ENGINEERING
TOMORROW

Programming Guide

Function Blocks with Beckhoff TwinCAT 3

drives.danfoss.com | VLT®

Contents

1 Introduction 4

1.1 Disclaimer 4

1.2 Purpose of the User Guide 4

1.3 Abbreviations 5

1.4 What are Function Blocks? 5

1.4.1 Advantages of Using Function Blocks 5

1.5 Overview of the Danfoss Library (VLT_ECAT_LIB_V1_12) 6

1.6 Basic Operation Function Block (VLT_ECAT_FC_BASIC) 6

1.7 Parameter Access Function Block (VLT_ECAT_FC_PARAM_ACCESS) 10

1.8 Diagnostics Function Block (VLT_ECAT_FC_DIAGNOSTICS) 12

2 Using FBs in TwinCAT 14

2.1 Importing Danfoss Library into a Project 14

2.2 Adding Danfoss ESI File to TwinCAT 18

2.2.1 Auto Scan 19

2.2.2 Manually Adding the EtherCAT Slave Device 22

2.3 Adding Additional Parameters in EtherCAT Slave 26

2.4 Configuring the EtherCAT Slave 30

2.5 Adding Function Blocks to the Main Program 31

2.6 Creating A Process Variable for the Function Block Instance 33

2.7 Creating A Mapping of Function Block Instance Variables and EtherCAT Slave's Variables 34

2.8 Identifying AMS_NET_ID of the EtherCAT Master Device 37

2.9 Identifying EtherCAT Slave Address 39

2.10 Saving and Building the Project 40

2.11 Transfering PLC Project to the Embedded PC 41

3 Examples 43

3.1 General Configuration of the Drive 43

3.2 Basic Operation Function Block 44

3.3 Parameter Access Function Block 48

3.4 Diagnostics Function Block 52

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 Beckhoff TwinCAT 3.1 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 Beckhoff TwinCAT 3.1 system.
- Procedure on using the library to communicate with Danfoss VLT® drives in TwinCAT 3.1, including examples.

Function blocks for Beckhoff TwinCAT 3 supporting Danfoss VLT® drives.

Danfoss VLT® drives with the following options:

- VLT® EtherCAT MCA 124

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
PVOID	Corresponds to data type UXINT, where UDINT on 32-bit platforms ULINT on 64-bit platforms
DUT	Device under test
LIB	Library
ECAT	EtherCAT
ESI	EtherCAT slave information

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_ECAT_LIB_V1_12)

The library *VLT_ECAT_LIB_V1_12* is a collection of predefined function blocks provided by Danfoss. Use these blocks as an aid to simplify programs, containing standard functionality for programming EtherCAT, Beckhoff systems and Danfoss drives.

The library contains the following FBs:

- **Basic operation block (VLT_ECAT_FC_BASIC)**: Dedicated to handling the basic operation of the drive and connected motor operations.
- **Parameter access block (VLT_ECAT_FC_PARAM_ACCESS)**: Dedicated to parameter read/write through an acyclic channel.
- **Diagnostics block (VLT_ECAT_FC_DIAGNOSTICS)**: Dedicated to read diagnostic information from the drive through an acyclic channel.

The function blocks are designed to work with FC profile only, and does not support the following features:

- DS402 CANopen Profile.
- Backward compatibility in TwinCAT Version.

1.6 Basic Operation Function Block (VLT_ECAT_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.

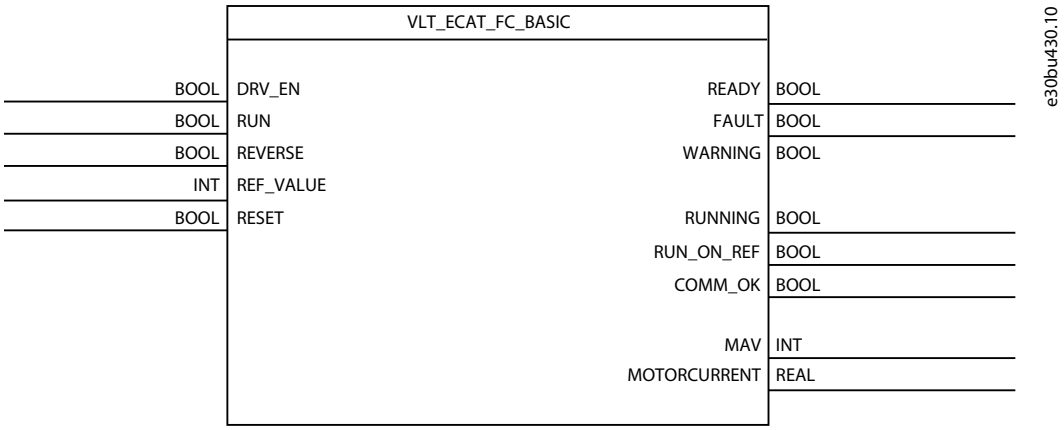


Illustration 1: Basic Operation Function Block Layout

Table 1: Input Parameter

Parameter	Type	Description
DRV_EN	BOOL	*TRUE* = enabling drive. Setting the drive in *Ready to Start* mode. The 4 bits, CTW.02 (DC brake), CTW.03 (coasting), CTW.04 (quick stop), CTW.05 (hold output frequency) are set to *TRUE* to move the drive to *READY* state.

Parameter	Type	Description
RUN	BOOL	*TRUE* = starts the motor run in the direction selected based on the *REVERSE* input variable. The CTW.06 (ramp start) bit is set to *TRUE* to start the motor.
REVERSE	BOOL	The direction of rotation of the motor. *FALSE* = activates the forward direction. *TRUE* = activates the reverse direction. The CTW.15 (reverse) bit is set to *TRUE* 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%–100.00%. Enter the reference value without decimal. For example: To run the drive at 56.75%, enter *REF_VALUE* as 5675 to achieve the motor speed.
RESET	BOOL	*TRUE* = resets the device failure and resets the *FAULT* output to 0. The CTW.07 (reset) bit is set to *TRUE* to reset the failure.

Table 2: Output Parameter

Parameter	Type	Description
READY	BOOL	*TRUE* = the drive is ready for operation. The 4 bits STW.00 (control ready), STW.01 (drive ready), STW.02 (coast stop), STW0.9 (bus control) are considered for *READY* state.
FAULT	BOOL	*TRUE* = a detected failure in the control block. To reset the *FAULT* output pin, the *RESET* input to be activated. The 3 bits, STW.03 (drive trips), STW.04 (shows error but not tripped), STW.06 (trip lock), are considered for *FAULT* state.
WARNING	BOOL	*TRUE* = 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 *WARNING* state.
RUNNING	BOOL	*TRUE* = 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	*TRUE* = 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	TRUE = The communication between the device and PLC is healthy.
MAV	INT	Main actual value in % (expressed in integer value). 0–10000. For example: if the MAV value is 9949, it means that the drive is running at 99.49%.
MOTORCUR-RENT	REAL	Motor current in Amps.

Table 3: Local Variables

Parameter	Type	Description	
COMM_STA-TUS	UINT	Connect the 'State' variable of the slave device and it holds the current EtherCAT state and link status of the EtherCAT slave device. COMM_STATUS variable should hold variable address “AT %I*” during declaration For example: COMM_STATUS AT %I*: UINT	
PCDREAD	BASIC_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 DFB, and it is recommended not use this input directly.	
	Parameter	**Type**	**Description**
	STATUS_WORD	UINT	Status word.
	MAIN_ACTUAL_VALUE	INT	Main actual value.
	MOTORCURRENT	DINT	Motor current in Amps.
PCDWRITE	BASIC_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.

NOTICE

CTW and STW mentioned in the function blocks refer to FC Profile CTW and FC Profile STW, respectively.

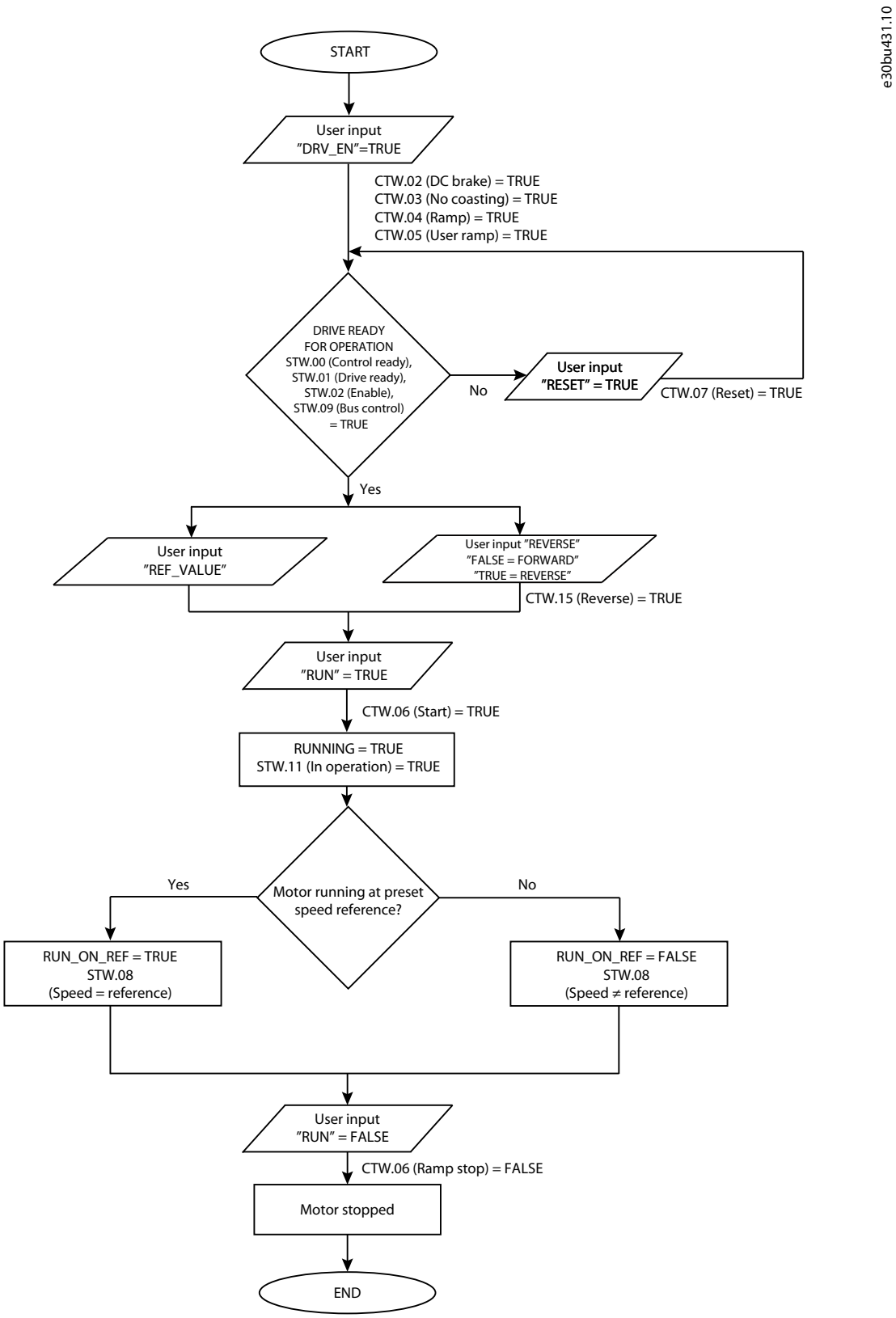


Illustration 2: Flow Chart for Basic Operation Of Drive Control

1.7 Parameter Access Function Block (VLT_ECAT_FC_PARAM_ACCESS)

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

- Carry out read/write operation (only 1 operation at a time).
- Read string type parameters.
- Write string type parameters.
- Read non-string data type parameters.
- Write non-string data type parameters.
- Show valid abort code on invalid read-write operations.
- User-configurable timeout.

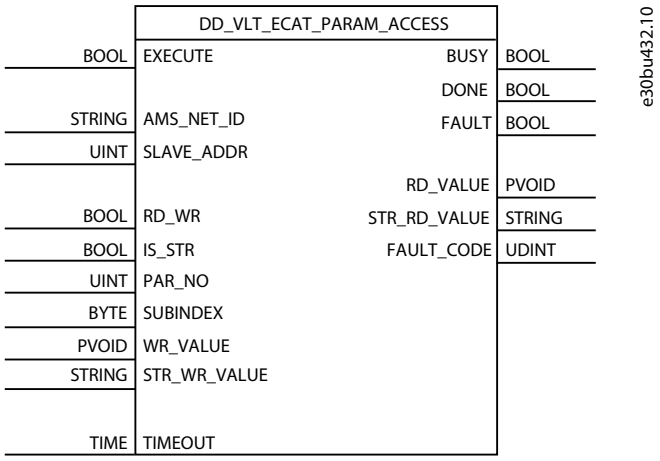


Illustration 3: Parameter Access Function Block Layout

The following elementary function blocks are embedded inside *VLT_ECAT_FC_PARAM_ACCESS* function block from TwinCAT PLC Library TC_EtherCAT library.

- FB_EcCoeSdoRead: a function block which allows data to be read from an EtherCAT slave through an SDO access.
- FB_EcCoeSdoWrite: a function block which allows an object from the object directory of an EtherCAT slave to be written with an SDO download.

Table 4: Input Parameter

Parameter	Type	Description
EXECUTE	BOOL	*TRUE* = the rising edge of this signal starts the requested operation. *FALSE* = 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.**
AMS_NET_ID	STRING	AMS NetId of the EtherCAT master device.
SLAVE_ADDR	UINT	EtherCAT slave address of the device.

Parameter	Type	Description
RD_WR	BOOL	Type of operation. *FALSE* = read operation. *TRUE* = write operation.
IS_STR	BOOL	Indicates operation to be performed based on the parameter data type. Set the input to: • *FALSE*: to perform the operation based on RD_WR input for non-string data type (UDINT, DINT, INT, and so on) parameter. • *TRUE*: to perform the operation based on RD_WR input for string data type parameter.
PAR_NO	UINT	Enter the drive parameter number in the integer format. Example: to read the running hours of *parameter 15-01 Running Hours*, enter PAR_NO as 1501. Reading the running hours in *parameter 15-01 Running Hours* 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 should be entered as *0*.
WR_VALUE	PVOID	Integer data to be written to the drive.
STR_WR_VALUE	STRING	String data are written to the drive.
TIMEOUT	TIME	Time to wait for a response, after this port sends a timeout error to the requesting device.

Table 5: Output Parameter

Parameter	Type	Description
BUSY	BOOL	*TRUE* = the client is processing the request and waiting for the response. *FALSE* = the client processed the request with status (Done or Error).
DONE	BOOL	*TRUE* = the request operation has ended (with or without a detected error). This signal is acknowledged by setting the *EXECUTE* signal to *Low*.
FAULT	BOOL	Activated when a detected error occurs in the request transmission. *TRUE* = the request operation is unsuccessful. To reset the detected error, activate the client again by setting the *EXECUTE* signal to *High*. Refer *FAULT_CODE* for more information.
RD_VALUE	PVOID	Holds the read data of pseudo data types depending on the read input parameter [(1)].
STR_RD_VALUE	STRING	Holds the read data of string type depending on the read input parameter.
FAULT_CODE	UDINT	Indicates the ADS fault code associated with the most recently executed command. Refer to the ADS return codes from the Beckhoff Information system: TwinCAT2⇒TwinCAT system⇒TwinCAT Connectivity⇒TwinCAT ADS⇒ADS return codes.

[1] *For pseudo data types, refer to TwinCAT system help file.*

1.8 Diagnostics Function Block (VLT_ECAT_FC_DIAGNOSTICS)

The function block provides functionality to read diagnostics information from the drive through SDO service.

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/emergency data when 1 or more alarms appear in alarm *16-90 Alarm Word*.
- TRIGGER ON ALARMS/WARNINGS: send extended diagnosis/emergency data if 1 or more alarms or warnings appear in alarm *16-90 Alarm Word* and *16-92 Warning Word*.
- EXEMPTION! Warning word 3 and alarm word 3 are not reflected on fieldbus.

The diagnostics function block supports the following functionalities:

- Show alarm word in 32-bit format (*16-90 Alarm Word*).
- Show warning word in 32-bit format (*16-92 Warning Word*).
- Display EMCY code.

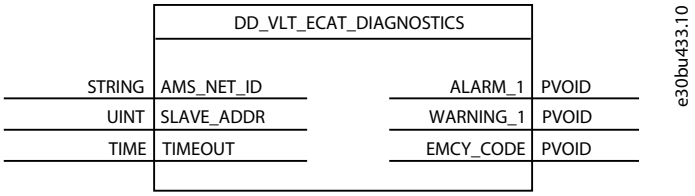


Illustration 4: Diagnostics Function Block Layout

The following elementary function blocks are embedded inside VLT_ECAT_FC_DIAGNOSTICS function block from TwinCAT PLC Library TC_EtherCAT library.

- FB_EcCoeSdoRead: a function block which allows data to be read from an EtherCAT slave through an SDO access.
- The following parameters are read through SDO and repetitive based on the user configuration provided in TIMEOUT input parameter of the function block.

Table 6: Parameter Description

Parameter number	Index (Hex)	Sub-index (Hex)	Description
16–90	269A	0	Alarm word 1
16–92	269C	0	Warning word 1
–	1003	1	Current Error: it has 2 16 words. MSW holds the information if a warning or alarm is set. LSW holds the EMCY error.

Table 7: Input Parameter

Parameter	Type	Description
AMS_NET_ID	STRING	AMS NetId of the EtherCAT master device.
SLAVE_ADDR	UINT	EtherCAT slave address of the device.
TIMEOUT	TIME	At every configured time interval, the requested operation is performed.

Table 8: Output Parameter

Parameter	Type	Description
ALARM_1	PVOID	The alarms read from the drive *parameter 16-90 Alarm Word* are shown.
WARNING_1	PVOID	The warnings read from the drive *parameter 16-92 Warning Word* are shown.
EMCY_CODE	PVOID	EMCY code values read from the LSB of the current error index: 1003, sub-index: 1.

2 Using FBs in TwinCAT

2.1 Importing Danfoss Library into a Project

Procedure

1. In the solution explorer, navigate to *TwinCAT project⇒PLC⇒PLCProject⇒References*. Right-click the *References*, and click *Add library....*

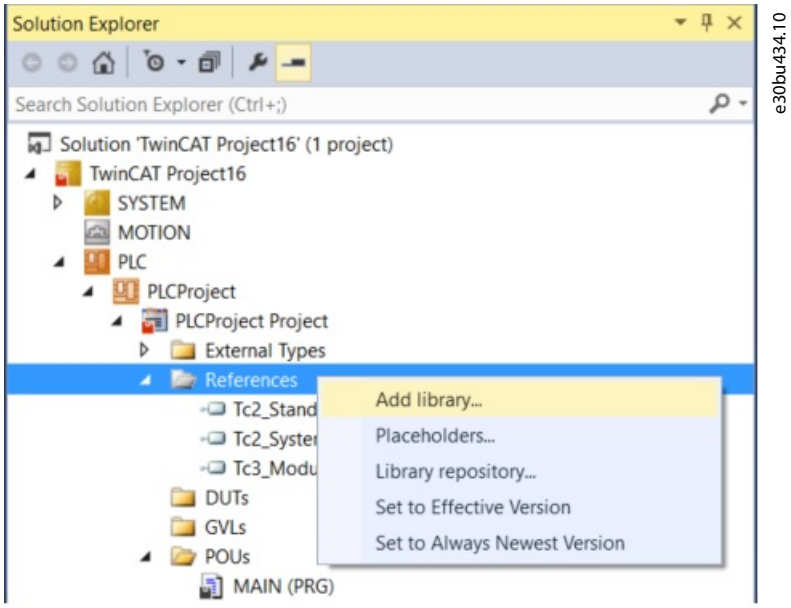
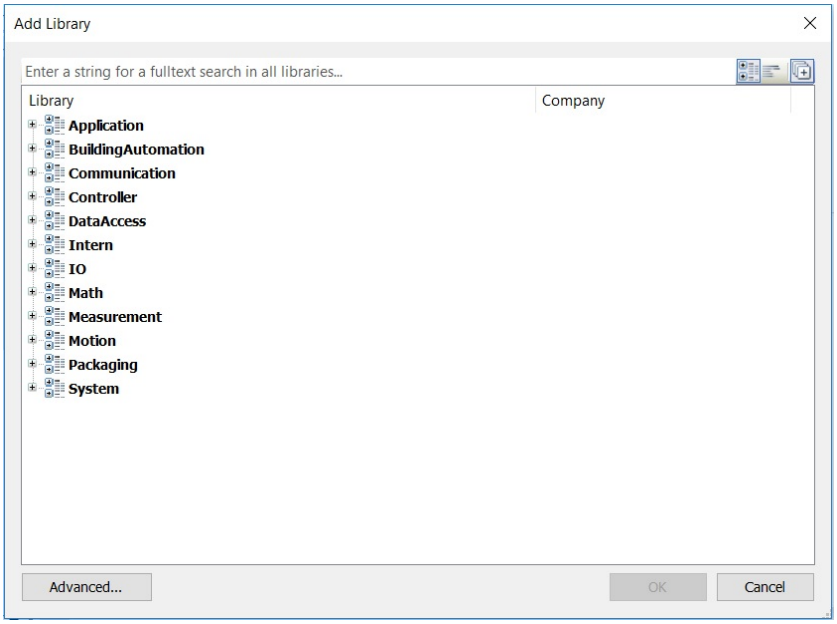


Illustration 5: Solution Explorer

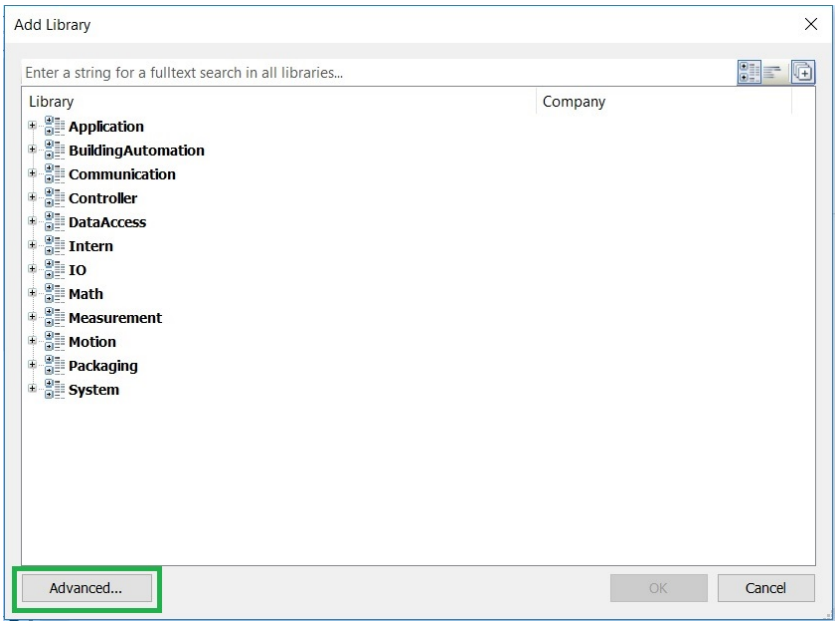
→ *Add Library* dialog box is shown.



e30bu435.10

Illustration 6: Add Library

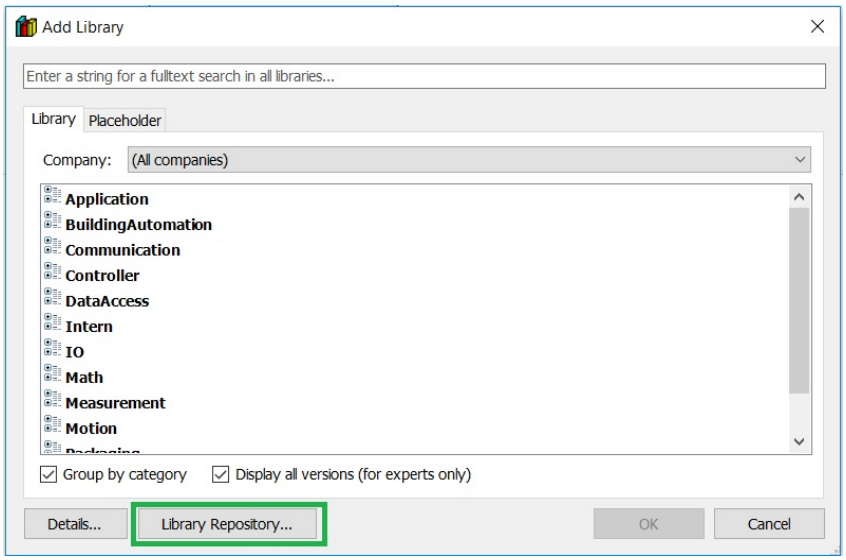
2. Click *Advanced*.



e30bu436.10

Illustration 7: Add Library

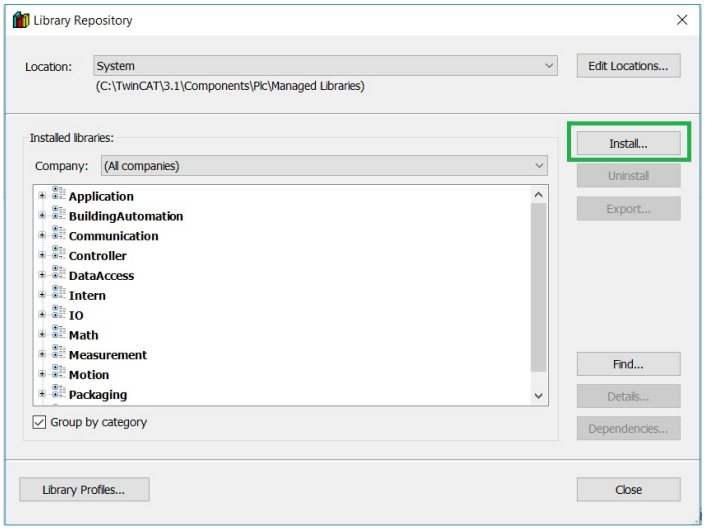
3. Click *Library Repository....*



e30bu437.10

Illustration 8: Add Library

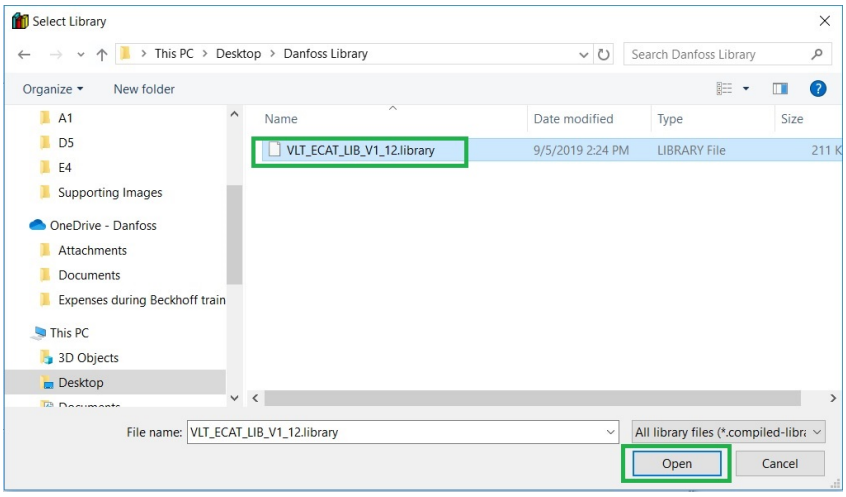
4. In the *Library Repository* dialog box, click *Install*.



e30bu438.10

Illustration 9: Library Repository

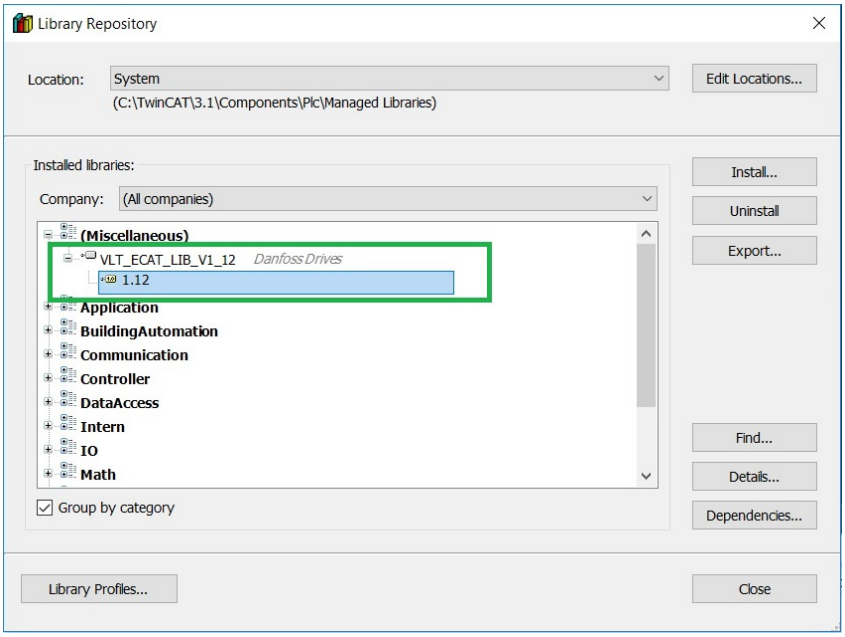
5. Navigate to the downloaded library installation file and click *Open*.



e30bu439.10

Illustration 10: Select Library

→ The library is successfully installed. It is added in the *Miscellaneous* category.



e30bu440.10

Illustration 11: Successful Installation of Library

2.2 Adding Danfoss ESI File to TwinCAT

Procedure

1. Download XML file from the Danfoss field configuration files website: https://www.danfoss.com/en/service-and-support/downloads/dds/fieldbus-configuration-files/#tab-downloads.
2. Unzip the *VLT_MCA124_EtherCAT_XML_2012-12-17* file and move the contents of this unzipped file to *C:\TwinCAT\3.1\Config\Io\EtherCAT* location as shown below.

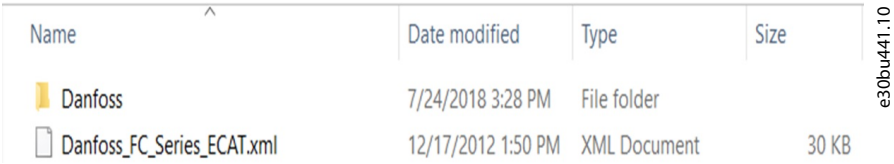


Illustration 12: ESI file

3. Make sure that the ESI file for the slave is available in TwinCAT. ESI file can be added in 2 approaches:
   - Auto scan.
   - Manually add the EtherCAT slave device.

2.2.1 Auto Scan

Procedure

1. In the solution explorer, navigate to *I/O* ⇒ *Devices*. Right-click *Devices* ⇒ *Scan*.

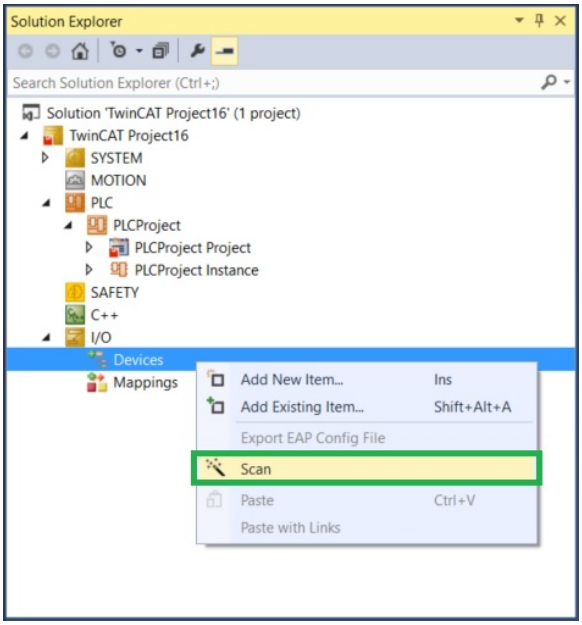


Illustration 13: Auto Scan

→ *TwinCAT System Manager* window appears.

2. Click *OK* to the message about not all devices can be found automatically.

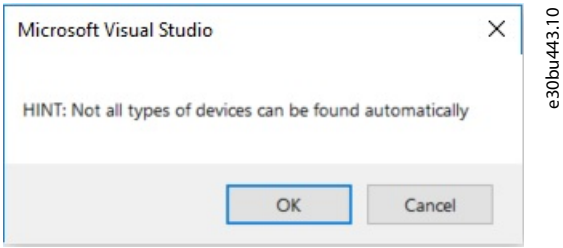


Illustration 14: Message Window

→ *TwinCAT System Manager* shows the found devices, and automatically select items that are EtherCAT enabled.

3. Click *OK*.

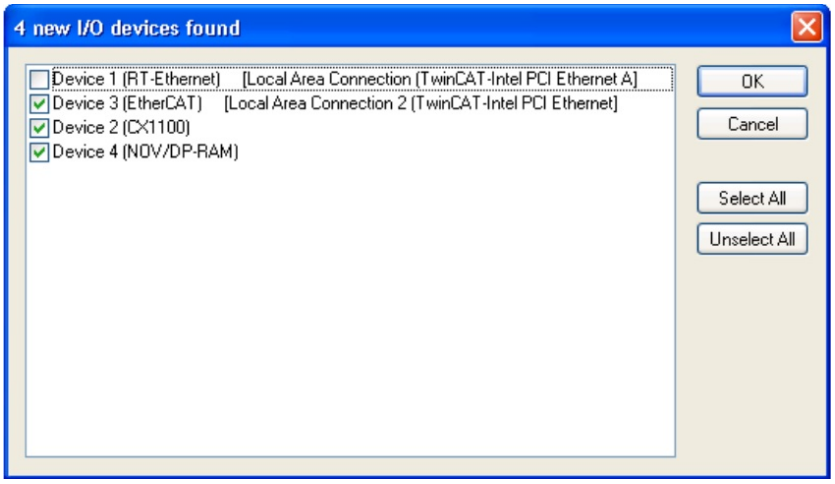


Illustration 15: I/0 Devices Found

→ TwinCAT asks to *Scan for boxes* or not.

4. Select *Yes* to find the EtherCAT devices that are connected to the PC/PLC.

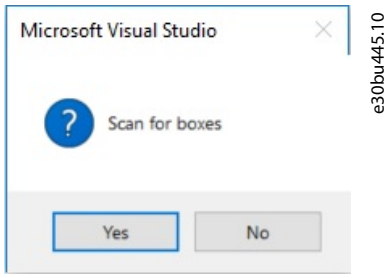


Illustration 16: Scan for Boxes

NOTICE

EtherCAT device is visible as *Device 3*.

→ During the scan operation, VLT® AutomationDrive FC 302 was found and is now able to be configured for the application.

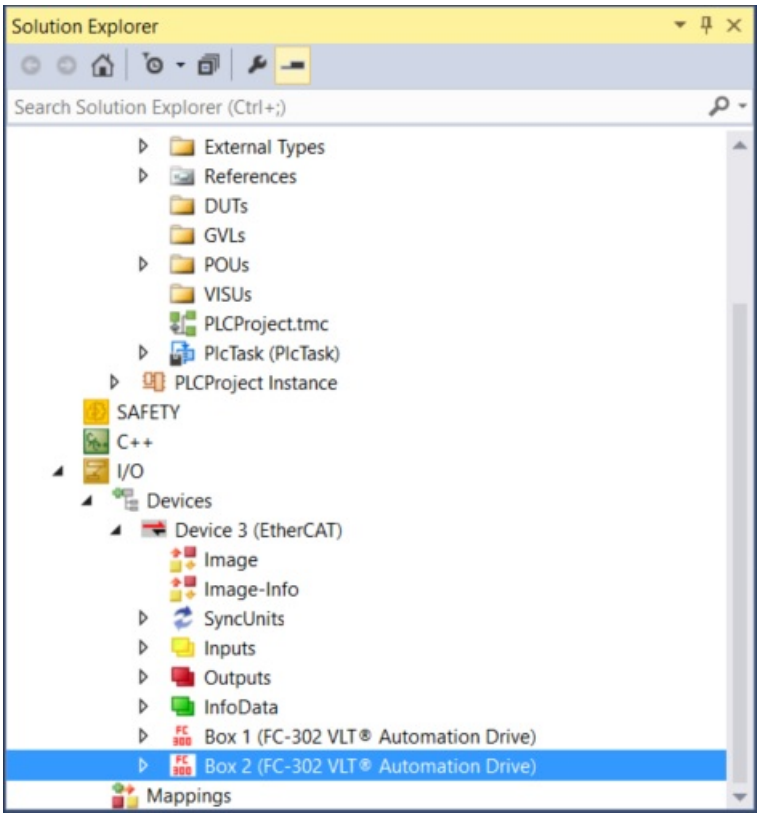


Illustration 17: EtherCAT Slave (FC 302) Added under EtherCAT Master

2.2.2 Manually Adding the EtherCAT Slave Device

Procedure

1. In the solution explorer, navigate to *I/O* ⇒ *Devices*. Right-click *Devices* ⇒ *Add New Item*.

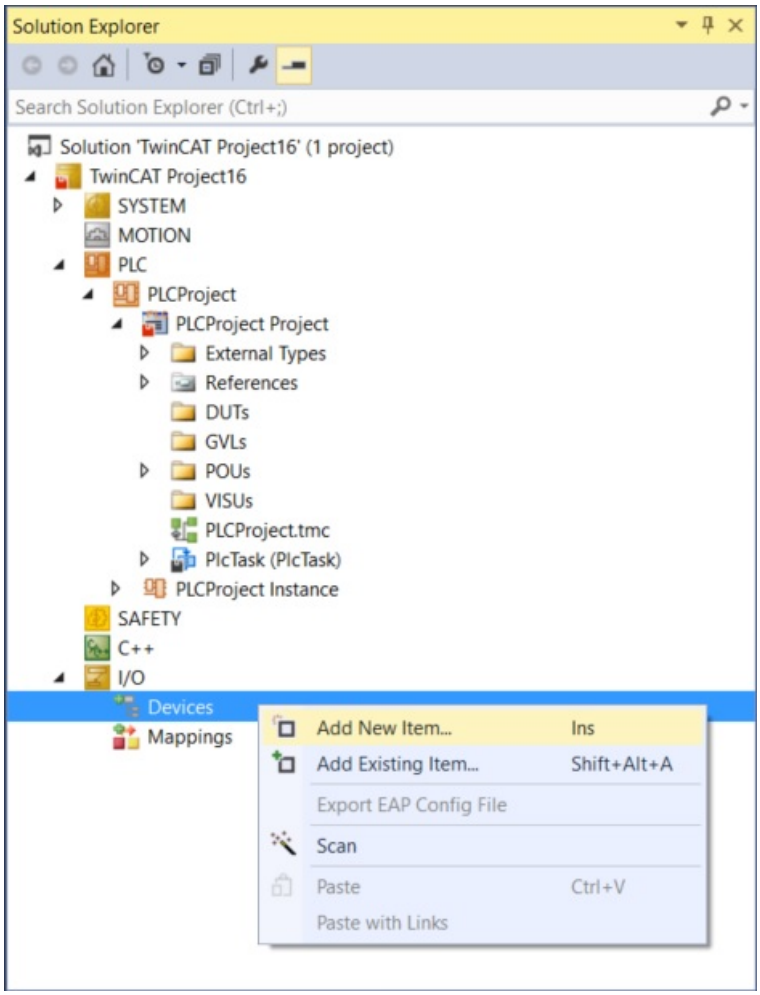


Illustration 18: Adding EtherCAT Master Manually

→ *Insert Device* dialog window appears.

2. In the box, select *EtherCAT* ⇒ *EtherCAT Master* and assign a name in the *Name* text box, click *OK*.

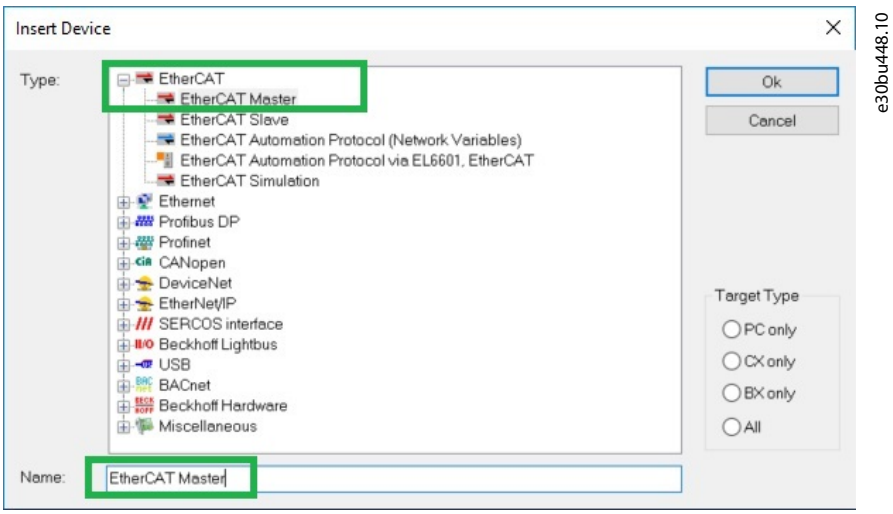


Illustration 19: Adding EtherCAT Master Device

3. Right-click *EtherCAT Master (EtherCAT)* ⇒ *Add New Item*.

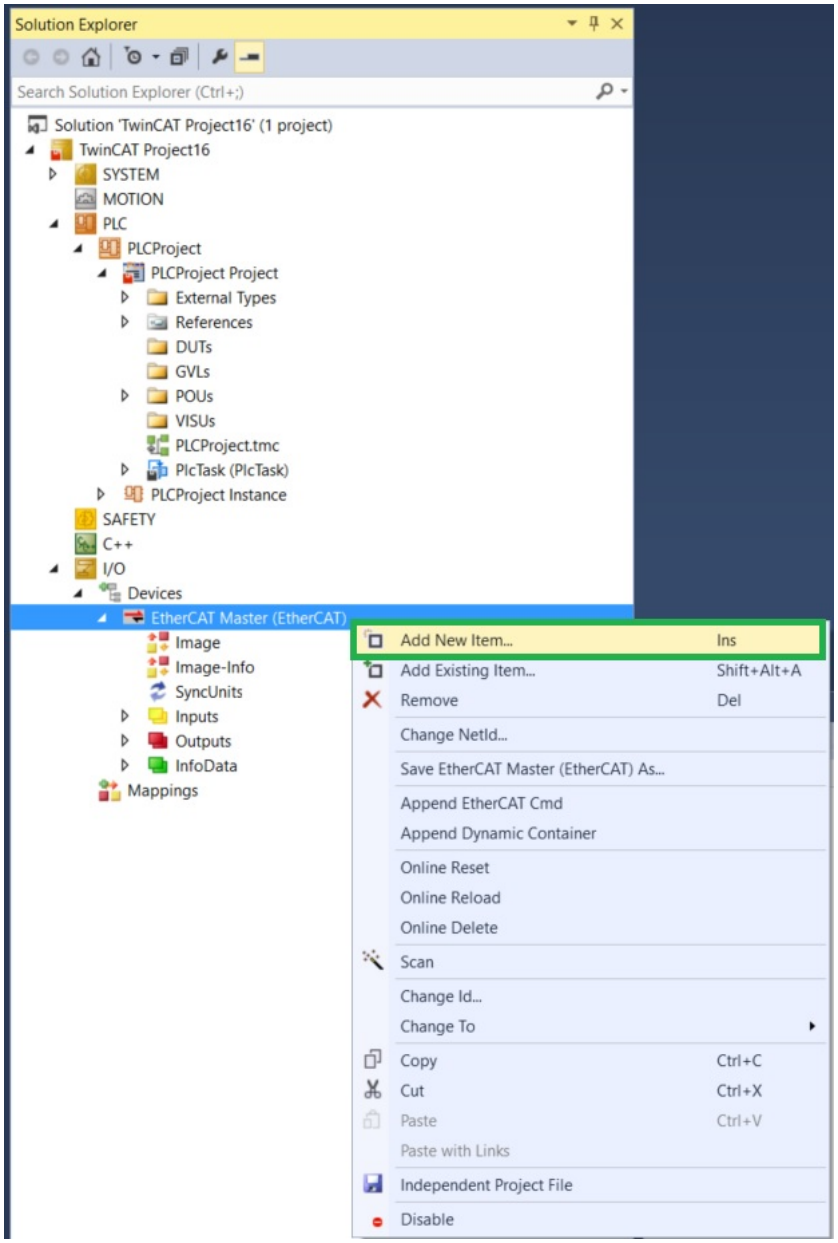
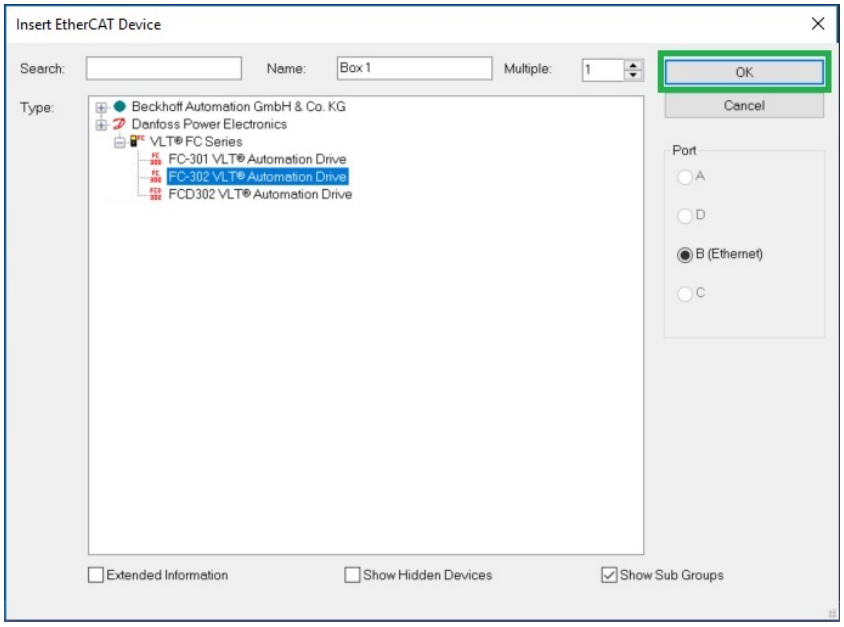


Illustration 20: Adding EtherCAT Slave Device

→ *Insert EtherCAT Device* window appears.

4. Navigate and add *Danfoss Power Electronics ⇒ VLT® FC Series ⇒ FC 302 VLT® Automation Drive*.



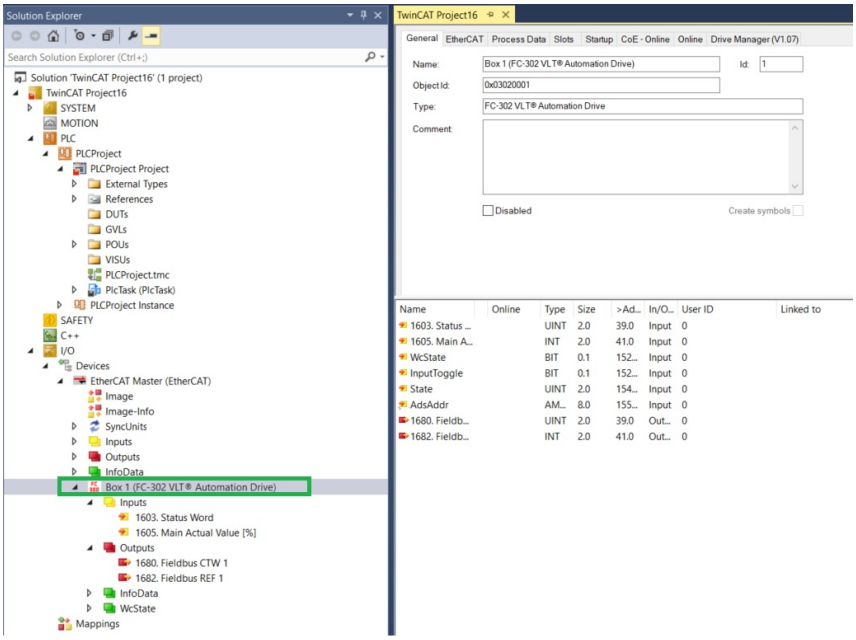
e30bu450.10

Illustration 21: Select the EtherCAT Slave Device

2.3 Adding Additional Parameters in EtherCAT Slave

Procedure

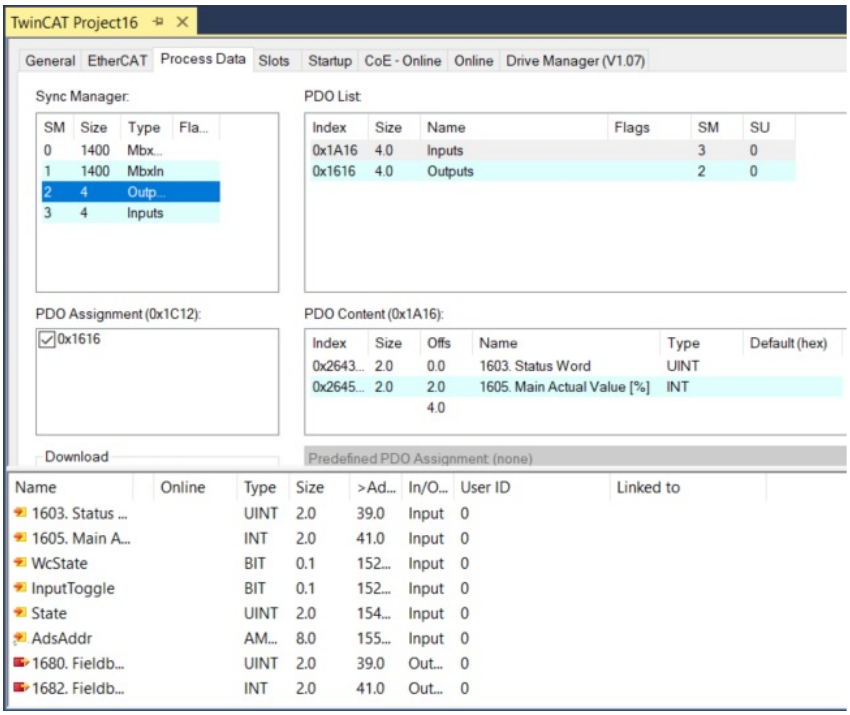
1. In the *Solution Explorer* window, navigate and open *TwinCAT Project ⇒ I/O ⇒ Devices ⇒ EtherCAT Master (EtherCAT) ⇒ Box 1*.



e30bu451.10

Illustration 22: Solution Explorer

2. Select *Process Data* tab.



e30bu452.10

Illustration 23: EtherCAT Slave Device Configuration

3. In the *PDO list*, select *Inputs*.
4. Right-click the PDO content, and select *Insert*.

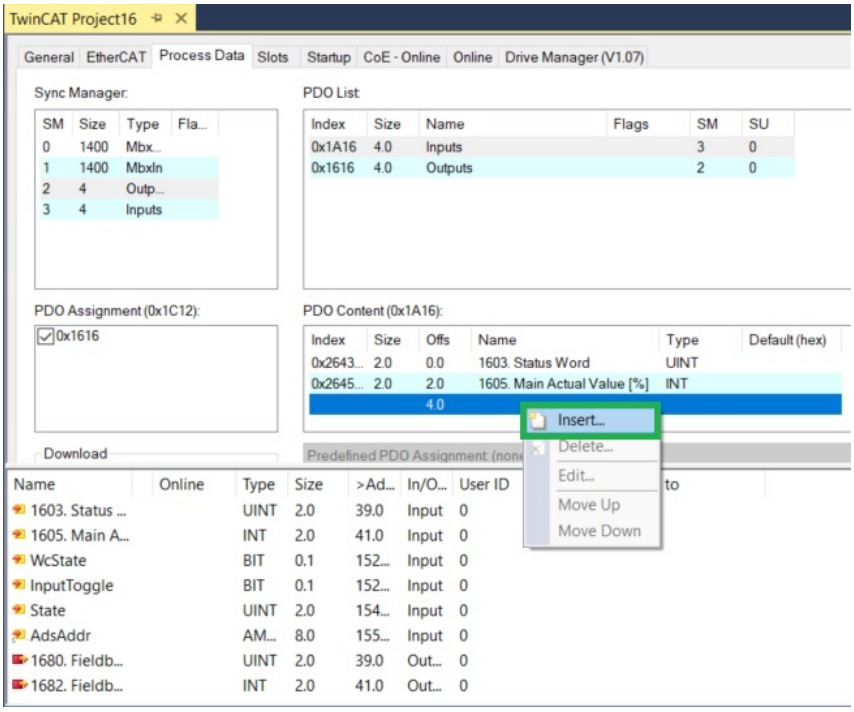


Illustration 24: EtherCAT Slave Device Process Data Configuration

→ *Edit Pdo Entry* dialog window appears.

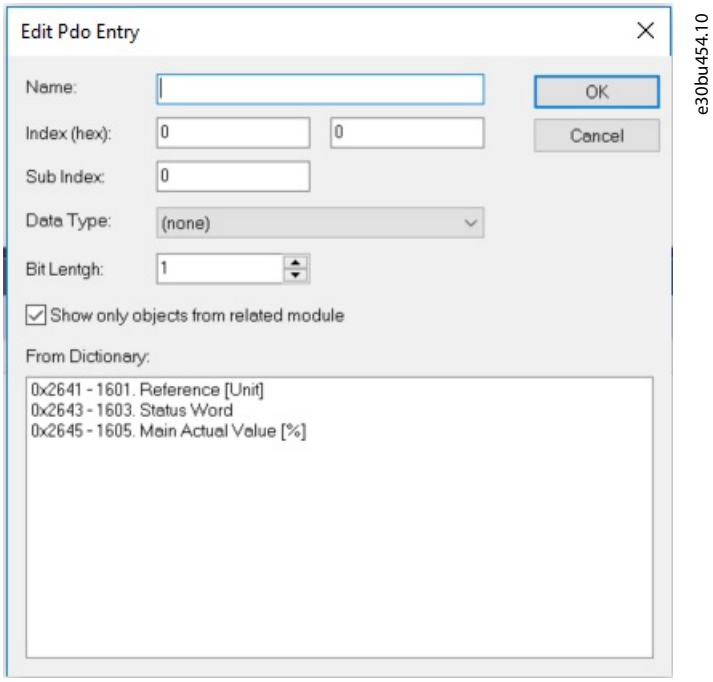


Illustration 25: EtherCAT Slave Device Process Data Configuration

5. Uncheck *Show only objects from related module*.

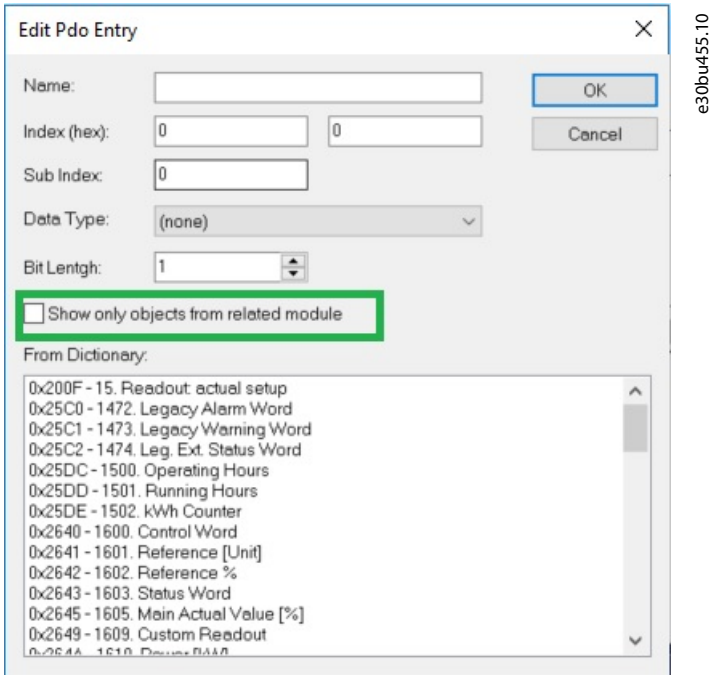


Illustration 26: EtherCAT Slave Device Configuration

6. Navigate and select *1614 Motor Current* parameter, and click *OK*.

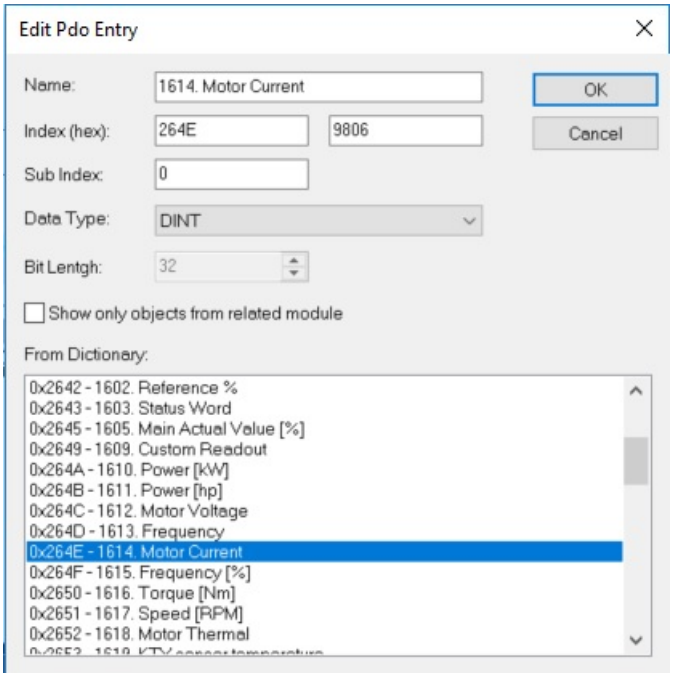


Illustration 27: EtherCAT Slave Device Configuration

→ *Motor current* parameter is added in the PDO content.

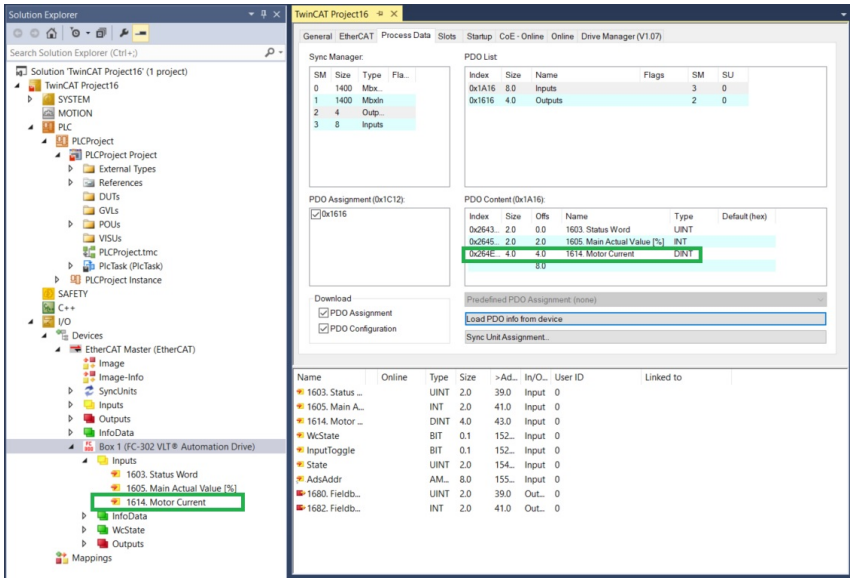
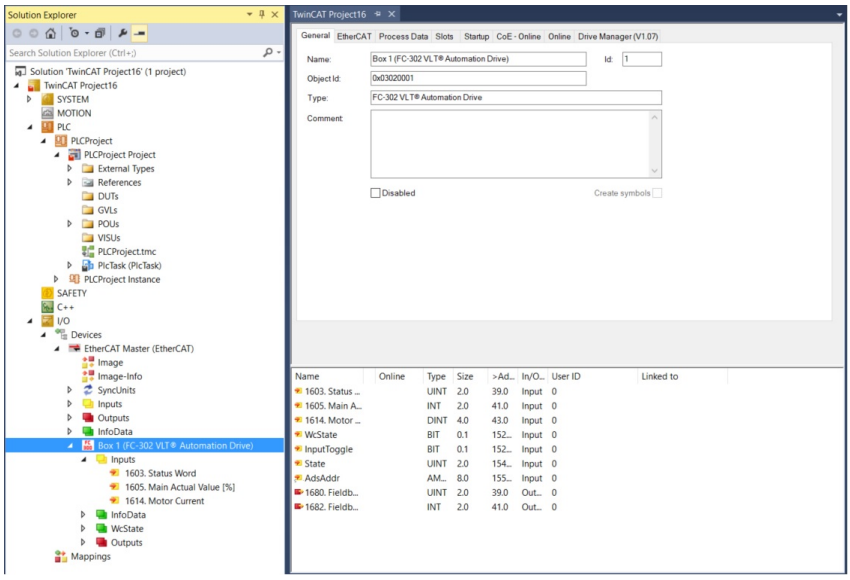


Illustration 28: EtherCAT Slave Device - Process Data Configuration

2.4 Configuring the EtherCAT Slave

Procedure

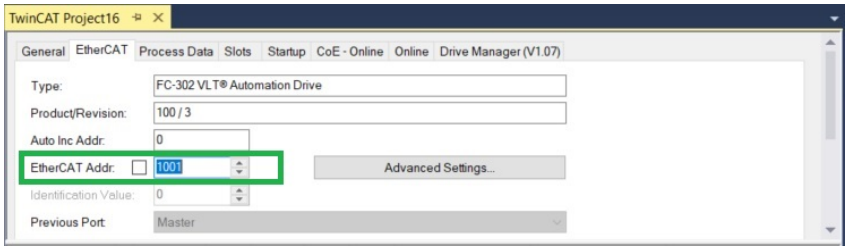
1. In the *Solution Explorer* window, navigate and open *TwinCAT Project ⇒ I/O ⇒ Devices ⇒ EtherCAT Master (EtherCAT) ⇒ Box 1*.



e30bu458.10

Illustration 29: EtherCAT Slave Configuration

2. Select *EtherCAT* tab. In this tab, the address of the actual device can be configured. The value can also be read out in *parameter 12-51 Configured Station Address*.



e30bu459.10

Illustration 30: EtherCAT Slave Address Configuration

2.5 Adding Function Blocks to the Main Program

Procedure

1. In the *Solution Explorer* window, navigate and open *TwinCAT Project ⇒ PLC ⇒ PLCProject ⇒ PLCProject Project ⇒ POUs ⇒ MAIN (PRG)*.

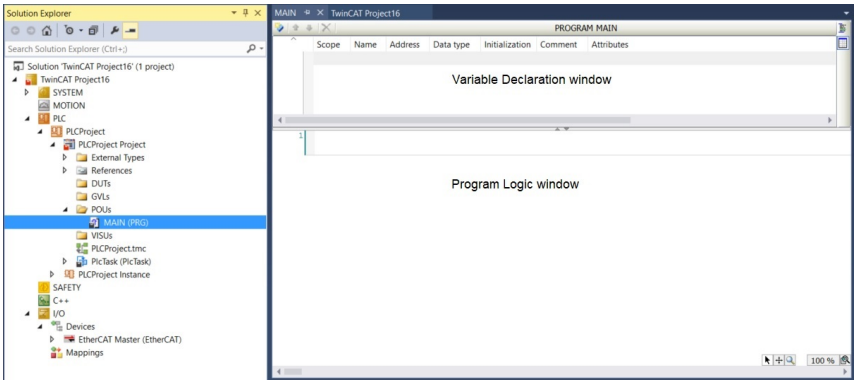


Illustration 31: Main Program Window

2. Right-click *Program logic window ⇒ Insert Empty Box*.

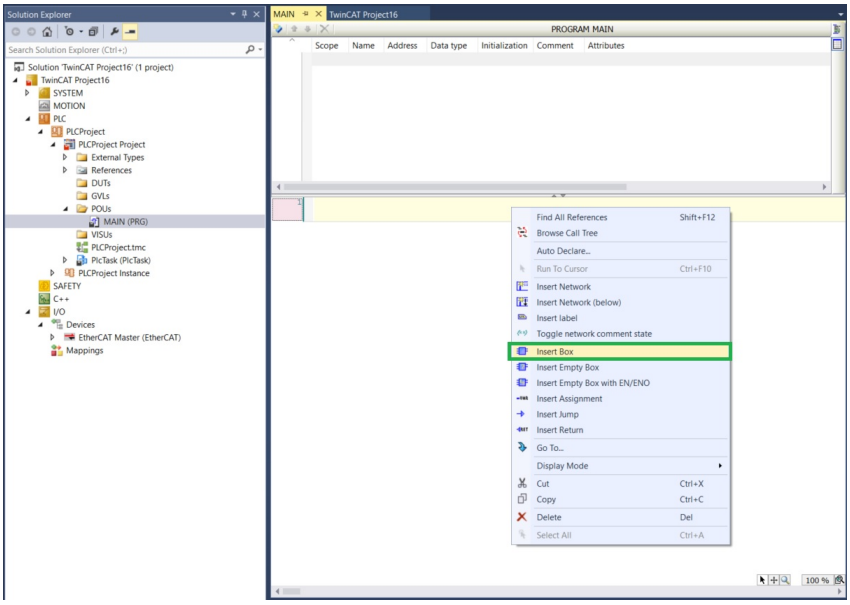
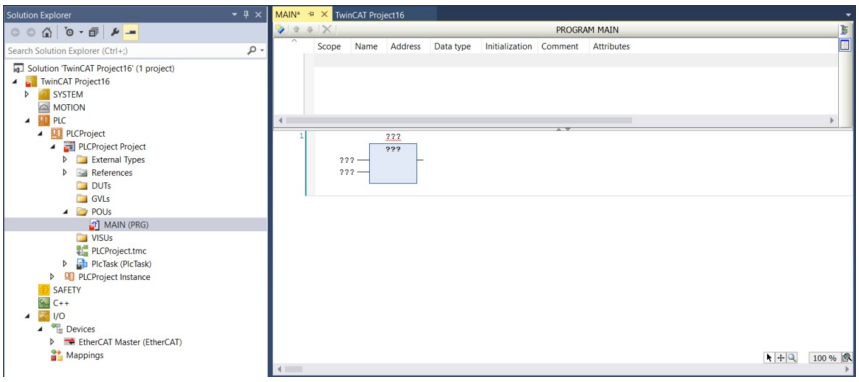


Illustration 32: Insert Empty Box

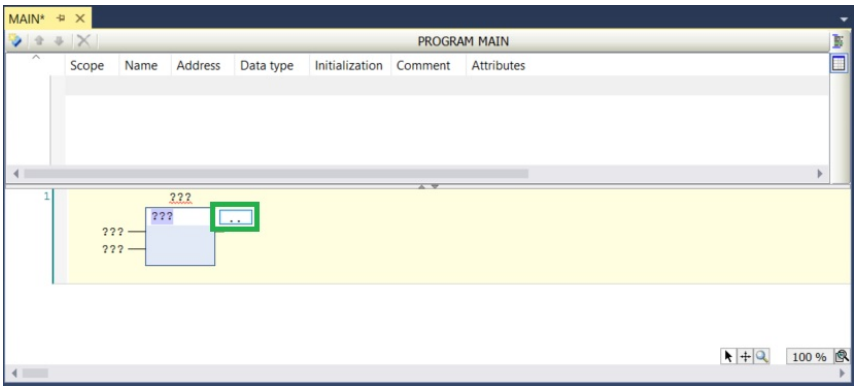
→ *EMPTY BOX* is inserted in the program logic window as shown below.



e30bu462.10

Illustration 33: Empty Box

3. Click the highlighted button to open *Input Assistant* dialog window.

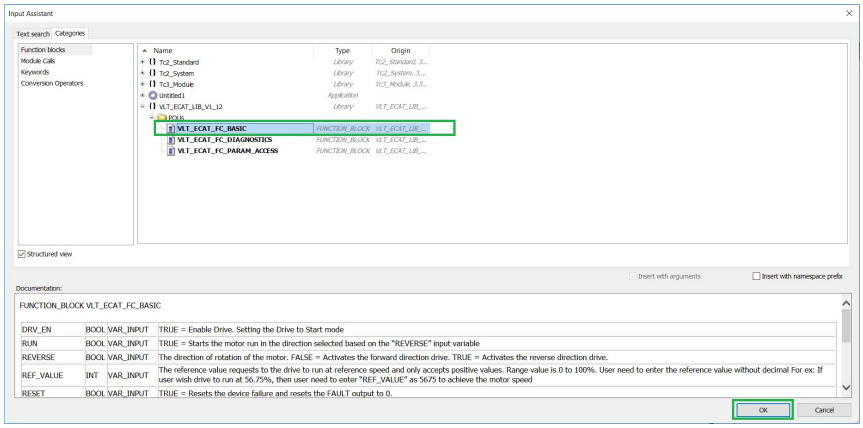


e30bu463.10

Illustration 34: Instantiate Function Block

→ *Input Assistant* window appears.

4. Navigate to *Categories* tab, and select *VLT_ECAT_FC_BASIC* function blocks.



e30bu464.10

Illustration 35: Input Assistant for Selecting the Function Block to Import

5. Click *OK*.

→ *Auto Declare* windows appear.

6. Enter the instance name as *Pump01_VLT302* and instance description.
7. Click *OK*.
8. *Pump01_VLT302* instance is added in the program logic window.

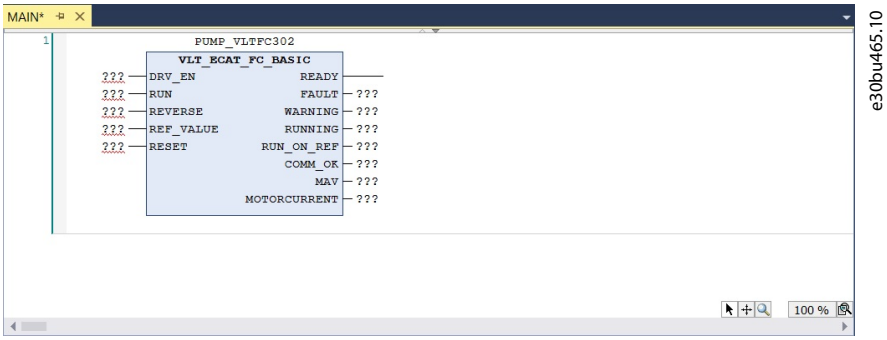


Illustration 36: Basic Operation Function Block

2.6 Creating A Process Variable for the Function Block Instance

Procedure

1. In the program logic window, create a variable *DRV_EN* for the input pin *DRV_EN* of *Pump01_VLT302* and press *Enter*.
   → *Auto Declare* window appears.
2. Click *OK*.
3. Repeat the same steps to create variables for I/O pins as shown below.

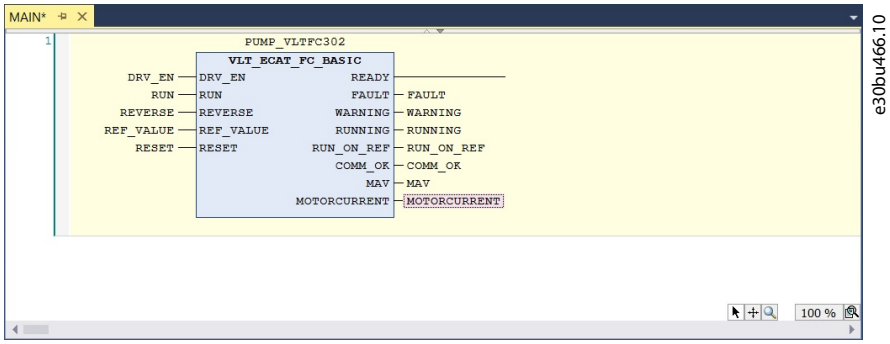
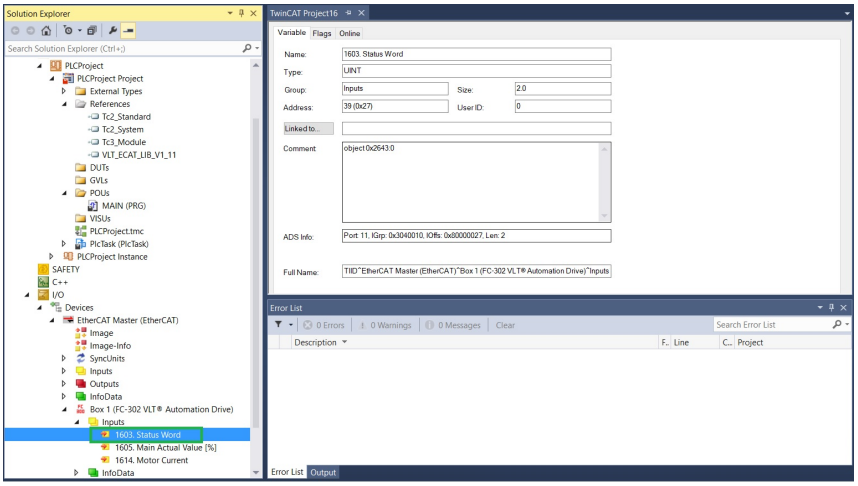


Illustration 37: Variable Assignment for Basic Operation Function Block

2.7 Creating A Mapping of Function Block Instance Variables and EtherCAT Slave's Variables

Procedure

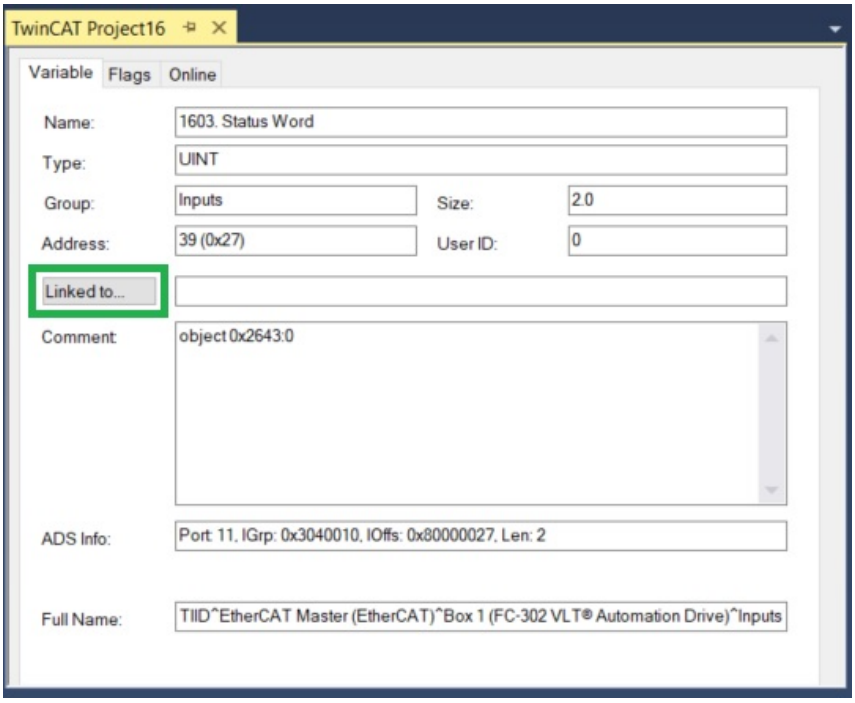
1. Navigate to *TwinCAT project16* ⇒ *I/O* ⇒ *Devices* ⇒ *EtherCAT Master (EtherCAT)* ⇒ *Box 1 (FC-302 VLT® Automation Drive)* ⇒ *Inputs* ⇒ *1603. Status Word*.



e30bu467.10

Illustration 38: Process Variable Mapping

2. Select the *Variable* tab.



e30bu468.10

Illustration 39: Mapping of the Process Variable

3. Click *Linked to...* button.

   → *Attach Variable* dialog window appears.

4. Navigate and select *PLC* ⇒ *PLCProject* ⇒ *PLCProject Instance* ⇒ *Main.* ⇒ *Inputs* ⇒ *StatusWord*.

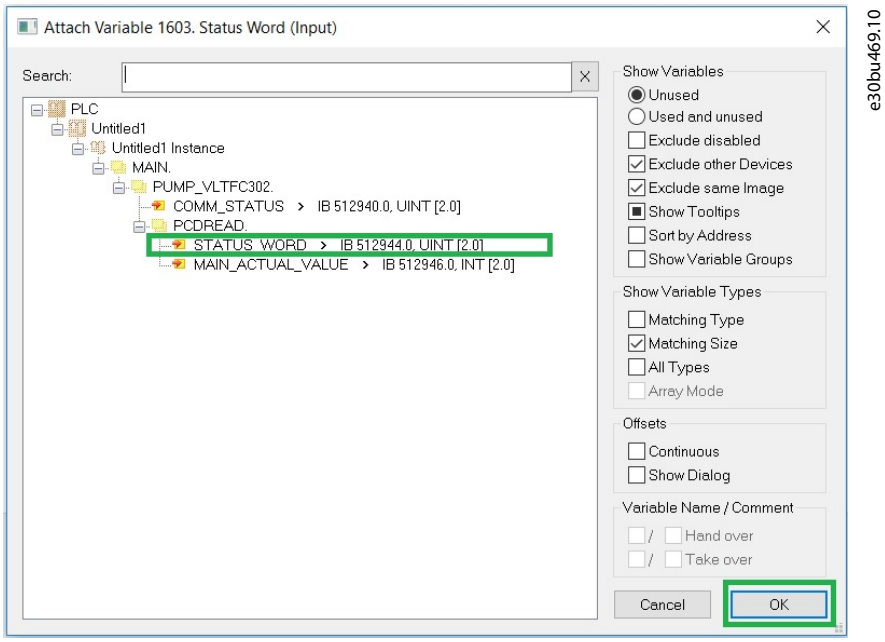


Illustration 40: Mapping of Process Variable

5. Click *OK*.

   → Status word of EtherCAT slave FC 302 is mapped with *Pump01_VLT302* function block instance.

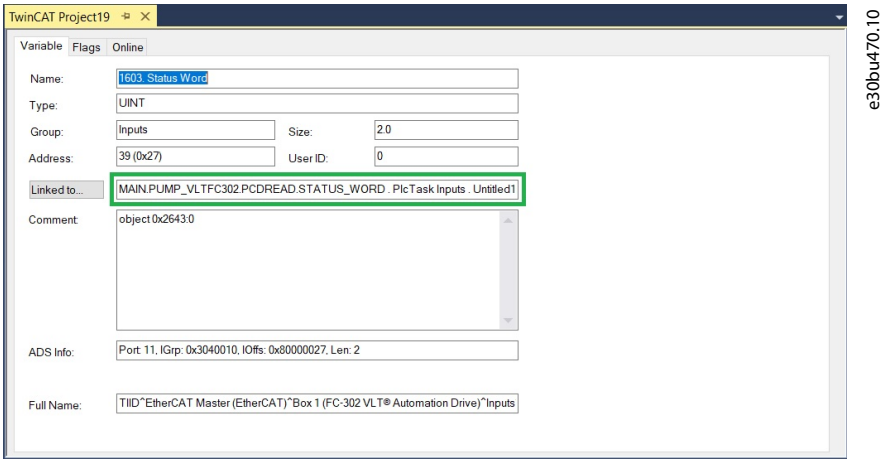


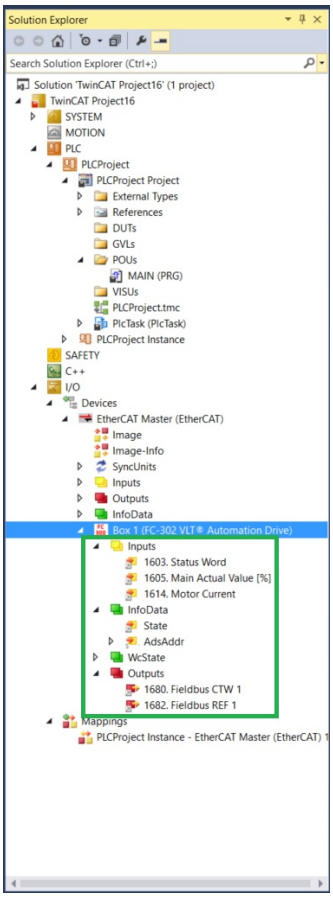
Illustration 41: Mapping of the Process Variable

6. Follow the same steps and map the following EtherCAT slave variables with function block instance variables.

Table 9: EtherCAT Slave Variables with Function Block Instance Variables

Type	EtherCAT slave	Function block instance
Input	1603. Status word	*PLC⇒PLCProject⇒PLCProject Instance⇒Main.⇒Function Block Instance.⇒PCDREAD.⇒STATUS_WORD*
Input	1605. Main actual value [%]	*PLC⇒PLCProject⇒PLCProject Instance⇒Main.⇒Function Block Instance.⇒PCDREAD.⇒MAIN_ACTUAL_VALUE*

Type	EtherCAT slave	Function block instance
Input	1614. Motor current	*PLC⇒PLCProject⇒PLCProject Instance⇒Main.⇒Function Block Instance.⇒PCDREAD.⇒MOTORCURRENT*
Output	1680. Fieldbus CTW 1	*PLC⇒PLCProject⇒PLCProject Instance⇒Main.⇒Function Block Instance.⇒PCDREAD.⇒CONTROL_WORD*
Output	1682. Fieldbus REF 1	*PLC⇒PLCProject⇒PLCProject Instance⇒Main.⇒Function Block Instance.⇒PCDREAD.⇒REF_VALUE*
Infodata	State	*PLC⇒PLCProject⇒PLCProject Instance⇒Main.⇒Function Block Instance.⇒COMM_STATUS*



e30bu471.10

Illustration 42: Mapping of the Process Variable

→ Once the EtherCAT slave variables mapped with function block instance, the icon of the EtherCAT slave variable changes to

.

2.8 Identifying AMS_NET_ID of the EtherCAT Master Device

Procedure

1. Double-click the EtherCAT master device *Device 2 (EtherCAT)*.

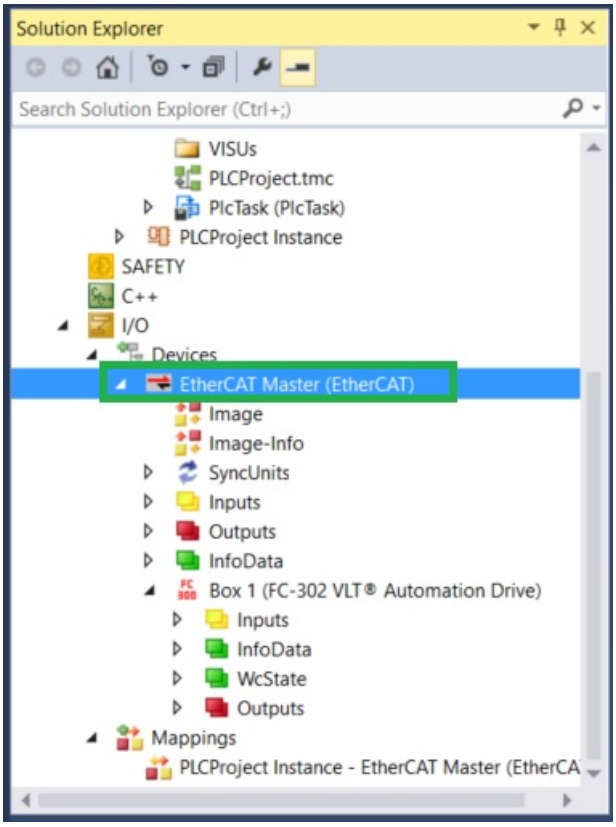
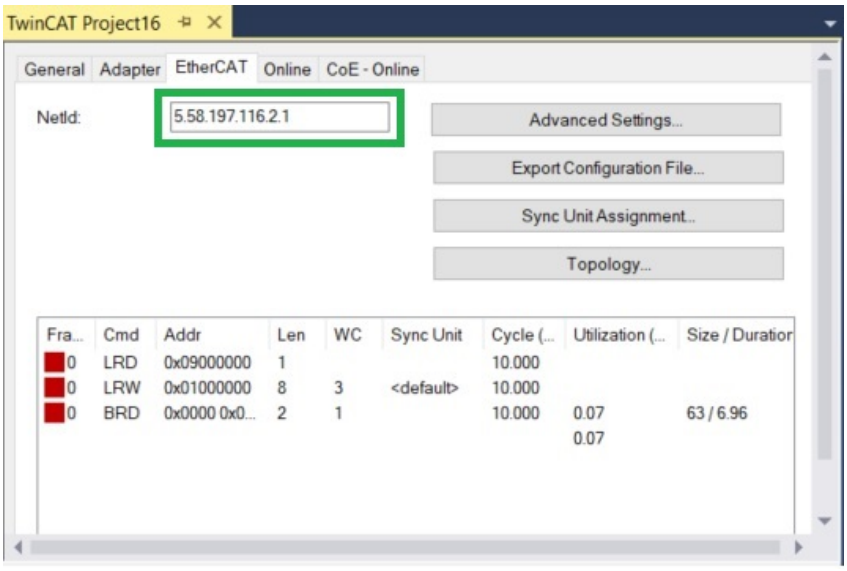


Illustration 43: EtherCAT Master Device Configuration

2. Navigate to *EtherCAT* tab and find the Netid, that is, AMS_NET_ID.



e30bu473.10

Illustration 44: EtherCAT Master Device Configuration

2.9 Identifying EtherCAT Slave Address

Procedure

1. Double-click the EtherCAT slave device *Box 1 (VLT® Automation Drive FC 302)*.

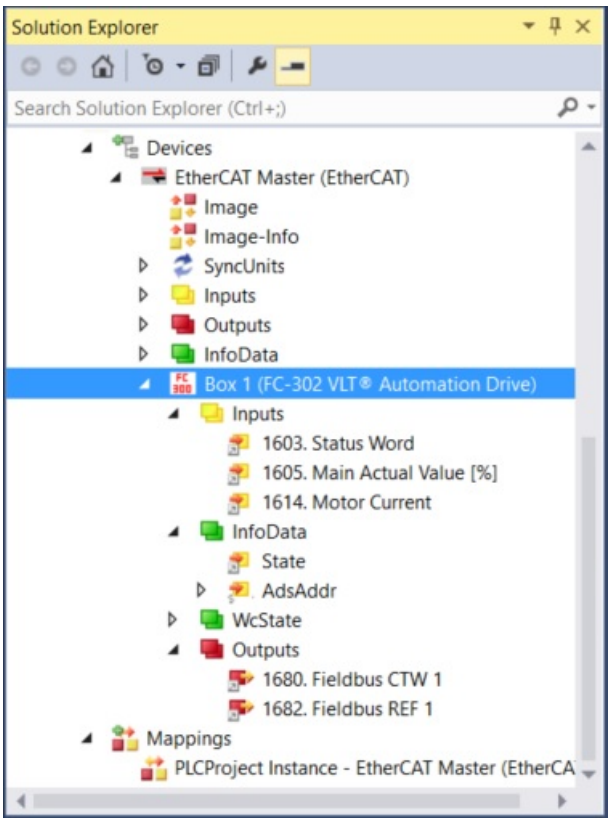


Illustration 45: EtherCAT Slave Address Configuration

2. Navigate to *EtherCAT* tab and find the slave address of the device in *EtherCAT Addr:* text field that is, SLAVE_ADDR.

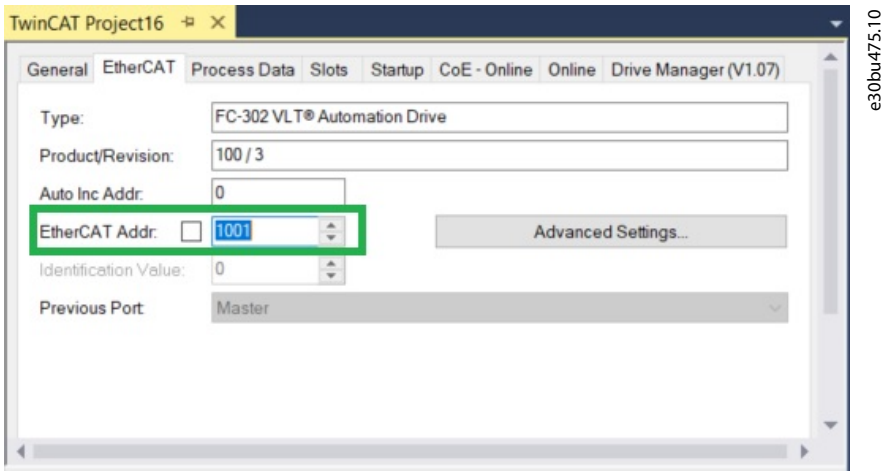


Illustration 46: EtherCAT Slave Address Configuration

2.10 Saving and Building the Project

Procedure

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

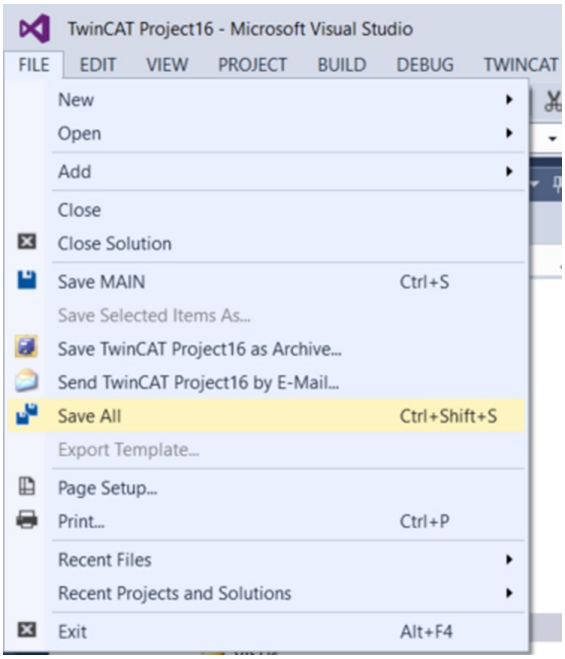


Illustration 47: Save the Project

2. To compile the project, click *BUILD* ⇒ *TwinCAT Project* from the menu bar.

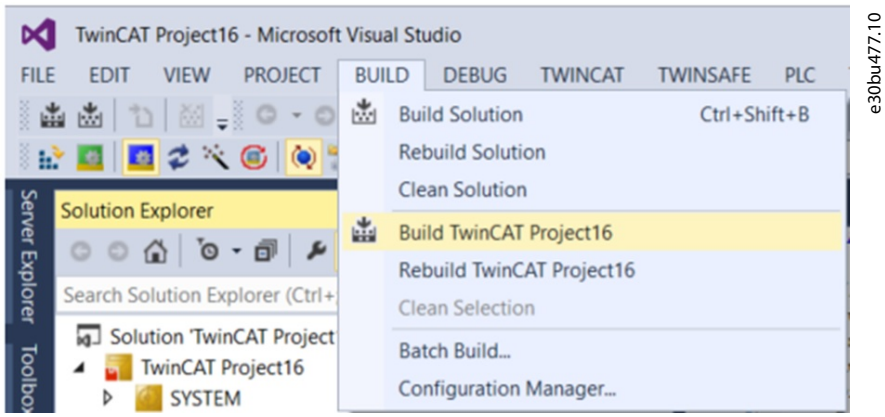
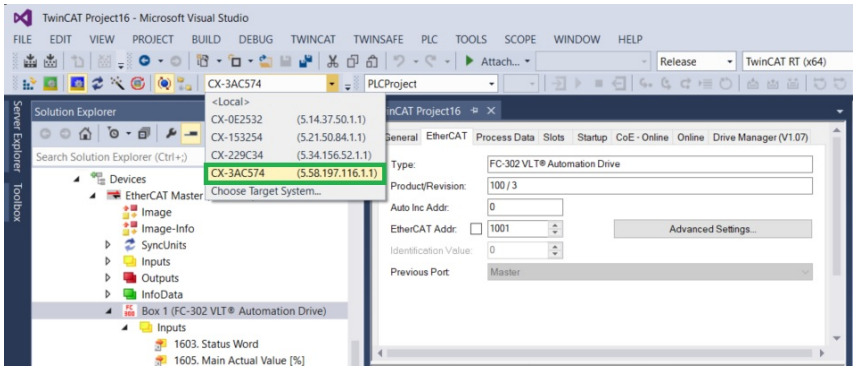


Illustration 48: Build the Project

2.11 Transfering PLC Project to the Embedded PC

Procedure

1. Ensure that the PLC project is error-free.
2. Select the PLC to be loaded and started in the drop-down list *Active PLC Project* in the *TwinCAT PLC Toolbar Options*.



e30bu478.10

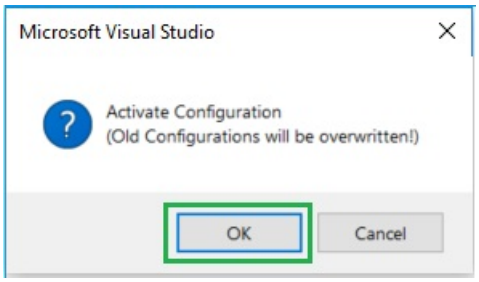
Illustration 49: Select the PLC to Download the Project

3. In the TwinCAT XAE Base Toolbar Options, click *Activate configuration* icon.

Activate configuration icon:

→ A dialog appears asking whether to activate the configuration.

4. Click *OK*.



e30bu479.10

Illustration 50: Activate Configuration Confirmation Window

→ A dialog appears asking whether to restart TwinCAT in Run mode.

5. Click *OK*.

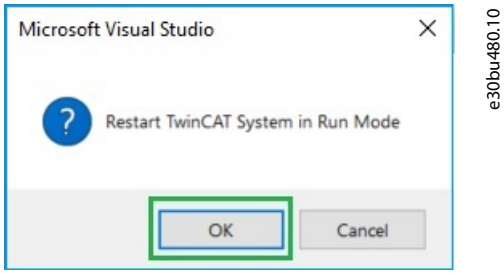


Illustration 51: Restart TwinCAT System Confirmation Window

→ The configuration is activated, and TwinCAT is set to Run mode. The taskbar shows the status icon. Activation also transfers the PLC project to the controller.

Status icon:

6. Select the command *Login* icon in the PLC menu or in the TwinCAT PLC Toolbar Options.

Login icon:

→ A dialog appears asking whether to create and load the application onto the controller.

7. Click *Yes*.

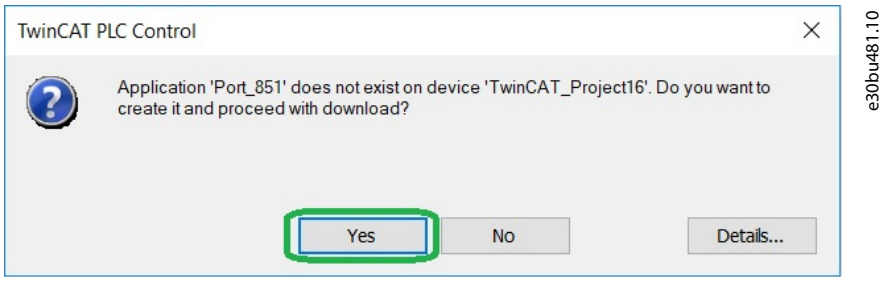


Illustration 52: PLC Download Confirmation Window

→ The PLC project is loaded onto the controller.

8. Select the command *Start* icon in the PLC menu or in the TwinCAT PLC Toolbar Options, or press [F5].

Start icon:

→ The program is running on the controller.

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 the PLC has control of the drive.

Table 10: Parameter Settings

Parameter	Value
Parameter 8-01 Control Site	*[0] Digital and Ctrl.word*, or *[2] Control word only*
Parameter 8-02 Control Word Source	*[3] Option A*

NOTICE

When *parameter 8-01 Control Site* is set to *[0] Digital and Ctrl. Word*, a connection between terminal 12/13 and terminal 27 is required to control the motor.

3. 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.

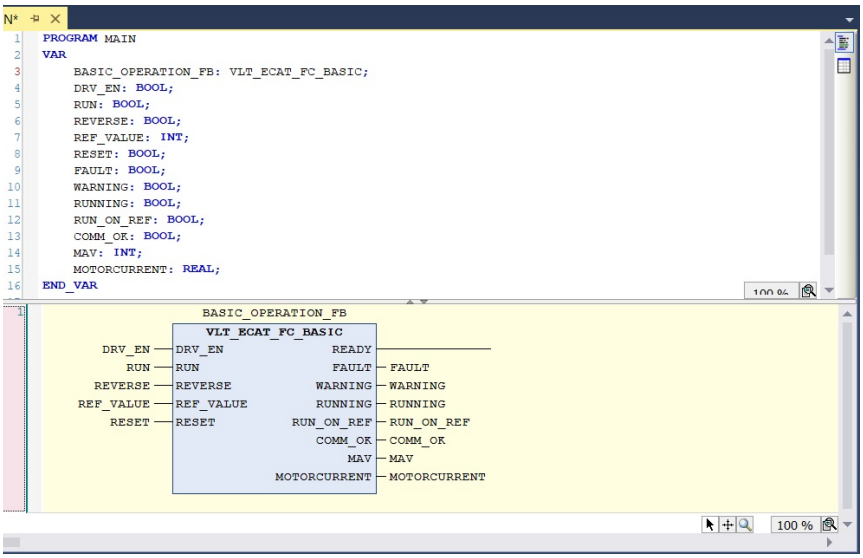
Table 11: Parameter Settings

Parameter	Value
Parameter 8-04 Control Word Timeout Function	*[0] Off*, or *[1] Freeze Output*, or *[2] Stop*, or *[3] Jogging*, or *[4] Max Speed*, or *[5] Stop and trip*

4. The add-on instruction 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.
5. Ensure physically that LCP mode is set to *Auto On* mode.

3.2 Basic Operation Function Block

Context:



e30bu482.10

Illustration 53: Example of Basic Operation Function Block

Procedure

1. Enable the drive to ready state.
   - **A** Verify function block output pin *COMM_OK* is set to *TRUE*. That is, communication between PLC and the drive is healthy.
   - **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 12: Pin Name and 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 *parameter 16-14 Motor Current*.

   - **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 13: Pin Name and 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 *parameter 16-14 Motor Current*.

   - **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 14: Pin Name and 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 *parameter 16-14 Motor Current*.

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 15: Pin Name and 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 *parameter 16-14 Motor Current*.

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 16: Pin Name and 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 *parameter 16-14 Motor Current*.

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

F Set the value *0* to the *REF_VALUE* input pin of the function block.

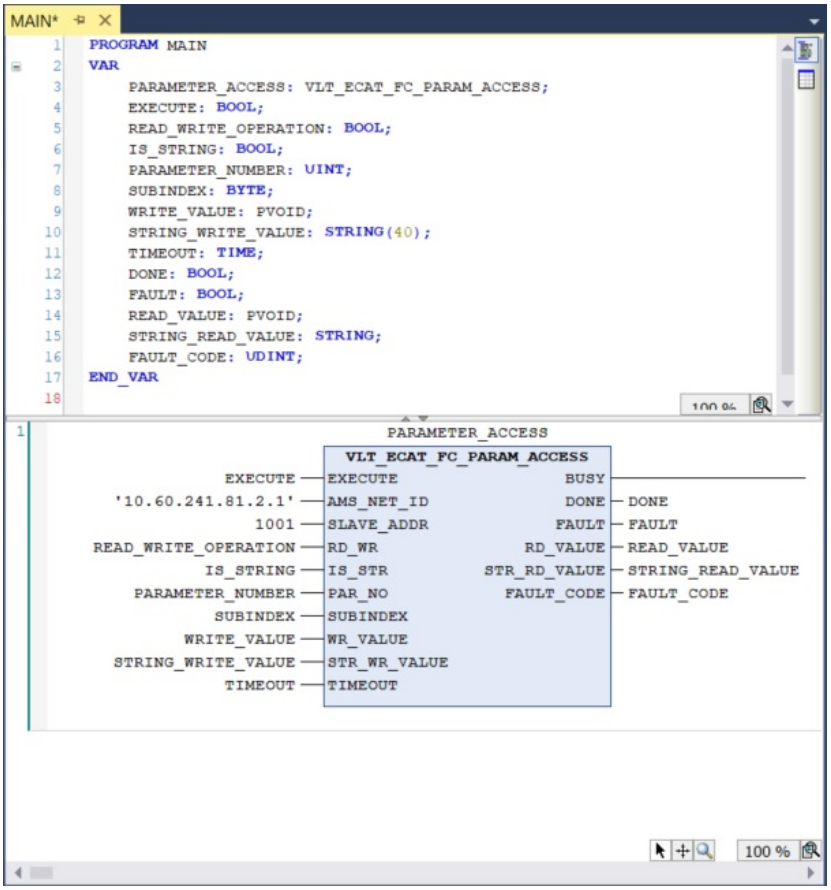
G Verify the following values with the function block output.

Table 17: Pin Name and Value

Pin name	Expected value
READY	TRUE
FAULT	FALSE
WARNING	FALSE
RUNNING	FALSE
RUN_ON_REF	FALSE
COMM_OK	TRUE
MAV	0x0000
MOTORCURRENT	Same as *parameter 16-14 Motor Current*.

3.3 Parameter Access Function Block

Context:



e30bu483.10

Illustration 54: Example of Parameter Access Function Block

- To identify AMS_NET_ID of EtherCAT master device, refer to 2.8 Identifying AMS_NET_ID of the EtherCAT Master Device.
- To identify the slave address of the EtherCAT slave device, refer to 2.9 Identifying EtherCAT Slave Address.

Procedure

1. 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 as shown.

Table 18: Pin Name and Set Value

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

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

Table 19: Pin Name and Value

Pin name	Expected value
FAULT_CODE	0x0000_0000
RD_VALUE	300

2. 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 as shown.

Table 20: Pin Name and Set Value

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

   - **C** Verify that *parameter 3-41 Ramp 1 Ramp Up Time* is 5.00 s using the 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 21: Parameter Value Settings

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

Parameter number	Parameter name	Value
310.4	PRESET REFERENCE	0.00

C Set the following values to the input pin of the function block to read the multiple element array type parameter.

Table 22: Pin Name and Set Value

Pin name	Set value for *parameter 3-10 [0] Preset Reference*
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	1
EXECUTE	TRUE

D Verify that the RD_VALUE output pin of the function blocks has value 0x09C4 (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 to read the multiple element array type parameter.

Table 23: Pin Name and Set Value

Pin name	Set value for *parameter 3-10 [1] Preset Reference*
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	2
EXECUTE	TRUE

G Verify that the RD_VALUE output pin of the function blocks has value 0x1194 (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 to read the multiple element array type parameter.

Table 24: Pin Name and Set Value

Pin name	Set value for *parameter 3-10 [4] Preset Reference*
RD_WR	FALSE
IS_STR	FALSE
PAR_NO	310
SUBINDEX	5
EXECUTE	TRUE

J Verify that the RD_VALUE output pin of the function blocks has value 0x0.

4. Write an array type parameter.

A Use the LCP of the drive to verify if *parameter 3-10 [3] Preset Reference* is showing a 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 to write the attribute value to the array element.

Table 25: Pin Name and Set Value

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

D Verify that 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 to write a string value to the drive.

Table 26: Pin Name and Value To Be Set

Pin name	Value to be set
RD_WR	TRUE
PAR_NO	1208
SUBINDEX	0
STR_WR_VALUE	*abcdefghijklmnopqrstuvwxyzabcdefghijklmn*
IS_STR	TRUE
EXECUTE	TRUE

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

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 to read the string type parameter.

Table 27: Pin Name and Value To Be Set

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

C Verify the *STR_RD_VALUE* output pin has the same value as *parameter 12-08 Host Name*.

3.4 Diagnostics Function Block

Context:

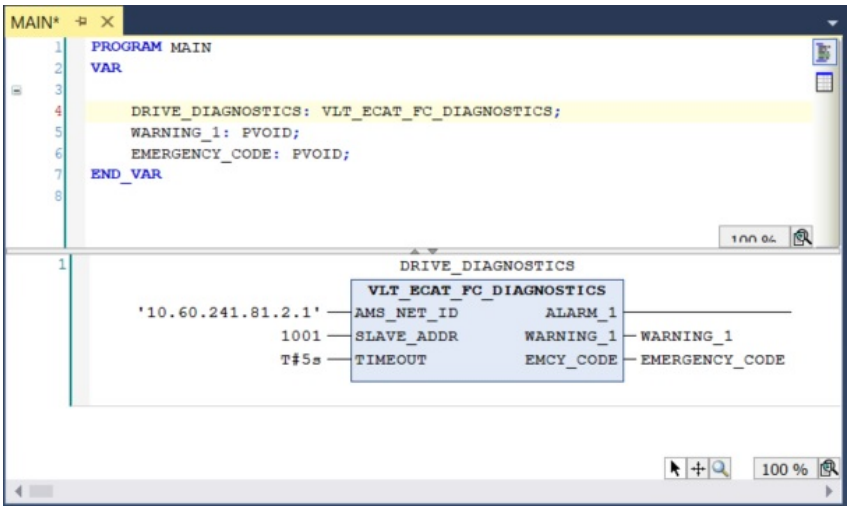


Illustration 55: Example of Diagnostics Function Block

- To identify AMS_NET_ID of EtherCAT master device, refer to 2.8 Identifying AMS_NET_ID of the EtherCAT Master Device.
- To identify the slave address of the EtherCAT slave device, refer to 2.9 Identifying EtherCAT Slave Address.

Procedure

1. Set *parameter 8-07 Diagnosis Trigger* to *[2] trigger alarm/warn* using the LCP.
2. Set *parameter 6-10 Terminal 53 Low Voltage* to *2 V* using the LCP.
3. Verify the following values with the function block output for live zero warning after 10 s.

Table 28: Pin Name and Value

Pin name	Expected value
ALARM_1	0x00010000
WARNING_1	0x00000000
EMCY_CODE	0x00001000

4. Verify whether the emergency message is shown in the error List log window of TwinCAT system manager 2 times. The 1st time is for the warning (*Hex: 1000, 01, 08 00 00 00 00*), and the 2nd time is for the alarm (*Hex: 1000, 01, 01 00 00 00 00*).
5. Set *parameter 6-10 Terminal 53 Low Voltage* to *0.07 V* using the LCP.
6. Press *Reset* to remove the live zero error alarm on the LCP display.
7. Press *Auto On* on the LCP display.
8. Verify the following values with the function block output.

Table 29: Pin Name and Value

Pin name	Expected value
ALARM_1	0x00000000
WARNING_1	0x00000000
EMCY_CODE	0x00000000

9. Verify whether the emergency message is shown in the error list log window of TwinCAT system manager as *Hex: 0000, 00, 00 00 00 00 00*.

Index

A

Auto scan 19, 22

B

Basic operation function block 6, 45

C

Control and monitoring 6

D

DDT 5
Diagnostics function block 12, 53
DUT 5

E

ESI 5, 18
EtherCAT Slave Information 5, 18

F

Failure management 6

P

Parameter access function block 10, 49
PDO 5

R

Reverse 6

S

SDO 5, 10
Speed regulation 6

ENGINEERING
TOMORROW

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