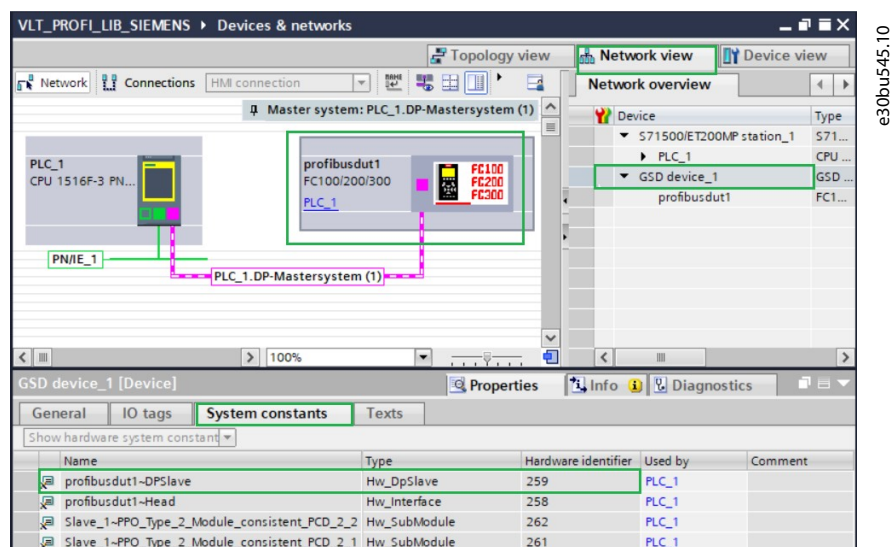


Illustration 32: Hardware ID for VLT_PROFI_FC_DIAGNOSTICS [FB302] Function Block

2.3.7 Finding PROFIBUS Device Hardware Identifier for VLT_PROFI_FC_FLEXIBLE [FC301]

Procedure

1. From **Devices & Network**⇒**Project tree**⇒**Devices**⇒**VLT_PROFI_LIB_SIEMENS**⇒**PLC_1 [CPU 1516F-3 PN/DP]**, double-click **Device Configuration**.
2. Select **Network view** tab, and double-click **profibusdut1** PROFIBUS slave device.

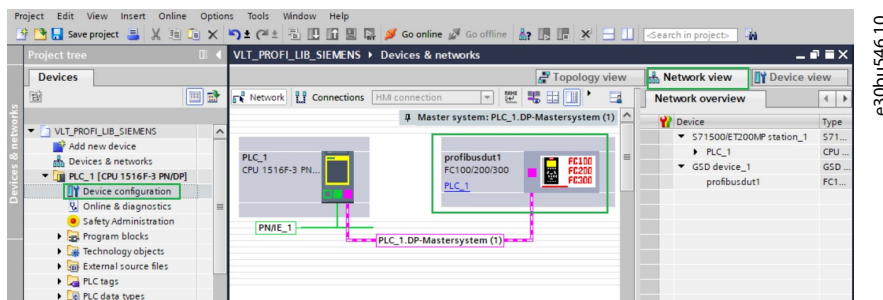


Illustration 33: Network View for PROFIBUS Slave Device

3. From **Device view**⇒**Device overview**, select **PPO Type 5 Module consistent PCD_2_2** on slot 2.
4. From **Inspector window** view, select **Properties** tab.
5. Select **System constants** tab and note down the hardware identifier of the drive for **VLT_PROFI_FC_FLEXIBLE [FC301]** function block as shown in the following illustration.

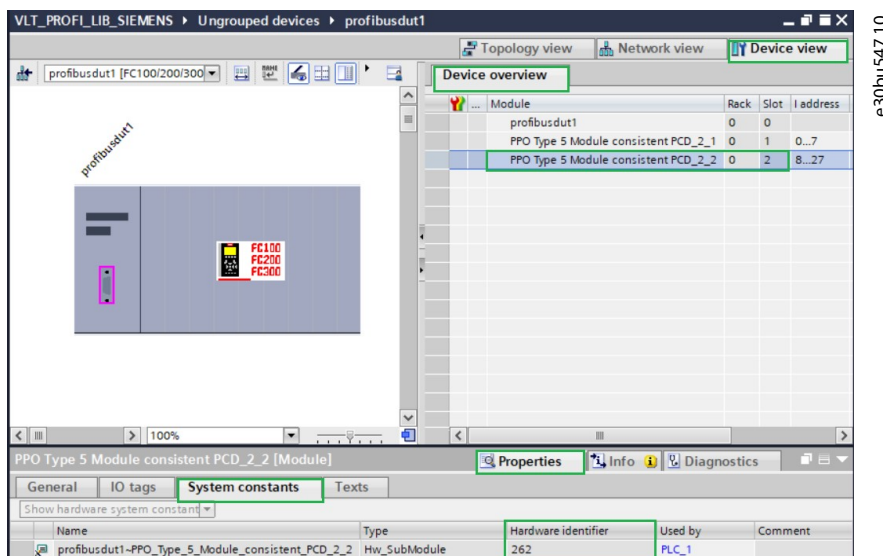


Illustration 34: Hardware ID for VLT_PROFI_FC_FLEXIBLE [FC301] Function Block

2.4 Mapping of PCDREAD & PCDWRITE with VLT_PCD_DATA [DB100]

Procedure

1. Drag and drop **VLT_PROFI_FC_FLEXIBLE [FC301]** function block into **Main [OB1]** as shown in the following illustration.

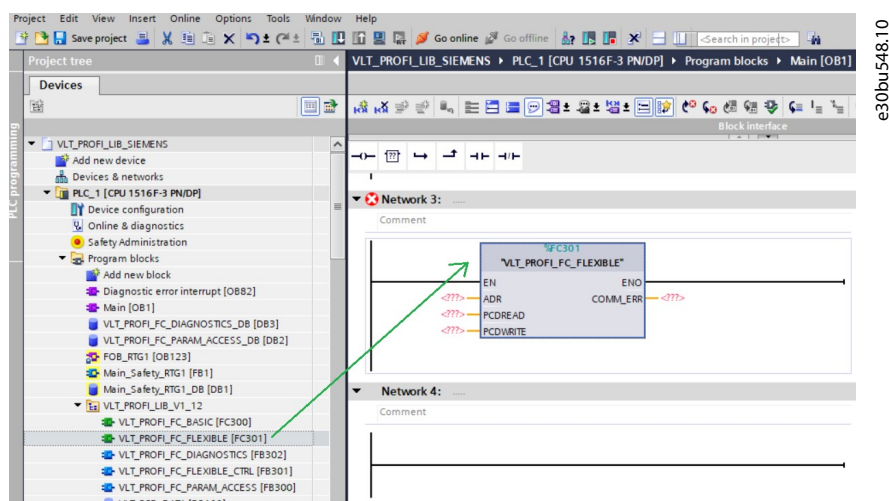


Illustration 35: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block

2. Double-click **PCDREAD** input pin question marks.

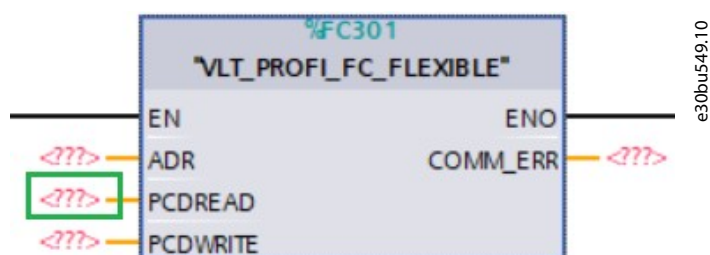


Illustration 36: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block

3. Enter **"VLT_PCD_DATA"** as the name, and select the **Global DB** and **DB 100** as shown in the following illustration.

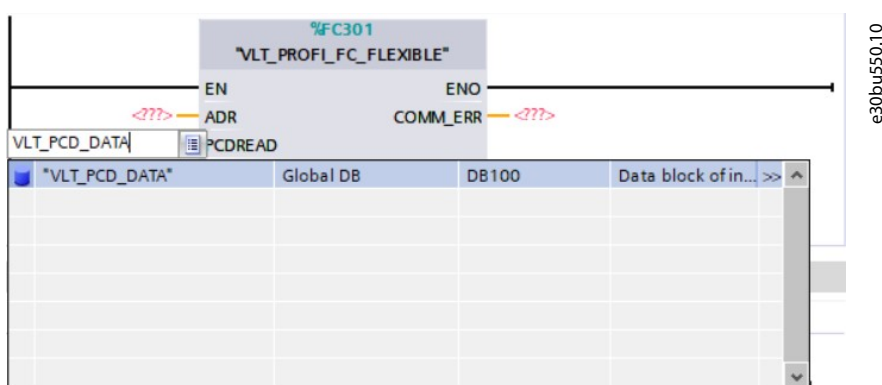


Illustration 37: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block Mapping with VLT_PCD_DATA [DB100]

4. Select *VLT_PCD_DATA*[]⇒*VLT_PCD_DATA*[1]⇒*PPO_FC*⇒*RD*.

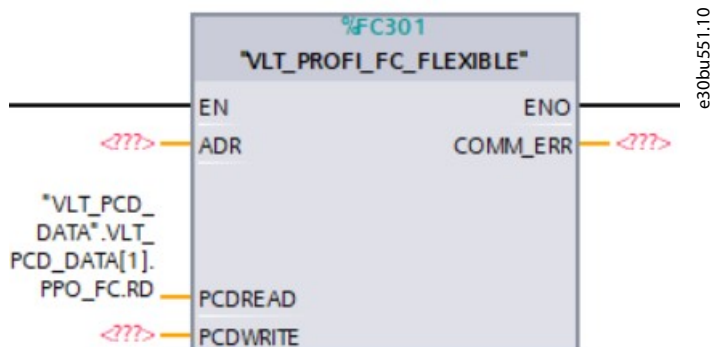


Illustration 38: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block Mapping with VLT_PCD_DATA [DB100]

5. To map the *PCDWRITE* pins with "*VLT_PCD_DATA*".*VLT_PCD_DATA*[1].*PPO_FC.WR*, repeat the same steps as shown in the following illustration.

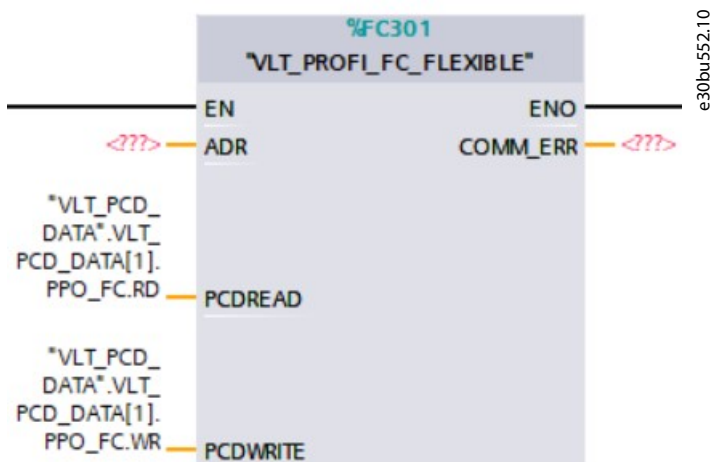


Illustration 39: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block Mapping with VLT_PCD_DATA [DB100]

6. Follow the same procedure to map *PCDREAD* & *PCDWRITE* with *VLT_PCD_DATA* [DB100] data block for *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

- *PCDREAD* pins with "*VLT_PCD_DATA*".*VLT_PCD_DATA*[1].*PPO_FC.RD*.
- *PCDWRITE* pins with "*VLT_PCD_DATA*".*VLT_PCD_DATA*[1].*PPO_FC.WR*.

2.5 Setting/Verifying the Value to Pin Using Watch Table

Procedure

1. Navigate to *Project tree*⇒*PLC*⇒*Watch and force tables*.
2. Open *Watch and force tables*.
3. Double-click *Add new watch table*.

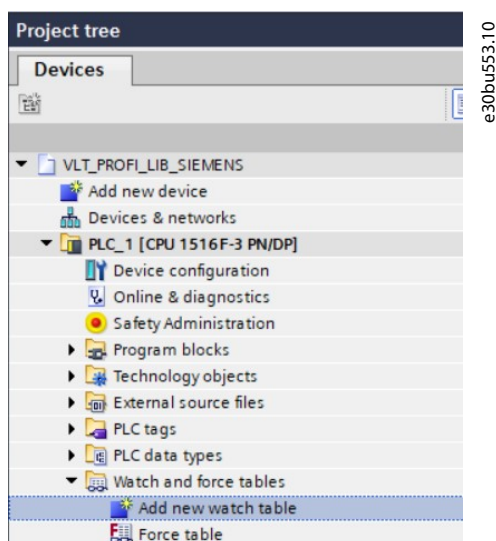


Illustration 40: Add New Watch Table

4. Add tags to monitor/modify in *Watch table_1* as shown in the following illustration.

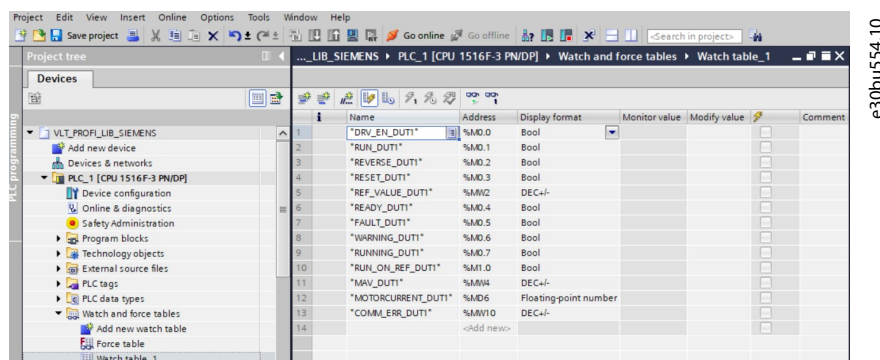


Illustration 41: Add Tags to Watch Table_1

5. Press *Go online* button.
6. Select *Monitor all* as shown in the following illustration.

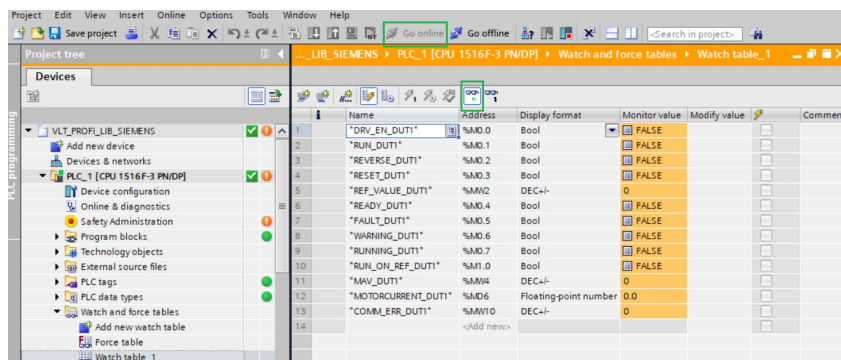


Illustration 42: Watch Table_1 with Monitor Values

7. Modify the value in *Modify value* field.
8. Click *Modify all selected values at once* to load the modification values.

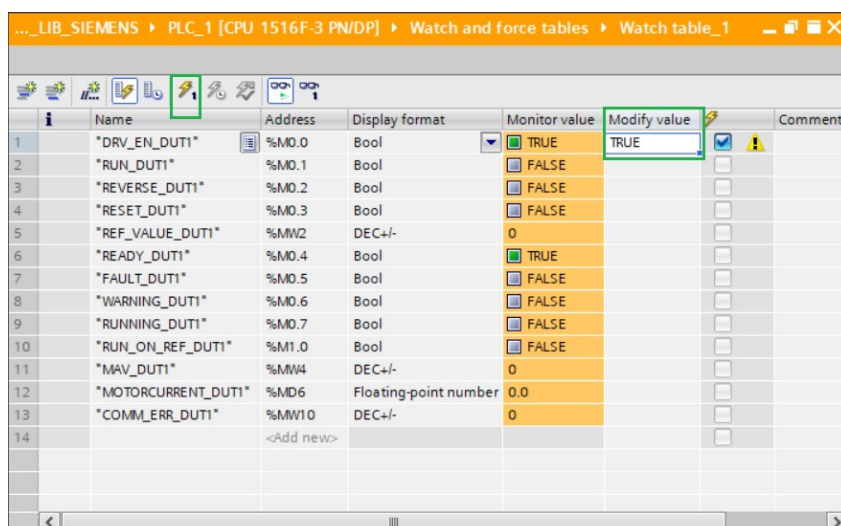


Illustration 43: Watch Table_1 with Modify Values

3 Examples

3.1 General Configuration of the Drive

Procedure

1. When the drive is commissioned, set *parameter 0-03 Regional Settings* before any other changes are made to the drive through the LCP.
2. Verify the following parameter settings to ensure the PLC has control of the drive.

Table 13: Parameter Settings

Parameter	Value
<i>Parameter 8-01 Control Site</i>	<i>[0] Digital and ctrl.word, or [2] Control word only</i>
<i>Parameter 8-02 Control Word Source</i>	<i>[3] Option A</i>

3. When *parameter 8-01 Control Site* is set to *[0] Digital and Ctrl. Word*, establish a connection between terminal 12/13 and terminal 27 to control the motor.
4. The default setting of the drive allows the drive to continue operation if the communication is lost to the PLC. If this operation is not wanted, change *parameter 8-04 Control Word Timeout Function* via the Main Menu.

Table 14: Parameter Settings

Parameter	Value
<i>Parameter 8-04 Control Word Timeout Function</i>	<i>[0] Off, or</i> <i>[1] Freeze output, or</i> <i>[2] Stop, or</i> <i>[3] Jogging, or</i> <i>[4] Max. speed, or</i> <i>[5] Stop and trip</i>

5. Verify that *parameter 8-10 Control Word Profile* is set correctly via the Main Menu.

The function block requires that *parameter 8-10 Control Word Profile* is set to *[0] FC Profile (DEFAULT)*. If *parameter 8-10 Control Word Profile* is set to *PROFIdrive Profile*, the function block does not work as expected and leads to malfunction.

6. Ensure physically that LCP mode is set to *Auto On* mode.

3.2 Basic Operation Function Block

Procedure

1. Drag and drop **VLT_PROFI_FC_BASIC [FC300]** function block into **Main [OB1]** as shown in the following illustration.

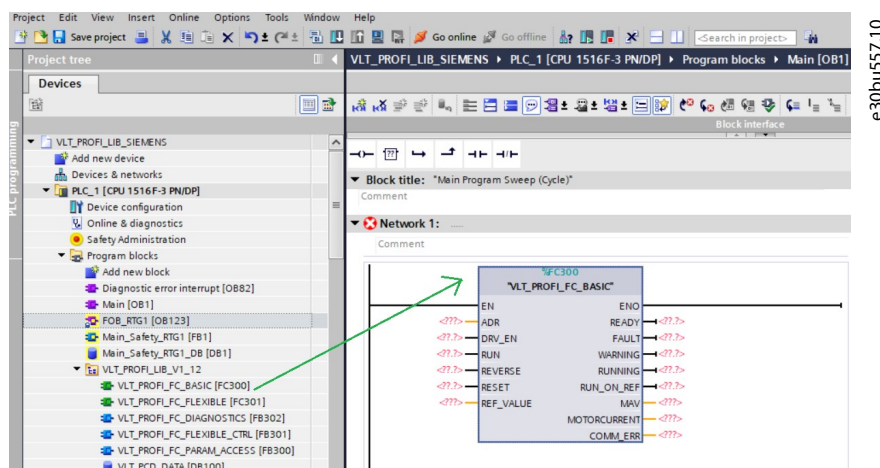


Illustration 44: VLT_PROFI_FC_BASIC [FC300] Function Block Instance

2. Create the variables for **VLT_PROFI_FC_BASIC [FC300]** functional block input and output pins. Refer to [2.2 Creating PLC Tags for Function Blocks](#).
3. Set hardware identifier for **VLT_PROFI_FC_BASIC [FC300]** function block.
 - A Note down the hardware identifier in TIA portal. Refer to [2.3.1 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_BASIC \[FC300\]](#).
 - B Go to **Main [OB1]** block.
 - C Enter the value of ADR inputs, as noted from hardware identifier as shown in the following illustration.

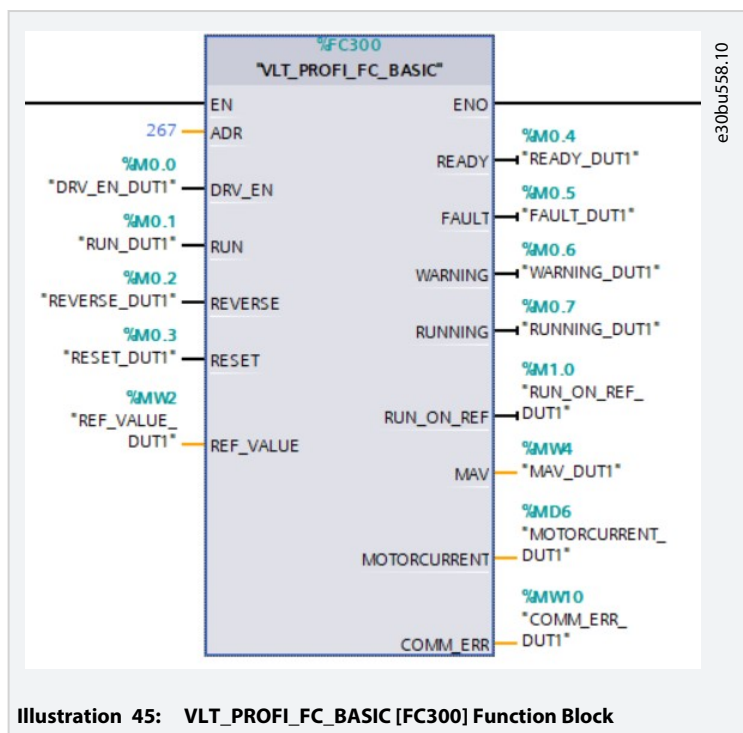


Illustration 45: VLT_PROFI_FC_BASIC [FC300] Function Block

4. Set parameter 9–16 PCD Read Configuration sub-index 2 and 3 using the drive LCP.

Table 15: PCD Read for Motor Current

PCD read config sub-index	Parameter name	Modify value in setup1
9–16 [2]	PCD Read Configuration	[1614] Motor current
9–16 [3]	PCD Read Configuration	[1614] Motor current

5. Enable the drive to ready state.

- A Verify the following values with the function block output.

Table 16: Pin Name and Expected Value

Pin name	Expected value
FAULT	FALSE
COMM_ERR	0 (0 means communication is OK)

- B Set the value *TRUE* to *DRV_EN* input pin of the function block.

- C Verify the following values with the function block output.

Table 17: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .
COMM_ERR	0

- D Set the value *TRUE* to the *RUN* input pin of the function block.

- E Set the value *10000* to the *REF_VALUE* input pin of the function block.

- F Verify the following values with the function block output.

Table 18: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	10000
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .

Pin name	Expected value
COMM_ERR	0

G Set the value *FALSE* to the *DRV_EN* input pin of the function block.

H Verify the following values with the function block output.

Table 19: Pin Name and Expected Value

Pin name	Expected value
READY	FALSE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .
COMM_ERR	0

I Set the value *FALSE* to the *RUN* input pin of the function block.

6. Start the motor in forward direction.

A Set the value *TRUE* to the *DRV_EN* input pin of the function block.

B Set the value *TRUE* to the *RUN* input pin of the function block.

C Set the value *10000* to the *REF_VALUE* input pin of the function block.

D Verify the following values with the function block output.

Table 20: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	10000
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .
COMM_ERR	0

7. Start the motor in reverse direction.

A Set the value *10000* to the *REF_VALUE* input pin of the function block.

B Set the value *TRUE* to the *REVERSE* input pin of the function block. Ensure that *parameter 4-10 Motor Speed Direction* is set to [2] *Both directions*.

C Wait until the motor ramps down and running in the reverse direction.

D Verify the following values with the function block output.

Table 21: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
MAV	-10000 [$\pm 1\%$]
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .
COMM_ERR	0

8. Stop the motor.

- A** Set the value *FALSE* to the *RUN* input pin of the function block.
- B** Set the value *0* to the *REF_VALUE* input pin of the function block.
- C** Verify the following values with the function block output.

Table 22: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
MAV	0
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .
COMM_ERR	0

3.3 Parameter Access Function Block

Procedure

1. Drag and drop `VLT_PROFI_FC_PARAM_ACCESS [FB300]` function block into `Main [OB1]` as shown in the following illustration.

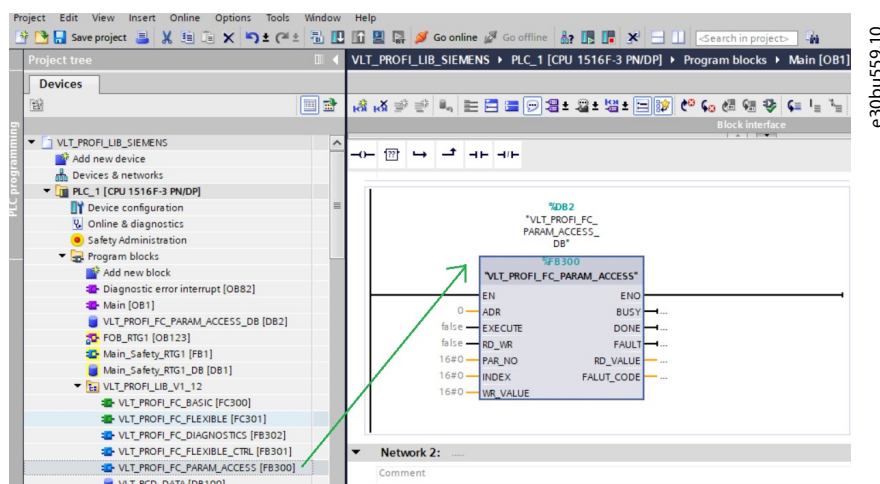


Illustration 46: VLT_PROFI_FC_PARAM_ACCESS [FB300] Function Block Instance

2. Create the variables for `VLT_PROFI_FC_PARAM_ACCESS [FB300]` functional block input and output pins. Refer to [2.2 Creating PLC Tags for Function Blocks](#).
3. Set hardware identifier for `VLT_PROFI_FC_PARAM_ACCESS [FB300]` function block.
 - A Note down the hardware identifier in TIA portal. Refer to [2.3.2 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_PARAM_ACCESS \[FB300\] and VLT_PROFI_FC_DIAGNOSTICS \[FB302\]](#).
 - B Go to `Main [OB1]` block.
 - C Enter the value of ADR inputs, as noted from hardware identifier as shown in the following illustration.

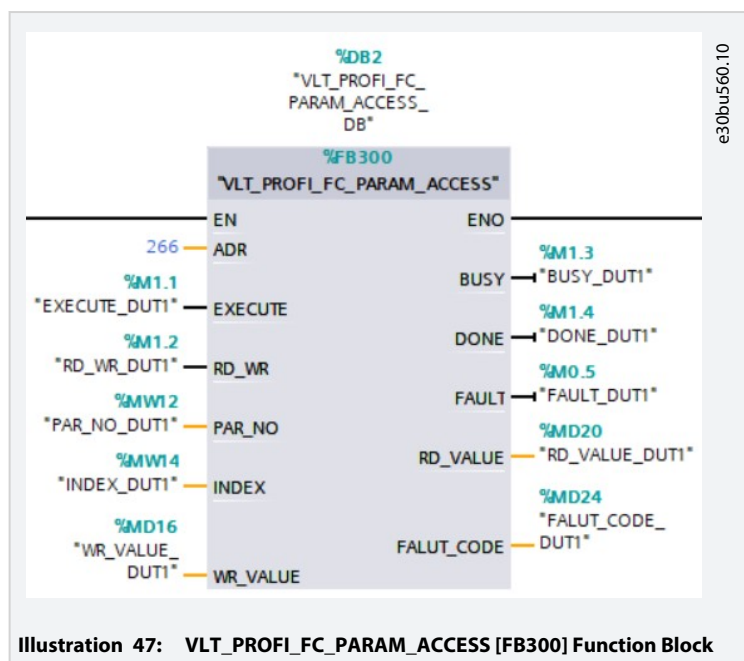


Illustration 47: VLT_PROFI_FC_PARAM_ACCESS [FB300] Function Block

4. Read non-array parameters.

- A Set the value *FALSE* to *EXECUTE* input pin of the function block.
- B Set *parameter 3-41 Ramp 1 Ramp Up Time* to value *3.00* using LCP.
- C Set the following values to the input pin of the function block to read the value in the *parameter 3-41 Ramp 1 Ramp Up Time*.

Table 23: Pin Name and Set Value

Pin Name	Set value for <i>parameter 3-41 Ramp 1 Ramp Up Time</i>
RD_WR	FALSE
PAR_NO	0x0155
INDEX	0x0000
EXECUTE	TRUE

- D Verify the following values with the function block output.

Table 24: Pin Name and Expected Value

Pin name	Expected value
DONE	TRUE
RD_VALUE	0x0000_012C
FAULT_CODE	0x0000_00FF

5. Write non-array parameters.

- A Set the value *FALSE* to *EXECUTE* input pin of the function block.
- B Set the following values to the input pin of the function block to write the value in the *parameter 3-41 Ramp 1 Ramp Up Time*.

Table 25: Pin Name and Set Value

Pin Name	Set value for <i>parameter 3-41 Ramp 1 Ramp Up Time</i>
RD_WR	TRUE
PAR_NO	0x0155
INDEX	0x0000
WR_VALUE	0x01F4
EXECUTE	TRUE

- C Verify *parameter 3-41 Ramp 1 Ramp Up Time* is *5.00 s* using LCP.
- D Verify the *DONE* output of the function block is set to *TRUE*.
- E Verify the *FAULT_CODE* output of the function block is set to *0x0000_00FF*.

6. Read array type parameters with index.

- A Set the value *FALSE* to *EXECUTE* input pin of the function block.
- B Set the value *10.00%* to the *parameter 3-10 [2] Preset Reference* using LCP.
- C Set the following values to the input pin of the function block to read the array parameter *parameter 3-10 [2] Preset Reference*.

Table 26: Pin Name and Set Value

Pin name	Set value for <i>parameter 3-10 [2] Preset Reference</i>
RD_WR	FALSE
PAR_NO	0x0136

Pin name	Set value for <i>parameter 3-10 [2] Preset Reference</i>
INDEX	0x0002
EXECUTE	TRUE

- D** Verify the following values with the function block output.

Table 27: Pin Name and Expected Value

Pin name	Expected value
RD_VALUE	0x0000_03E8
DONE	TRUE
FALUT_CODE	0x0000_00FF

- 7.** Write an array type parameter with index.

- A** Set the value *FALSE* to *EXECUTE* input pin of the function block.
- B** Set the following values to the input pin of the function block to write the value in the non-array *parameter 3-10 [3] Preset Reference*.

Table 28: Pin Name and Set Value

Pin name	Set value for <i>parameter 3-10 [3] Preset Reference</i>
RD_WR	TRUE
PAR_NO	0x0136
INDEX	0x0003
WR_VALUE	0x07D0
EXECUTE	TRUE

- C** Verify on LCP that *parameter 3-10 [3] Preset Reference* is set to 20.00 s.
- D** Verify the *DONE* output of the function block is set to *TRUE*.
- E** Verify the *FAULT_CODE* output of the function block is set to *0x0000_00FF*.

- FAULT_CODE* value is *0x0000_00FF* means that the request has been completed successfully.
- FAULT_CODE* detailed information can be found in:
 - PROFIBUS Operating Instruction Guide VLT
 - PROFINET Operating Instruction Guide VLT

3.4 Diagnostics Function Block

Procedure

1. Drag and drop `VLT_PROFI_FC_DIAGNOSTICS [FB302]` function block into *Diagnostics error interrupt [OB82]* block as shown in the following illustration.

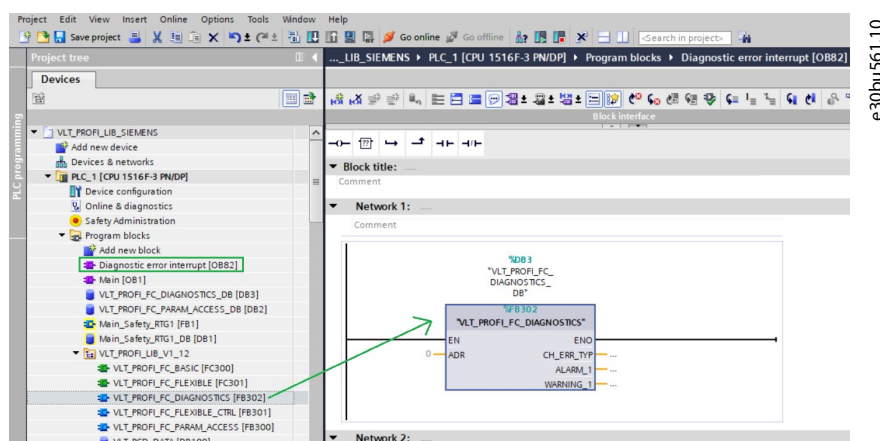


Illustration 48: VLT_PROFI_FC_DIAGNOSTICS [FB302] Function block Instance

2. Create the variables for `VLT_PROFI_FC_DIAGNOSTICS [FB302]` functional block input and output pins. Refer to [2.2 Creating PLC Tags for Function Blocks](#).
3. Set hardware identifier for `VLT_PROFI_FC_DIAGNOSTICS [FB302]` function block.
 - A Note down the hardware identifier in TIA portal. Refer to [2.3.2 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_PARAM_ACCESS \[FB300\] and VLT_PROFI_FC_DIAGNOSTICS \[FB302\]](#).
 - B Go to *Diagnostics error interrupt [OB82]* block.
 - C Enter the value of ADR inputs, as noted from hardware identifier as shown in the following illustration.

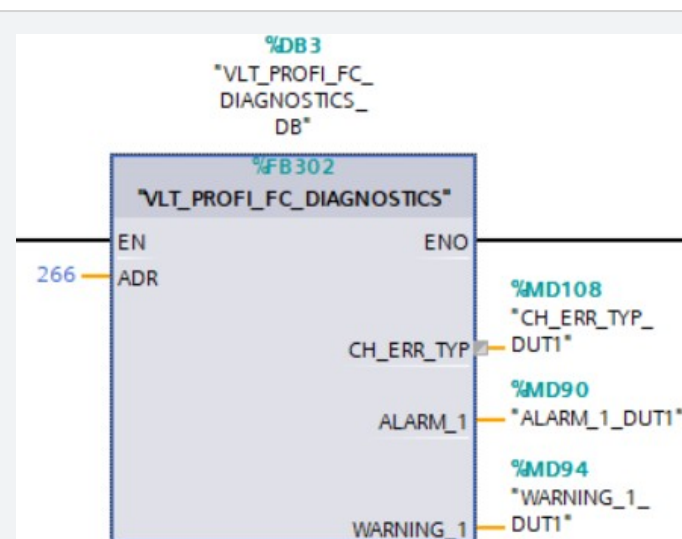


Illustration 49: VLT_PROFI_FC_DIAGNOSTICS [FB302] Function Block

- Set the *parameter 8-07 Diagnosis trigger* to [2] *Trigger alarm/warn* using LCP.
- `CH_ERR_TYP` output is only applicable for PROFINET devices. For PROFIBUS devices, it always shows as zero value `16#0000`.

4. Generate Live Zero Error Warning (W2).

- A** Set the following values in the drive parameters for *Live Zero Error Warning (W2)* using LCP.

Table 29: Live Zero Error Configuration Settings Using LCP

Parameter	Modify value
Parameter 6-00 Live Zero Timeout Time	99 s
Parameter 6-01 Live Zero Timeout Function	[5] Stop and trip
Parameter 6-10 Terminal 53 Low Voltage	2.00 V

- B** Verify the following values in *VLT_PROFI_FC_DIAGNOSTICS [FB302]* function block outputs.

Table 30: Pin Name and Expected Value

Pin name	Expected values
ALARM_1	16#00000000
WARNING_1	16#00010000

- C** Verify whether the value in *CH_ERR_TYP* function block output terminal is 16#9012 for PROFINET and 16#0000 for PROFIBUS.

5. Remove Live Zero Error Warning (W2).

- A** Set parameter 6-01 Live Zero Timeout Function to [0] Off using LCP.

- B** Verify whether *Live Zero Error Warning (W2)* is removed on LCP.

- C** Verify the following values in *VLT_PROFI_FC_DIAGNOSTICS [FB302]* function block outputs.

Table 31: Pin Name and Expected Value

Pin name	Expected values
ALARM_1	16#00000000
WARNING_1	16#00000000

- D** Verify whether the value in *CH_ERR_TYP* function block output terminal is 16#0000 for PROFINET and 16#0000 for PROFIBUS.

6. Generate Live Zero Error Alarm (A2).

- A** Set the following values in the drive parameters for *Live Zero Error Alarm (A2)* using LCP.

Table 32: Live Zero Error Configuration Settings Using LCP

Parameter	Modify value
Parameter 6-00 Live Zero Timeout Time	1 s
Parameter 6-01 Live Zero Timeout Function	[5] Stop and trip
Parameter 6-10 Terminal 53 Low Voltage	2.00 V

- B** Verify the following values in *VLT_PROFI_FC_DIAGNOSTICS [FB302]* function block outputs.

Table 33: Pin Name and Expected Value

Pin name	Expected values
ALARM_1	16#00010000
WARNING_1	16#00000000

- C** Verify whether the value in *CH_ERR_TYP* function block output terminal is 16#9012 for PROFINET and 16#0000 for PROFIBUS.

7. Remove Live Zero Error Alarm (A2).

- A** Set parameter 6-01 Live Zero Timeout Function to [0] Off using LCP.
- B** Verify whether Live Zero Error Alarm (A2) is removed on LCP.
- C** Verify the following values in *VLT_PROFI_FC_DIAGNOSTICS [FB302]* function block outputs.

Table 34: Pin Name and Expected Value

Pin name	Expected values
ALARM_1	16#00000000
WARNING_1	16#00000000

- D** Verify whether the value in *CH_ERR_TYP* function block output terminal is 16#0000 for PROFINET and 16#0000 for PROFIBUS.

3.5 Flexible Function Block

Procedure

1. Drag and drop **VLT_PROFI_FC_FLEXIBLE [FC301]** function block into **Main OB1** as shown in the following illustration.

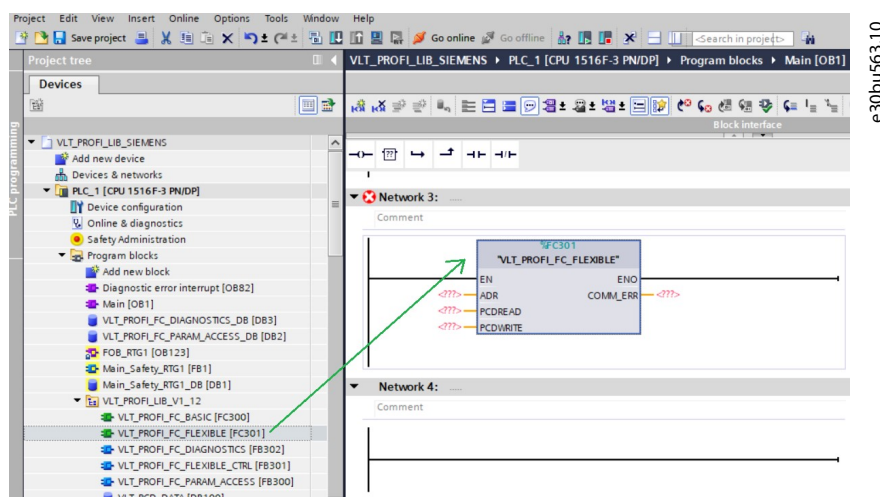


Illustration 50: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block Instance

2. Create the variables for **VLT_PROFI_FC_FLEXIBLE [FC301]** functional block input and output pins. Refer to [2.2 Creating PLC Tags for Function Blocks](#).
3. Map **VLT_PCD_DATA DB[100] ⇒ VLT_PCD_DATA[1]** data block with **PCDREAD** and **PCDWRITE** in-out pin. Refer to [2.4 Mapping of PCDREAD & PCDWRITE with VLT_PCD_DATA \[DB100\]](#).
4. Set hardware identifier for **VLT_PROFI_FC_FLEXIBLE [FC301]** function block.
 - A Note down the hardware identifier in TIA portal. Refer to [2.3.3 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_FLEXIBLE \[FC301\]](#).
 - B Go to **Main [OB1]** block.
 - C Enter the value of ADR inputs, as noted from hardware identifier as shown in the following illustration.

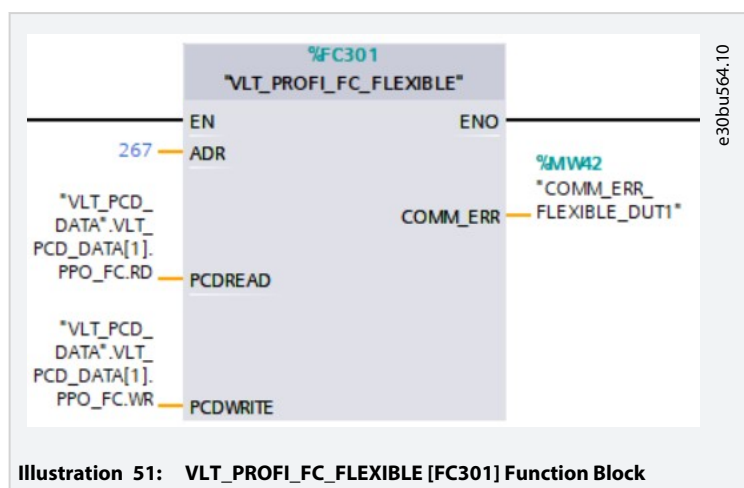


Illustration 51: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block

5. Open **VLT_PCD_DATA DB[100]** data block ⇒ **VLT_PCD_DATA[VLT_PCD_DATA[1]] ⇒ PPO_FC ⇒ RD ⇒ STW**, **MAV** and **PCD02–PCD09** values can be monitored using **Monitor All** button as shown in the following illustration.

PLC programming

VLT_PROFI_LIB_SIEMENS > PLC_1 [CPU 1516F-3 PN/DP] > Program blocks > VLT_PROFI_LIB_V1

Keep actual values Snapshot Copy snapshots to start values

VLT_PCD_DATA

Name	Data type	Start value	Monitor value
Static			
VLT_PCD_DATA	Array[1..10] of *PP...		
VLT_PCD_DATA[1]	*PPO_FC		
PPO_FC	Struct		
RD	*PPO_READ_FCPRO...		
STW	Struct		
BIT_00_CTRL_RDY	Bool	false	TRUE
BIT_01_DRV_RDY	Bool	false	TRUE
BIT_02_COAST_RDY	Bool	false	TRUE
BIT_03_TRIP	Bool	false	FALSE
BIT_04_ERROR	Bool	false	FALSE
BIT_05_RESERVED	Bool	false	FALSE
BIT_06_TRIPLOCK	Bool	false	FALSE
BIT_07_WARNING	Bool	false	FALSE
BIT_08_RUNNING_REF	Bool	false	FALSE
BIT_09_CTRL_SOURCE	Bool	false	TRUE
BIT_10_F_LIMIT_OK	Bool	false	TRUE
BIT_11_IN_OPR	Bool	false	FALSE
BIT_12_IN_AUTOSTART	Bool	false	FALSE
BIT_13_VOLT_MAX	Bool	false	FALSE
BIT_14_TORQ_MAX	Bool	false	FALSE
BIT_15_TIMER_EXD	Bool	false	FALSE
MAV	Int	0	0
PCD_02	Int	0	0
PCD_03	Int	0	0
PCD_04	Int	0	0
PCD_05	Int	0	0
PCD_06	Int	0	0
PCD_07	Int	0	0
PCD_08	Int	0	0
PCD_09	Int	0	0

e30bu565.10

Illustration 52: VLT_PCD_DATA [DB100] Data Block with Read

- Open *VLT_PCD_DATA* DB[100] data block⇒*VLT_PCD_DATA*⇒*VLT_PCD_DATA[1]*⇒*PPO_FC*⇒*WR*⇒*CTW*, *REF_VALUE* and *PCD02*–*PCD09* values can be controlled using *Monitor All* button as shown in the following illustration.

The screenshot shows the 'VLT_PCD_DATA' data block in the Siemens TIA Portal. The block is organized into a table with columns for Name, Data type, Start value, and Monitor value. The variables are listed as follows:

Name	Data type	Start value	Monitor value
Static			
VLT_PCD_DATA	Array(1..10) of *PP...		
VLT_PCD_DATA[1]	*PPO_FC		
PPO_FC	Struct		
RD	*PPO_READ_FCPRO...		
WR	*PPO_WRITE_FCPRO...		
CTW	Struct		
BIT_00_PRESET_REF_1	Bool	false	FALSE
BIT_01_PRESET_REF_2	Bool	false	FALSE
BIT_02_DC_BRAKE_RA...	Bool	TRUE	TRUE
BIT_03_COAST_INV	Bool	TRUE	TRUE
BIT_04_QSTP_RMP	Bool	TRUE	TRUE
BIT_05_HLD_RMP	Bool	TRUE	TRUE
BIT_06_START	Bool	false	FALSE
BIT_07_RESET	Bool	false	FALSE
BIT_08_JOG	Bool	false	FALSE
BIT_09_RAMP_1_2	Bool	false	FALSE
BIT_10_DATA_VALID	Bool	TRUE	TRUE
BIT_11_RLY_1_ON	Bool	FALSE	FALSE
BIT_12_RLY_2_ON	Bool	FALSE	FALSE
BIT_13_SETUP_SEL_1	Bool	FALSE	FALSE
BIT_14_SETUP_SEL_2	Bool	FALSE	FALSE
BIT_15_REVERSE	Bool	false	FALSE
REF_VALUE	Int	0	0
PCD_02	Int	0	0
PCD_03	Int	0	0
PCD_04	Int	0	0
PCD_05	Int	0	0
PCD_06	Int	0	0
PCD_07	Int	0	0
PCD_08	Int	0	0
PCD_09	Int	0	0

Illustration 53: VLT_PCD_DATA [DB100] Data Block with Write

7. To check that the communication is OK, verify the following values with the *VLT_PROFI_FC_FLEXIBLE [FC301]* function block output.

Table 35: Pin Name and Expected Value

Pin name	Expected value
COMM_ERR	0 (0 means the communication is OK)

8. Start the motor in forward direction.

- A** Open *VLT_PCD_DATA [DB100]* data block and enable *Monitor all*, it shows the actual values which the tags currently have in the CPU.
- B** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW*, modify the following values in *Monitor Value* field.

Table 36: Tag Name and Set Value

Tag name	Set values in VLT_PCD_DATA [DB100] data block
BIT_00_PRESET_REF_1	FALSE
BIT_01_PRESET_REF_2	FALSE
BIT_02_DC_BRAKE_RAMP	TRUE
BIT_03_COAST_INV	TRUE
BIT_04_QSTP_RMP	TRUE
BIT_05_HLD_RMP	TRUE

Tag name	Set values in VLT_PCD_DATA [DB100] data block
BIT_06_START	TRUE
BIT_07_RESET	FALSE
BIT_08_JOG	FALSE
BIT_09_RAMP_1_2	FALSE
BIT_10_DATA_VALID	TRUE
BIT_11_RLY_1_ON	FALSE
BIT_12_RLY_2_ON	FALSE
BIT_13_SETUP_SEL_1	FALSE
BIT_14_SETUP_SEL_2	FALSE
BIT_15_REVERSE	FALSE
REF_VALUE	16384

- C** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *RD* ⇒ *STW*, verify the following values.

Table 37: Tag Name and Expected Value

Tag name	Expected value
BIT_00_CTRL_RDY	TRUE
BIT_01_DRV_RDY	TRUE
BIT_02_COAST_RDY	TRUE
BIT_03_TRIP	FALSE
BIT_04_ERROR	FALSE
BIT_05_RESERVED	FALSE
BIT_06_TRIPLOCK	FALSE
BIT_07_WARNING	FALSE
BIT_08_RUNNING_REF	TRUE
BIT_09_CTRL_SOURCE	TRUE
BIT_10_F_LIMIT_OK	TRUE
BIT_11_IN_OPR	TRUE
BIT_12_IN_AUTOSTART	FALSE
BIT_13_VOLT_MAX	FALSE
BIT_14_TORQ_MAX	FALSE
BIT_15_TIMER_EXD	FALSE
MAV	16384

- 9.** Start the motor in reverse direction.

- A** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW*, modify the following values in *Monitor Value* field.

Table 38: Tag Name and Set Value

Tag name	Set values in VLT_PCD_DATA [DB100] data block
BIT_00_PRESET_REF_1	FALSE
BIT_01_PRESET_REF_2	FALSE
BIT_02_DC_BRAKE_RAMP	TRUE
BIT_03_COAST_INV	TRUE
BIT_04_QSTP_RMP	TRUE
BIT_05_HLD_RMP	TRUE
BIT_06_START	TRUE
BIT_07_RESET	FALSE
BIT_08_JOG	FALSE
BIT_09_RAMP_1_2	FALSE
BIT_10_DATA_VALID	TRUE
BIT_11_RLY_1_ON	FALSE
BIT_12_RLY_2_ON	FALSE
BIT_13_SETUP_SEL_1	FALSE
BIT_14_SETUP_SEL_2	FALSE
BIT_15_REVERSE	FALSE

- B** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *REF_VALUE*, enter the *REF_VALUE* input -16384 in *Monitor Value* field.
- C** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *RD* ⇒ *STW*, verify the following values.

Table 39: Tag Name and Expected Value

Tag name	Expected value
BIT_00_CTRL_RDY	TRUE
BIT_01_DRV_RDY	TRUE
BIT_02_COAST_RDY	TRUE
BIT_03_TRIP	FALSE
BIT_04_ERROR	FALSE
BIT_05_RESERVED	FALSE
BIT_06_TRIPLOCK	FALSE
BIT_07_WARNING	FALSE
BIT_08_RUNNING_REF	TRUE
BIT_09_CTRL_SOURCE	TRUE
BIT_10_F_LIMIT_OK	TRUE
BIT_11_IN_OPR	TRUE
BIT_12_IN_AUTOSTART	FALSE

Tag name	Expected value
BIT_13_VOLT_MAX	FALSE
BIT_14_TORQ_MAX	FALSE
BIT_15_TIMER_EXD	FALSE
MAV	-16384

10. Reset the drive.

- A** Set the following parameter values in the drive using LCP.

Table 40: Parameter Settings

Parameter	Modify value
Parameter 6-00 Live Zero Timeout Time	99 s
Parameter 6-01 Live Zero Timeout Function	[5] Stop and trip
Parameter 6-10 Terminal 53 Low Voltage	2.00 V

- B** Verify whether *Live Zero Error Warning* [W2] is generated on LCP.

- C** From *VLT_PCD_DATA* [DB100]⇒*Static*⇒*VLT_PCD_DATA*⇒*VLT_PCD_DATA*[1]⇒*PPO_FC*⇒*RD*⇒*STW*, verify the following values.

Table 41: Tag Name and Expected Value

Tag name	Expected value
BIT_00_CTRL_RDY	TRUE
BIT_01_DRV_RDY	TRUE
BIT_02_COAST_RDY	TRUE
BIT_03_TRIP	FALSE
BIT_04_ERROR	FALSE
BIT_05_RESERVED	FALSE
BIT_06_TRIPLOCK	FALSE
BIT_07_WARNING	TRUE
BIT_08_RUNNING_REF	TRUE
BIT_09_CTRL_SOURCE	TRUE
BIT_10_F_LIMIT_OK	TRUE
BIT_11_IN_OPR	TRUE
BIT_12_IN_AUTOSTART	FALSE
BIT_13_VOLT_MAX	FALSE
BIT_14_TORQ_MAX	FALSE
BIT_15_TIMER_EXD	FALSE
MAV	-16384

- D** Wait for 99 s.

- E** Verify whether *Live Zero Error Alarm* [A2] is generated on LCP.

F Wait until the motor stops.

G From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *RD* ⇒ *STW*, verify the following values.

Table 42: Tag Name and Expected Value

Tag name	Expected value
BIT_00_CTRL_RDY	FALSE
BIT_01_DRV_RDY	FALSE
BIT_02_COAST_RDY	FALSE
BIT_03_TRIP	TRUE
BIT_04_ERROR	FALSE
BIT_05_RESERVED	FALSE
BIT_06_TRIPLOCK	FALSE
BIT_07_WARNING	FALSE
BIT_08_RUNNING_REF	FALSE
BIT_09_CTRL_SOURCE	TRUE
BIT_10_F_LIMIT_OK	TRUE
BIT_11_IN_OPR	FALSE
BIT_12_IN_AUTOSTART	FALSE
BIT_13_VOLT_MAX	FALSE
BIT_14_TORQ_MAX	FALSE
BIT_15_TIMER_EXD	FALSE
MAV	0

H Set parameter 6-01 Live Zero Timeout Function to [0] Off using LCP.

I From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW* ⇒ *BIT_07_RESET*, set the value *TRUE* in Monitor Value field.

J Verify whether Live Zero Error Alarm [A2] is removed on LCP.

K From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *RD* ⇒ *STW*, verify the following values.

Table 43: Tag Name and Expected Value

Tag name	Expected value
BIT_00_CTRL_RDY	TRUE
BIT_01_DRV_RDY	TRUE
BIT_02_COAST_RDY	TRUE
BIT_03_TRIP	FALSE
BIT_04_ERROR	FALSE
BIT_05_RESERVED	FALSE
BIT_06_TRIPLOCK	FALSE
BIT_07_WARNING	FALSE

Tag name	Expected value
BIT_08_RUNNING_REF	TRUE
BIT_09_CTRL_SOURCE	TRUE
BIT_10_F_LIMIT_OK	TRUE
BIT_11_IN_OPR	TRUE
BIT_12_IN_AUTOSTART	FALSE
BIT_13_VOLT_MAX	FALSE
BIT_14_TORQ_MAX	FALSE
BIT_15_TIMER_EXD	FALSE
MAV	-16384

- L** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW* ⇒ *BIT_07_RESET*, set the value *FALSE* in *Monitor Value* field.

11. Stop the motor.

- A** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW* ⇒ *BIT_06_START*, set the value *FALSE* in *Monitor Value* field.
- B** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW* ⇒ *REF_VALUE*, set the value *0* in *Monitor Value* field.
- C** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *RD* ⇒ *STW*, verify the following values.

Table 44: Tag Name and Expected Value

Tag name	Expected value
BIT_00_CTRL_RDY	TRUE
BIT_01_DRV_RDY	TRUE
BIT_02_COAST_RDY	TRUE
BIT_03_TRIP	FALSE
BIT_04_ERROR	FALSE
BIT_05_RESERVED	FALSE
BIT_06_TRIPLOCK	FALSE
BIT_07_WARNING	FALSE
BIT_08_RUNNING_REF	FALSE
BIT_09_CTRL_SOURCE	TRUE
BIT_10_F_LIMIT_OK	TRUE
BIT_11_IN_OPR	FALSE
BIT_12_IN_AUTOSTART	FALSE
BIT_13_VOLT_MAX	FALSE
BIT_14_TORQ_MAX	FALSE
BIT_15_TIMER_EXD	FALSE
MAV	0

12. Set PCD Write & PCD Read on the drive.

- A** Set parameter 9-15 PCD Write Configuration sub-index 2–9 and parameter 9-16 PCD Read Configuration sub-index 2–9 using LCP.

Table 45: PCD Write and Read Configuration

PCD write and read configuration	Parameter name	Modify value in setup 1
9–15 [2]	PCD Write Configuration	[312] Catch up/slow down value
9–15 [3]	PCD Write Configuration	[412] Motor speed low limit [HZ]
9–15 [4]	PCD Write Configuration	[341] Ramp 1 ramp-up time
9–15 [5]	PCD Write Configuration	[341] Ramp 1 ramp-up time
9–15 [6]	PCD Write Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–15 [7]	PCD Write Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–15 [8]	PCD Write Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–15 [9]	PCD Write Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–16 [2]	PCD Read Configuration	[1614] Motor current
9–16 [3]	PCD Read Configuration	[1614] Motor current
9–16 [4]	PCD Read Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–16 [5]	PCD Read Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–16 [6]	PCD Read Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–16 [7]	PCD Read Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–16 [8]	PCD Read Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)
9–16 [9]	PCD Read Configuration	Depends on the drive configuration. (Default setting is <i>None</i> .)

13. Read process data.

- A** From `VLT_PCD_DATA [DB100]` data block $\Rightarrow$ `VLT_PCD_DATA [DB100]  $\Rightarrow$  VLT_PCD_DATA[1]  $\Rightarrow$  PPO_FC  $\Rightarrow$  RD  $\Rightarrow$  PCD_03`, read the low word value of motor current.
- B** From `VLT_PCD_DATA [DB100]` data block $\Rightarrow$ `VLT_PCD_DATA [DB100]  $\Rightarrow$  VLT_PCD_DATA[1]  $\Rightarrow$  PPO_FC  $\Rightarrow$  RD  $\Rightarrow$  PCD_02`, read the high word value of motor current.

14. Write process data.

- A** From `VLT_PCD_DATA [DB100]` data block $\Rightarrow$ `VLT_PCD_DATA [DB100]  $\Rightarrow$  VLT_PCD_DATA[1]  $\Rightarrow$  PPO_FC  $\Rightarrow$  WR  $\Rightarrow$  PCD_03`, set the value as 80.
- B** The expected value in the drive parameter 4-12 Motor Speed Low Limit [HZ] is 8.0 Hz.

3.6 Flexible Control Function Block

Procedure

1. Drag and drop `VLT_PROFI_FC_FLEXIBLE [FC301]` function block into *Main OB1* as shown in the following illustration.

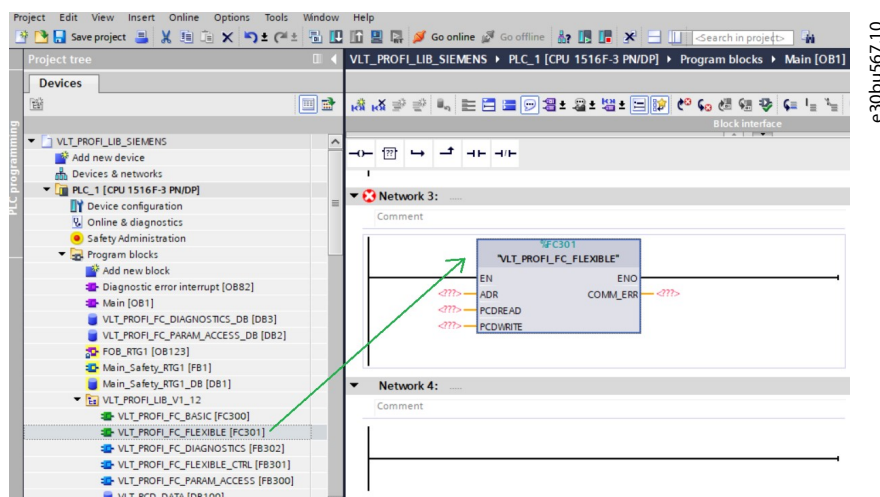


Illustration 54: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block Instance

2. Create the variables for `VLT_PROFI_FC_FLEXIBLE [FC301]` functional block input and output pins. Refer to [2.2 Creating PLC Tags for Function Blocks](#).
3. Map `VLT_PCD_DATA DB[100] ⇒ VLT_PCD_DATA[1]` data block with `PCDREAD` and `PCDWRITE` in-out pin. Refer to [2.4 Mapping of PCDREAD & PCDWRITE with VLT_PCD_DATA \[DB100\]](#).
4. Set hardware identifier for `VLT_PROFI_FC_FLEXIBLE [FC301]` function block.
 - A Note down the hardware identifier in TIA portal. Refer to [2.3.3 Finding PROFINET Device Hardware Identifier for VLT_PROFI_FC_FLEXIBLE \[FC301\]](#).
 - B Go to *Main [OB1]* block.
 - C Enter the value of ADR inputs, as noted from hardware identifier as shown in the following illustration.

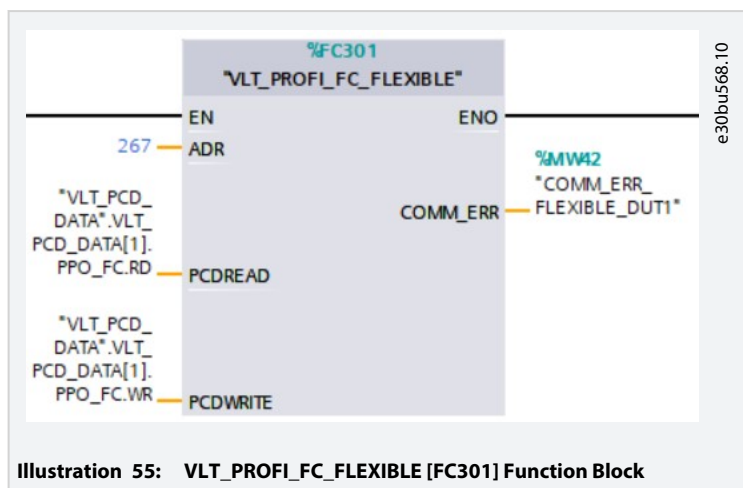


Illustration 55: VLT_PROFI_FC_FLEXIBLE [FC301] Function Block

5. Drag and drop `VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]` function block into *Main OB1* as shown in the following illustration.

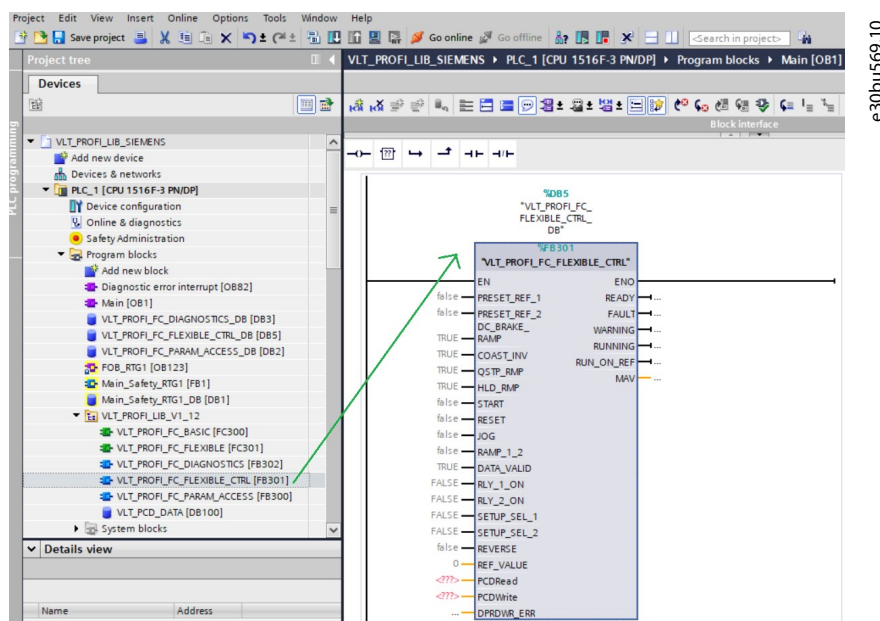
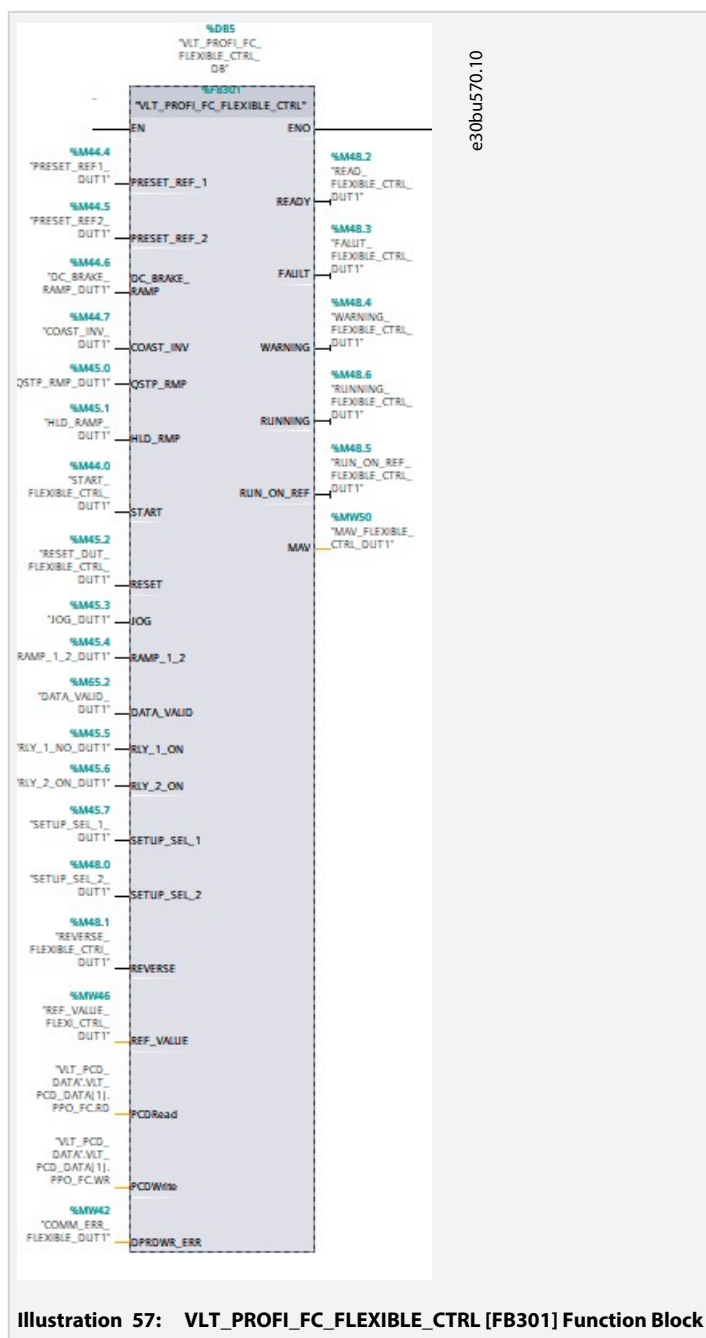


Illustration 56: VLT_PROFI_FC_FLEXIBLE_CTRL [FB301] Function Block Instance

6. Create the variables for VLT_PROFI_FC_FLEXIBLE_CTRL [FB301] functional block input and output pins. Refer to [2.2 Creating PLC Tags for Function Blocks](#).
7. Map VLT_PCD_DATA DB[100] ⇒ VLT_PCD_DATA[1] data block with PCDREAD and PCDWRITE in-out pin. Refer to [2.4 Mapping of PCDREAD & PCDWRITE with VLT_PCD_DATA \[DB100\]](#).
8. Go to VLT_PROFI_FC_FLEXIBLE [FC301] function block, COMM_ERR output pin tag is created as COMM_ERR_FLEXIBLE_DUT1. Use the same tag (COMM_ERR_FLEXIBLE_DUT1) as the input of DPRDWR_ERR as shown in the following illustration.



9. To check that the communication is OK, verify the following values with the *VLT_PROFI_FC_FLEXIBLE [FC301]* function block output.

Table 46: Pin Name and Expected Value

Pin name	Expected value
COMM_ERR	0 (0 means the communication is OK)

10. Enable the fieldbus control.

- A** Set value *TRUE* for *DATA_VALID* input terminal on the *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- B** From *VLT_PCD_DATA [DB100]* ⇒ *Static* ⇒ *VLT_PCD_DATA* ⇒ *VLT_PCD_DATA[1]* ⇒ *PPO_FC* ⇒ *WR* ⇒ *CTW*, verify the following values.

Table 47: Pin Name and Expected Value

Pin name	Expected value
BIT_10_DATA_VALID	TRUE

- C** Verify the following output values of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 48: Pin Name and Expected Value

Pin name	Expected value
READY	FALSE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
DPRDWR_ERR	0

- 11.** Enable the drive to *Ready* state.

- A** Set the following values to the *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block input pins.

Table 49: Pin Name and Set Value

Pin name	Set value
PRESET_REF_1	FALSE
PRESET_REF_2	FALSE
DC_BRAKE_RAMP	TRUE
COAST_INV	TRUE
QSTP_RMP	TRUE
HLD_RMP	TRUE
START	FALSE
RESET	FALSE
JOG	FALSE
RAMP_1_2	FALSE
DATA_VALID	TRUE
RLY_1_ON	FALSE
RLY_2_ON	FALSE
SETUP_SEL_1	FALSE
SETUP_SEL_2	FALSE
REVERSE	FALSE
REF_VALUE	0

- B** Verify the following output values of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 50: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
DPRDWR_ERR	0

- C** Set the following values in *parameter 3-10 Preset Reference* using the drive LCP.

Table 51: Set the Values in the Drive Using LCP

Parameter number	Value
3-10 [0]	00.00
3-10 [1]	00.00
3-10 [2]	00.00
3-10 [3]	00.00

- 12.** Start the motor in forward direction.

- A** Set the value *TRUE* to the *START* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- B** Set the value *8192* to the *REF_VALUE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- C** Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 52: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	8192
DPRDWR_ERR	0

- D** Set the value *FALSE* to the *COAST_INV* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- E** Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 53: Pin Name and Expected Value

Pin name	Expected value
READY	FALSE
FAULT	FALSE

Pin name	Expected value
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
DPRDWR_ERR	0

F Set the value *FALSE* to the *START* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

13. Start the motor in forward direction.

A Set the value *TRUE* to the *COAST_INV* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

B Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 54: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
DPRDWR_ERR	0

C Set the value *TRUE* to the *START* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

D Set the value *12288* to the *REF_VALUE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

E Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 55: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	12288
DPRDWR_ERR	0

F Set the value *16384* to the *REF_VALUE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

G Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 56: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	16384
DPRDWR_ERR	0

14. Start the motor in reverse direction.

- A** Set the value *TRUE* to the *REVERSE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- B** Wait until the motor ramps down and runs in the reverse direction.
- C** Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 57: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	-16384
DPRDWR_ERR	0

- D** Set the value *FALSE* to the *START* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- E** Set the value *FALSE* to the *REVERSE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.
- F** Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

Table 58: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
DPRDWR_ERR	0

- G** Set the value *TRUE* to the *START* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL [FB301]* function block.

- H** Set the value -16384 to the *REF_VALUE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.
- I** Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 59: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	-16384
DPRDWR_ERR	0

15. Reset the drive.

- A** Set the following parameter values in the drive using LCP.

Table 60: Parameter Settings

Parameter	Modify value
<i>Parameter 6-00 Live Zero Timeout Time</i>	20 s
<i>Parameter 6-01 Live Zero Timeout Function</i>	[5] Stop and trip
<i>Parameter 6-10 Terminal 53 Low Voltage</i>	2.00 V

- B** Verify whether *Live Zero Error Warning* [W2] is generated on LCP.
- C** Verify the following values in *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 61: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	TRUE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	-16384
DPRDWR_ERR	0

- D** Wait for 20 s.
- E** Verify whether *Live Zero Error Alarm* [A2] is generated on LCP.
- F** Wait until the motor stops.
- G** Verify the following values in *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 62: Pin Name and Expected Value

Pin name	Expected value
READY	FALSE
FAULT	TRUE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0
DPRDWR_ERR	0

- H** Set parameter 6-01 Live Zero Timeout Function to [0] Off using LCP.
- I** Set the value *TRUE* to the *RESET* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.
- J** Verify whether *Live Zero Error Alarm* [A2] is removed on LCP.
- K** Verify the following values in *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 63: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
MAV	-16384
DPRDWR_ERR	0

- L** Set the value *FALSE* to the *RESET* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

16. Stop the motor.

- A** Set the value *FALSE* to the *START* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.
- B** Set the value 0 for *REF_VALUE* input pin of *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.
- C** Verify the following output values in *VLT_PROFI_FC_FLEXIBLE_CTRL* [FB301] function block.

Table 64: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
MAV	0

Pin name	Expected value
DPRDWR_ERR	0

Index

A

Acyclic communication 8

B

Basic operation function block 5, 5, 37

C

Control and monitoring 5

D

Diagnostics function block 5, 9, 44

F

Failure management 5

Flexible control function block 5, 12, 56

Flexible function block 5, 10, 47

G

General configuration 36

L

Library 5

M

MCA 101 4

MCA 120 4

P

Parameter access function block 5, 8, 41

R

Reverse 5

S

Speed regulation 5

ENGINEERING
TOMORROW

Danfoss

.....
Danfoss can accept no responsibility for possible errors in catalogues, brochures and other printed material. Danfoss reserves the right to alter its products without notice. This also applies to products already on order provided that such alterations can be made without subsequential changes being necessary in specifications already agreed. All trademarks in this material are property of the respective companies. Danfoss and the Danfoss logotype are trademarks of Danfoss A/S. All rights reserved.
.....

Danfoss A/S
Ulsnaes 1
DK-6300 Graasten
vlt-drives.danfoss.com

