

Programming Guide

Function Blocks with B & R Automation Studio

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 (VLTEPLV100)	5
1.6	Basic Operation Function Block (VLT_EPL_FC_BASIC)	5
1.7	Parameter Access Function Block (VLT_EPL_FC_PARAM_ACCESS)	10
2	Using FBs in B & R Automation Studio	12
2.1	Importing Danfoss Library into a Project	12
2.2	Adding Danfoss XDD File	17
2.3	Adding Slave Device (Controlled Node)	19
2.4	Instantiating Function Block	21
2.5	Adding Process Variable	27
2.6	Building and Downloading the Project to the PLC	31
3	Examples	35
3.1	General Configuration of the Drive	35
3.2	Basic Operation Function Block	36
3.3	Parameter Access Function Block	40

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 B & R Automation Studio V4.3 or V4.4 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 B & R Automation Studio V4.3 or V4.4 system.
- Procedure on using the library to communicate with Danfoss VLT® drives in B & R Automation Studio V4.3 or V4.4 system, including examples.

Function blocks for B & R Automation Studio supporting Danfoss VLT® drives.

Danfoss VLT® drives with the following options:

- VLT® POWERLINK MCA 123

The manual is intended for use by qualified personnel.

1.3 Abbreviations

Abbreviation	Description
FB	Function block
CAN	Controller area network
CPU	Central processor unit
DUT	Device under test
ETH	Ethernet
IP	Internet protocol
PLC	Programming logic control
SS	Sub slot
SDO	Service data object
XDD	XML device description

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 (VLTEPLV100)

The *VLTEPLV100* 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 B & R Automation Studio systems and Danfoss drives.

The library contains the following FBs:

- **Basic operation function block (VLT_EPL_FC_BASIC):** Dedicated to handling the basic operation of the drive and connected motor operations.
- **Parameter access function block (VLT_EPL_FC_PARAM_ACCESS):** Dedicated to parameter read/write through an acyclic channel.

The function blocks are designed to work with Danfoss proprietary *FC profile* only.

1.6 Basic Operation Function Block (VLT_EPL_FC_BASIC)

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_EPL_FC_BASIC			
BOOL	DRV_EN	READY	BOOL
BOOL	RUN	FAULT	BOOL
BOOL	REVERSE	WARNING	BOOL
INT	REF_VALUE		
BOOL	RET	RUNNING	BOOL
		RUN_ON_REF	BOOL
UINT	COMM_STATUS	COMM_OK	BOOL
		MAV	INT
		MOTORCURRENT	REAL
BASIC_VLT_IN_DDT	PCDREAD	PCDWRITE	BASIC_VLT_OUT_DDT

e30bu572.10

Illustration 1: Basic Operation Function Block Layout

Table 1: Input Parameter

Parameter	Type	Description
DRV_EN	BOOL	<i>TRUE</i> = enabling drive to <i>Ready</i> state. The 4 bits, CTW.02 (DC brake), CTW.03 (coasting), CTW.04 (quick stop), CTW.05 (hold output frequency) are set to <i>TRUE</i> to move the drive to <i>READY</i> state.
RUN	BOOL	<i>TRUE</i> = starts the motor run in the direction selected based on the <i>REVERSE</i> input variable. The CTW.06 (ramp start) bit is set to <i>TRUE</i> to start the motor.
REVERSE	BOOL	The direction of rotation of the motor. <i>FALSE</i> = activates the forward direction. <i>TRUE</i> = activates the reverse direction. The CTW.15 (reverse) bit is set to <i>TRUE</i> to change the direction of the motor.
REF_VALUE	INT	The reference value requests to the drive to run at reference speed and only accepts positive values. The range value 0–10000 equal to 0.00% to 100.00%. Enter the reference value without decimal. For example: to run the drive at 56.75%, enter <i>REF_VALUE</i> as 5675 to achieve the motor speed.
RESET	BOOL	<i>TRUE</i> = resets the device failure and resets the <i>FAULT</i> output to <i>FALSE</i> . The CTW.07 (reset) bit is set to <i>TRUE</i> to reset the failure.
COMM_STATUS	BOOL	Indicates that the node is present on the bus.

Parameter	Type	Description	
PCDREAD	BA-SIC_VLT_IN_DDT	Process data sent from the drive contains information about the current state of the drive. Holds a structure with the data obtained from the drive. This input is reserved for the function block, and it is recommended not using this input directly.	
	Parameter	Type	Description
	STATUS_WORD	UINT	Status word.
	MAIN_ACTUAL_VALUE	INT	Main actual value.
	MOTORCURRENT	DINT	Motor current in Amps.

Table 2: Output Parameter

Parameter	Type	Description
READY	BOOL	<i>TRUE</i> = the drive is ready for operation. The 4 bits STW.00 (control ready), STW.01 (drive ready), STW.02 (coast stop), STW.09 (bus control) are considered for <i>READY</i> state.
FAULT	BOOL	<i>TRUE</i> = a detected failure in the control block. To reset the <i>FAULT</i> output pin, activate the <i>RESET</i> input. The 3 bits, STW.03 (drive trips), STW.04 (displays error but not tripped), STW.06 (trip lock), are considered for <i>FAULT</i> state.
WARNING	BOOL	<i>TRUE</i> = a warning has been activated for the drive. It cannot be reset because the signal remains active until the cause of the warning is removed. The STW.07 (warning) bit is considered for <i>WARNING</i> state.
RUNNING	BOOL	<i>TRUE</i> = the drive is running and has an output frequency (MAV>0). The STW.11 (in operation) bit is considered for motor running status.
RUN_ON_REF	BOOL	<i>TRUE</i> = the actual motor speed reaches the preset speed reference. The STW.08 (speed = reference) bit is considered for motor running on preset speed reference.
COMM_OK	BOOL	<i>TRUE</i> = the communication between the device and PLC is OK.
MAV	INT	Main actual value in %, expressed in integer value 0–10000. For example: the MAV value is 9949 means that the drive is running at 99.49%.
MOTORCURRENT	REAL	Motor current in Amps.

Parameter	Type	Description	
PCDWRITE	BA-SIC_VLT_OUT_DDT	Process data sent from the PLC to the drive.	
		Holds a structure with data sent to the drive. The drive can be controlled with this output variable.	
	Parameter	Type	Description
	CONTROL_WORD	UINT	Control word.
	REF_VALUE	INT	Reference value.

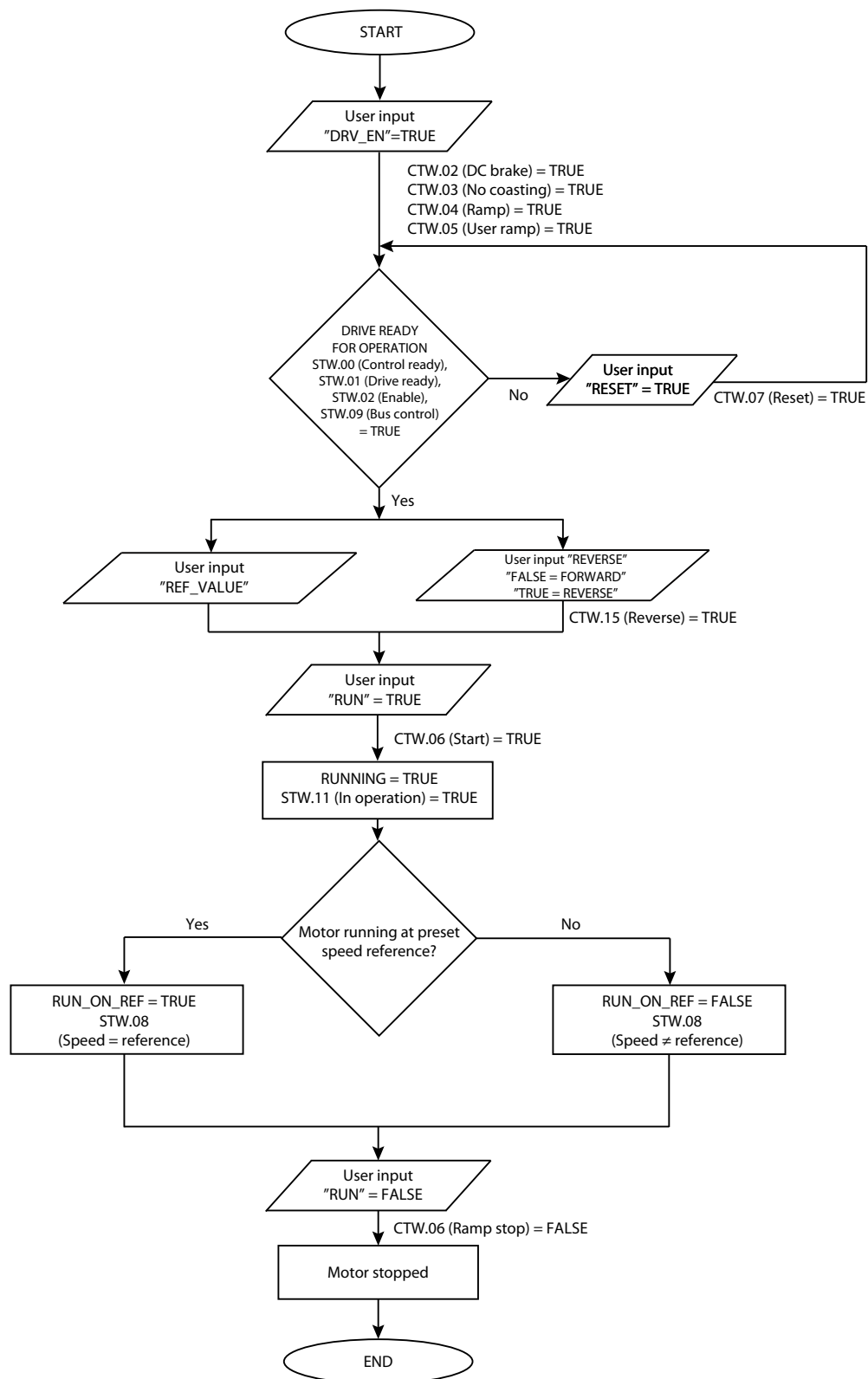


Illustration 2: Flow Chart for Basic Operation Of Drive Control

1.7 Parameter Access Function Block (VLT_EPL_FC_PARAM_ACCESS)

The function block provides functionality to read and write Danfoss drive parameters through SDO service.

- Carry out read/write operation (only one operation at a time).
- Read string type parameters.
- Write string type parameters.
- Read non-string data type parameters.
- Write non-string data type parameters.
- Read array type parameters.
- Write array type parameters.
- Show valid abort code on invalid read-write operations.

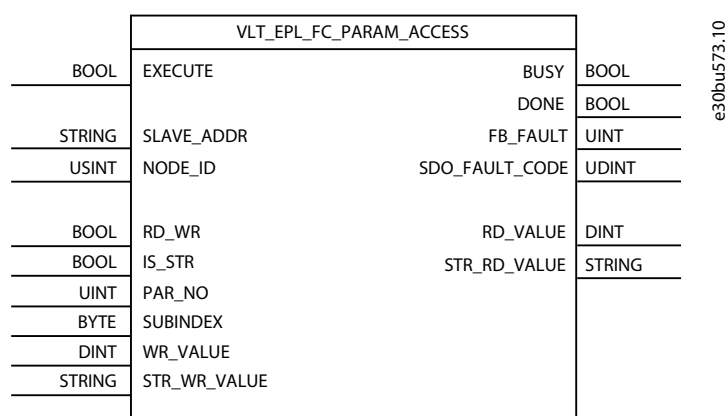


Illustration 3: Parameter Access Function Block Layout

The following elementary function blocks are embedded inside *VLT_EPL_FC_PARAM_ACCESS* function block.

- *EpISDORead*: a function block which allows data to be read from an Ethernet POWERLINK slave through an SDO access.
- *EpISDOWrite*: a function block which allows an object from the object directory of an Ethernet POWERLINK slave to be written with an SDO download.

Table 3: Input Parameter

Parameter	Type	Description
EXECUTE	BOOL	<i>TRUE</i> = the rising edge of this signal starts the requested operation. <i>FALSE</i> = triggers an ACK of the end-of-operation notification and the client is ready for the next cycle. NOTICE When the signal is activated, it copies the parameters to the function, so modifying the parameters has no effect.
SLAVE_ADDR	STRING	The device name of the POWERLINK interface. The syntax is of the following format: SL<x>. SS<y>. IF<z>. For example: SS1.1F1.ST1.
NODE_ID	UINT	Node address of the drive is entered in this input. Node address ranges: 1–255.

Parameter	Type	Description
RD_WR	BOOL	Type of operation. <i>FALSE</i> = read operation. <i>TRUE</i> = write operation.
IS_STR	BOOL	Indicates operation to be performed based on the parameter data type. Set the input to: <i>FALSE</i>: to perform the operation based on <i>RD_WR</i> input for non-string data type (UDINT, DINT, INT, and so on) parameter. <i>TRUE</i>: to perform the operation based on <i>RD_WR</i> input for string data type parameter.
PAR_NO	UINT	Enter the drive parameter number in the integer format. Example: to read the running hours, enter <i>PAR_NO</i> as 1501. Reading the running hours in <i>parameter 15-01 Running Hours</i> is internally calculated by 2000 h+ parameter number in hex number = 2000 h +5DD=index 25DDh.
SUBINDEX	BYTE	Sub-index of the object that is to be read/write. If no sub-index available for the particular parameter, it is entered as 0.
WR_VALUE	DINT	Integer data to be written to the drive.
STR_WR_VALUE	STRING	String data are written to the drive.

Table 4: Output Parameter

Parameter	Type	Description
BUSY	BOOL	<i>TRUE</i> = the client is processing the request and waiting for the response. <i>FALSE</i> = the client processed the request with status (Done or Error).
DONE	BOOL	<i>TRUE</i> = the request operation has ended (with or without a detected error). This signal is acknowledged by setting the <i>Start</i> signal to <i>Low</i> .
FB_FAULT	UINT	The requested operation was unsuccessful. Check the function block configuration, then reactivate the client by setting the <i>EXECUTE</i> signal to <i>High</i> . Refer to the error number part in help file of EplSDORRead or EplSDOWrite function block for more information.
SDO_FAULT_CODE	UDINT	The requested operation was unsuccessful and showed fault codes. Refer to the SDO abort code part in help file of EplSDORRead or EplSDOWrite function block for more information.
RD_VALUE	DINT	Holds the read data of integer values depending on the read input parameter.
STR_RD_VALUE	STRING	Holds the read data of string type depending on the read input parameter.

2 Using FBs in B & R Automation Studio

2.1 Importing Danfoss Library into a Project

Procedure

1. From *Logical View*, navigate to *DS_FB_Project* (project name)⇒*Libraries*.

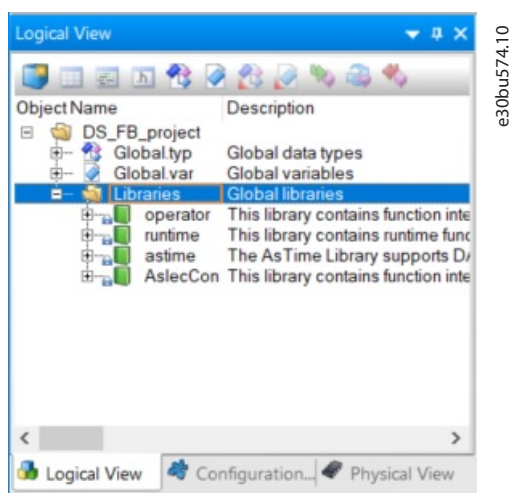
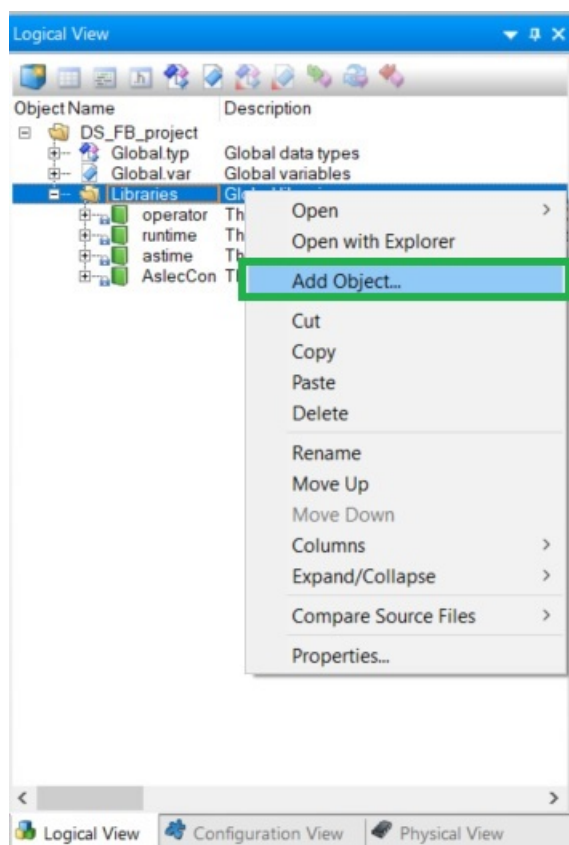


Illustration 4: Logical View

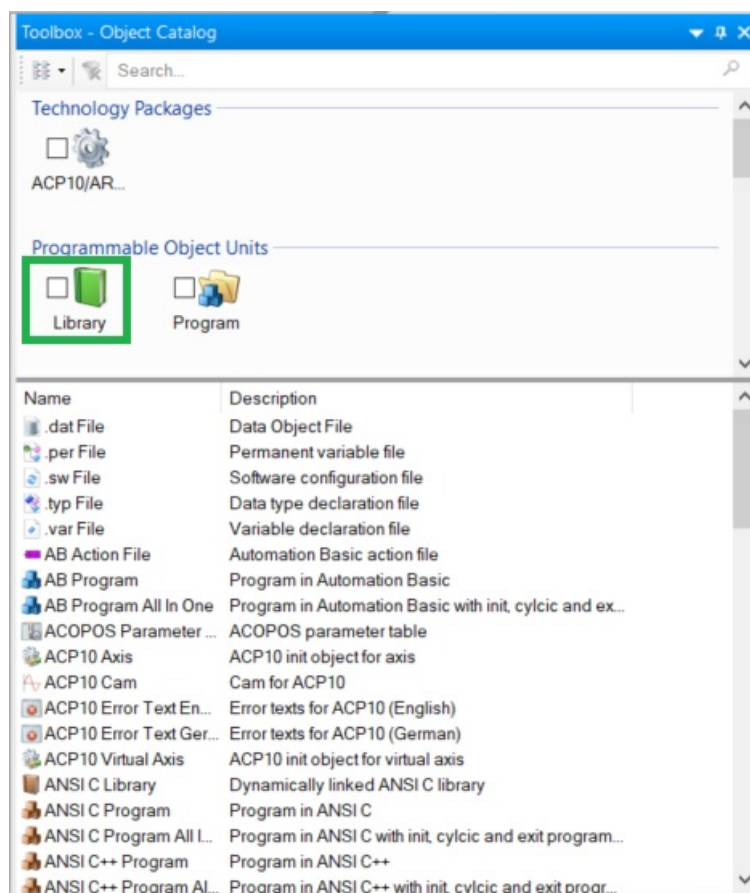
2. Right-click *Libraries*, and select *Add Object....*



e30bu575.10

Illustration 5: Add Object (PLC Library)

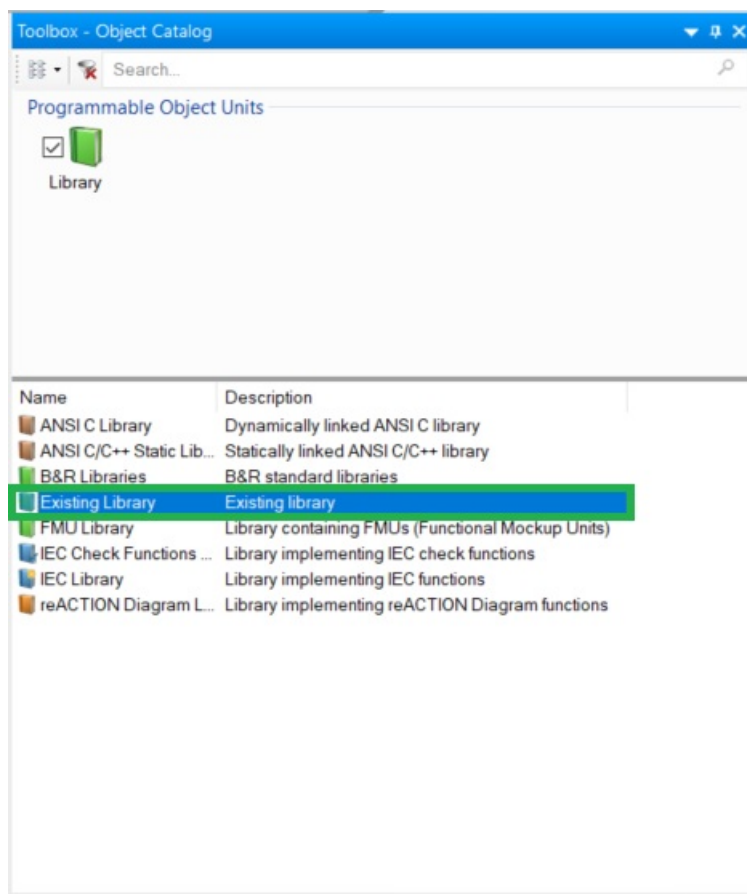
3. Select *Library* option from *Toolbox–Object Catalog*.



e30bu576:10

Illustration 6: Library Option in Object Catalog

4. Double-click *Existing Library* window.



e30bu577.10

Illustration 7: Existing Library Option

5. Navigate to the library downloaded path, and click *Finish*.

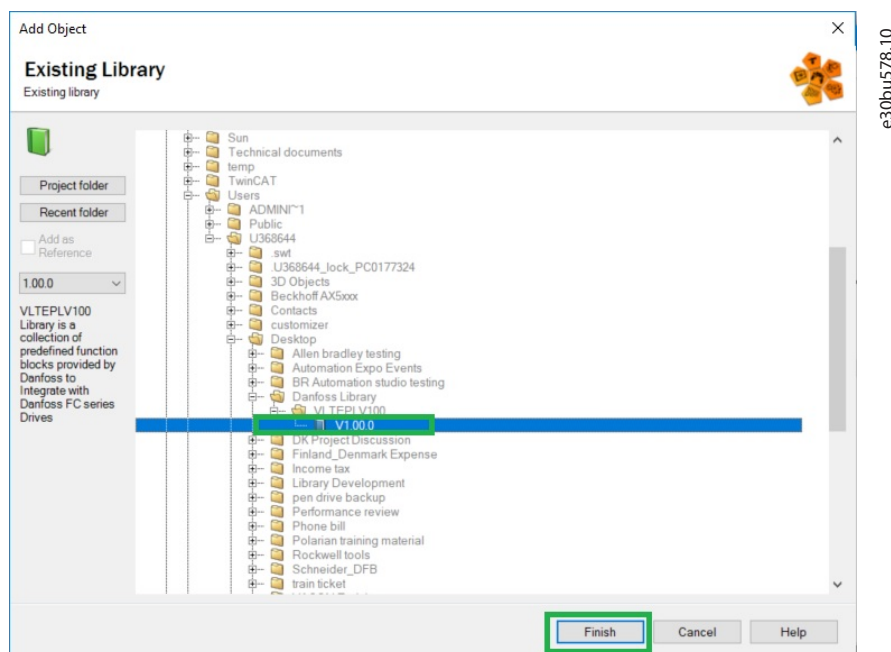


Illustration 8: Selecting the Existing Library Path

6. Verify whether the libraries are successfully imported.

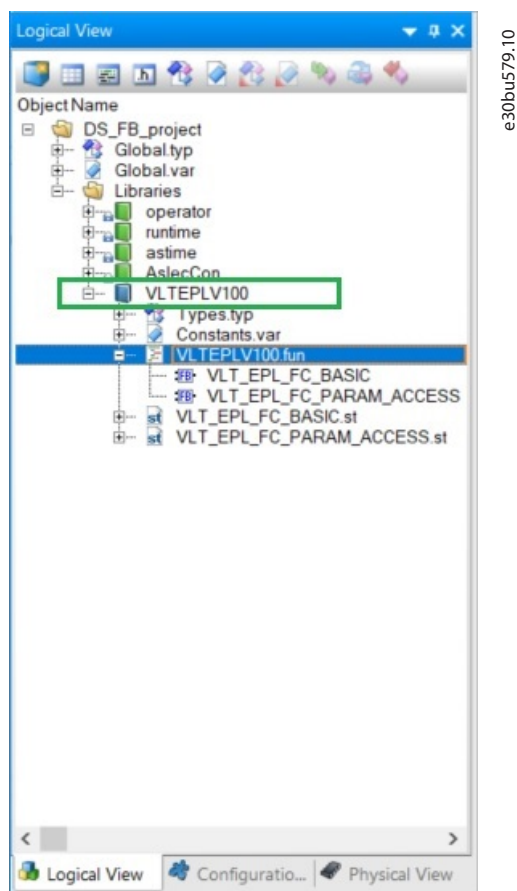


Illustration 9: Verification of Library Installation

NOTICE

To use *VLTEPLV100* library, import the following relevant libraries from B & R standard library manually.

- AsEPL
- AsBrStr

2.2 Adding Danfoss XDD File

Context:

The following steps explain how to add Danfoss XDD file in the project.

Procedure

1. Click **Tools** menu, and select **Import Fieldbus Device....**

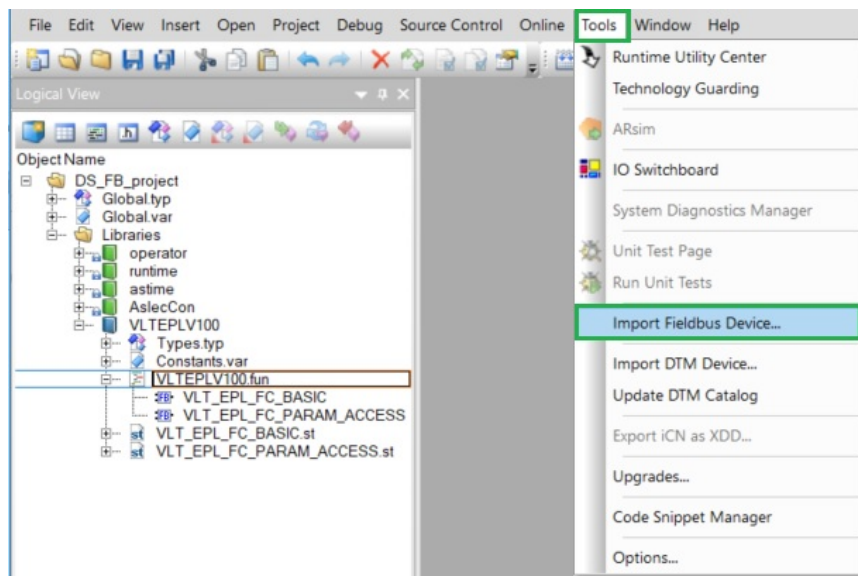


Illustration 10: Importing XDD file

2. Select the file and click **Open**.

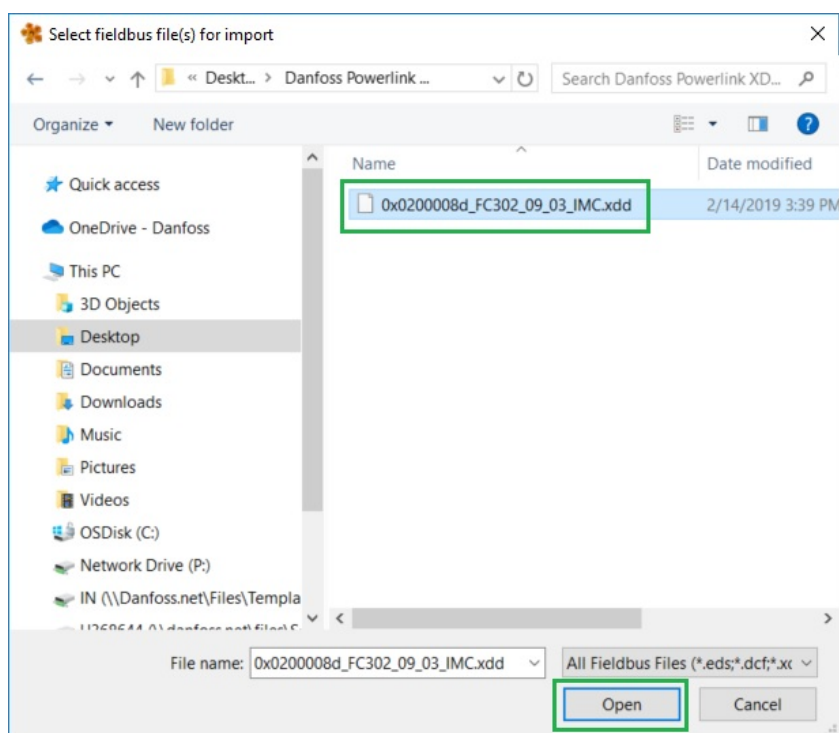


Illustration 11: Selecting the XDD File

2.3 Adding Slave Device (Controlled Node)

Context:

The following steps explain how to add a slave device (controlled node) in the project.

Procedure

1. From *Physical View*, navigate to the *PLC*⇒*PLK* (POWERLINK port).
2. Right-click *PLK*⇒*Add Hardware Module...*

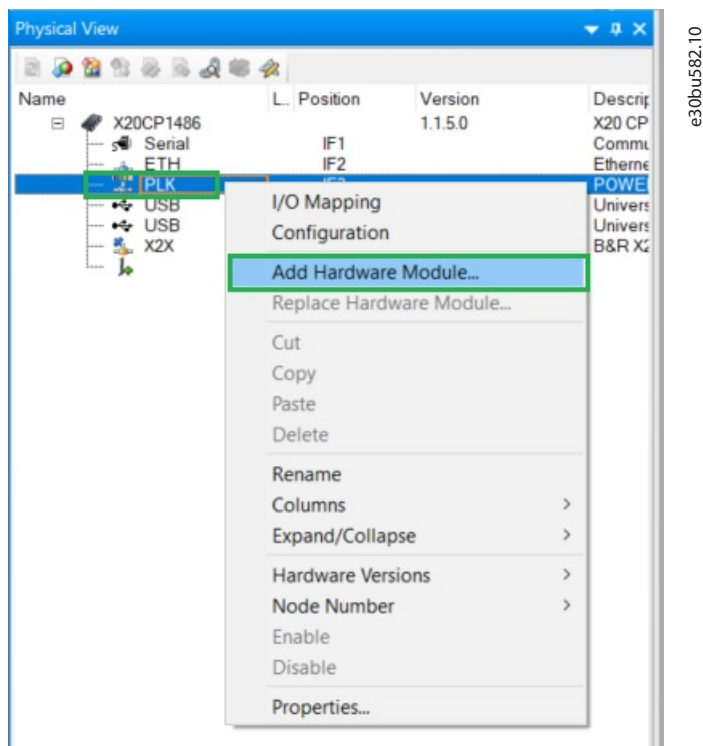
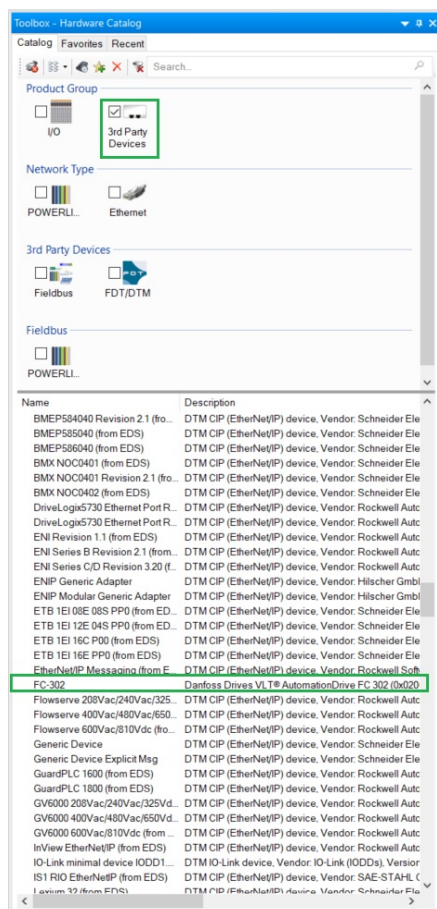


Illustration 12: Add Slave Device (Controlled Node)

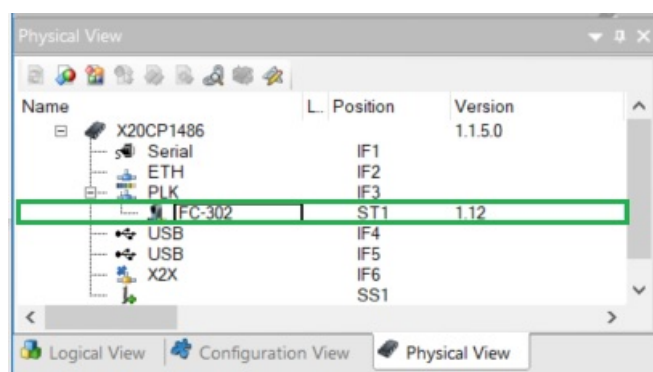
3. In *Toolbox-Hardware Catalogue*, select *3rd Party Devices*.
4. Select *FC-302*.



e30bu583.10

Illustration 13: Toolbox Hardware Catalog Window

- Double-click FC-302, or drag and drop FC-302 under PLK port.



e30bu584.10

Illustration 14: Controlled Node Successfully Added

2.4 Instantiating Function Block

Context:

The following steps explain how to instantiate function block in the project.

Procedure

1. From *Logical View*, right-click the project and select *Add Object...*

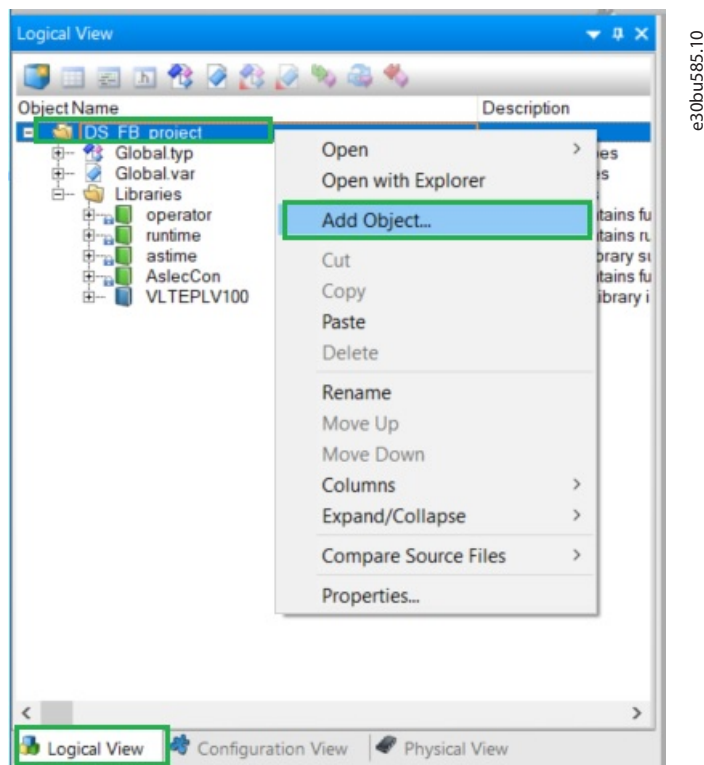


Illustration 15: Adding IEC 61131 Program

2. In *Toolbox-Hardware Catalogue*, select *Program*, and then select *LD program* (ladder logic).

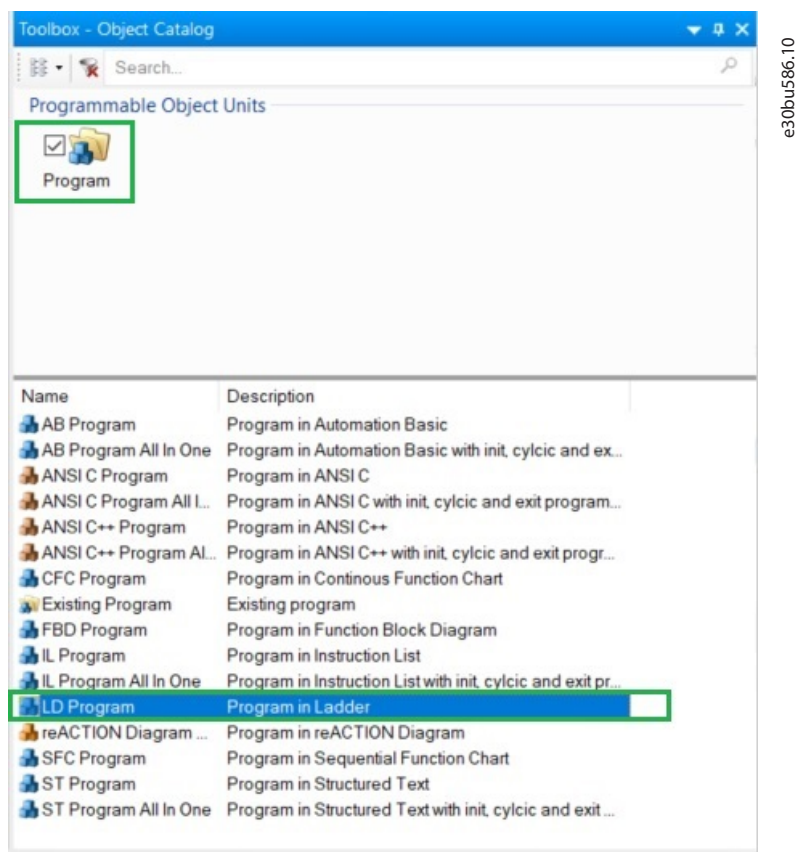
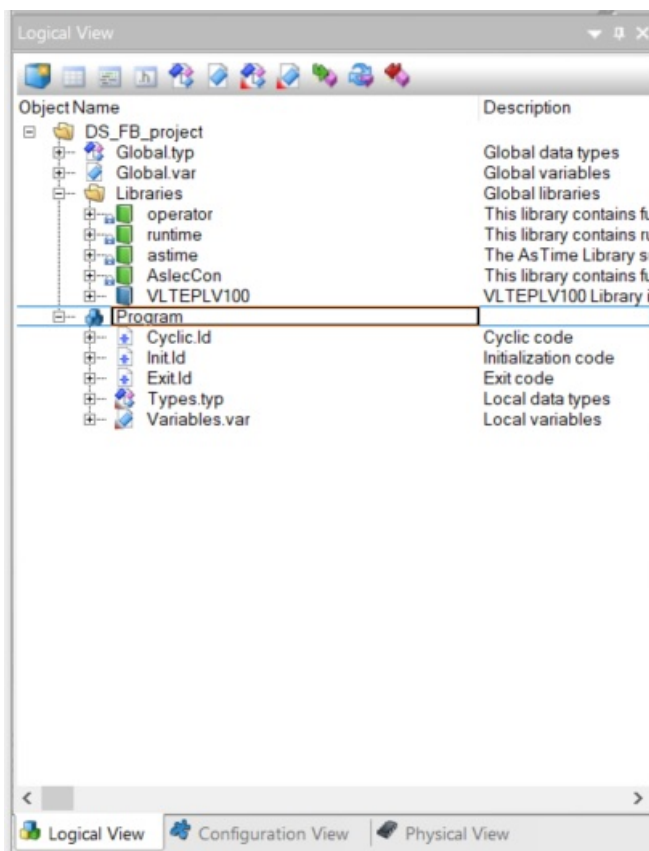


Illustration 16: Adding 61131-3 Program

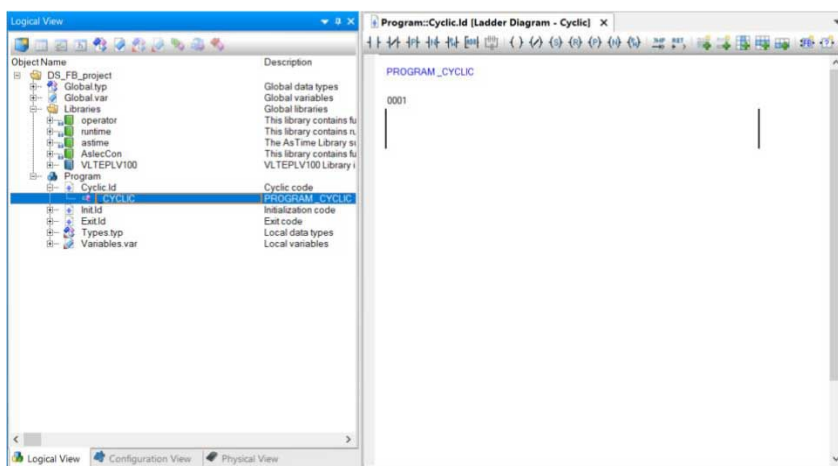
3. Double-click the LD program, or drag and drop the LD program under the current project.



e30bu587.10

Illustration 17: Ladder Logic Program Added to the Project

4. Select *Program* and double-click *Cyclic.id*.



e30bu588.10

Illustration 18: Selecting Cyclic.id

5. Instantiate *Normally Open Contact* and *Coil* in the network, and assign a local variable as *EnableIn* and *EnableOut* respectively.

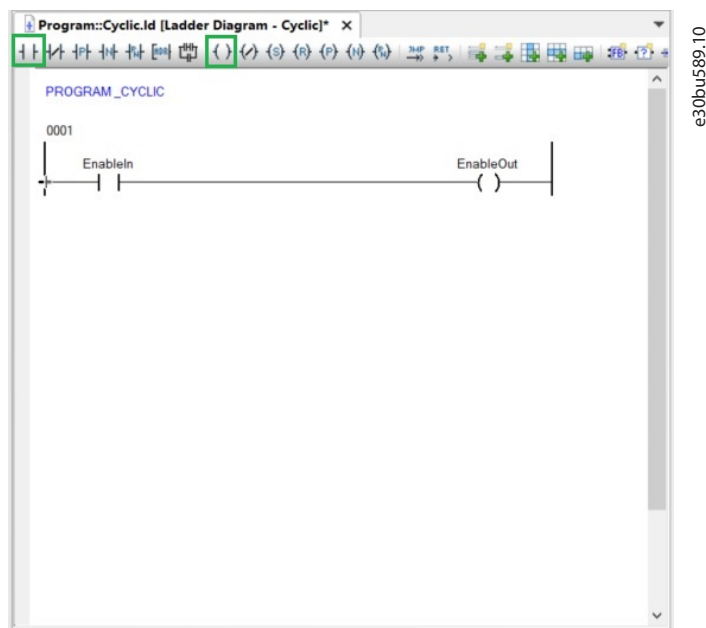


Illustration 19: Selecting Cyclic.Id

6. Select insert function/function block.

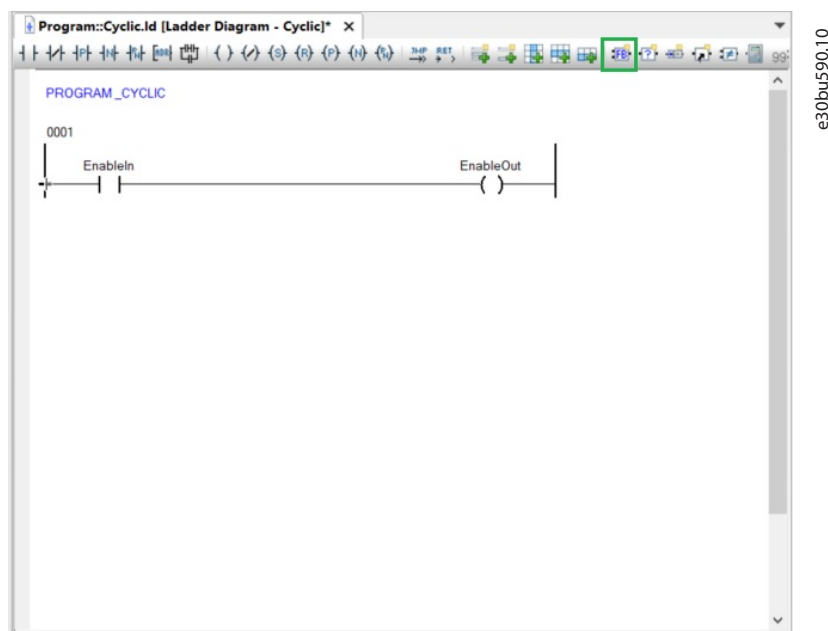


Illustration 20: Selecting Insert Function/Function Block

7. Select Data Type dialog appears, select the function block from the imported library.

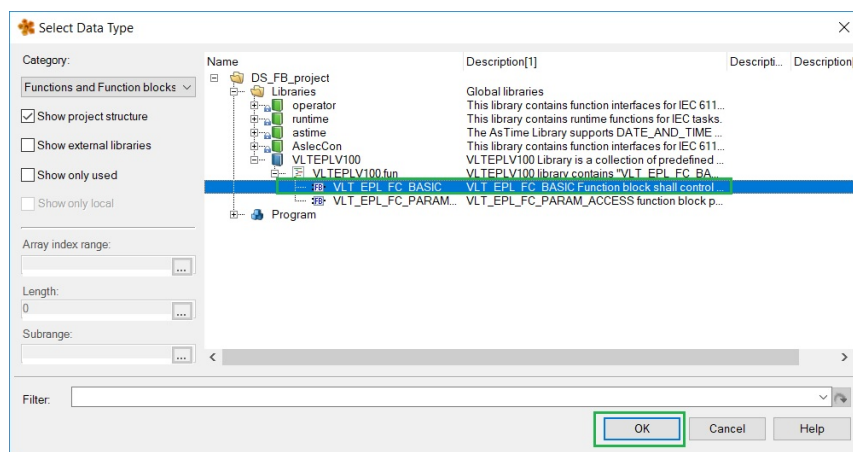


Illustration 21: Selecting Data Type

8. Click OK.

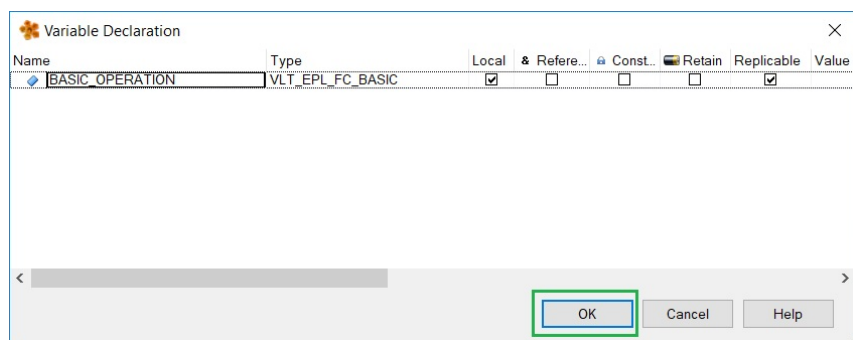


Illustration 22: Variable Declaration

9. Create and assign a variable for each input and output pin of the function block.

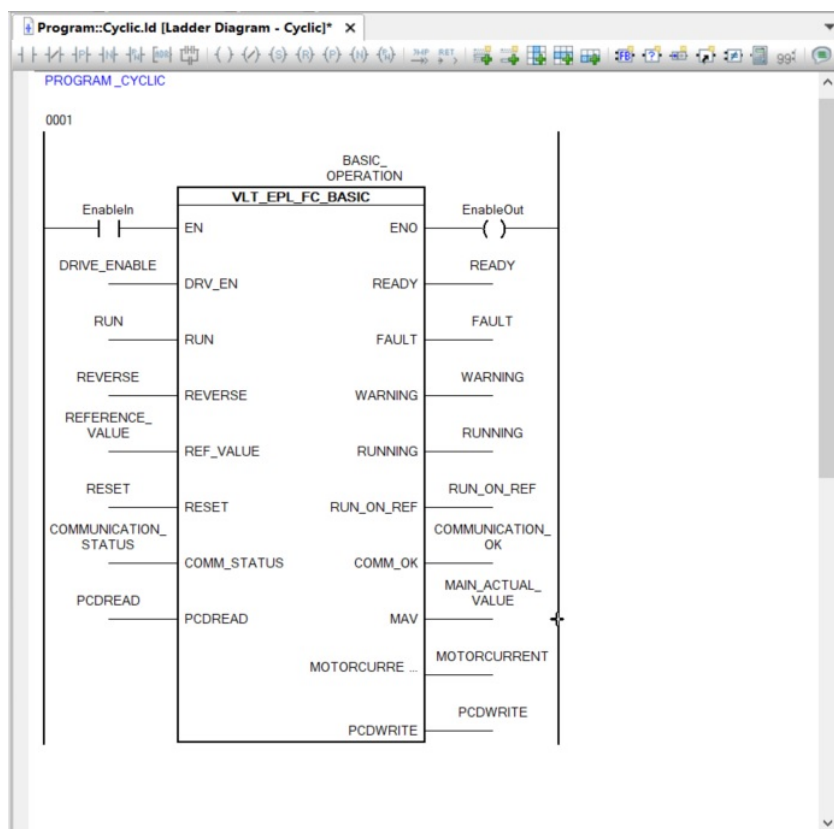


Illustration 23: Local Variable Assignment

10. Save all changes.

2.5 Adding Process Variable

Context:

The following steps explain how to add the process variables in the controlled node.

Procedure

1. From *Physical View*, navigate to the controlled node.
2. Right-click the controlled node and select *Configuration*.

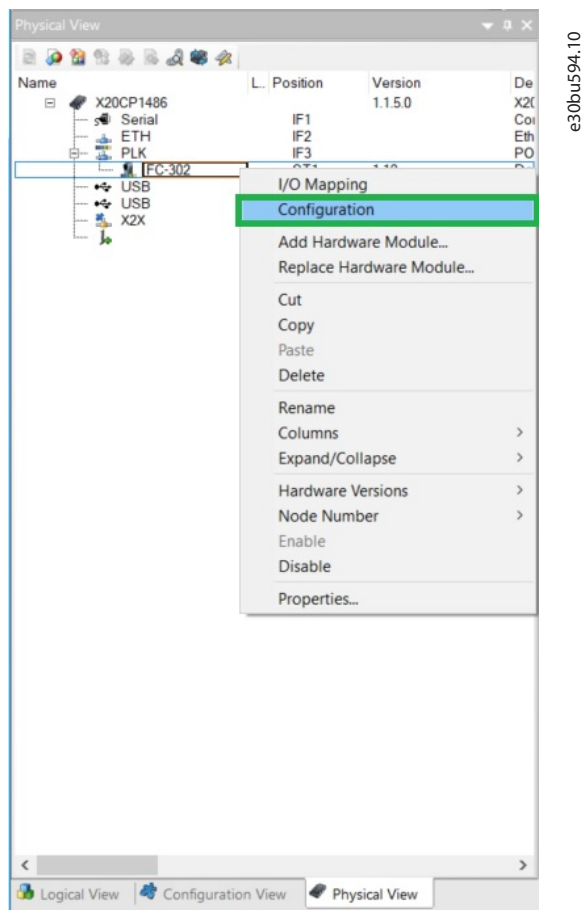
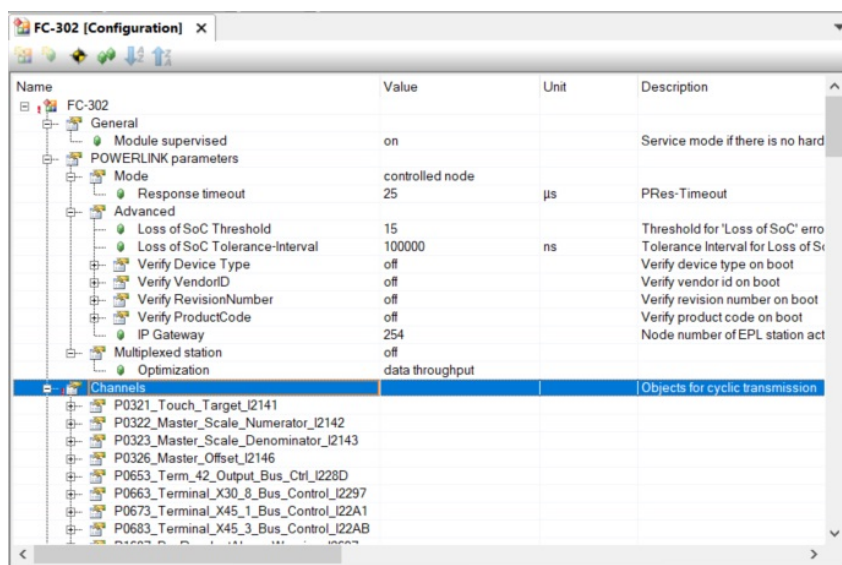


Illustration 24: Controlled Node Configuration

3. In the channel, change the value of parameters as the following table.

Table 5: Parameter Values

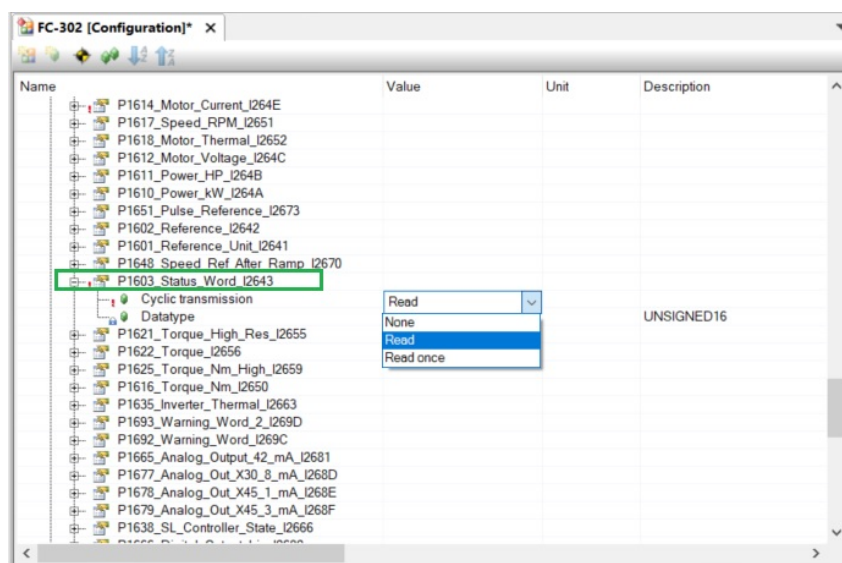
Parameter	Value
P1603_Status_Word_I2643	Read
P1605_Main_Actual_Value_I2645	Read
P1614_Motor_Current_I264E	Read
P1680_Fieldbus_CTW_1_I2690	Write
P1682_Fieldbus_REF_1_I2692	Write



e30bu595.10

Illustration 25: Parameters to be Added

4. Change the parameter value to read or write.



e30bu596.10

Illustration 26: Changing the Parameter Values

5. Save the project.
6. Right-click the controlled node.
7. Click the properties.

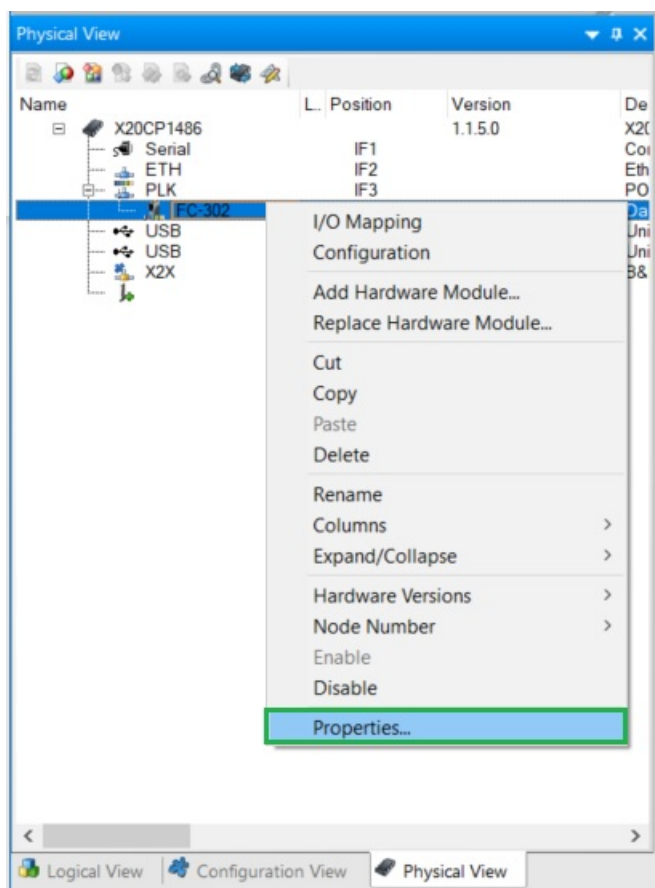


Illustration 27: Adding Process Variables

8. Double-click the process variable field.
9. Click the ellipsis icon.

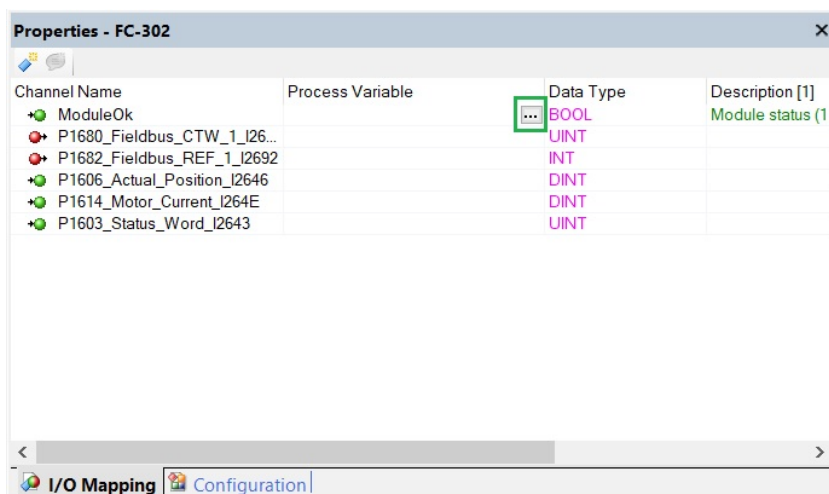


Illustration 28: Adding Process Variables

10. Add the link variable in the process variable from the function block instance as the following table.

Table 6: Parameter Link Variables

Parameter	Link variable to be added
Module ok	::Program:DUT_1_BASIC.COMM_STATUS
P1603_Status_Word_I2643	::Program:DUT_1_BASIC.PCDREAD.STATUS_WORD
P1605_Main_Actual_Value_I2645	::Program:DUT_1_BASIC.PCDREAD.MAIN_ACTUAL_VALUE
P1614_Motor_Current_I264E	::Program:DUT_1_BASIC.PCDREAD.MOTORCURRENT
P1680_Fieldbus_CTW_1_I2690	::Program:DUT_1_BASIC.PCDWRITE.CONTROL_WORD
P1682_Fieldbus_REF_1_I2692	::Program:DUT_1_BASIC.PCDWRITE.REF_VALUE

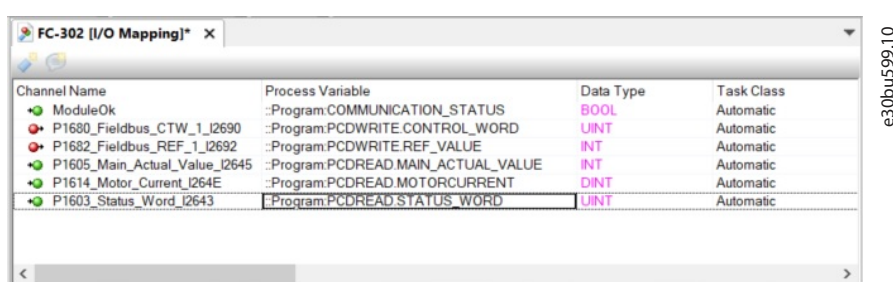


Illustration 29: Adding Process Variables

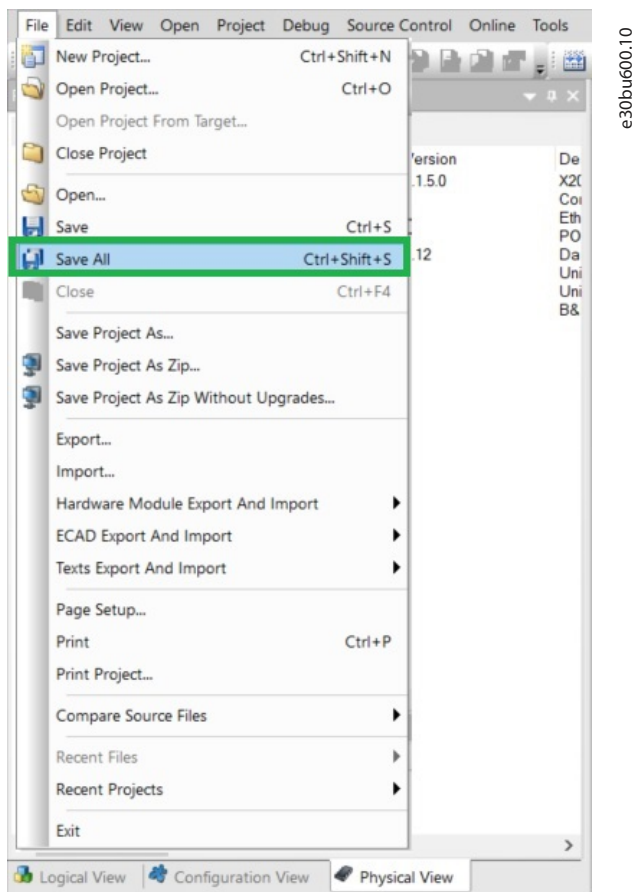
2.6 Building and Downloading the Project to the PLC

Context:

The following steps explain how to build and download the configuration to the PLC. Ensure that the PLC is connected online and in *RUN* state.

Procedure

1. To save the project, select *File*⇒*Save All* from the menu bar.

**Illustration 30: Save All Configuration**

2. To build the configuration, select *Project*⇒*Build Configuration* from the menu bar.

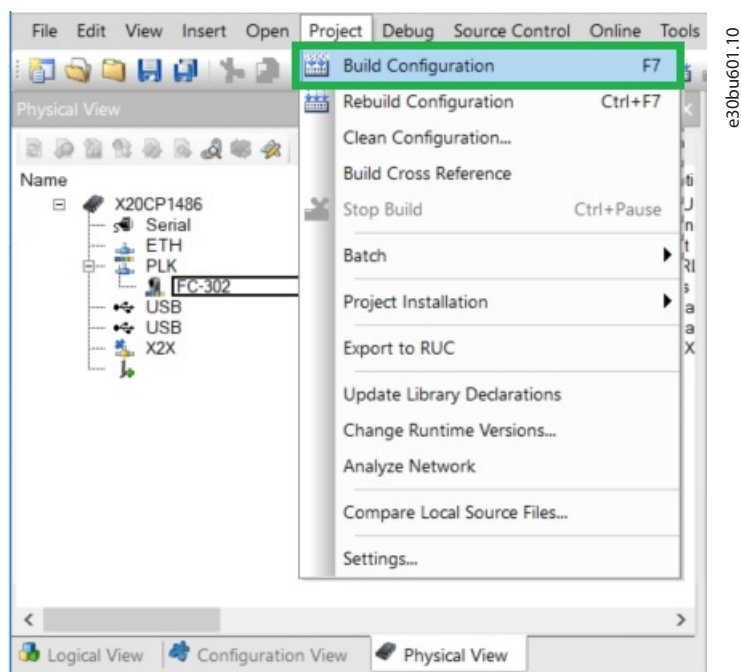


Illustration 31: Build Configuration

→ After the configuration is built successfully, *Project Build* window appears.

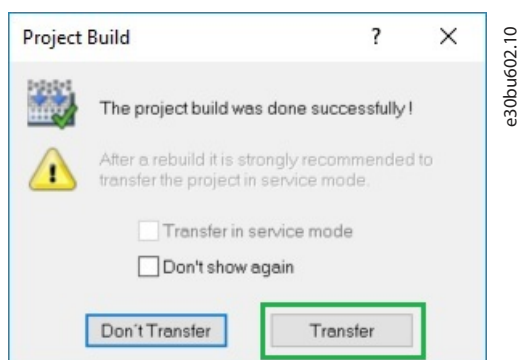


Illustration 32: Project Build Window

3. Click *Transfer*.

→ *Transfer to target* window appears.

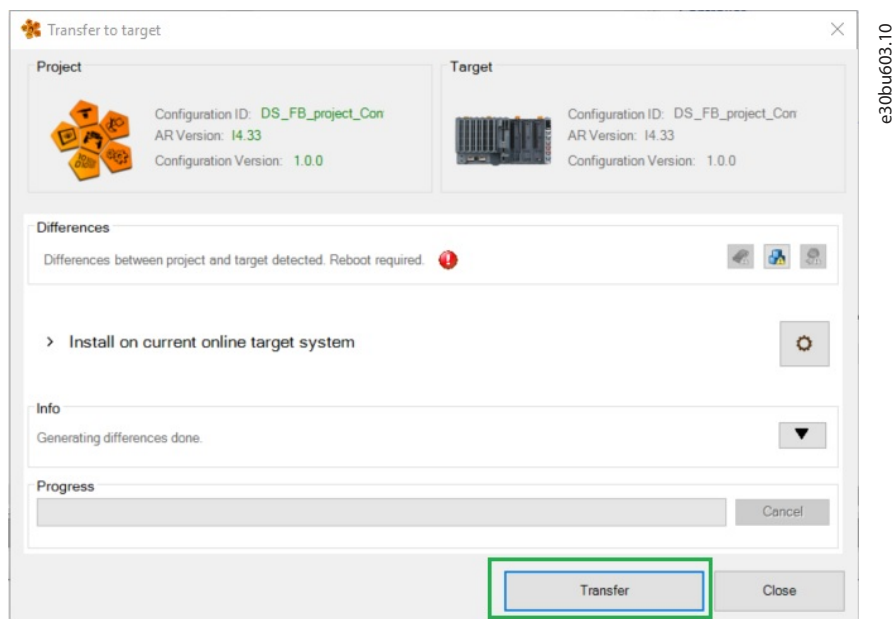


Illustration 33: Transfer to Target Window

4. Click *Transfer*.
5. After transferring to PLC successfully, click *Close*.

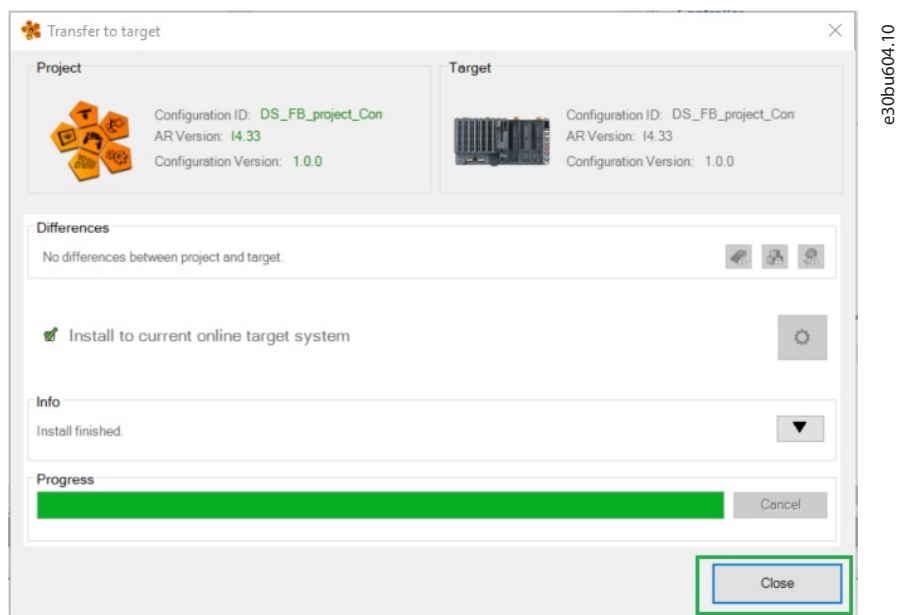


Illustration 34: Transfer to Target

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 LCP.
2. Verify the following parameter settings to ensure that the PLC has control of the drive.

Table 7: 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 8: 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. 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 *[7] CANopen DSP 402*, the function block does not work as expected and leads to malfunction. Verify that *parameter 8-10 Control Word Profile* is set correctly via the Main Menu.
6. Ensure physically that LCP mode is set to *Auto On* mode.

3.2 Basic Operation Function Block

Context:

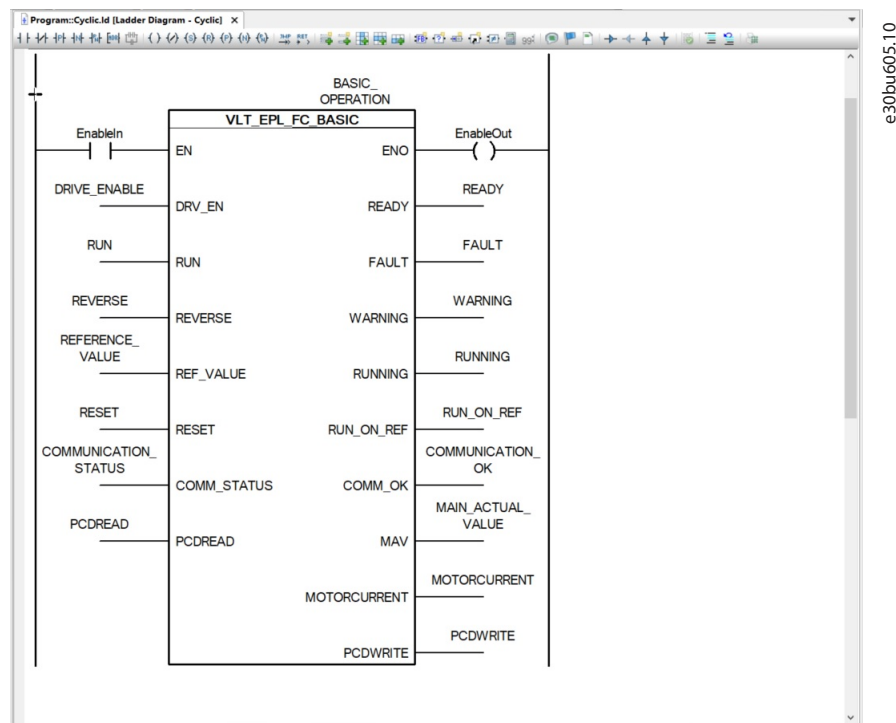


Illustration 35: Basic Operation Function Block

Procedure

1. Enable the drive to ready state.
 - A Set the input variable *EnableIn* to *TRUE*.
 - 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 9: Pin Name and Expected Value

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

- 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 10: Pin Name and Expected Value

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

- 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 11: Pin Name and Expected Value

Pin name	Expected value
READY	FALSE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE

Pin name	Expected value
RUN_ON_REF	FALSE
COMM_OK	TRUE
MAV	0
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .

- I Set the value *FALSE* to the *RUN* input pin of the function block.
2. 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 12: Pin Name and Expected Value

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

3. 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 *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 13: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	TRUE
RUN_ON_REF	TRUE
COMM_OK	TRUE
MAV	-10000 [±1%]

Pin name	Expected value
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .

4. 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 14: Pin Name and Expected Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
COMM_OK	TRUE
MAV	16#0000
MOTORCURRENT	Same as <i>parameter 16-14 Motor Current</i> .

3.3 Parameter Access Function Block

Context:

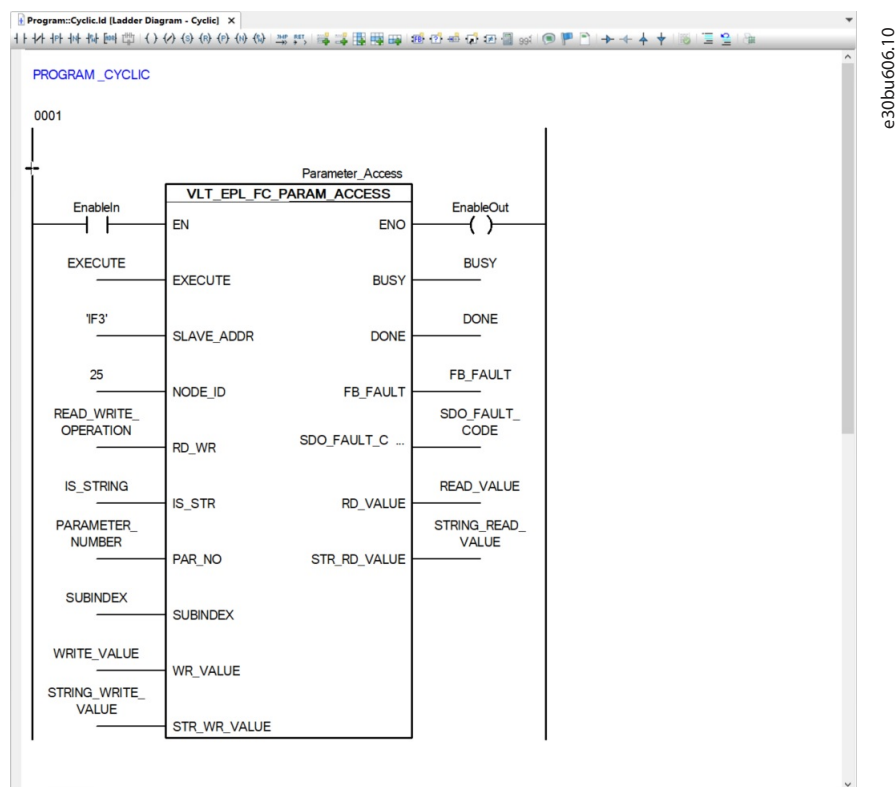


Illustration 36: Parameter Access Function Block

Procedure

1. Read non-array parameters.

- A** Set the value *TRUE* to the *EnableIn* input variable which is connected to the *EN* input pin.
- B** Set the value *FALSE* to the *EXECUTE* input pin of the function block.
- C** Set *parameter 3-41 Ramp 1 Ramp Up Time* to value *3.00* using LCP.
- D** Set the following values to the input pin of the function block.

Table 15: Pin Name and Set Value

Pin Name	Set value for <i>parameter 3-41 Ramp 1 Ramp Up Time</i>
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	341
SUBINDEX	0
EXECUTE	TRUE

E Verify the following values with the function block output.

Table 16: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	TRUE
FB_FAULT	0
SDO_FAULT_CODE	16#0000_0000
RD_VALUE	300

2. Write non-array parameters.

- A** Set the value *FALSE* to the *EXECUTE* input pin of the function block.
- B** Set the following values to the input pin of the function block.

Table 17: Pin Name and Set Value

Pin Name	Set value for <i>parameter 3-41 Ramp 1 Ramp Up Time</i>
RD_WR	TRUE
IS_STR	FALSE
PAR_NO	341
SUBINDEX	0
WR_VALUE	500
EXECUTE	TRUE

C Verify the following values with the function block output.

Table 18: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	DONE
FB_FAULT	0
FAULT_CODE	16#0000_0000

D Verify *parameter 3-41 Ramp 1 Ramp Up Time* is 5.00 s using LCP.

3. Read array type parameters.

A Set the value *FALSE* to *EXECUTE* input pin of the function block.

B Set the following parameters in *parameter 3-10 Preset Reference*.

Table 19: Parameter Value Settings

Parameter number	Parameter name	Value
310.0	PRESET REFERENCE	25.00
310.1	PRESET REFERENCE	45.00
310.4	PRESET REFERENCE	0.00

C Set the following values to the input pin of the function block.

Table 20: Pin Name and Set Value

Pin name	Set value for <i>parameter 12-21 Process Data Write Config</i>
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	1
EXECUTE	TRUE

D Verify the following values with the function block output.

Table 21: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	TRUE
FB_FAULT	0
SDO_FAULT_CODE	16#0000_0000
RD_VALUE	2500

E Set the value *FALSE* to *EXECUTE* input pin of the function block.

F Set the following values to the input pin of the function block.

Table 22: Pin Name and Set Value

Pin name	Set value for <i>parameter 12-21 Process Data Write Config</i>
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	2
EXECUTE	TRUE

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

Table 23: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	TRUE
FB_FAULT	0
SDO_FAULT_CODE	16#0000_0000
RD_VALUE	4500

- H** Set the value *FALSE* to *EXECUTE* input pin of the function block.
I Set the following values to the input pin of the function block.

Table 24: Pin Name and Set Value

Pin name	Set value for <i>parameter 12-21 Process Data Write Config</i>
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	5
EXECUTE	TRUE

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

Table 25: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	TRUE
FB_FAULT	0
SDO_FAULT_CODE	16#0000_0000
RD_VALUE	0

- 4.** Write an array type parameter.
A Use the LCP of the drive to verify if *parameter 3-10 [3] Preset Reference* is showing value other than 0.10%. If the value is 0.10%, change it to any other value.

- B Set the value *FALSE* to *EXECUTE* input pin of the function block.
- C Set the following values to the input pin of the function block.

Table 26: Pin Name and Set Value

Pin name	Set value for <i>parameter 3-10 [3] Preset Reference</i>
RD_WR	TRUE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	4
WR_VALUE	10
EXECUTE	TRUE

- D Verify on LCP that *parameter 3-10 [3] Preset Reference* is set to 0.10%.

5. Write a string type parameter.

- 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.

Table 27: Pin Name and Value To Be Set

Pin name	Value to be set
RD_WR	TRUE
IS_STR	TRUE
PAR_NO	1208
SUBINDEX	0
STR_WR_VALUE	<i>abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuv</i>
EXECUTE	TRUE

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

Table 28: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	TRUE
FB_FAULT	0
SDO_FAULT_CODE	0

- D Verify on LCP if the value of *parameter 12-08 Host Name* is set as *abcdefghijklmnopqrstuvwxyzabcdefghijklmnopqrstuv*.

6. Read a string type parameter.

- 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.

Table 29: Pin Name and Value To Be Set

Pin name	Value to be set
RD_WR	FALSE
IS_STR	TRUE
PAR_NO	1208
SUBINDEX	0
EXECUTE	TRUE

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

Table 30: Pin Name and Expected Value

Pin name	Expected value
BUSY	FALSE
DONE	TRUE
FB_FAULT	0
SDO_FAULT_CODE	0
STR_RD_VALUE	Same as <i>parameter 12-08 Host Name</i> .

Index

B

Basic operation function block 5, 5, 37

C

Control and monitoring 5

Controlled node 28

F

Failure management 5

G

General configuration 35

L

Library 5

M

MCA 123 4

P

Parameter access function block 5, 10, 41

Process variables 28

R

Reverse 5

S

SDO 10

Speed regulation 5

X

XDD file 18

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

