ENGINEERING
TOMORROW

Application Guide

iC7 Integration into Siemens TIA Portal

iC7 Series

drives.danfoss.com | iC7

Contents

1 Introduction

2 Safety

3 Function Block Description

4 Configure the iC7 Drive

5 Get Started with Danfoss Function Blocks

6 Faceplate for HMI Application

7 Drive Parameters

8 Determining the iC7 Drive Object Parameter Number

9 Troubleshooting

1 Introduction

1.1 Version History

This guide is regularly reviewed and updated. All suggestions for improvement are welcome.

The original language of this guide is English.

Table 1: Version History

Version	Remarks
AB479721468891, version 0101	First release.

1.2 Purpose of this Application Guide

This application guide is intended for qualified personnel such as:

- Automation engineers.
- Commissioning engineers who have experience of operating with parameters and basic knowledge of the iC7 drive.
- The engineers must have knowledge with Siemens TIA Portal and skills for using Step 7, WinCC (TIA Portal).

This application guide provides information about how to integrate the Danfoss iC7 drives into Siemens TIA portal V17. The library and examples are made in Siemens TIA portal V17, but it can be updated to operate in a newer version of the Siemens TIA portal.

The function blocks and faceplate are not protected and can be modified to serve the specific requirements by the end-user.

Danfoss takes no responsibility for losses due to code faults in these function blocks or for wrong use.

This application guide does not include:

- A description of the used Siemens equipment and programing tools that are necessary to handle the application.
- How to create a network connection between the controller/PLC and the device/PC, and the installation of the Siemens TIA Portal.

NOTE: Refer to the *Siemens programming and operating guide* for further information.

1.3 Additional Resources

Additional resources are available to help understand the features, safely install, and operate the iC7 products:

- Operating guide iC7 Series PROFINET RT OS7PR.
- Operating guide iC7 Series Modbus TCP OS7MT.
- Safety guides, which provide important safety information related to installing iC7 drives.
- Installation guides, which cover the mechanical and electrical installation of drives.
- Design guides, which provide technical information to understand the capabilities of the iC7 drives for integration into motor control and monitoring systems.
- Application guides, which provide instructions on setting up the drive for a specific end use.
- Other supplemental publications, drawings, and guides are available at www.danfoss.com

Latest versions of Danfoss product guides are available for download at https://www.danfoss.com/en/service-and-support/documentation/.

1.4 General

Configuration examples and illustrations made in this guide are based on the default settings of the drive. Special parameter settings to fit the current application can be possible, but it can also harm the description made in this application guide.

1.5 Abbreviations and Acronyms

Term	Definition
CTW	Control word
DB	Data block
FB	Function block
GSD	General station description
MAV	Main actual value (Process value)
N/A	Not applicable
N2	Normalized value [1]
PD	Process data signal
PLC	Programmable logic controller
PV	Process value (Main actual value)
REF	Reference (Setpoint)
RPM	Revolution per minute
SP	Setpoint (Reference value)
STW	Status word
TIA	Totally integrated automation [2]
UDT	User-defined types

1) According to PROFIdrive specification (IEC 61800-7-203).

2) Siemens automation platform, for example, for S7-1200, S7-1500 controllers and WinCC HMI systems.

1.6 Trademarks

- Siemens® is a registered trademark of Siemens Trademark GmbH & Co. KG.
- Simatic IT® is a registered trademark of Siemens AG.
- WinCC® is a registered trademark of Siemens AG.
- TIA Portal® is a registered trademark of Siemens AG.
- PROFIBUS® and PROFINET® are registered trademarks of PROFIBUS and PROFINET International (PI).
- PROFIdrive® is a registered trademark licensed by PROFIBUS and PROFINET International (PI)
- MODBUS® is a registered trademark of Schneider Electric USA, Inc.
- Ethernet® is a registered trademark of Xerox Corporation.

1.7 Disclaimer

The library 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 non-infringement. In no event must the authors or any legal entity part of the 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.

2 Safety

2.1 Safety Symbols

The following symbols are used in Danfoss documentation.

⚠ DANGER

Indicates a hazardous situation which, if not avoided, will result in death or serious injury.

⚠ WARNING

Indicates a hazardous situation which, if not avoided, could result in death or serious injury.

⚠ CAUTION

Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.

NOTICE

Indicates information considered important, but not hazard-related (for example, messages relating to property damage).

The guide also includes ISO warning symbols related to hot surfaces and burn hazard, high voltage and electrical shock, and referring to the instructions.

⚠	ISO warning symbol for hot surfaces and burn hazard
⚠	ISO warning symbol for high voltage and electrical shock
📖	ISO action symbol for referring to the instructions

2.2 General Safety Considerations

The end-user is responsible for addressing any regulatory requirements to the installation. Refer to the *iC7 Design Guide* for further information of the mechanical installation considerations.

⚠ WARNING

POTENTIAL DAMAGE TO SYSTEMS

Some parameters in system configuration can significantly change the system response and even make the system non-functioning. Potential damage to other systems, for example motor, or other components related to motor control can occur if the parameters are not set correctly.

- Make sure to understand the parameters fully before changing any of the parameters in the drive.

3 Function Block Description

3.1 Advantages of Using Function Blocks

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

This application guide is applicable for the following options for Danfoss iC7 drive:

- PROFINET RT OS7PR
- Modbus TCP OS7MT

Basic skeleton:

- Function blocks provide the basic infrastructure towards the user.
- Free up time to focus on the complex and application-specific implementation of the iC7 drive.
- Reuse of a function block several times in a program without rewriting the function block.
- Easy to use, knowledge of the internal operation of the drive or complex algorithms is not required.

Pretested function:

- The function block is pretested for operation and functionality.

Extensibility:

- Function blocks can be extended in future by Danfoss. It is possible to incorporate the function blocks with minimal modification in the existing program.

NOTE: Function blocks without any memory capabilities (Functions/FCs) are mentioned as function blocks in this guide.

3.2 Overview of the Danfoss Library

- Basic operation function block *DD:iC7:PNET:Basic [FC400]*: Dedicated for basic motor control via the iC7 drive with PROFINET RT interface.
- Basic operation function block *DD:iC7:MBUS:Basic [FB400]*: Dedicated for basic motor control via the iC7 drive with Modbus TCP interface.
- Faceplate *DD:iC7:Basic*: Dedicated for use with basic motor control and monitoring function block.

The function blocks only support the iC Speed profile and are only intended for use with Simatic S7-1200 series and Simatic S7-1500 series controller systems. Use the following minimum firmware versions in the Simatic controllers:

- S7-1200 CPU as of FW 4.2.
- S7-1500 CPU as of FW 2.5.

3.3 Basic Operation Function Block

The function block provides the following features:

- Control and monitoring:
  - Control the motor and monitor the status and feedback from the drive.
- Direction:
  - Control the forward and reverse direction of the motor.
- Speed control:
  - Allow the speed reference to the drive to be changed.

4 Configure the iC7 Drive

4.1 Preconditions for Start Usage of the Function Block and Faceplate

The following preconditions are required before the integration of the iC7 drive and the use of the iC7 library can occur.

- Download and install the GSD file for the individual application. This allows the addition to the hardware catalog in the TIA portal.
- Download the iC7 library for start usage of the function block and faceplate for the iC7 drive.

4.2 Installation of General Station Description File

Before the function blocks can be used in a real application, the basic steps for iC7 series integration into the TIA portal are needed.

NOTE: Install WinCC if a faceplate addition is needed. WinCC Advanced is used in the following examples.

All iC7 drive applications (Automation, HVACR, AQUA, and so forth) have their own dedicated general station description (GSD) file allocated. Import the GSD file into the TIA Portal before the configuration of each drive in the network can begin. The GSD file is available on the Danfoss website.

1. Download the general station description (GSD) file and save the file locally.

   The files can be installed from the following link: https://www.danfoss.com/en/service-and-support/downloads/dds/fieldbus-configuration-files/.

2. Verify that the GSD file matches the firmware and application version used in the drive.

   A newer GSD version is backward compatible to previous firmware and application version, as long as the existing features are used. The new features, which are described in the newer GSD file are not compatible with older firmware and application versions. To match the dedicated drive firmware and application version, it is always recommended to download and install the dedicated GSD file.

3. Select *Option* menu, and select *Manage general station description files (GSD)* to install the GSD file.

   The following example show how the GSD file for the iC7-Automation product is installed.

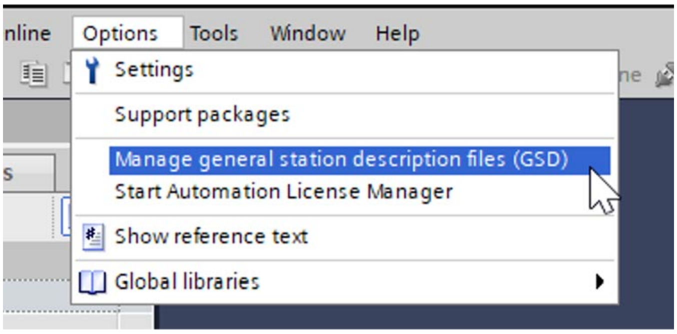


Figure 1: Manage General Station Description File

4. Select the 3 dots in the upper right corner in the dialog box, and locate the file folder.

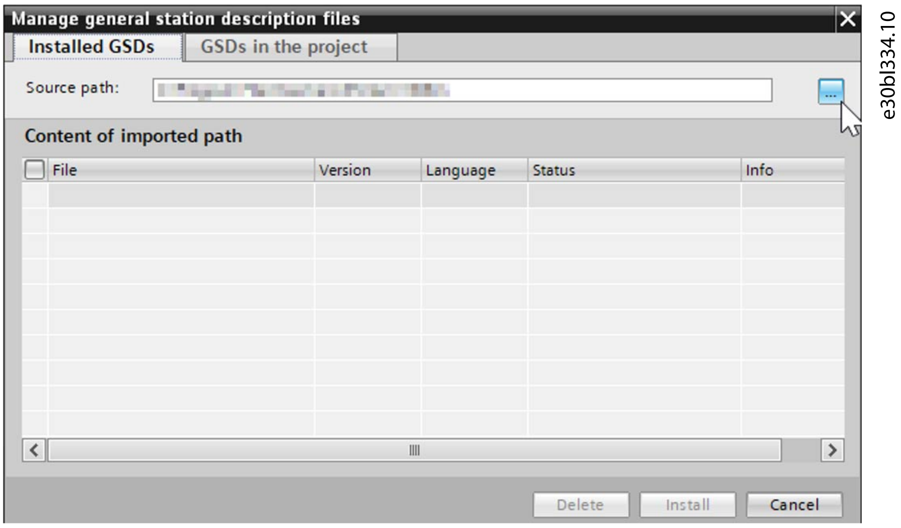
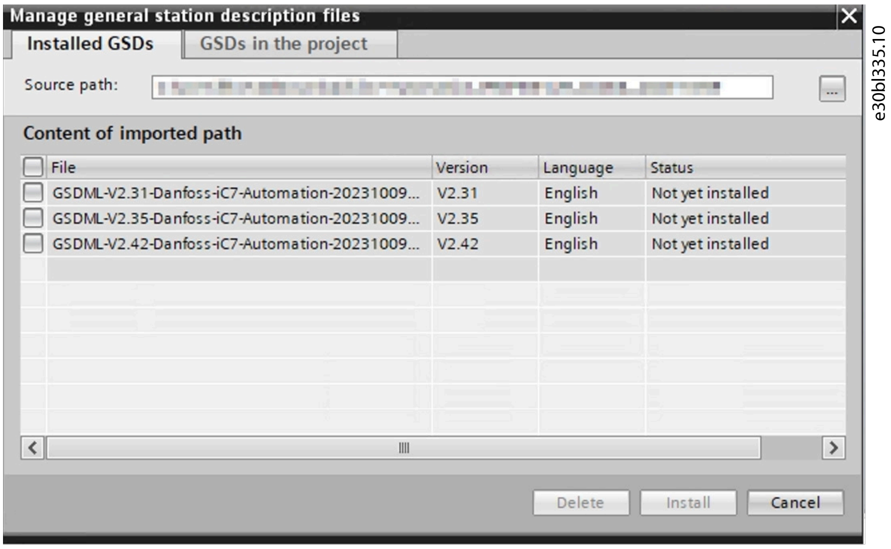


Figure 2: Select the Location of the General Station Description File

5. 3 different GSD files appear in the dialog box. It is recommended to always install the latest version of the GSD files.

The files differ in the PROFINET specified GSDML schema version numbers and the build date of the files.

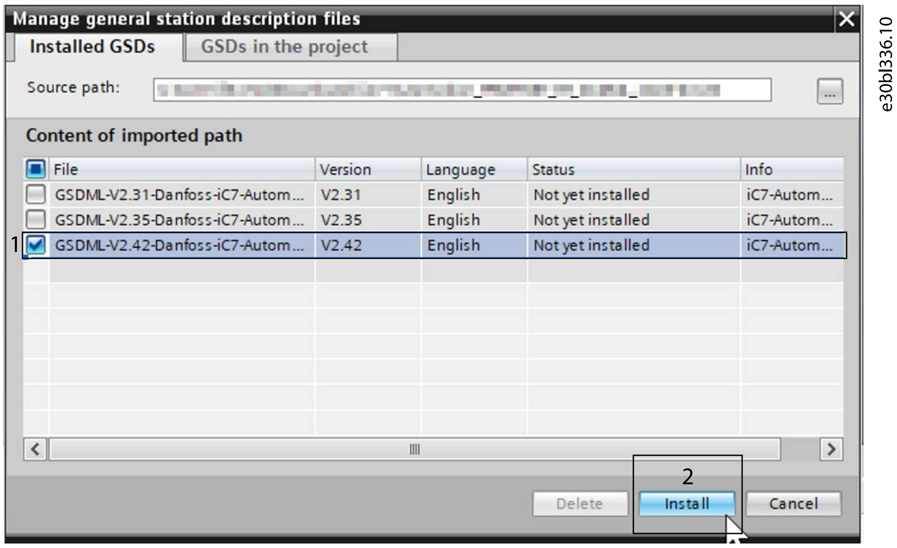
Legacy support to other controller systems, and older versions are also provided.



☐ File	Version	Language	Status
☐ GSDML-V2.31-Danfoss-iC7-Automation-20231009...	V2.31	English	Not yet installed
☐ GSDML-V2.35-Danfoss-iC7-Automation-20231009...	V2.35	English	Not yet installed
☐ GSDML-V2.42-Danfoss-iC7-Automation-20231009...	V2.42	English	Not yet installed

Figure 3: Content of Imported Path

6. Checkmark the preferred file, and select the *Install* button.



Content of imported path

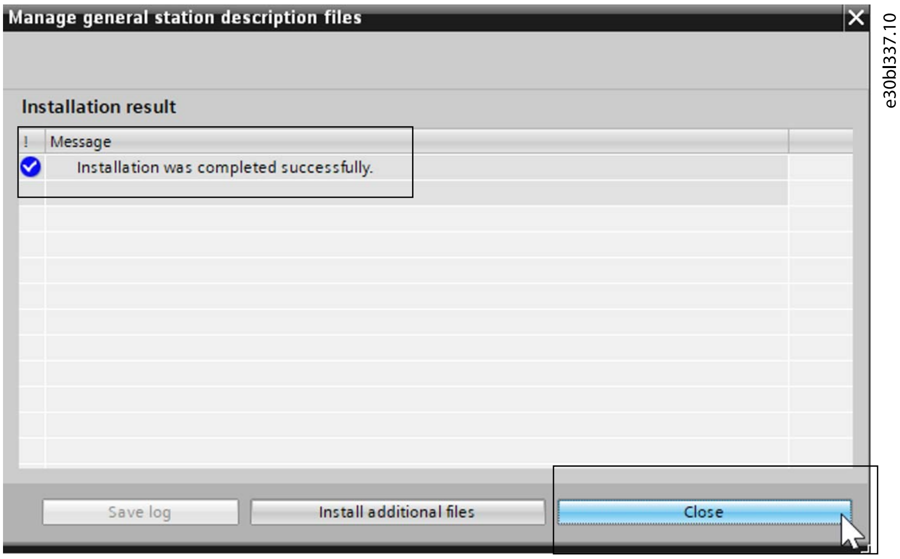
File	Version	Language	Status	Info
□ GSDML-V2.31-Danfoss-iC7-Autom...	V2.31	English	Not yet installed	iC7-Autom...
□ GSDML-V2.35-Danfoss-iC7-Autom...	V2.35	English	Not yet installed	iC7-Autom...
☑ GSDML-V2.42-Danfoss-iC7-Autom...	V2.42	English	Not yet installed	iC7-Autom...

Figure 4: Selecting and Installing the GSD File

1	Selected GSD file and version	**2**	Install button

➲ A message box appears when the GSD file is successfully installed.

7. Select the *Close* button for update of the hardware catalog.



Installation result

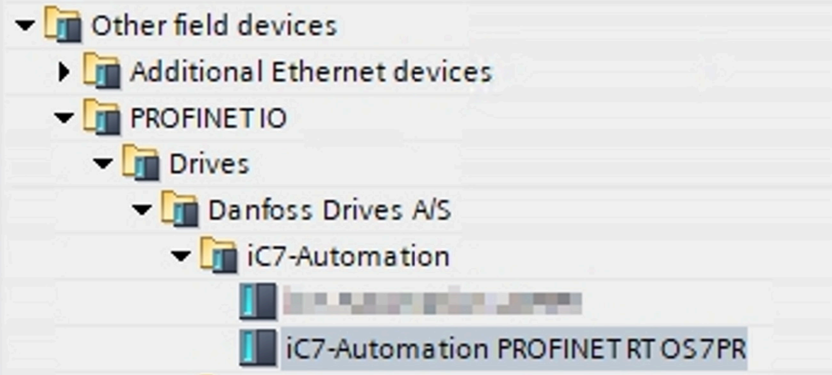
!	Message
✓	Installation was completed successfully.

Figure 5: Status of the Installation Result

8. When the hardware catalog is updated, the iC7 drive is available in the folder *Other field devices→ PROFINET IO→ Drives→ Danfoss Drives A/S→iC7 product series.*

The *iC7 Product series* is the Automation, HVACR, AQUA, and so forth.

NOTE: It is possible to install multiple instances of the same GSD file, but with different release dates. It is vital that the correct instance is selected before moving on. The following example showing the instance of the current GSD installed for the iC7-Automation Series.

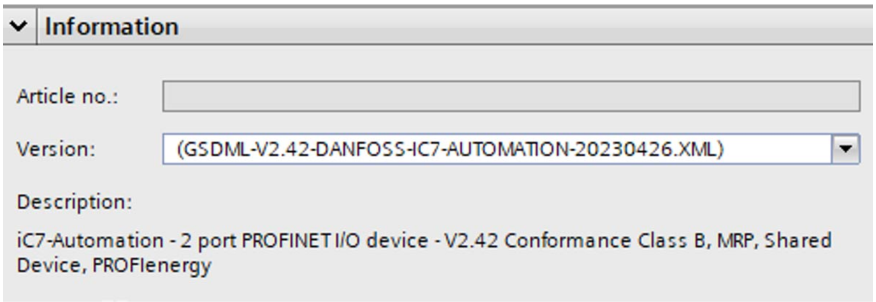


- Other field devices
  - Additional Ethernet devices
  - PROFINET IO
    - Drives
      - Danfoss Drives A/S
        - iC7-Automation
          - [redacted]
          - iC7-Automation PROFINET RT OS7PR

Figure 6: Example (iC7 Automation Series): Instance of Current GSD File

9. If multiple instances of the same GSD file are installed, each individual version can be selected from the drop-down list in the information window. The version to select depends on the firmware and application version in the drive.

The compatibility matrix is part of the GSD library for the iC7 drive series, refer to the following link https://www.danfoss.com/en/service-and-support/downloads/dds/fieldbus-configuration-files/.



Information

Article no.:

Version: (GSDML-V2.42-DANFOSS-IC7-AUTOMATION-20230426.XML)

Description:

iC7-Automation - 2 port PROFINET I/O device - V2.42 Conformance Class B, MRP, Shared Device, PROFIenergy

Figure 7: Example: Selected Version in Information Window

4.3 Adding the iC7 Drive to the TIA Project

1. Drag and drop the iC7 drive from the hardware catalog to the network view window.

Change the name of the device if needed.

2. Assign the drive to the controller for PROFINET connection.

In the following example, the iC7 drive is assigned to PLC_1 with the given default name *ic7-Automation*.

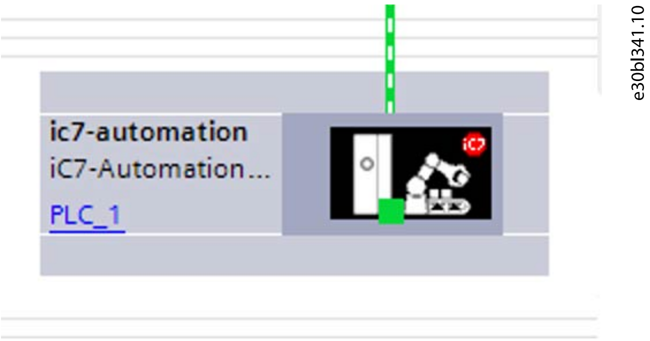


Figure 8: iC7 Drive Assigned to PLC_1

3. Verify that all devices in the network have unique device names.

   Assign the device name to the iC7 drive before the connection can be established in *RUN* mode.

4. To start the configuration of the iC7 drive, select the *Device view*.

5. Select the correct application in the drive and add it to the slot application.

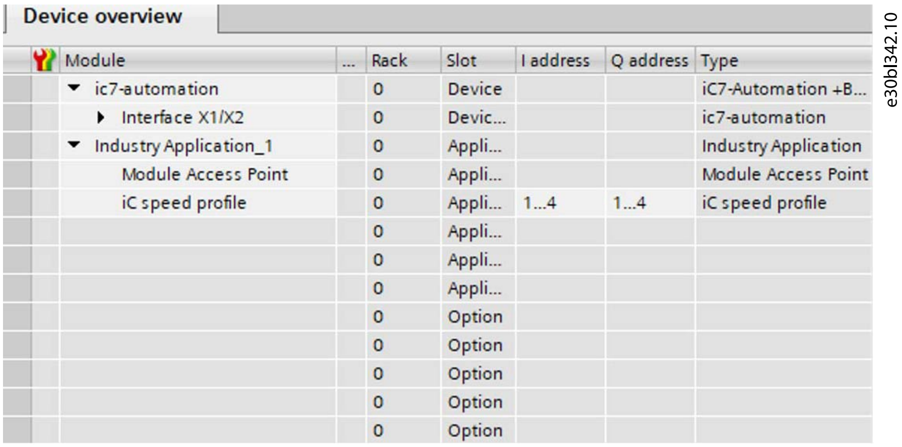
   The folder *Applications* show the option of applications available for the corresponding iC7 drive series.

6. Verify that the corresponding application is installed on the drive.

7. As default the iC profile is selected, the default profile can be changed to the PROFIdrive profile.

> **NOTICE**
>
> The support for the functions blocks is no longer available, if the default profile is changed to PROFIdrive profile.

The following example is given with the Industry application selected.



Device overview

Module	...	Rack	Slot	I address	Q address	Type
▾ ic7-automation		0	Device			iC7-Automation +B...
▸ Interface X1/X2		0	Devic...			ic7-automation
▾ Industry Application_1		0	Appli...			Industry Application
Module Access Point		0	Appli...			Module Access Point
iC speed profile		0	Appli...	1...4	1...4	iC speed profile
		0	Appli...			
		0	Appli...			
		0	Appli...			
		0	Option			
		0	Option			
		0	Option			
		0	Option			
		0	Option			

Figure 9: Device Overview

8. Ensure that the I/O address range is known by the function block support.

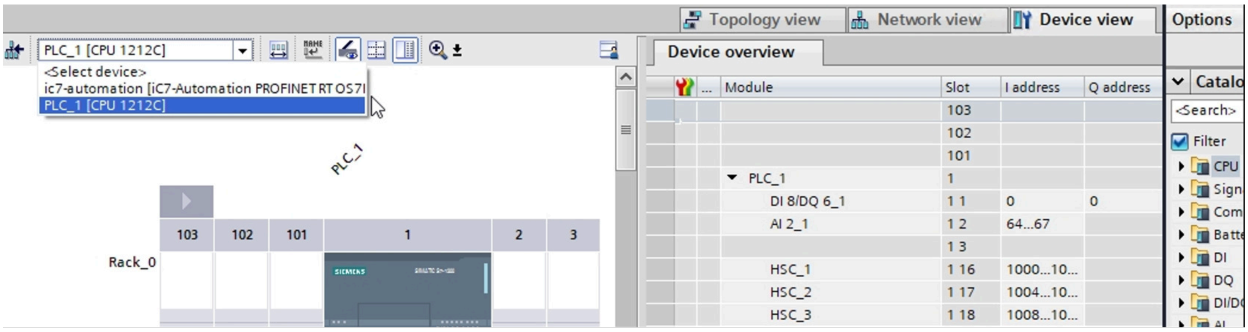
   The tool automatically set the I/O address range. The I/O address range can be changed, but must be unique in the entire project.

Keep the default settings of the I/O address range is recommended.

4.4 Optional Configuration Features

Each individual option can be selected from the folder *Options*, if option support installed in the drive is needed. The placement of the option in the drive depends on which slot the option is inserted in. The following example demonstrates the PROFINET usage together with the basic I/O option.

1. Select the *Device view tab* in the *Device configuration menu*.
2. Select the *iC7 drive* in the selection list.



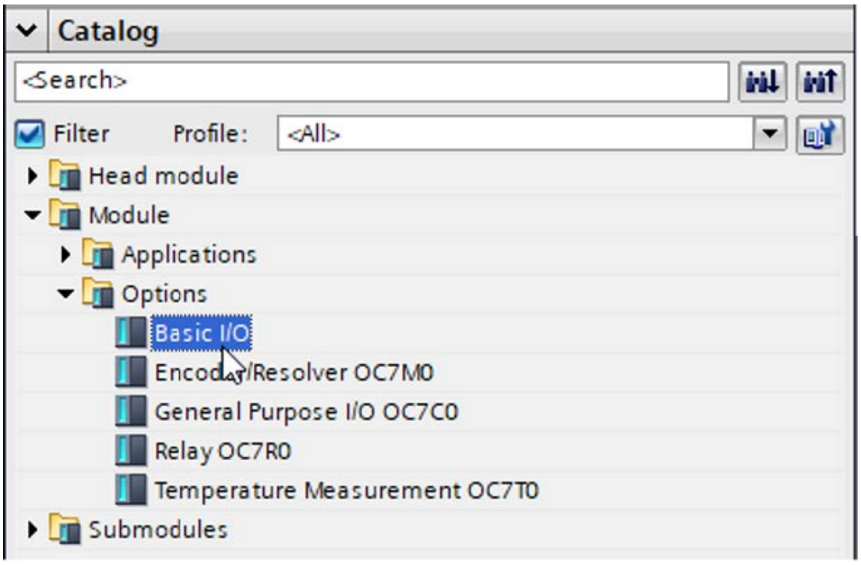
...	Module	Slot	I address	Q address
		103		
		102		
		101		
	▼ PLC_1	1		
	DI 8/DQ 6_1	1 1	0	0
	AI 2_1	1 2	64...67	
		1 3		
	HSC_1	1 16	1000...10...	
	HSC_2	1 17	1004...10...	
	HSC_3	1 18	1008...10...	

Figure 10: Device Selection List

3. Expand the module folder *Options* in the hardware catalog and select the option *Basic I/O*.

The selected module can be added to all 5 module slots. The exception is that the Basic I/O module can only be added to the 1st option slot, and only once. If more options are present, the slot representation of the option depends on the location in the drive. Always add additional options to the next available option slot in the module list.

NOTE: Refer to the *Operating Guide iC7 Series PROFINET RT OS7PR* for further information about the option placement and allocation in the iC7 drive.



Catalog

<Search>

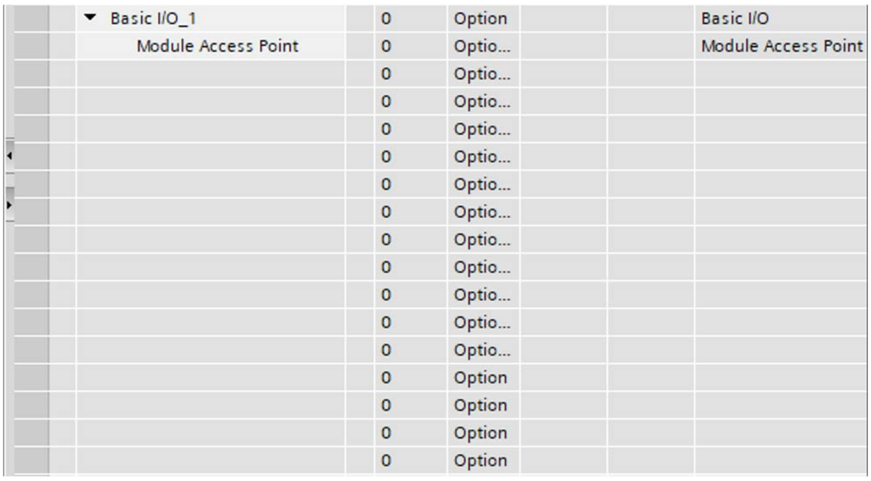
☑ Filter Profile: <All>

- Head module
- Module
  - Applications
  - Options
    - Basic I/O
    - Encoder/Resolver OC7M0
    - General Purpose I/O OC7C0
    - Relay OC7R0
    - Temperature Measurement OC7T0
- Submodules

Figure 11: Selecting Basic I/O in the Hardware Catalog

4. Add the basic I/O option to the 1st module option slot in the iC7 device.

> The basic I/O can only be present in the 1st option slot, also if additional options are preinstalled or added to the drive. Only the module and module access point is present for the basic I/O option.



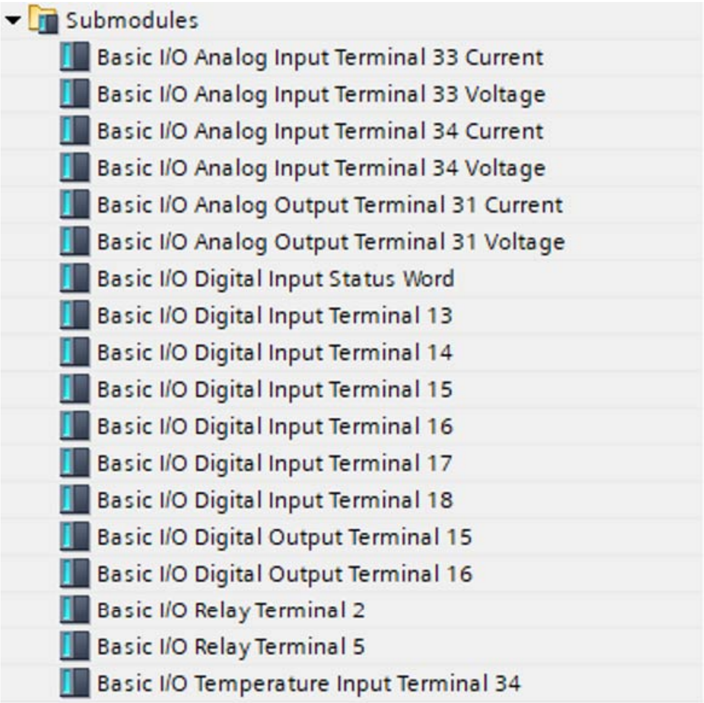
Basic I/O_1	0	Option	Basic I/O
Module Access Point	0	Optio...	Module Access Point
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Optio...	
	0	Option	
	0	Option	
	0	Option	
	0	Option	

Figure 12: Module Access Point

> ➡ The module and *Module Access Point* are now present for the basic I/O option.

5. To add I/O data to the basic I/O module, select the folder *Submodules* in the hardware catalog and select the corresponding feature from this list.

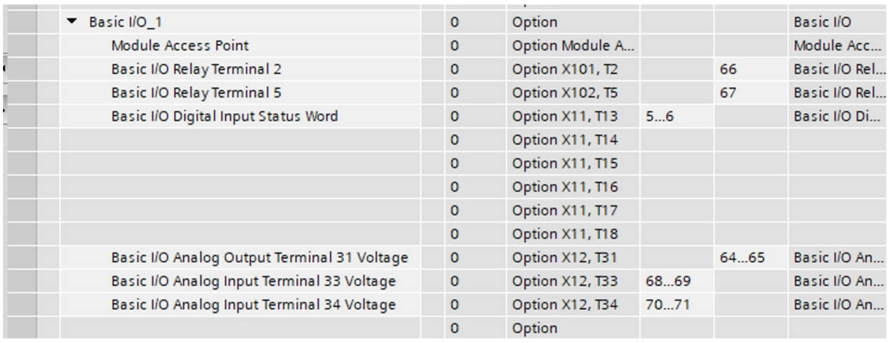
> Each function can only be placed in the dedicated sub-slot of the option module, based on the terminal for the function. Misplacing a function is not possible.



- Submodules
  - Basic I/O Analog Input Terminal 33 Current
  - Basic I/O Analog Input Terminal 33 Voltage
  - Basic I/O Analog Input Terminal 34 Current
  - Basic I/O Analog Input Terminal 34 Voltage
  - Basic I/O Analog Output Terminal 31 Current
  - Basic I/O Analog Output Terminal 31 Voltage
  - Basic I/O Digital Input Status Word
  - Basic I/O Digital Input Terminal 13
  - Basic I/O Digital Input Terminal 14
  - Basic I/O Digital Input Terminal 15
  - Basic I/O Digital Input Terminal 16
  - Basic I/O Digital Input Terminal 17
  - Basic I/O Digital Input Terminal 18
  - Basic I/O Digital Output Terminal 15
  - Basic I/O Digital Output Terminal 16
  - Basic I/O Relay Terminal 2
  - Basic I/O Relay Terminal 5
  - Basic I/O Temperature Input Terminal 34

Figure 13: Basic I/O Data to the Basic I/O Module

The following illustration shows an example of a basic I/O configuration.



▼ Basic I/O_1	0	Option			Basic I/O
Module Access Point	0	Option Module A...			Module Acc...
Basic I/O Relay Terminal 2	0	Option X101, T2		66	Basic I/O Rel...
Basic I/O Relay Terminal 5	0	Option X102, T5		67	Basic I/O Rel...
Basic I/O Digital Input Status Word	0	Option X11, T13	5...6		Basic I/O Di...
	0	Option X11, T14			
	0	Option X11, T15			
	0	Option X11, T16			
	0	Option X11, T17			
	0	Option X11, T18			
Basic I/O Analog Output Terminal 31 Voltage	0	Option X12, T31		64...65	Basic I/O An...
Basic I/O Analog Input Terminal 33 Voltage	0	Option X12, T33	68...69		Basic I/O An...
Basic I/O Analog Input Terminal 34 Voltage	0	Option X12, T34	70...71		Basic I/O An...
	0	Option			

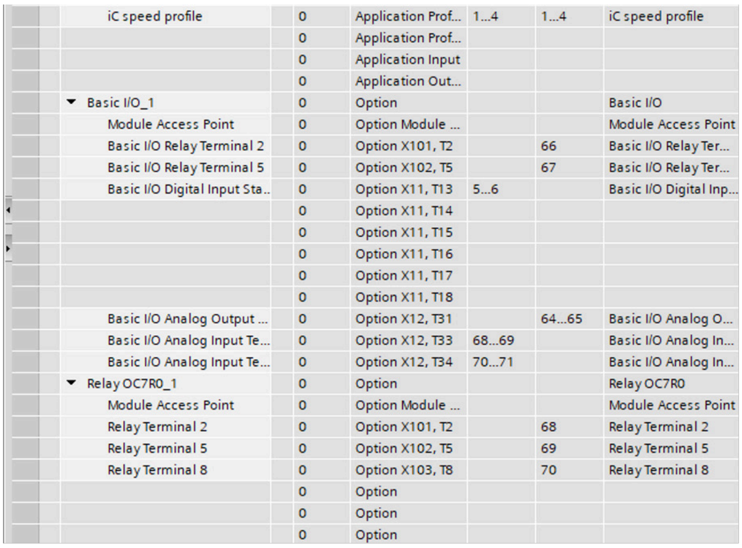
Figure 14: Example of Basic I/O Configuration

6. Check the I/O address ranges for the different functions.

   Ensure to get the information to and from the basic I/O option via the PROFINET interface.

7. If another option besides the basic I/O is added or installed in the drive, the basic I/O option is still present in the 1st option slot. The next installed option in the option slot is assigned to the next available option slot in the module list of the device.

   The principle to add submodules from the hardware catalog to the option sub-slots are the same as with the basic I/O option. In the following example, both basic I/O and relay option OC7R0 are installed in the drive.



iC speed profile	0	Application Prof...	1...4	1...4	iC speed profile
	0	Application Prof...			
	0	Application Input			
	0	Application Out...			
Basic I/O_1	0	Option			Basic I/O
Module Access Point	0	Option Module ...			Module Access Point
Basic I/O Relay Terminal 2	0	Option X101, T2		66	Basic I/O Relay Ter...
Basic I/O Relay Terminal 5	0	Option X102, T5		67	Basic I/O Relay Ter...
Basic I/O Digital Input Sta..	0	Option X11, T13	5...6		Basic I/O Digital Inp...
	0	Option X11, T14			
	0	Option X11, T15			
	0	Option X11, T16			
	0	Option X11, T17			
	0	Option X11, T18			
Basic I/O Analog Output ...	0	Option X12, T31		64...65	Basic I/O Analog O...
Basic I/O Analog Input Te...	0	Option X12, T33	68...69		Basic I/O Analog In...
Basic I/O Analog Input Te...	0	Option X12, T34	70...71		Basic I/O Analog In...
Relay OC7R0_1	0	Option			Relay OC7R0
Module Access Point	0	Option Module ...			Module Access Point
Relay Terminal 2	0	Option X101, T2		68	Relay Terminal 2
Relay Terminal 5	0	Option X102, T5		69	Relay Terminal 5
Relay Terminal 8	0	Option X103, T8		70	Relay Terminal 8
	0	Option			
	0	Option			
	0	Option			

Figure 15: Example of Basic I/O and Relay Option Installed in the Drive

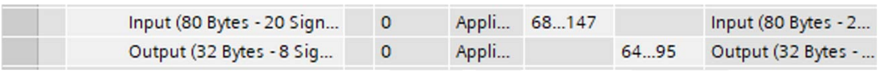
NOTE: If the basic I/O option not is installed in the drive, but only additional options are installed in the drive, the 1st option must be added to the 1st available option slot in the device module configuration. For more information about the option support, placement in the drive, and module slot number. Refer to the *Operating Guide iC7 Series PROFINET RT OS7PR* .

4.4.1 Adding Additional Process Data

If needed additional I/O signals can be added to the drive configuration.

For the purpose, the folder *Submodules* is available. It is possible to add process data to the configuration and freely select between the different input and output data in this folder. Up to 20 signals of 4 bytes can be selected.

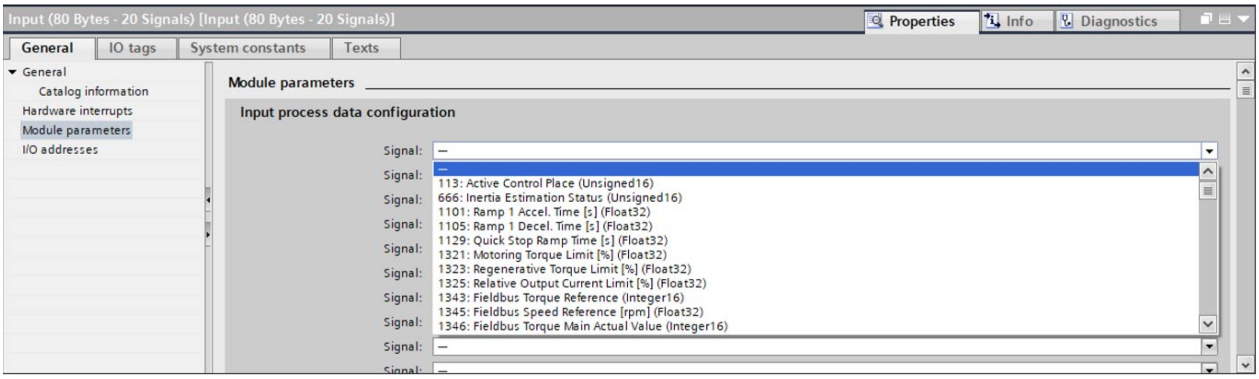
In the following example, 20 signals are selected as input data and 8 signals are selected as output data.



Input (80 Bytes - 20 Sign...	0	Appli...	68...147		Input (80 Bytes - 2...
Output (32 Bytes - 8 Sig...	0	Appli...		64...95	Output (32 Bytes - ...

Figure 16: Selection of Input/Output Data

The data for the I/O signals are configured in the *Properties* tab of each submodule, where the *Module Parameters* are available. The following example shows the drop-down list available for the 1st input signal where the individual process data can be selected.



Input (80 Bytes - 20 Signals) [Input (80 Bytes - 20 Signals)]

Module parameters

Input process data configuration

Signal: –

Signal: –

113: Active Control Place (Unsigned16)
666: Inertia Estimation Status (Unsigned16)
1101: Ramp 1 Accel. Time [s] (Float32)
1105: Ramp 1 Decel. Time [s] (Float32)
1129: Quick Stop Ramp Time [s] (Float32)
1321: Motoring Torque Limit [%] (Float32)
1323: Regenerative Torque Limit [%] (Float32)
1325: Relative Output Current Limit [%] (Float32)
1343: Fieldbus Torque Reference (Integer16)
1345: Fieldbus Speed Reference [rpm] (Float32)
1346: Fieldbus Torque Main Actual Value (Integer16)

Signal: –

e30bl350.10

Figure 17: Individual Process Data

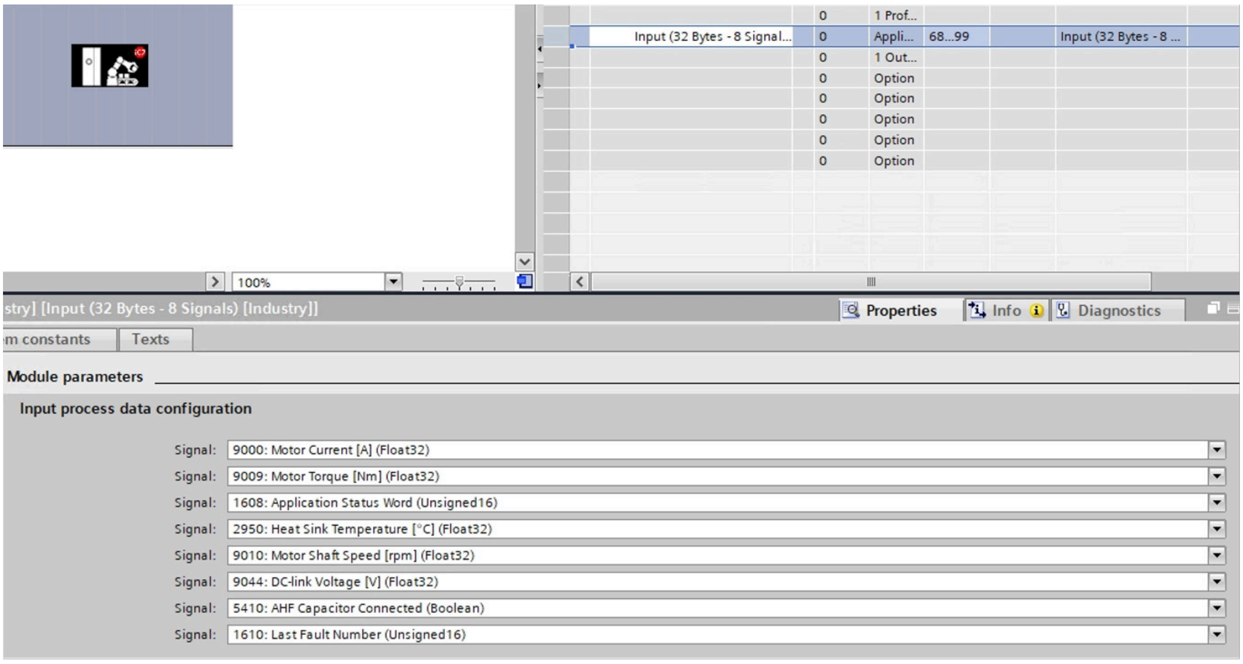
The number in front of the signal name is the corresponding parameter number in the drive.

The procedure is the same for the remaining input signals, where the same drop-down list is available. It is the same procedure for the output signals, but the list of signals are different.

The buffer size adapts to the data type of the selected signals. Exception is when a Boolean data type is mapped, only bit 0 is used in the selected signal address, and the remaining 7 bits are not used. The corresponding data type must match what is configured for each signal to match the corresponding process data values.

NOTE: For example, if selecting the process data *1101: Ramp 1 Accel. Time [s] (Float 32)* as the 1st input signal, then the data type must be selected as *REAL*, to get the corresponding value into the PROFINET controller.

The following illustration is an example of a configuration with 8 input signals. 32 bytes (8 x 4 bytes) are reserved in the receive buffer, but only 25 bytes are used.



	0	1 Prof...		
Input (32 Bytes - 8 Signal...	0	Appli...	68...99	Input (32 Bytes - 8 ...
	0	1 Out...		
	0	Option		
	0	Option		
	0	Option		
	0	Option		
	0	Option		

stry] [Input (32 Bytes - 8 Signals) [Industry]]

Module parameters

Input process data configuration

Signal: 9000: Motor Current [A] (Float32)
Signal: 9009: Motor Torque [Nm] (Float32)
Signal: 1608: Application Status Word (Unsigned16)
Signal: 2950: Heat Sink Temperature [°C] (Float32)
Signal: 9010: Motor Shaft Speed [rpm] (Float32)
Signal: 9044: DC-link Voltage [V] (Float32)
Signal: 5410: AHF Capacitor Connected (Boolean)
Signal: 1610: Last Fault Number (Unsigned16)

e30bl639.10

Figure 18: Example of Input Process Data Configuration (8 Input Signals)

NOTE: Output data is only accepted by the drive from 1 reference source. If multiple sources want to control the drive input data, the 1st connected source gets the access rights. All other sources get ignored and an occurrence appears in the drive.

NOTE: Reference sources are:

- Analog inputs.
- Digital inputs either as pulse input or digital potentiometer.
- Reference from a fieldbus.
- Up to 8 preset references (selectable by parameter, fieldbus, or digital inputs).
- Local reference from control panel.

NOTE: For more information on how process data is handled. Refer to the *Operating Guide iC7 Series PROFINET RT OS7PR* .

5 Get Started with Danfoss Function Blocks

5.1 Installing the iC7 Library

When the iC7 drive is fully configured in the *Devices & Networks view*, the next step is to utilize the function blocks for the iC7 drive.

Install the library for the iC7 drive support from the link https://www.danfoss.com/en/service-and-support/downloads/dds/plc-libraries.

NOTE: The library is created in TIA Portal V17, and is supported from V17 and forward compatible versions only.

1. Press the button *Open global library* in the *Global libraries* section.

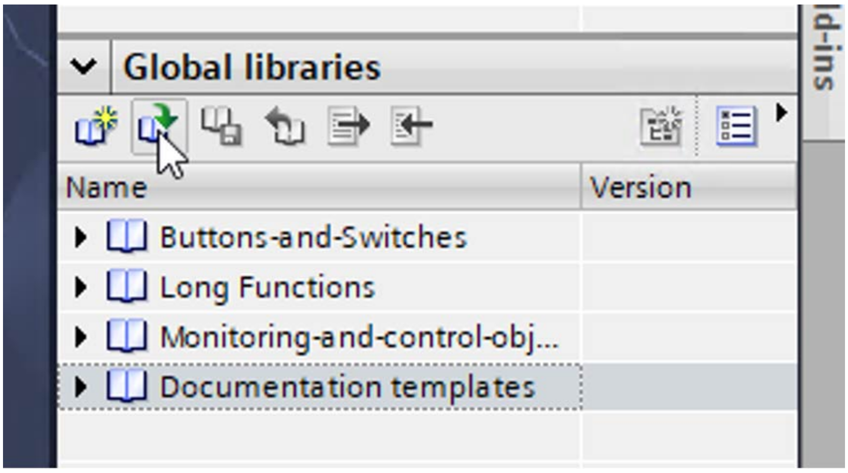


Figure 19: Global Libraries

2. Select the folder where the library is stored and change the *Files of types* to *Compressed libraries* to get the file visible.

The following example shows the pop-up window.

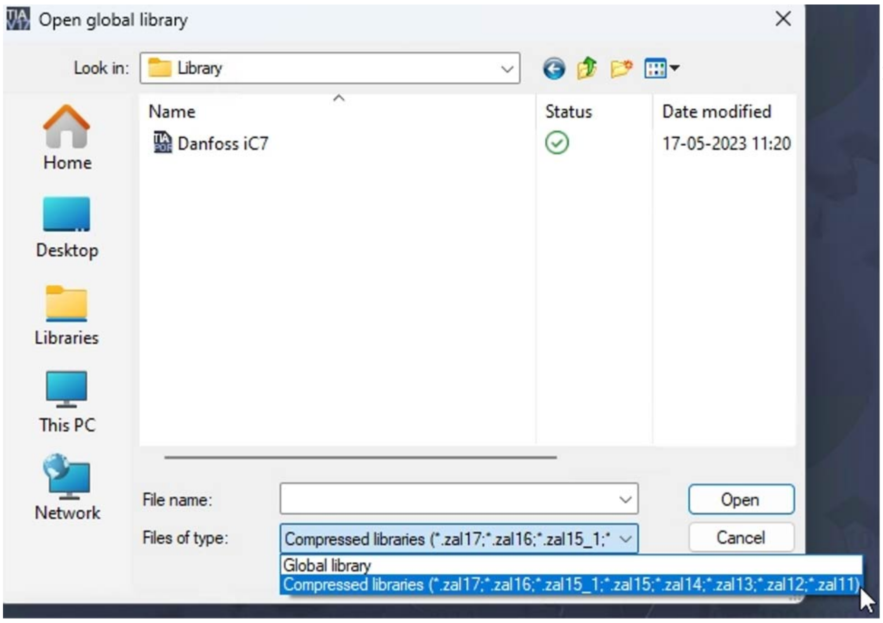


Figure 20: Change Files of Type to Compressed Library

3. Select compressed library file and press the *Open* button.
4. To store the extracted library, select a folder and press the *Select folder* button to get the library extracted.

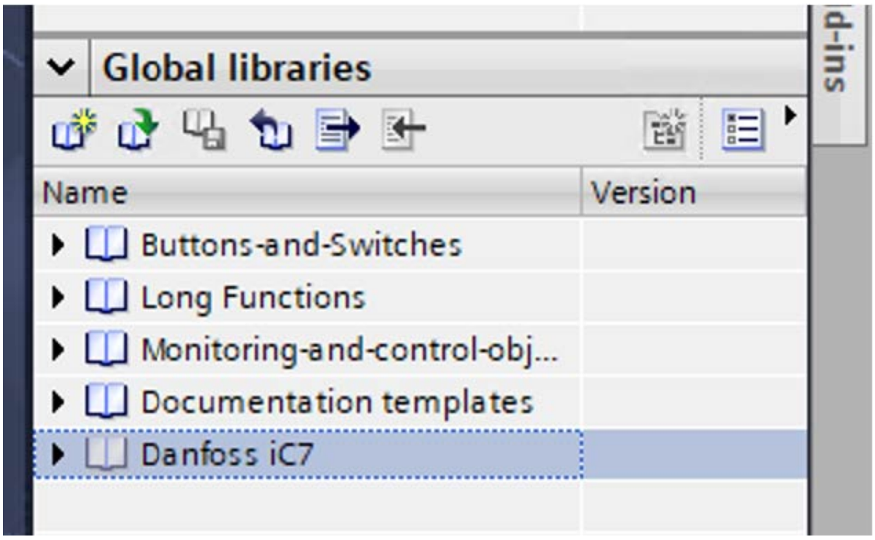


Figure 21: Library Visible in the Global Library

➡ If the extraction of the library is successful, the library becomes visible in the *Global libraries* section.

5. Expand the library and select the folder *Types* to get access to the different function blocks and related artifacts for the iC7 drive.

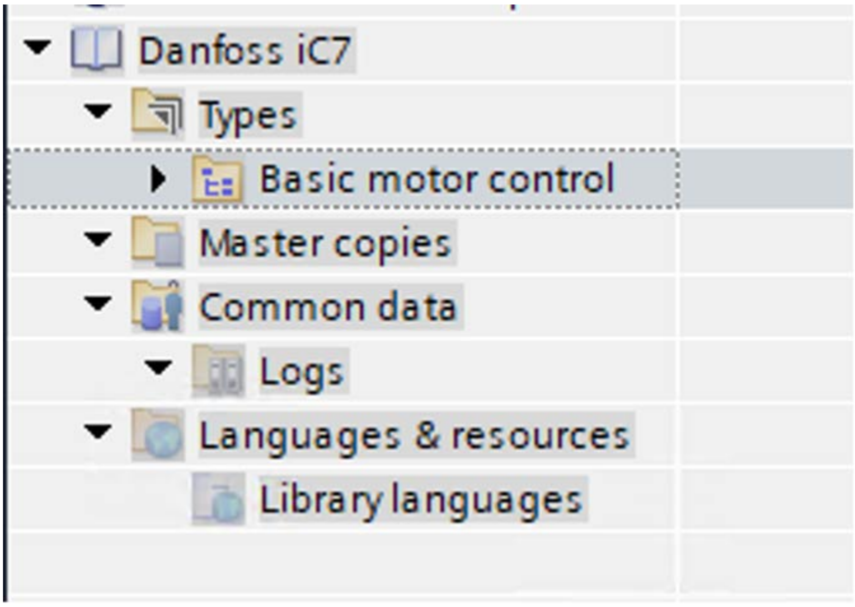


Figure 22: Access to the Different Function Blocks

6. Expand the *Basic motor control* folder to get access to the function block and related artifacts.

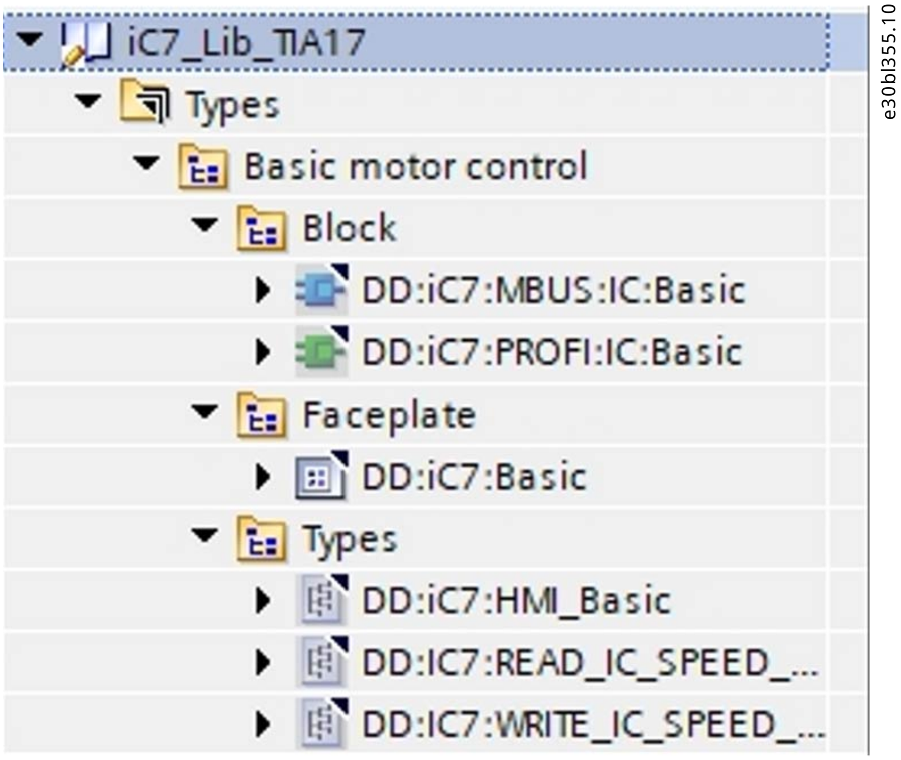


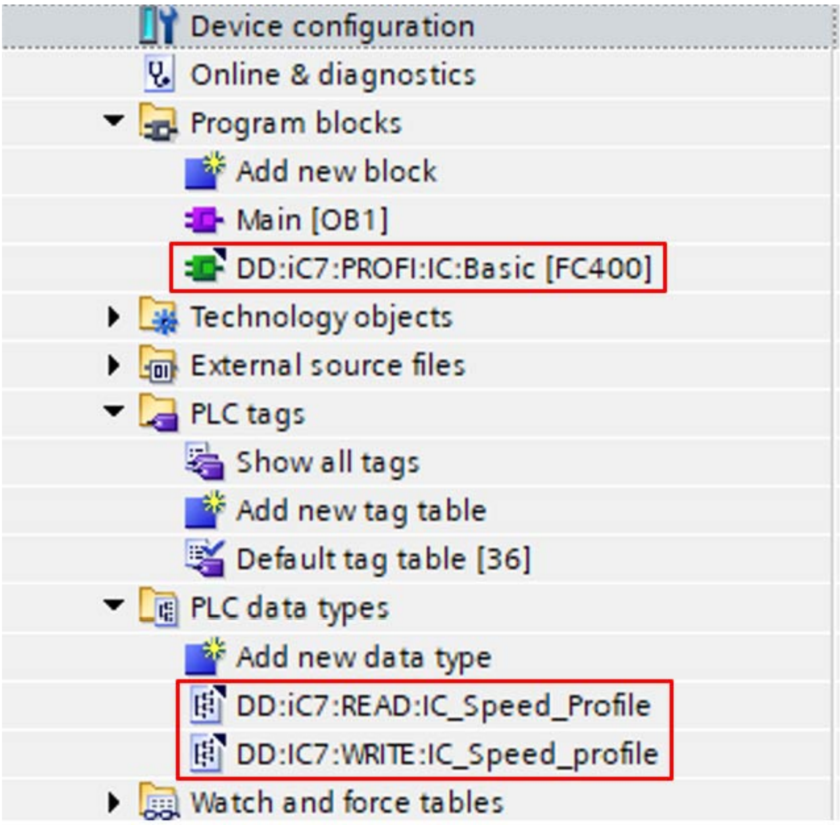
Figure 23: Basic Motor Control

5.2 Configuration of the Basic Drive Control and Monitoring Function Block

1. Copy/paste or drag/drop the function block *DD:iC7:PROFI:IC:Basic* to the folder *Program blocks* under the controller, which must control the iC7 drive.
2. Copy/paste or drag/drop the 2 data types (UDT) *DD:iC7:READ:IC_Speed_profile* and *DD:iC7:WRITE:IC_Speed_profile* to the folder *PLC data types*.

When the function and 2 data types are copied, the function is visible as *FC400*. The function is visible under the folder *Program blocks* and the 2 data types are visible in the folder *PLC data types*. The following illustration shows the view when the 2 data types are copied to the program blocks and PLC data types folder.

NOTE: The user-defined data types are normally copied together with the function block. The UDT in the library folder *Types* is automatically added to the folder *PLC data types*, when the function block is copied.



Device configuration

Online & diagnostics

Program blocks

Add new block

Main [OB1]

DD:iC7:PROFI:IC:Basic [FC400]

Technology objects

External source files

PLC tags

Show all tags

Add new tag table

Default tag table [36]

PLC data types

Add new data type

DD:iC7:READ:IC_Speed_Profile

DD:IC7:WRITE:IC_Speed_profile

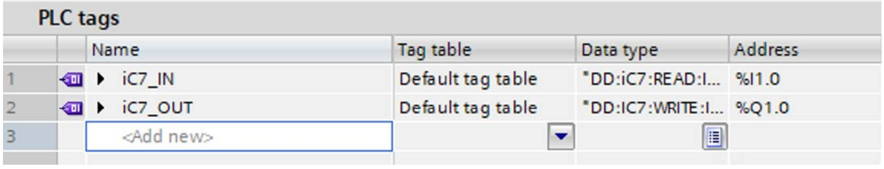
Watch and force tables

Figure 24: Add Files in the Program Blocks and PLC Data Types Folders

3. Create I/O tags for the process data to/from the drive. Use the user-defined PLC data types as illustrated in Figure 24.
   - a. Create a tag that corresponds to the input address from the drive.

     The I/O address is given from the *Device view* in the window *Device configuration*, refer to 4.3 Adding the iC7 Drive to the TIA Project for more information.
   - b. Create a similar tag for the output address to the drive.

     The following illustration shows an example of the 2 created input/output tags.



PLC tags

		Name	Tag table	Data type	Address
1		iC7_IN	Default tag table	"DD:iC7:READ:I...	%I1.0
2		iC7_OUT	Default tag table	"DD:IC7:WRITE:I...	%Q1.0
3		<Add new>			

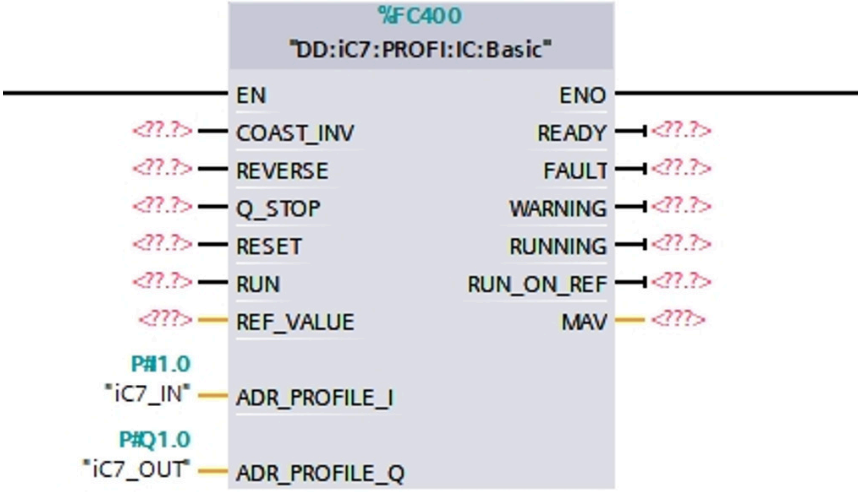
e30bl357.10

Figure 25: Example of Created Tags for Input/Output Address

4. Add the function *FC400* to 1 of the program blocks created or directly in an organization block.
   a. Assign the *ADR_PROFILE_I* pin to the input (read) tag.
   b. Assign the *ADR_PROFILE_Q* pin to the output (write) tag.

   > Refer to step 3 for creating the input (read) and output (write) tags.

> The following illustration shows an example of the assigned input/output pins.



e30bl358.10

Figure 26: Adding Function FC400 to Program/Organization Block

5. Define the remaining I/O pins, reference value *REF_VALUE*, and main actual value *MAV*, digital I/O values, flags, and so forth, to control the motor with the drive. Refer to table Table 2.

Table 2: Definition of I/O Data

Input/Output	I/O data definition	Data type	Description
Input	EN	N/A	Enabling of the block for data execution.
	COAST_INV	Bool	Coast stop inverse. This will coast the motor if disabled.
	REVERSE	Bool	Reverse the motor direction. Enabled, the motor starts spinning in the reverse direction.
	Q_STOP	Bool	Quick stop. Perform a quick stop ramp when enabled. **NOTE:** This function is depending on index ***5.7.3.x Quick stop settings***.
	RESET	Bool	Reset an occurrence. When enabled a present fault can be reset, if the conditions for the reset of the fault/faults are met.
	RUN	Bool	Motor start/stop. Enabled, the motor will start/run. Disabled, the motor performs a ramp stop. **NOTE:** This function depends on index ***5.8.6.2.x Ramp 1 settings***[1].
	REF_VALUE	REAL	Added value gives the reference value in percentage based on N2 definition (-200% - +200%). **NOTE:** This function depends on the index ***5.8.3.x Speed limits***[1].

Table 2: Definition of I/O Data (continued)

Input/Output	I/O data definition	Data type	Description
Output	ENO	N/A	Mirror the EN function.
	READY	Bool	Enabled, the motor is ready to start and run. Disabled, the motor cannot start, since the raised conditions for a start command are not fulfilled. **NOTE:** Refer to section 9.1 Troubleshooting with the Basic Drive Function Block for clarification on root-cause finding for missing start condition.
	FAULT	Bool	Enabled, a fault/alarm is present, and the motor is stopped. Disabled, the drive is fully operational.
	WARNING	Bool	Enabled, a warning is present, but it is still possible to control the motor, maybe with limited performance. Disabled, no warning is present and the motor is fully operational.
	RUNNING	Bool	The motor is started and is heading or performing the requested reference value.
	RUN_ON_REF	Bool	The main actual value matches the requested reference value, which means that the motor is now running at the requested reference value.
	MAV	REAL	The estimated rotor speed of the motor is controlled in open loop conditions. **NOTE:** This value depends on index ***4.3.4.x settings***[1].

[1] The index number and name depends on the current application installed in the iC7drive. The number in the table is valid for iC7-Automation with Industry Application installed.

The following illustration shows an example of the function block when all I/O pins are configured.

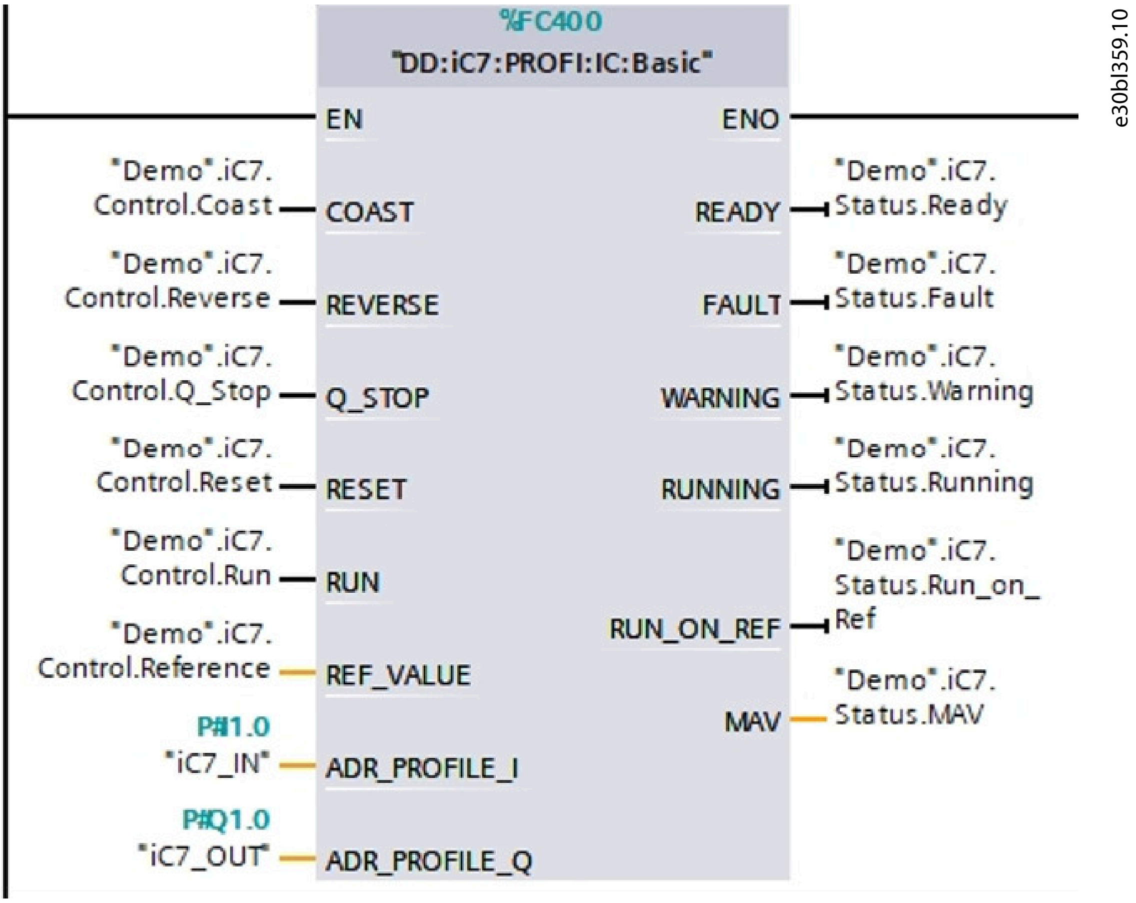


Figure 27: Example of the Function Block with Configured I/O Pins

6. Test the function block by creating a variable table with the requested I/O information.

5.3 Function Block for use with Modbus TCP Communication

The procedure is the same as for the PROFINET function block. Copy the Modbus TCP function block from the library and paste it into the *Program blocks* folder as shown in the following illustration.

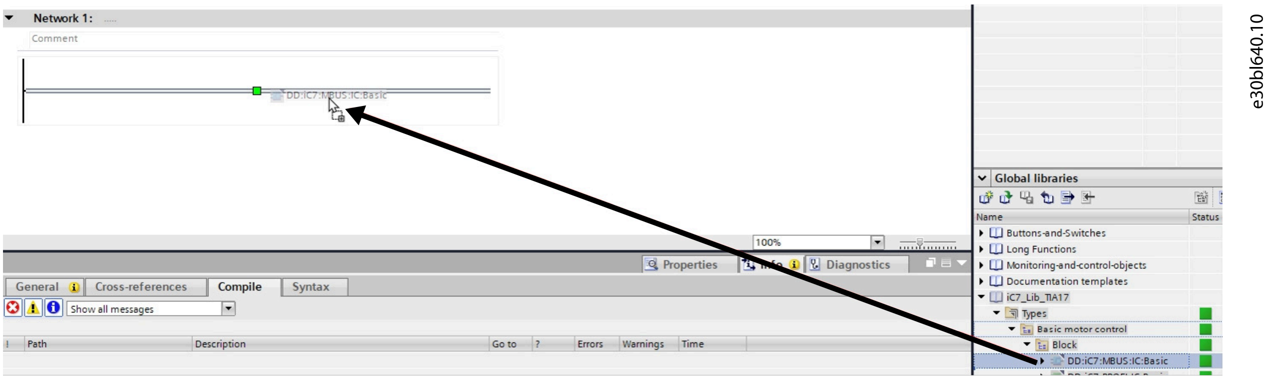
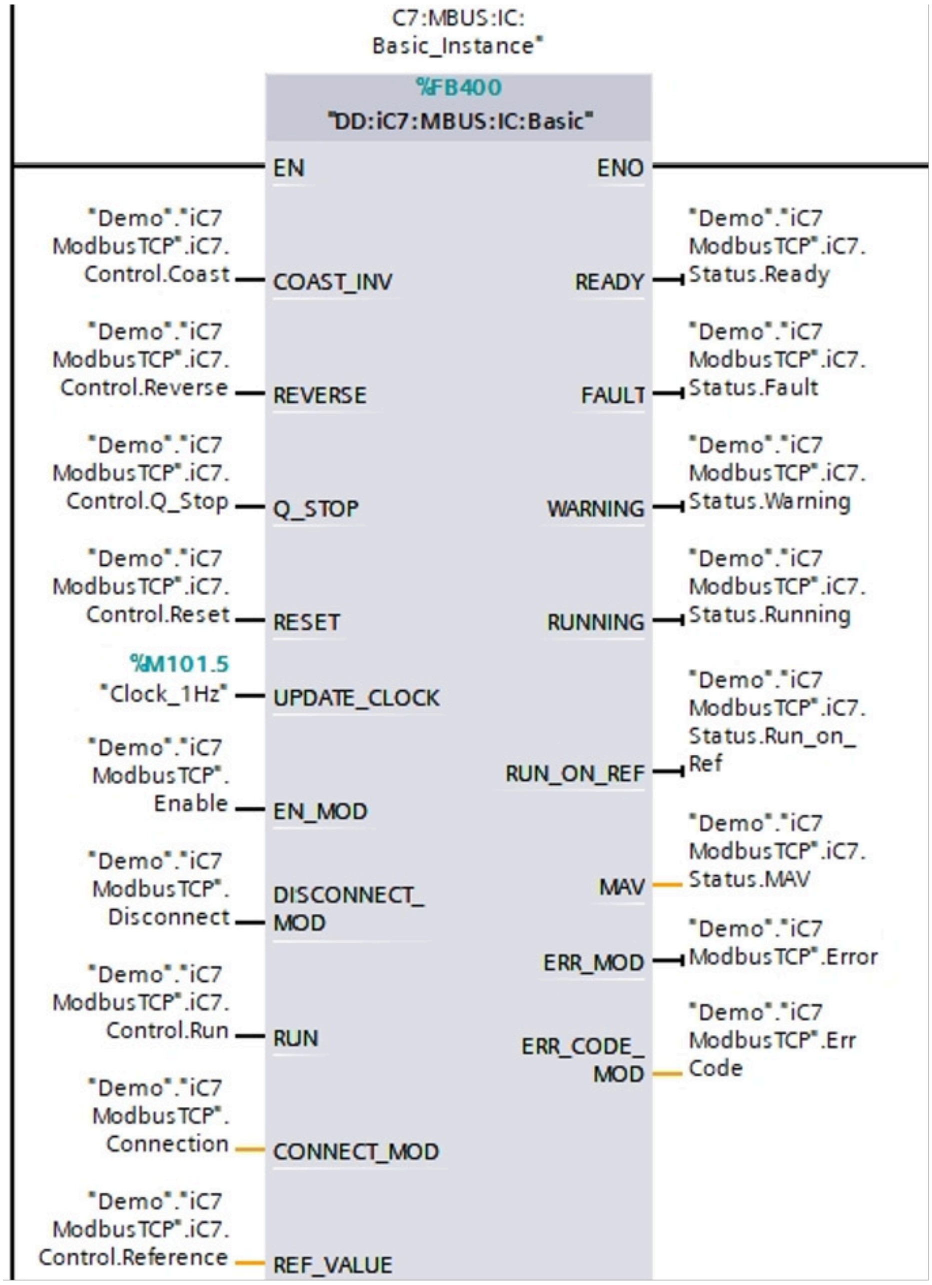


Figure 28: Modbus TCP Function Block Added to the Program Blocks Folder

The Modbus TCP interface protocol is available in the iC7 drive, if the corresponding license is installed in the drive. If multiple protocol licenses are installed in the drive, and Modbus TCP is not the default communication protocol, then the Modbus TCP protocol must be selected accordingly. Use the MyDrive® Insight PC tool.

All Simatic S7-1200 and Simatec S7-1500 series controllers supporting the Modbus TCP protocol. Beside the control and monitoring commands, which are similar as for the function block for usage with PROFINET RT communication, additional pins and configuration data are also needed.

The following illustration is an example of the function block when all I/O pins are configured. Refer also to table Table 3.


C7:MBUS:IC:
Basic_Instance"
%FB400
"DD:iC7:MBUS:IC:Basic"
EN
ENO
"Demo"."iC7 ModbusTCP".iC7.Control.Coast
COAST_INV
READY
"Demo"."iC7 ModbusTCP".iC7.Status.Ready
"Demo"."iC7 ModbusTCP".iC7.Control.Reverse
REVERSE
FAULT
"Demo"."iC7 ModbusTCP".iC7.Status.Fault
"Demo"."iC7 ModbusTCP".iC7.Control.Q_Stop
Q_STOP
WARNING
"Demo"."iC7 ModbusTCP".iC7.Status.Warning
"Demo"."iC7 ModbusTCP".iC7.Control.Reset
RESET
RUNNING
"Demo"."iC7 ModbusTCP".iC7.Status.Running
%M101.5
"Clock_1Hz"
UPDATE_CLOCK
"Demo"."iC7 ModbusTCP".iC7.Status.Run_on_Ref
RUN_ON_REF
"Demo"."iC7 ModbusTCP".Enable
EN_MOD
"Demo"."iC7 ModbusTCP".iC7.Status.MAV
MAV
"Demo"."iC7 ModbusTCP".Disconnect
DISCONNECT_MOD
ERR_MOD
"Demo"."iC7 ModbusTCP".Error
"Demo"."iC7 ModbusTCP".iC7.Control.Run
RUN
ERR_CODE_MOD
"Demo"."iC7 ModbusTCP".Err Code
"Demo"."iC7 ModbusTCP".Connection
CONNECT_MOD
"Demo"."iC7 ModbusTCP".iC7.Control.Reference
REF_VALUE

e30bl408.10

Figure 29: Example of Function Block with Configured I/O Pins

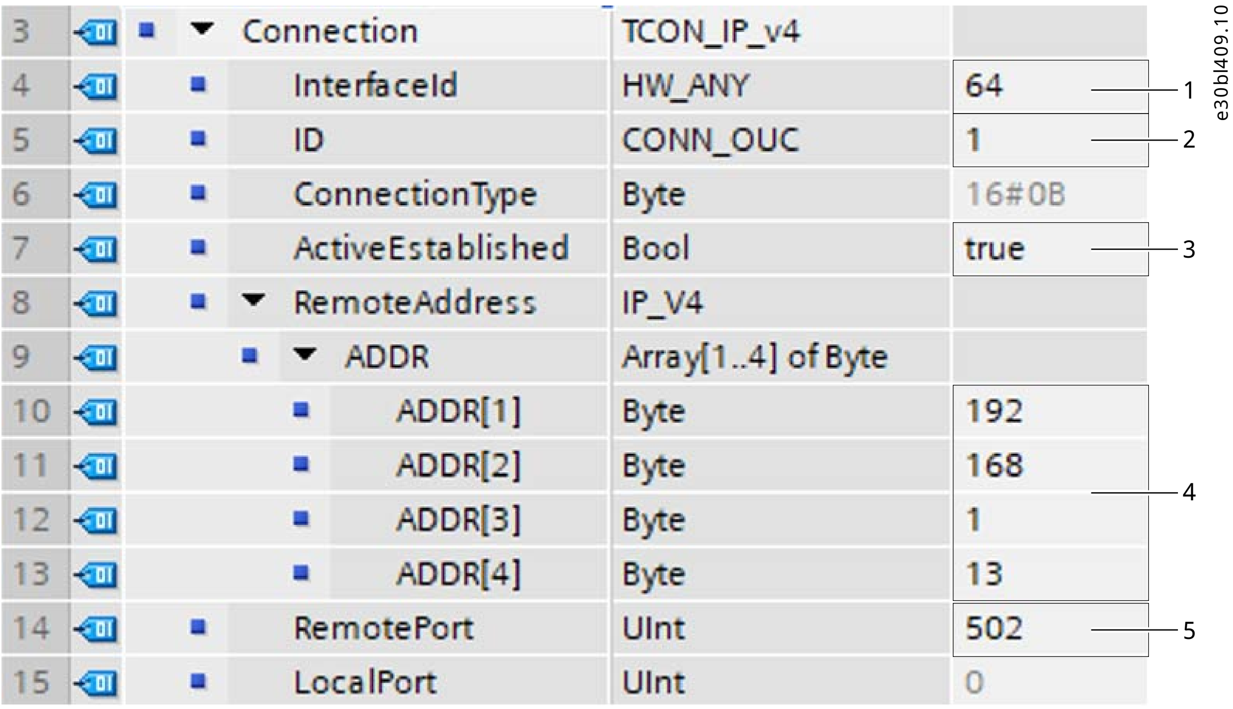
Table 3: Definition of I/O Data

Input/Output	I/O data definition	Data type	Description
Input	UPDATE_CLOCK	Bool	As the Modbus protocol is a request/response protocol, a request to update the output data must be initiated. A clock update from the controller memory is needed here to perform background updates of the output variables. Besides the update on each clock ticks, the output is also updated every time the input data is changed.
	EN_MOD	Bool	Enable the Modbus TCP client communication handler in the Simatic controller, and a connection to the iC7 drive is established when it is set to true.
	DISCONNECT_MOD	Bool	This pin is linked directly to the Modbus TCP client communication handler in the Simatic controller. It is possible to connect (false) or disconnect (true) the establishment of the communication handler to the Modbus server.
	CONNECT_MOD	TCON_IP_v4	Pointer to the Modbus server connection of the iC7 drive. Further information of the structure of this data type is available from Siemens, for example in the TIA Portal.
Output	ERR_MOD	Bool	An error occurred during the Modbus TCP communication.
	ERR_CODE_MOD	Word	Fault code from the Modbus TCP client function block of the Simatic controller. Further information of the individual fault codes is available from Siemens, for example in the TIA Portal.

NOTE: No hardware configuration in TIA Portal is needed to operate the iC7 drive with Modbus TCP. Refer to the *Operating Guide iC7 Series Modbus TCP OS7MT* for further information of the Modbus TCP interface of the iC7 drive.

The following is an example of the structure of the CONNECT_MOD pin defined as the data type TCON_IP_v4.

NOTE: The type TCON_IP_v4 structure is a standard component in Siemens TIA Portal, for example Modbus TCP communication. Refer to the *Siemens TIA Portal documentation* for more information on the TCON_IP_v4 structure and definitions.



3	Connection	TCON_IP_v4	
4	InterfaceId	HW_ANY	64
5	ID	CONN_OUC	1
6	ConnectionType	Byte	16#0B
7	ActiveEstablished	Bool	true
8	RemoteAddress	IP_V4	
9	ADDR	Array[1..4] of Byte	
10	ADDR[1]	Byte	192
11	ADDR[2]	Byte	168
12	ADDR[3]	Byte	1
13	ADDR[4]	Byte	13
14	RemotePort	UInt	502
15	LocalPort	UInt	0

Figure 30: Example of the CONNECT_MOD Pin Structure

1	64: Unique HW ID of the industrial Ethernet (PROFINET/ Modbus TCP) interface of the controller (master).	2	1: Unique ID of the Modbus TCP connection within the client.
3	True: Set to *true* for active connection establishment.	4	192;168;1;13: IP address of the Modbus TCP server (iC7 drive X1/X2 interface).
5	502: Port number for Modbus TCP server (iC7 drive) communication. The default value is 502.		

NOTE: The unique HW ID is accessible from the *System constants* tab of the default tag table.

NOTE: The unique ID uniquely identifies a connection within the Simatic S7 CPU. Each individual instance of the instruction *MB_CLINET*, which is used for Modbus TCP communication to the drive, must use a unique ID. If client connections must be established to multiple drives or servers, then this ID must be unique for each instance used.

A sample project is available, based on a S7-1200 CPU with iC7 drives operating in iC Speed profile with Industry application installed in the iC7 drive. The TIA17 project file name is *iC7_SampleProject.zap.*

6 Faceplate for HMI Application

6.1 Faceplate for iC7 Drive Control and Monitoring

Before you begin:

- Establish an active HMI connection between the HMI device and the controller (PLC).
- Ensure the PLC data type DD:iC7:HMI_Basic is available in the project.

> **NOTE:** The UDT is placed in the iC7 library and can be copied to the active project.

A faceplate for HMI support of the 2 function blocks for iC7 drive control and monitoring is also available in the library. To add the faceplate to the HMI application, use the following example.

1. Add a new HMI device to the TIA project.

   In the example, a PC with WinCC RT Advanced is used. HMI panels supported by WinCC can also be used.

2. Drag and drop the faceplate of the iC7 library to the HMI screen.

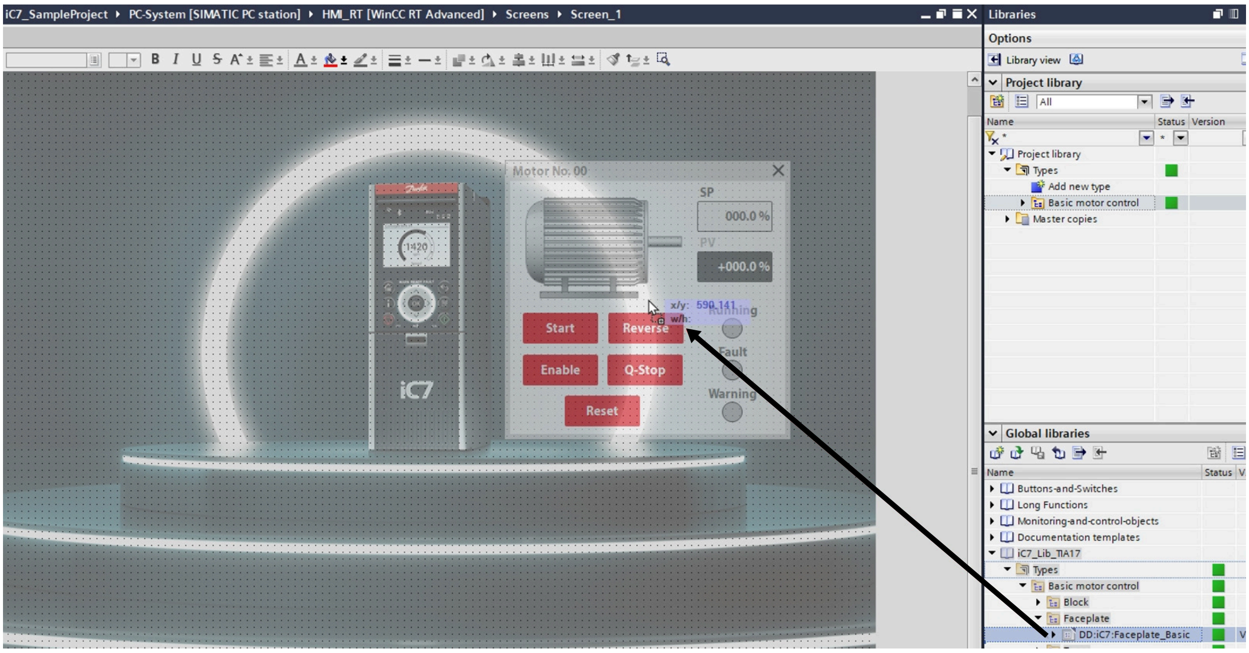
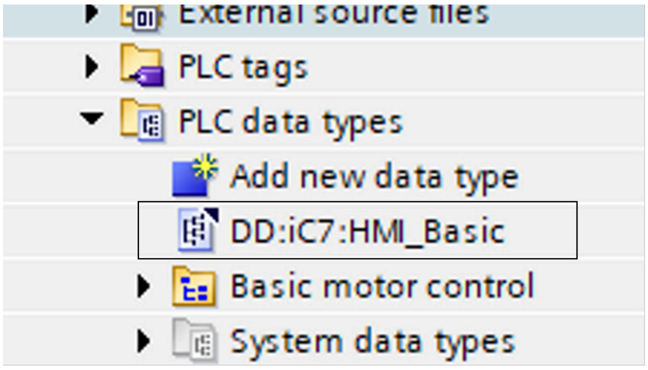


Figure 31: Adding the Faceplate from the iC7 Library to the HMI Screen

3. The PLC data type DD:iC7:HMI_Basic must be available in the project. The UDT is placed in the iC7 library and can be copied to the active project.

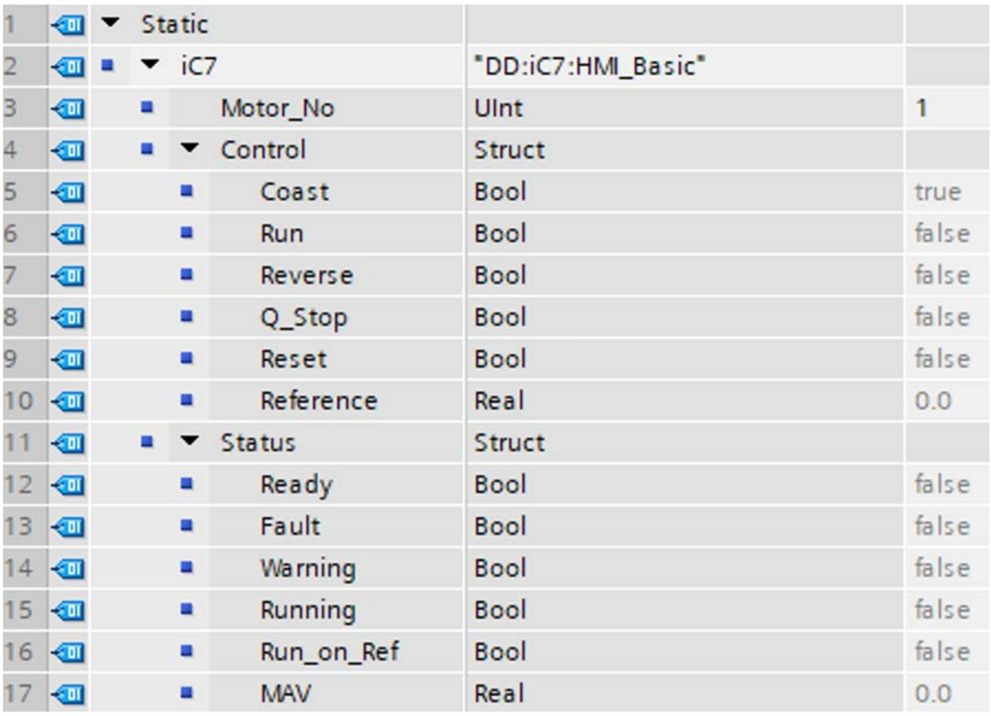
   Use the copy/paste or drag/drop function to add the UDT from the library to the active project.



External source files
- PLC tags
- PLC data types
  - Add new data type
  - DD:iC7:HMI_Basic
  - Basic motor control
  - System data types

Figure 32: PLC Data Types

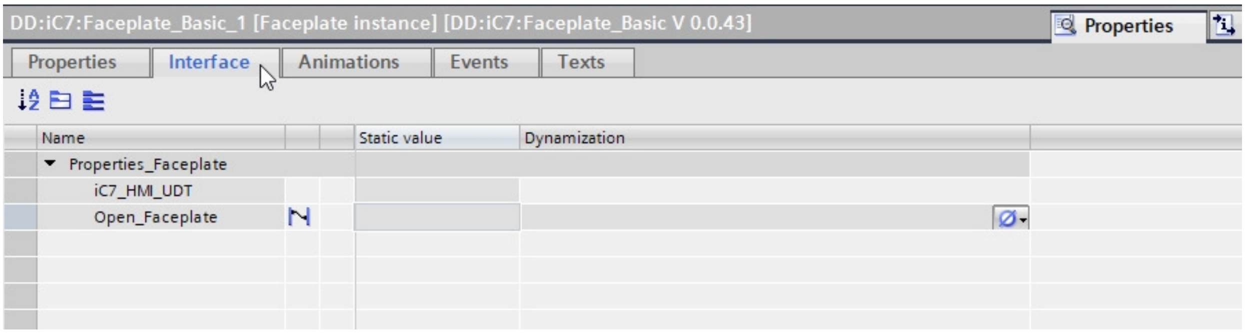
4. To ensure that the HMI can access the data in the controller, declare the UDT for the HMI part in, for example, a global DB.



1	Static		
2	iC7	"DD:iC7:HMI_Basic"	
3	Motor_No	UInt	1
4	Control	Struct	
5	Coast	Bool	true
6	Run	Bool	false
7	Reverse	Bool	false
8	Q_Stop	Bool	false
9	Reset	Bool	false
10	Reference	Real	0.0
11	Status	Struct	
12	Ready	Bool	false
13	Fault	Bool	false
14	Warning	Bool	false
15	Running	Bool	false
16	Run_on_Ref	Bool	false
17	MAV	Real	0.0

Figure 33: Example Declare the UDT for the HMI part in Global DB

5. Select the tab *Interface* in the properties of the faceplate.



DD:iC7:Faceplate_Basic_1 [Faceplate instance] [DD:iC7:Faceplate_Basic V 0.0.43]

Name	Static value	Dynamization
Properties_Faceplate		
iC7_HMI_UDT		
Open_Faceplate		

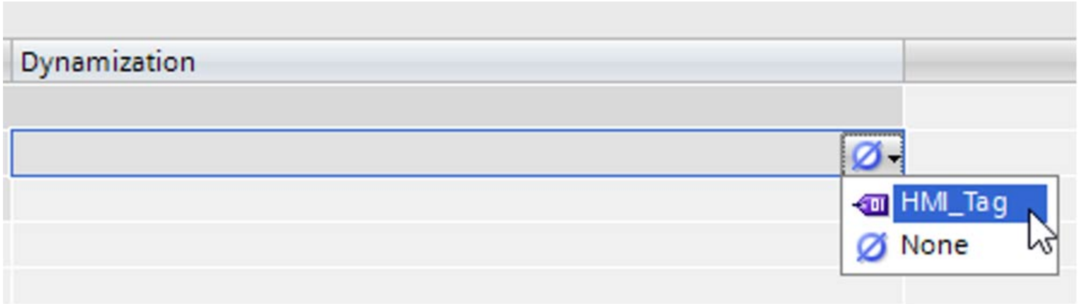
Figure 34: Faceplate Properties

6. Declare 2 HMI tags that match the properties of the faceplate. Define the tag for *iC7_HMI_UDT* in combination with the UDT defined for the faceplate. Define the tag for the *Open_Faceplate* as an internal HMI tag only.

   - iC7 HMI UDT
   - Open Faceplate

7. Add the declared data from the global DB to the new tag in *iC7_HMI_UDT* entry field of the faceplate.

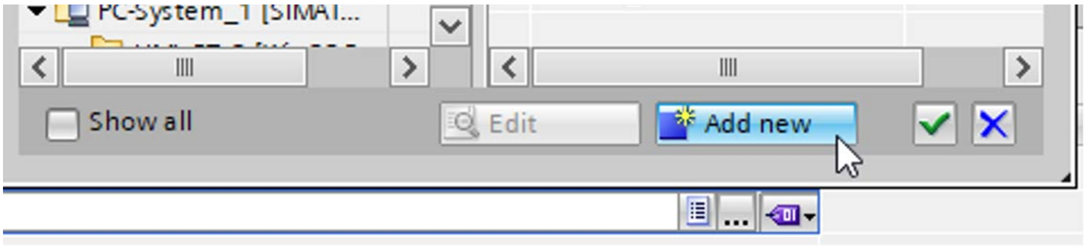
   In the drop-down field, select *HMI_Tag*.



e30bl415.10

Figure 35: Add the Declared Data to New Tag iC7_HMI_UDT

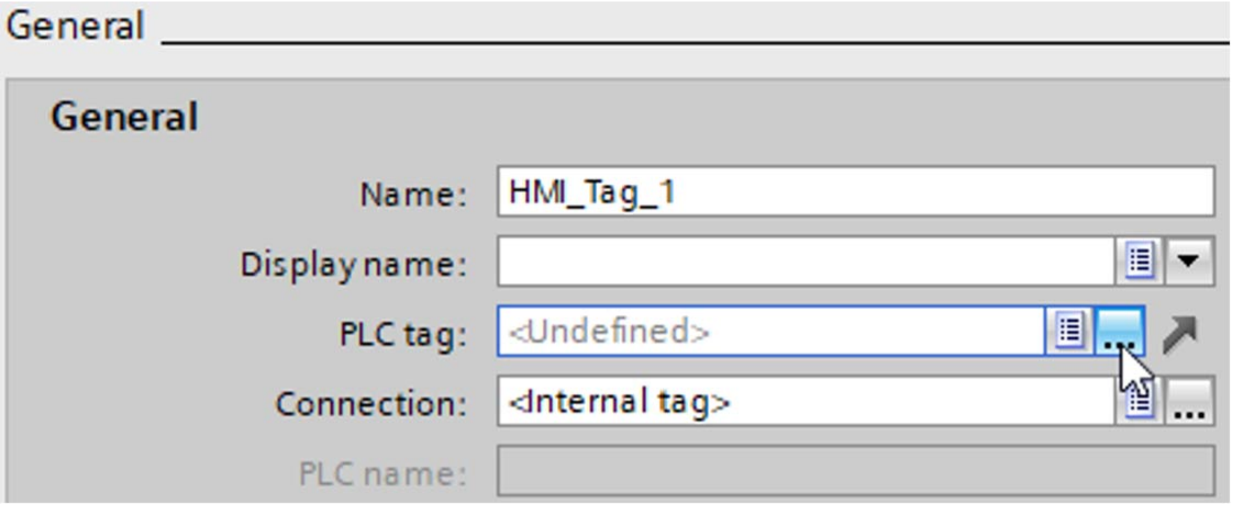
8. Add the declared data to the new tag in *iC7_HMI_UDT* entry field of the faceplate.
9. Press the button with the *(3 dots)*.
10. Select *Add new* in the pop-up screen.



e30bl416.10

Figure 36: Add New Tag

11. Select the button with the *(3 dots)* on the line *PLC tag*.



General

General

Name: HMI_Tag_1

Display name:

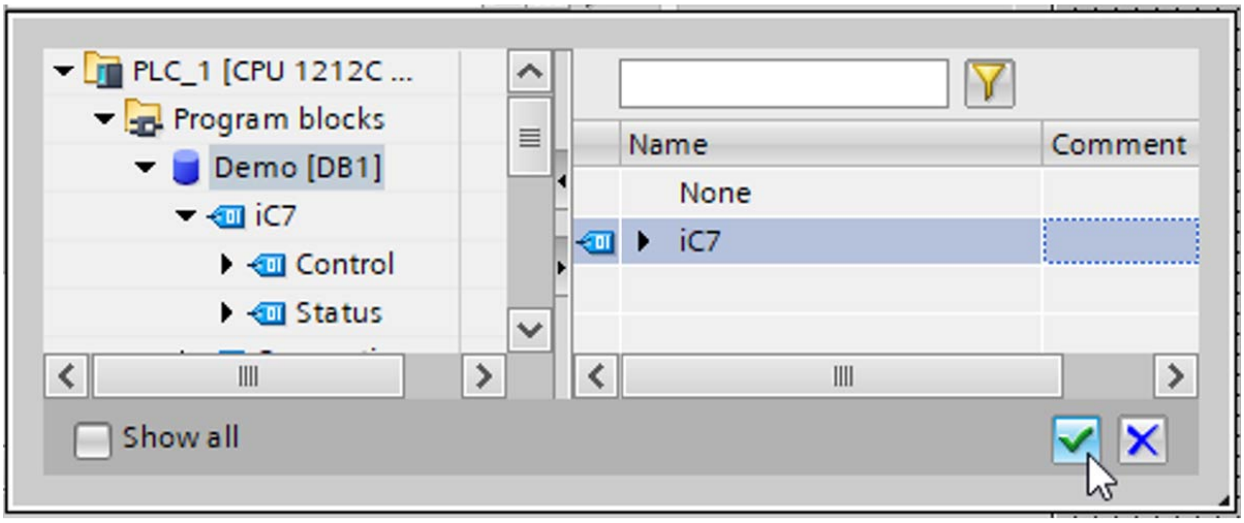
PLC tag: <Undefined>

Connection: <Internal tag>

PLC name:

Figure 37: Select the Suspension Point by the PLC Tag

12. Locate the tag where the UDT is declared, for example, in the global DB and press the *checkmark*.



PLC_1 [CPU 1212C ...
Program blocks
Demo [DB1]
iC7
Control
Status

Name	Comment
None	
iC7	

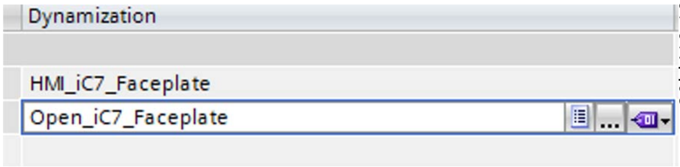
☐ Show all

Figure 38: Locate the Tag for the UDT

> This adds the PLC tag to the HMI tag inclusive the connection point. Before closing the view, change the name of the HMI tag if needed. The new created HMI tag is visible in the tag definition for the faceplate.

13. The *Open_iC7_Faceplate* definition needs, for example, an internal HMI tag for the purpose to open the face from. An example can be from a button on the HMI.

> Define the internal tag, use the same procedure as for the previous HMI tag. An internal HMI tag is sufficient.



Dynamization

HMI_iC7_Faceplate

Open_iC7_Faceplate

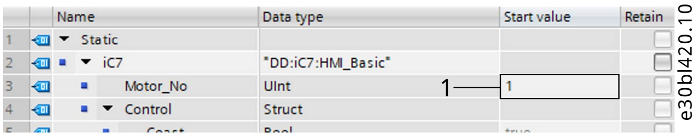
Figure 39: Open_iC7_Faceplate

Both HMI tags are now created.

14. The I/O data from the example global DB, where the HMI UDT is defined, can now be mapped to the function block.

Refer to chapter 5.2 Configuration of the Basic Drive Control and Monitoring Function Block for more information of the function block for usage with PROFINET RT communication.

NOTE: The field *Motor_No* can be used when multiple motors must be controlled from multiple faceplates. If needed, enter the motor number manually in the field.



	Name	Data type	Start value	Retain
1	Static			□
2	iC7	"DD:iC7:HMI_Basic"		□
3	Motor_No	UInt	1	□
4	Control	Struct		□
5	Coast	Bool	true	□

Figure 40: I/O Data Mapped to the Function Block

1	Field for entering motor number manually.

6.1.1 Operation of the Faceplate

The faceplate is ready to operate when the connection between the PLC and HMI is established. The drive must be ready for control of the motor, this is the mandatory condition to control the motor from the drive. A green rectangular shape around the [Start/Stop] button indicates that it is possible to control the motor from the drive. If the green rectangular shape is not present, it is not possible to start the motor. To achieve this condition, press the [Enable/Coast] button. This releases the control of the motor from the faceplate, when [Coast] is visible as illustrated in Figure 41.

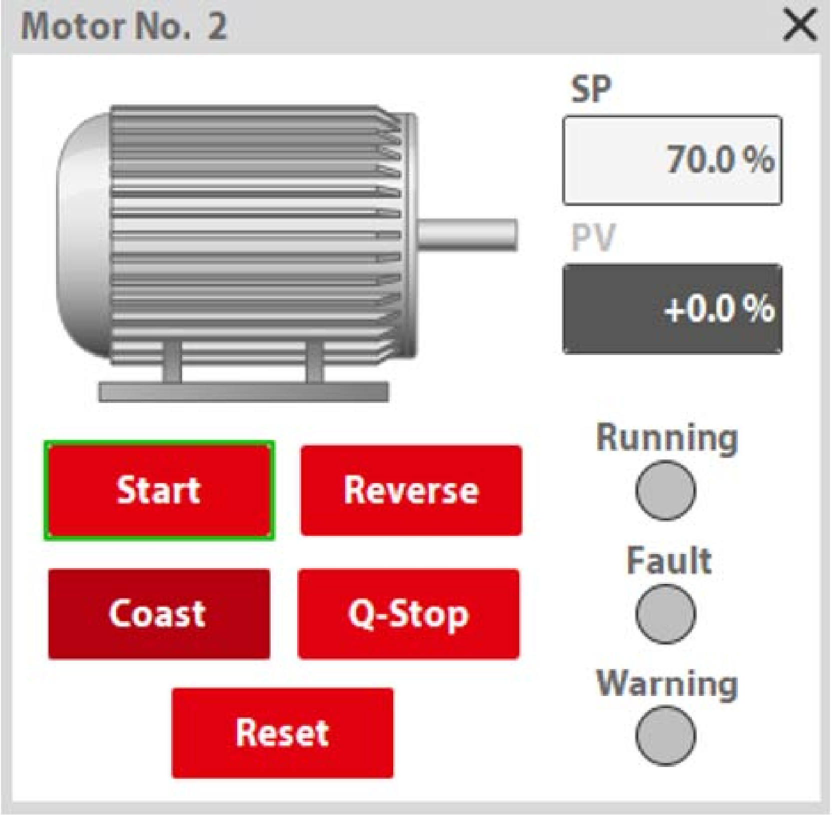


Figure 41: Start/Stop of the Motor from the Faceplate

The field *SP* is setting the speed reference value of the motor. When the speed reference (setpoint) of the motor speed is reached, a green rectangular shape around the field *SP* is visible. The green rectangular shape indicates that the actual motor speed (PV) is now on the requested motor speed (SP). The following illustration shows an example of setting the speed reference value. 70% is requested in the field *SP* and the requested motor speed of 70% is achieved in the field *PV*.

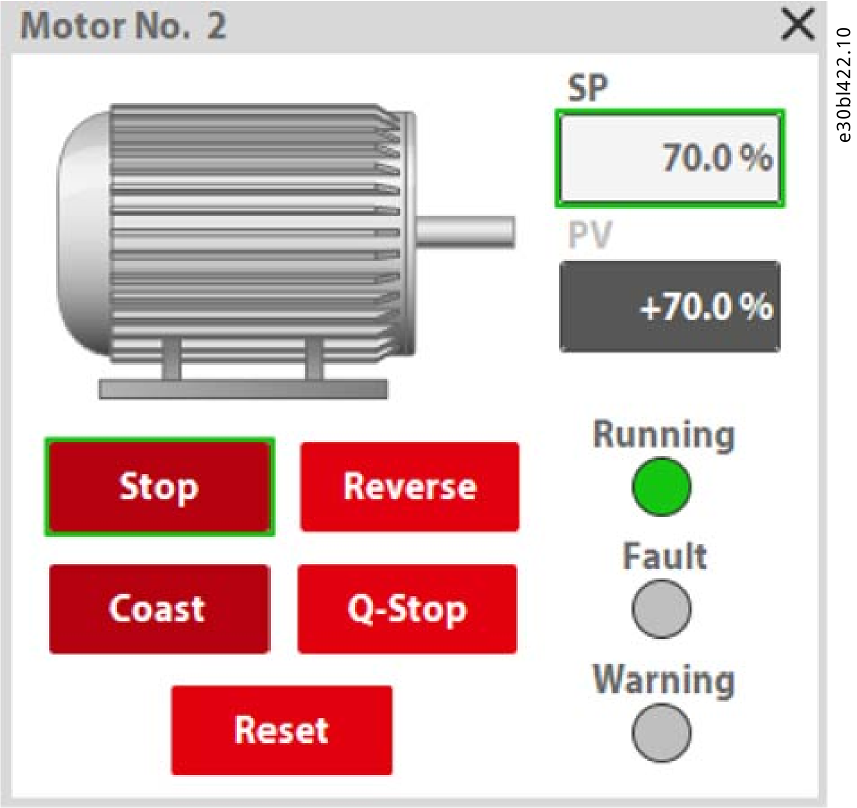


Figure 42: Setting the Speed Reference Value (SP)

The remaining buttons in Figure 43 have no limitations for control.

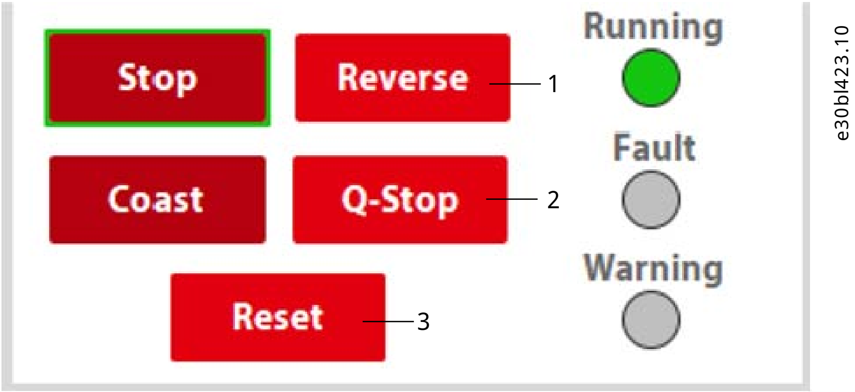


Figure 43: Button Overview

1	[Reverse/Forward] button can change the direction of the motor.	2	[Qstop] button executes a quick stop of the motor.
3	The [Reset] button resets an alarm if present.		

The status of the motor and/or drive is monitored via 3 indicators in Figure 44.

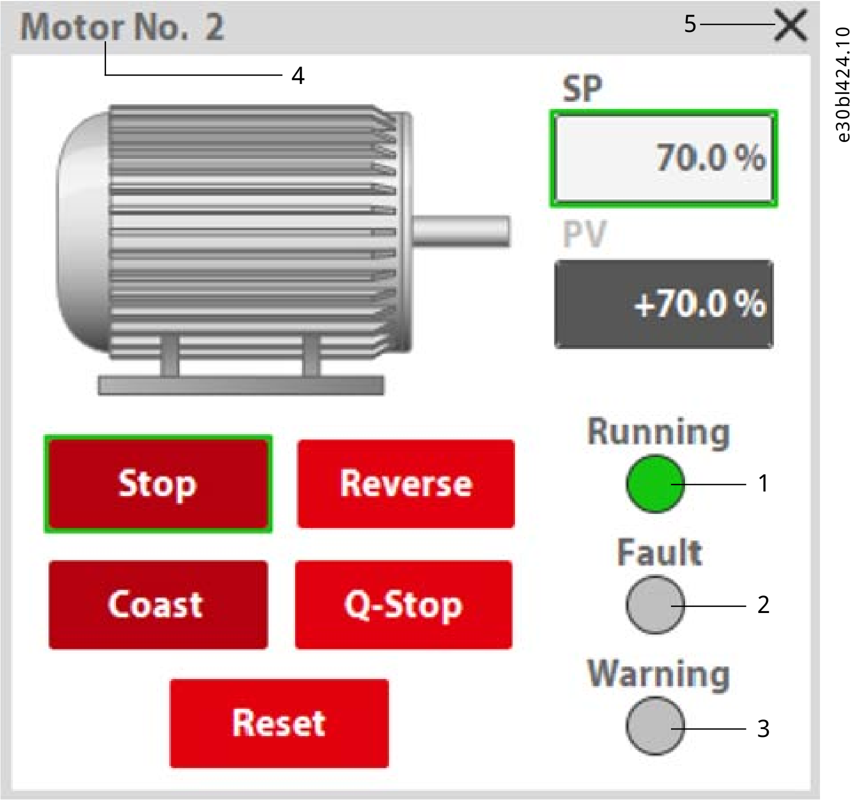


Figure 44: Status Overview

1	Running: The motor is magnetized and can turn if a given reference (SP) is given.	2	Fault: A fault is reported from the iC7 drive. The motor is coasted and cannot be controlled until the fault is removed from the iC7 drive.
3	Warning: A warning occurs from the iC7 drive. The motor can still be controlled, but with limited performance.	4	Motor number: Indicates which motor is controlled with the opened faceplate.
5	Close: Close the current opened faceplate.		

NOTE: Refer to the application guide for the specific iC7 drive, Industry, Motion, and so forth, for more information on occurrences of faults and warnings.

NOTE: In the same TIA17 sample project with the filename *iC7_PN_MTCP_S7_1200.zap*, there is also an example of how the faceplate is used in WinCC Advanced.

7 Drive Parameters

7.1 Drive Parameters Reading and Writing

To access parameters in the drive, the acyclic PROFINET communication channel in the drive is used. This is also known as the module access point in the device.

Since the iC7 drive series is conform to the PROFIdrive standard and certified accordingly, the access to the module access point is as per the PROFIdrive specification. This means that standard functions that support parameter access point via the PROFIdrive standard, can be used without any limitations. For this purpose, it is recommended to use the function blocks already given by Siemens for the Simatic S7-1200 and Simatic S7-1500 controller series. These function blocks are made according to the PROFIdrive specification, and can be used 1–1 with the iC7 drive series.

Function blocks and supporting material, refer to https://support.industry.siemens.com/cs/document/109479553/simatic-s7-1200-s7-1500-and-simotion-acyclic-data-exchange.

The following description only describes how the functions blocks are used with the iC7 drive series. For a detailed description on the usage of each function block for acyclic data exchange, refer to the guide given by Siemens.

NOTE: The acyclic data exchange channel is independent of the profile selection.

7.2 Acyclic Data Exchange Handling

It is recommended to use a function block from the resource manager library when acyclic data exchange is needed. If 2 or more acyclic data requests are simultaneously requested to a device (for example, a iC7 drive), conflicts can occur when processing the request in the device and a new request is sent. In this case, the function block *LAcycCom_ResourceManager (FB 30501)* can be used with the iC7 drive. This block handles a central buffer management for arbitration of acyclic data services, when communicating via the acyclic data channel. For more information about the resource management library for S7-1200 series and S7-1500 series controllers, refer to the Siemens guide *Acyclic Data Exchange for S7-1200/S7-1500 – Library LAcycCom – Resource Management*.

NOTE: The Resource Manager is mandatory to add, before it is possible to operate the read and write function blocks from the LAcycCom library. The request buffer is needed with all function blocks for the acyclic read and write operations.

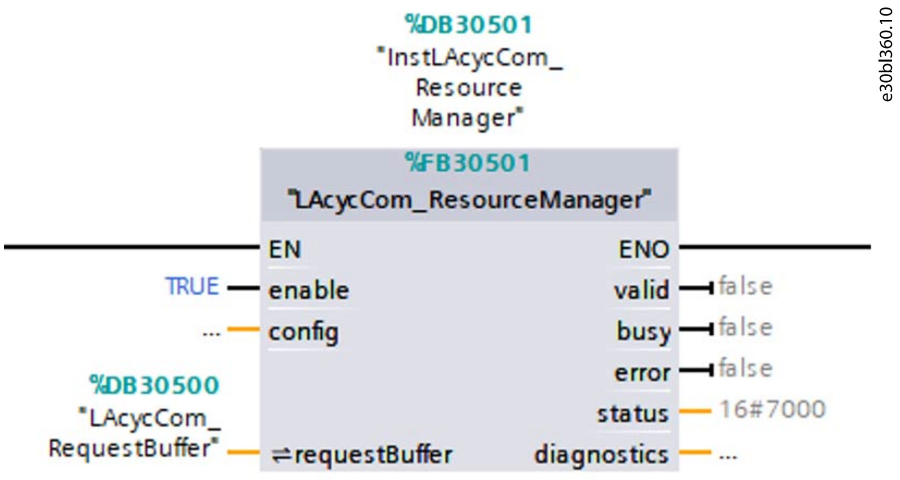


Figure 45: Resource Manager

NOTE: Make sure the same request buffer *RequestBuffer* defined DB with this function block must be used across all LAcycCom function blocks for proper operation.

7.3 Single Parameter Reading

Use the *LAcycCom_ReadDriveSingleParam (FB 30510)* if only a single drive parameter must be read.

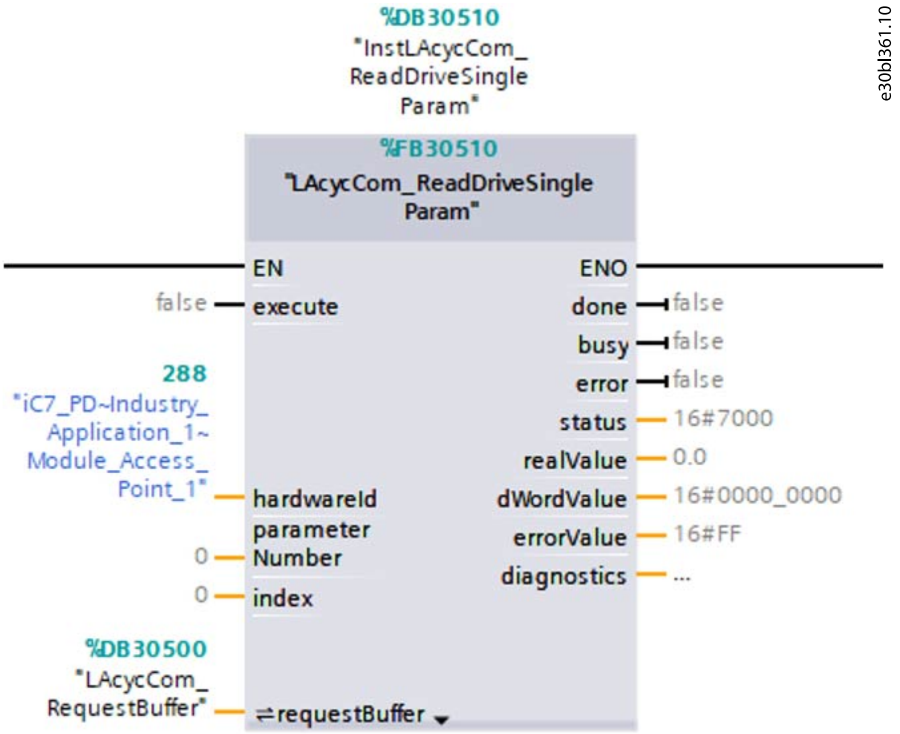
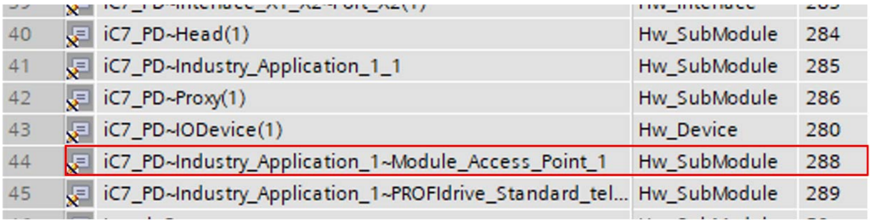


Figure 46: Read Drive Single Parameter

The hardware ID of the module access point *hardwareId* for the drive is available in the tag table. Select *Show all tags* and the tab *System constants* to get access to the hardware IDs of all the devices configured in the project. It is always the Module Access Point that must be identified as the hardware ID for acyclic data exchange.

NOTE: To handle acyclic communication to and from iC7 parameter reserved for options, the same procedure is used. Find the hardware ID in the tag list of the system constants, where the corresponding Module Access Point is located for each individual option.



40	iC7_PD~Head(1)	Hw_SubModule	284
41	iC7_PD~Industry_Application_1_1	Hw_SubModule	285
42	iC7_PD~Proxy(1)	Hw_SubModule	286
43	iC7_PD~IODevice(1)	Hw_Device	280
44	iC7_PD~Industry_Application_1~Module_Access_Point_1	Hw_SubModule	288
45	iC7_PD~Industry_Application_1~PROFIdrive_Standard_tel...	Hw_SubModule	289

Figure 47: Example of Identification of Module Access Point

For more information about single parameter read access with the function block *LAcycCom_ReadDriveSingleParam (FB 30510)*, refer to the Siemens guide *Acyclic Data Exchange for S7-1200/S7-1500 – Library LAcycCom – Drive Based Functionalities*.

Refer to 8.1 Localize Parameter Numbers with MyDrive® Insight for information on drive parameter object number, on how to determine the parameter number IDs.

The I/O data for handling the function block is provided as part of the instance DB. No parameter number or any other attribute is given in the example. The value from a parameter read is given in *REAL* or double word, all 32-bit values.

7.4 Single Parameter Writing

Use the function block *LAcycCom_WriteDriveSingleParam (FB 30511)*, if only a single parameter must be written to the iC7 drive.

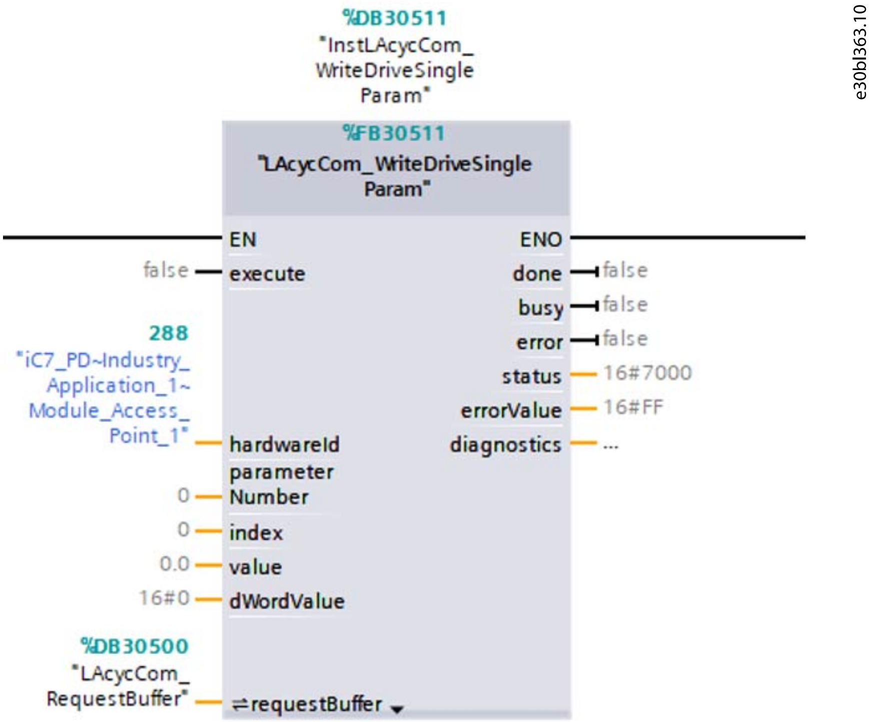


Figure 48: Write Drive Single Parameter

The hardware ID *hardwareId* is similar as for the *ReadDriveSingleParam* function block. Add the parameter input values in *REAL* or double word for all 32-bit values. Most of the parameter data types in the iC7 drive series are 32-bit values, mainly as *REAL* types.

NOTE: To write *1* in a boolean parameter (P1.3.3, and so forth), use the value *65536.0* in *REAL* format or *16#0001_0000* in double word format.

7.5 Multiple Parameter Reading

The major difference between single parameter read function block and multiple parameters read function block *LAcycCom_ReadDriveParams (FB 30512)* is the introduction of a dataset. The dataset determines which parameter object IDs are requested and stores the parameter values accordingly.

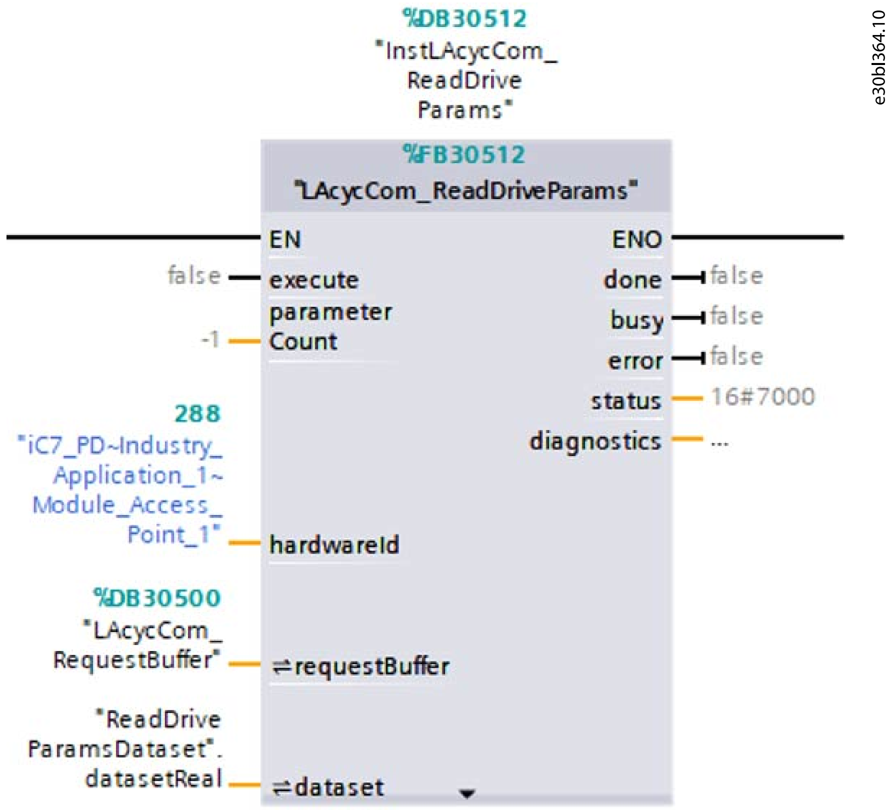
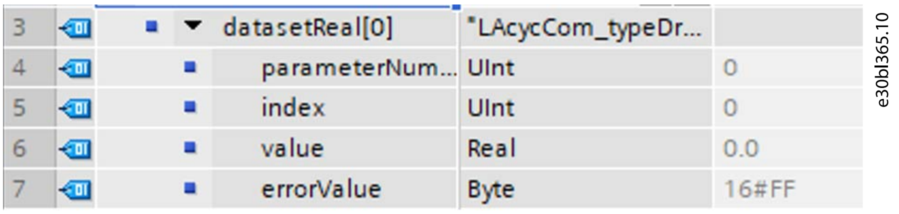


Figure 49: Read Drive Multiple Parameters

Each array in the dataset DB consists of a parameter number, index of the parameter, value, and an error value. Before execution, fill-in only the parameter number *parameterNumber* and index. The parameter value or error value are given after a successful execution of the function block.



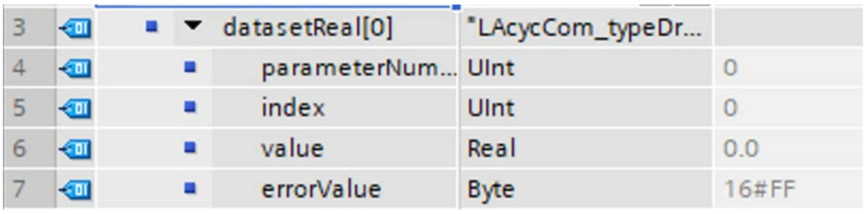
3	▼ datasetReal[0]	"LAcycCom_typeDr...	
4	parameterNum...	UInt	0
5	index	UInt	0
6	value	Real	0.0
7	errorValue	Byte	16#FF

Figure 50: Example of Dataset

NOTE: A maximum of 39 individual parameters can be read using 1 parameter dataset. In this case, the maximum telegram length of 240 bytes is used.

7.6 Multiple Parameters Writing

There is no difference between the multiple parameters read and the multiple parameters write function block *LAcycCom_WriteDriveParams (FB 30513)* on the I/O definition. The same is for the dataset definition used for the function blocks. The main difference is that an array of data type REAL, DWORD, CHAR, or BYTE on the In/Out dataset definition *Value* with corresponding values must be provided.



3	▼ datasetReal[0]	"LAcycCom_typeDr...	
4	parameterNum...	UInt	0
5	index	UInt	0
6	value	Real	0.0
7	errorValue	Byte	16#FF

Figure 51: Example of Dataset

NOTE: A maximum of 19 individual parameters can be written with 1 request. In this case, the maximum telegram length of 240 bytes is used.

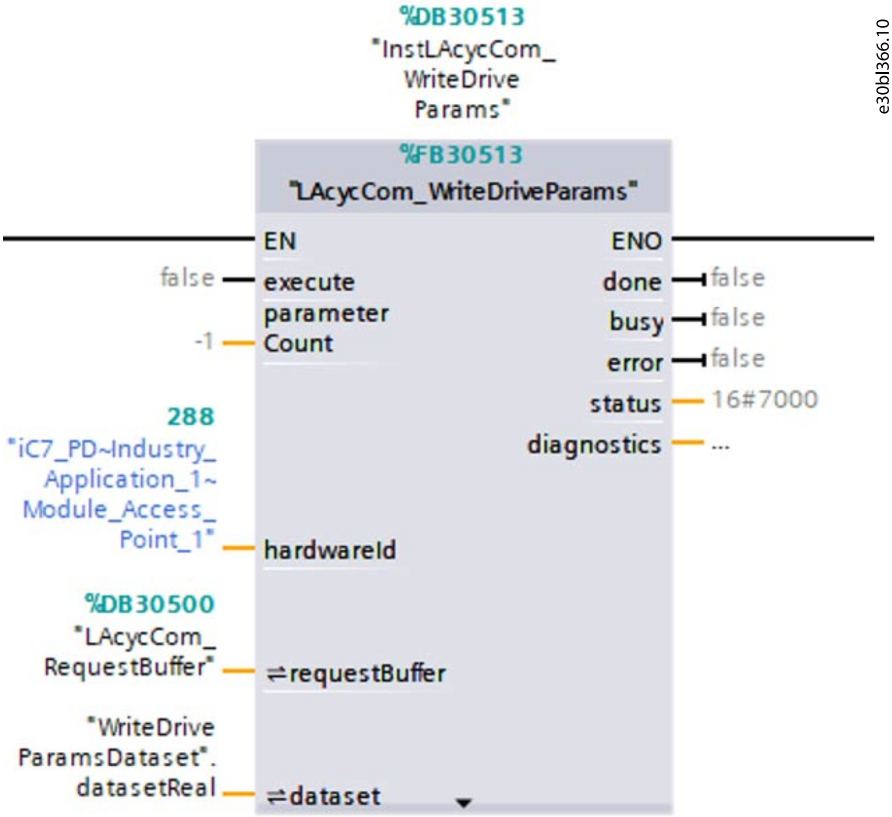


Figure 52: Write Drive Multiple Parameters

The sample project in the *LAcycCom library* is a good starting point to look into how these function blocks are used in TIA Portal.

8 Determining the iC7 Drive Object Parameter Number

8.1 Localize Parameter Numbers with MyDrive® Insight

All parameters in the iC7 drive have a unique index number, which is only available internally in the drive. Some of these index numbers are converted to a unique parameter number, which can be used to get access externally, for example, from a fieldbus. To localize a parameter number from a corresponding index number, use the Danfoss commissioning tool MyDrive® Insight. The latest version of the MyDrive® Insight tool can be downloaded from the link https://suite.mydrive.danfoss.com/content/tools/?id=24.

1. Open MyDrive® Insight.
2. Establish a communication from the MyDrive® Insight tool to the drive.

   > As default, a link local IPv4 port (169.254.x.x) is used to establish the Ethernet connection.

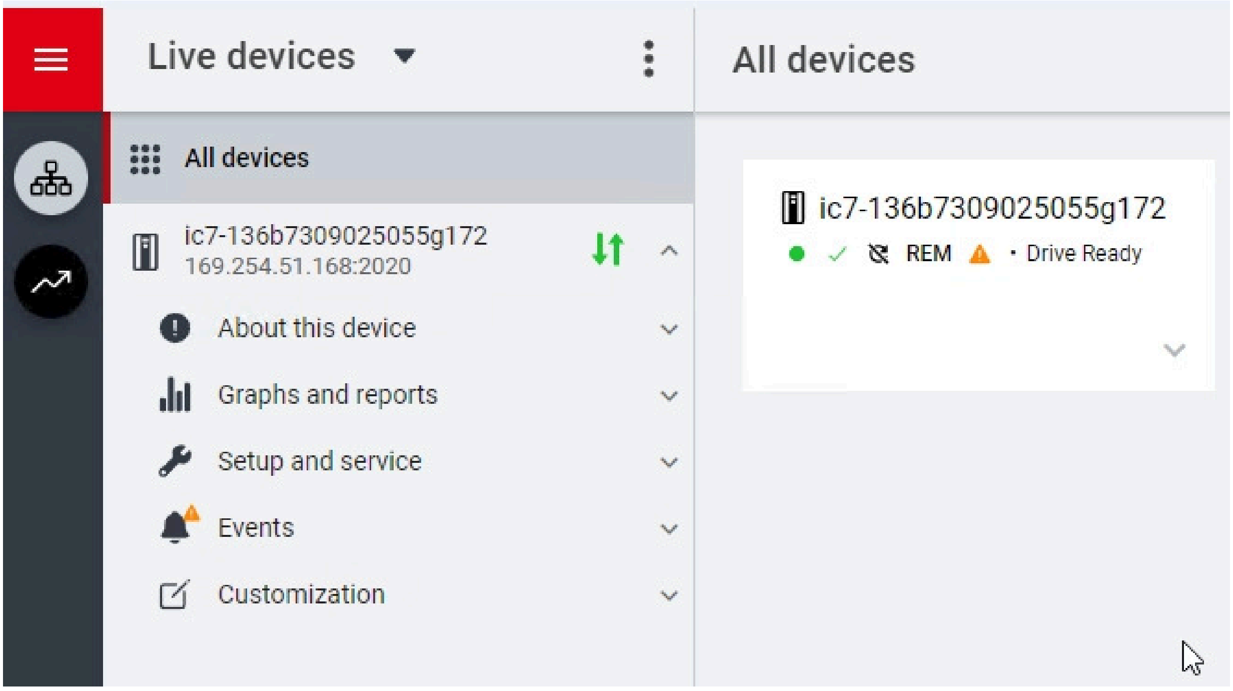


Figure 53: MyDrive® Insight Connected to the Drive

3. Select the connected drive in the *Live devices*.
4. Browse to the parameter live list by selecting the menu item *Setup and service*→*Parameters*→*Live*.
5. In the upper right corner of the parameter list, select the *3 dots* to get access to the columns configuration and select *Edit columns*.

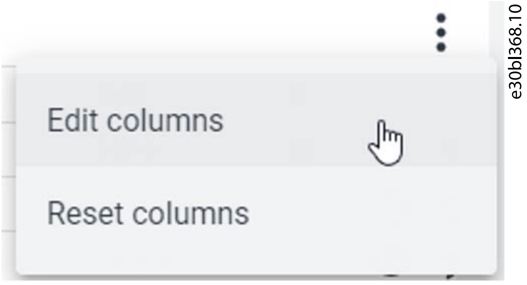


Figure 54: Edit Columns

6. Select the column *Number* in the list of available column items.
7. Add a checkmark and press *OK*.

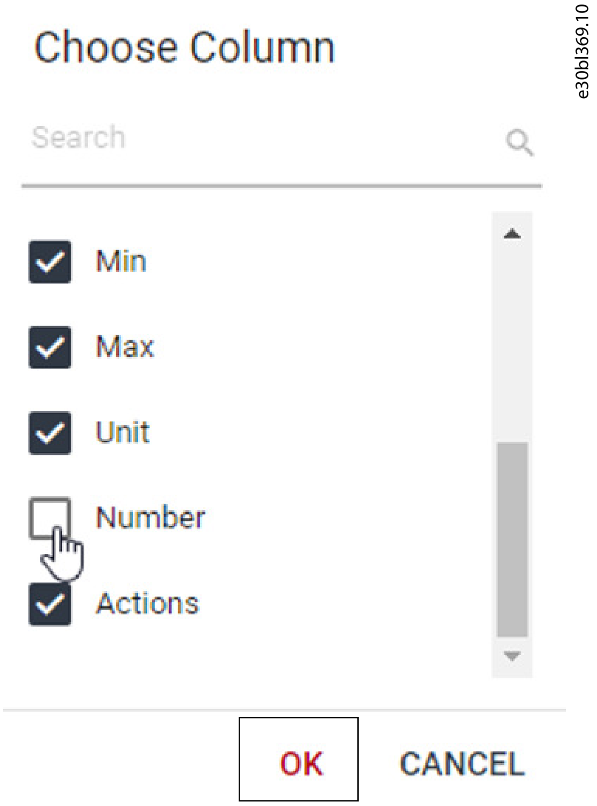


Figure 55: List of Available Columns Items

8. The new added column *Number* shows the corresponding parameter number for the index number.

The parameter number is needed for acyclic access to the drive. For example, to readout the actual motor current stated in index 4.1.1, address the corresponding parameter 9000.

Table 4: Index Number

Index	Name	Value	Default	Min	Max	Unit	Num...
4.1.1	Motor current	0.00		0.00	12.00	A	9000

9 Troubleshooting

9.1 Troubleshooting with the Basic Drive Function Block

The following flow chart can be used for troubleshooting with the basic drive function block.

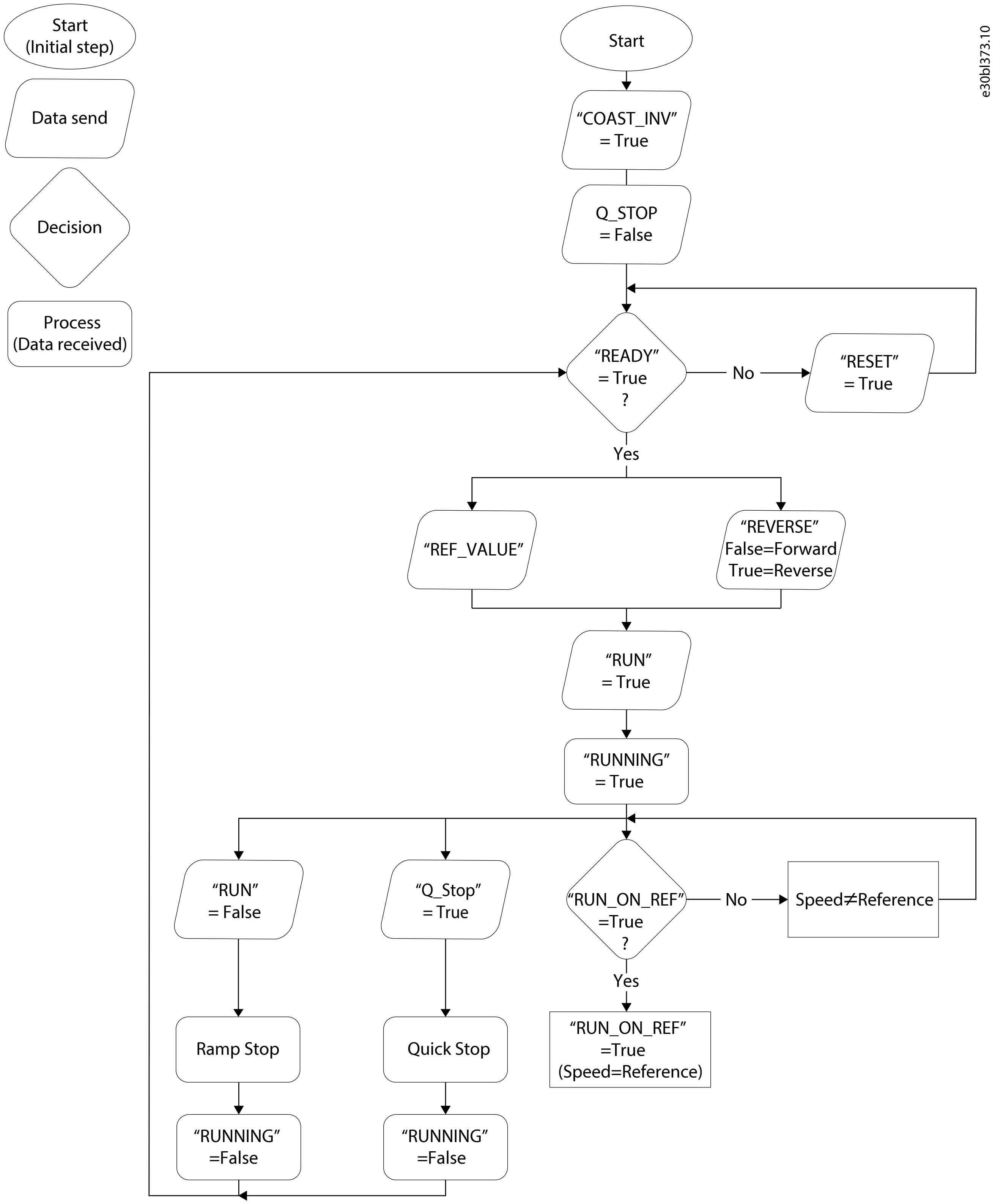


Figure 56: Troubleshooting with the Basic Drive Function Block

ENGINEERING
TOMORROW

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

Any information, including, but not limited to information on selection of product, its application or use, product design, weight, dimensions, capacity or any other technical data in product manuals, catalog descriptions, advertisements, etc. and whether made available in writing, orally, electronically, online or via download, shall be considered informative, and is only binding if and to the extent, explicit reference is made in a quotation or order confirmation. Danfoss cannot accept any responsibility for possible errors in catalogs, brochures, videos and other material. Danfoss reserves the right to alter its products without notice. This also applies to products ordered but not delivered provided that such alterations can be made without changes to form, fit or function of the product. All trademarks in this material are property of Danfoss A/S or Danfoss group companies. Danfoss and the Danfoss logo are trademarks of Danfoss A/S. All rights reserved.

M00437

1 3 6 R 0 3 4 5