

EcoStruxure Machine Expert TeSys island Library Guide

06/2019

EIO0000003855.00

www.schneider-electric.com



The information provided in this documentation contains general descriptions and/or technical characteristics of the performance of the products contained herein. This documentation is not intended as a substitute for and is not to be used for determining suitability or reliability of these products for specific user applications. It is the duty of any such user or integrator to perform the appropriate and complete risk analysis, evaluation and testing of the products with respect to the relevant specific application or use thereof. Neither Schneider Electric nor any of its affiliates or subsidiaries shall be responsible or liable for misuse of the information contained herein. If you have any suggestions for improvements or amendments or have found errors in this publication, please notify us.

You agree not to reproduce, other than for your own personal, noncommercial use, all or part of this document on any medium whatsoever without permission of Schneider Electric, given in writing. You also agree not to establish any hypertext links to this document or its content. Schneider Electric does not grant any right or license for the personal and noncommercial use of the document or its content, except for a non-exclusive license to consult it on an "as is" basis, at your own risk. All other rights are reserved.

All pertinent state, regional, and local safety regulations must be observed when installing and using this product. For reasons of safety and to help ensure compliance with documented system data, only the manufacturer should perform repairs to components.

When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

Failure to use Schneider Electric software or approved software with our hardware products may result in injury, harm, or improper operating results.

Failure to observe this information can result in injury or equipment damage.

© 2019 Schneider Electric. All rights reserved.

Table of Contents



	Safety Information	7
	About the Book	11
Part I	General Information	17
Chapter 1	Presentation of the Library	19
	General Information	19
Chapter 2	Common Inputs and Outputs	21
	Referencing Inputs to the TeSys island Library	22
	Behavior of Function Blocks with the Input i_xExecute	27
	Behavior of Function Blocks with the Input i_xEnable	28
	Errorhandling of the Function Blocks	28
Part II	Avatar Description	29
Chapter 3	Avatar Description	31
	Avatar Description	31
Part III	Data Unit Types	33
Chapter 4	Enumerations	35
	ET_Result	36
	ET_SafeStop	42
	ET_TripMessage	43
	ET_WarningMessage	44
Chapter 5	Structures	45
	ST_MinorFltRegister	46
	ST_TripRegister	47
	ST_VoltageRegister	48
Part IV	Global Variables	49
Chapter 6	Global Constants List	51
	Global Constants List (GCL)	51
Chapter 7	Global Parameter List	53
	Global Parameter List (GPL)	53
Part V	Program Organization Units (POU)	55
Chapter 8	Function Block Description	57
8.1	General Function Block Description	58
	Avatar Function Block Description	58
8.2	System Control Function Blocks	60
	SystemControl - Functional Description	60

8.3	System Diagnostic Function Blocks	62
	ReadSystemDiagnostics - Functional Description	62
8.4	System Energy Function Blocks	65
	SystemEnergyBasic - Functional Description	66
	SystemPowerBasic - Functional Description	68
	SystemVoltageBasic - Functional Description	71
	SystemVoltageEnhanced - Functional Description	73
8.5	System Asset Management Function Blocks	75
	SystemMaintenanceData - Functional Description	76
	SystemProductData - Functional Description	77
8.6	Control Function Blocks	79
	AnalogIO - Functional Description	80
	DigitalIO - Functional Description	82
	EnhancedStatus - Functional Description	84
	MotorOneDirection - Functional Description	86
	MotorOneDirectionSafeStopWCatland2 - Functional Description	88
	MotorTwoDirections - Functional Description	90
	MotorTwoDirectionsSafeStopWCatland2 - Functional Description	92
	MotorTwoSpeeds - Functional Description	95
	MotorTwoSpeedsSafeStopWCatland2 - Functional Description ..	98
	MotorTwoSpeedsTwoDirections - Functional Description	101
	MotorTwoSpeedsTwoDirectionsSafeStopWCatland2 - Functional Description	105
	MotorYDOneDirection - Functional Description	108
	MotorYDTwoDirections - Functional Description	111
	PowerInterface - Functional Description	114
	PowerInterfaceIO - Functional Description	116
	PowerSupply - Functional Description	119
	Resistor - Functional Description	121
	Switch - Functional Description	123
	SwitchSafeStopWCatland2 - Functional Description	125
	Transformer - Functional Description	127
8.7	Energy Function Blocks	129
	Energy - Functional Description	130
	Power - Functional Description	133

8.8	Diagnostic Function Blocks	135
	ReadDiagnostics - Functional Description	136
	ReadTripCounters - Functional Description	138
	ReadTripRegisters - Functional Description	140
	ReadWarningCounters - Functional Description	142
8.9	Asset Management Function Blocks	144
	MaintenanceData - Functional Description	145
	ProductData - Functional Description	147
Glossary	149
Index	151

Safety Information



Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, service, or maintain it. The following special messages may appear throughout this documentation or on the equipment to warn of potential hazards or to call attention to information that clarifies or simplifies a procedure.



The addition of this symbol to a “Danger” or “Warning” safety label indicates that an electrical hazard exists which will result in personal injury if the instructions are not followed.



This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

DANGER

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

WARNING

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

CAUTION

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

NOTICE

NOTICE is used to address practices not related to physical injury.

PLEASE NOTE

Electrical equipment should be installed, operated, serviced, and maintained only by qualified personnel. No responsibility is assumed by Schneider Electric for any consequences arising out of the use of this material.

A qualified person is one who has skills and knowledge related to the construction and operation of electrical equipment and its installation, and has received safety training to recognize and avoid the hazards involved.

QUALIFICATION OF PERSONNEL

A qualified person is one who has the following qualifications:

- Skills and knowledge related to the construction and operation of electrical equipment and the installation.
- Knowledge about providing machine functionality in software implementation.
- Received safety-related training to recognize and avoid the hazards involved.

The qualified person must be able to detect possible hazards that may arise from parameterization, modifying parameter values and generally from mechanical, electrical, or electronic equipment. The qualified person must be familiar with the standards, provisions, and regulations for the prevention of industrial accidents, which they must observe when designing and implementing the system.

PROPER USE

This product is a library to be used together with the control systems and servo amplifiers intended solely for the purposes as described in the present documentation as applied in the industrial sector.

Always observe the applicable safety-related instructions, the specified conditions, and the technical data.

Perform a risk evaluation concerning the specific use before using the product. Take protective measures according to the result.

Since the product is used as a part of an overall system, you must ensure the safety of the personnel by means of the design of this overall system (for example, machine design).

Any other use is not intended and may be hazardous.

BEFORE YOU BEGIN

Do not use this product on machinery lacking effective point-of-operation guarding. Lack of effective point-of-operation guarding on a machine can result in serious injury to the operator of that machine.

WARNING

UNGUARDED EQUIPMENT

- Do not use this software and related automation equipment on equipment which does not have point-of-operation protection.
- Do not reach into machinery during operation.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.

Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.

In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

Ensure that appropriate safeties and mechanical/electrical interlocks related to point-of-operation protection have been installed and are operational before placing the equipment into service. All interlocks and safeties related to point-of-operation protection must be coordinated with the related automation equipment and software programming.

NOTE: Coordination of safeties and mechanical/electrical interlocks for point-of-operation protection is outside the scope of the Function Block Library, System User Guide, or other implementation referenced in this documentation.

START-UP AND TEST

Before using electrical control and automation equipment for regular operation after installation, the system should be given a start-up test by qualified personnel to verify correct operation of the equipment. It is important that arrangements for such a check be made and that enough time is allowed to perform complete and satisfactory testing.

WARNING

EQUIPMENT OPERATION HAZARD

- Verify that all installation and set up procedures have been completed.
- Before operational tests are performed, remove all blocks or other temporary holding means used for shipment from all component devices.
- Remove tools, meters, and debris from equipment.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Follow all start-up tests recommended in the equipment documentation. Store all equipment documentation for future references.

Software testing must be done in both simulated and real environments.

Verify that the completed system is free from all short circuits and temporary grounds that are not installed according to local regulations (according to the National Electrical Code in the U.S.A, for instance). If high-potential voltage testing is necessary, follow recommendations in equipment documentation to prevent accidental equipment damage.

Before energizing equipment:

- Remove tools, meters, and debris from equipment.
- Close the equipment enclosure door.
- Remove all temporary grounds from incoming power lines.
- Perform all start-up tests recommended by the manufacturer.

OPERATION AND ADJUSTMENTS

The following precautions are from the NEMA Standards Publication ICS 7.1-1995 (English version prevails):

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.
- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments actually required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Book



At a Glance

Document Scope

This document describes the library TeSys island.

The TeSys island library provides function blocks for EcoStruxure Machine Expert environment to control the avatar and device modules of a TeSys island.

Validity Note

This document has been updated for the release of EcoStruxure™ Machine Expert V1.1.

The technical characteristics of the devices described in the present document also appear online. To access the information online:

Step	Action
1	Go to the Schneider Electric home page www.schneider-electric.com .
2	In the Search box type the reference of a product or the name of a product range. <ul style="list-style-type: none">• Do not include blank spaces in the reference or product range.• To get information on grouping similar modules, use asterisks (*).
3	If you entered a reference, go to the Product Datasheets search results and click on the reference that interests you. If you entered the name of a product range, go to the Product Ranges search results and click on the product range that interests you.
4	If more than one reference appears in the Products search results, click on the reference that interests you.
5	Depending on the size of your screen, you may need to scroll down to see the datasheet.
6	To save or print a datasheet as a .pdf file, click Download XXX product datasheet .

The characteristics that are presented in the present document should be the same as those characteristics that appear online. In line with our policy of constant improvement, we may revise content over time to improve clarity and accuracy. If you see a difference between the document and online information, use the online information as your reference.

WARNING

LOSS OF CONTROL

- The designer of any control scheme must consider the potential failure modes of control paths and, for certain critical control functions, provide a means to achieve a safe state during and after a path failure. Examples of critical control functions are emergency stop and overtravel stop, power outage and restart.
- Separate or redundant control paths must be provided for critical control functions.
- System control paths may include communication links. Consideration must be given to the implications of unanticipated transmission delays or failures of the link.
- Observe all accident prevention regulations and local safety guidelines.¹
- Each implementation of this equipment must be individually and thoroughly tested for proper operation before being placed into service.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

¹ For additional information, refer to NEMA ICS 1.1 (latest edition), "Safety Guidelines for the Application, Installation, and Maintenance of Solid State Control" and to NEMA ICS 7.1 (latest edition), "Safety Standards for Construction and Guide for Selection, Installation and Operation of Adjustable-Speed Drive Systems" or their equivalent governing your particular location.

Before you attempt to provide a solution (machine or process) for a specific application using the POU's found in the library, you must consider, conduct and complete best practices. These practices include, but are not limited to, risk analysis, functional safety, component compatibility, testing and system validation as they relate to this library.

WARNING

IMPROPER USE OF PROGRAM ORGANIZATION UNITS

- Perform a safety-related analysis for the application and the devices installed.
- Ensure that the Program Organization Units (POUs) are compatible with the devices in the system and have no unintended effects on the proper functioning of the system.
- Use appropriate parameters, especially limit values, and observe machine wear and stop behavior.
- Verify that the sensors and actuators are compatible with the selected POU's.
- Thoroughly test all functions during verification and commissioning in all operation modes.
- Provide independent methods for critical control functions (emergency stop, conditions for limit values being exceeded, etc.) according to a safety-related analysis, respective rules, and regulations.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Only use software approved by Schneider Electric for use with this equipment.
- Update your application program every time you change the physical hardware configuration.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Incomplete file transfers, such as data files, application files and/or firmware files, may have serious consequences for your machine or controller. If you remove power, or if there is a power outage or communication interruption during a file transfer, your machine may become inoperative, or your application may attempt to operate on a corrupted data file. If an interruption occurs, reattempt the transfer. Be sure to include in your risk analysis the impact of corrupted data files.

WARNING

UNINTENDED EQUIPMENT OPERATION, DATA LOSS, OR FILE CORRUPTION

- Do not interrupt an ongoing data transfer.
- If the transfer is interrupted for any reason, re-initiate the transfer.
- Do not place your machine into service until the file transfer has completed successfully, unless you have accounted for corrupted files in your risk analysis and have taken appropriate steps to prevent any potentially serious consequences due to unsuccessful file transfers.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Care must be taken and provisions made for use of this library for machine control to avoid inadvertent consequences of commanded machine operation, state changes, or alteration of data memory or machine operating elements.

WARNING

UNINTENDED EQUIPMENT OPERATION

- Place operator devices of the control system near the machine or in a place where you have full view of the machine.
- Protect operator commands against unauthorized access.
- If remote control is a necessary design aspect of the application, ensure that there is a local, competent, and qualified observer present when operating from a remote location.
- Configure and install the Run/Stop input, if so equipped, or, other external means within the application, so that local control over the starting or stopping of the device can be maintained regardless of the remote commands sent to it.

Failure to follow these instructions can result in death, serious injury, or equipment damage.

Related Documents

Document title	Reference
TeSys™ island System Guide	www.schneider-electric.us/en/download/ ;
TeSys™ island Operating Guide	www.schneider-electric.us/en/download/ ;
TeSys™ island DTM Library Online Help	www.schneider-electric.us/en/download/ ;
Modicon M241 Logic Controller Hardware Guide	EIO0000001456 (ENG) ; EIO0000001457 (FRE) ; EIO0000001458 (GER) ; EIO0000001459 (SPA) ; EIO0000001460 (ITA) ; EIO0000001461 (CHS)
Modicon M251 Logic Controller Hardware Guide	EIO0000001486 (ENG) ; EIO0000001487 (FRE) ; EIO0000001488 (GER) ; EIO0000001489 (SPA) ; EIO0000001490 (ITA) ; EIO0000001491 (CHS)
Modicon M262 Logic/Motion Controller Hardware Guide	EIO0000003659 (ENG) ; EIO0000003660 (FRE) ; EIO0000003661 (GER) ; EIO0000003662 (SPA) ; EIO0000003663 (ITA) ; EIO0000003664 (CHS)
EcoStruxure Machine Expert Functions and Libraries User Guide	EIO0000002829 (ENG) ; EIO0000002830 (FRE) ; EIO0000002831 (GER) ; EIO0000002832 (ITA) ; EIO0000002833 (SPA) ; EIO0000002834 (CHS) ;
EcoStruxure Machine Expert Programming Guide	EIO0000002854 (ENG) ; EIO0000002855 (FRE) ; EIO0000002856 (GER) ; EIO0000002857 (ITA) ; EIO0000002858 (SPA) ; EIO0000002859 (CHS) ;

Terminology Derived from Standards

The technical terms, terminology, symbols and the corresponding descriptions in this manual, or that appear in or on the products themselves, are generally derived from the terms or definitions of international standards.

In the area of functional safety systems, drives and general automation, this may include, but is not limited to, terms such as *safety*, *safety function*, *safe state*, *fault*, *fault reset*, *malfunction*, *failure*, *error*, *error message*, *dangerous*, etc.

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2015	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2013	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2015	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2016	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

In addition, terms used in the present document may tangentially be used as they are derived from other standards such as:

Standard	Description
IEC 60034 series	Rotating electrical machines
IEC 61800 series	Adjustable speed electrical power drive systems
IEC 61158 series	Digital data communications for measurement and control – Fieldbus for use in industrial control systems

Finally, the term *zone of operation* may be used in conjunction with the description of specific hazards, and is defined as it is for a *hazard zone* or *danger zone* in the *Machinery Directive (2006/42/EC)* and *ISO 12100:2010*.

NOTE: The aforementioned standards may or may not apply to the specific products cited in the present documentation. For more information concerning the individual standards applicable to the products described herein, see the characteristics tables for those product references.

Part I

General Information

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	Presentation of the Library	19
2	Common Inputs and Outputs	21

Chapter 1

Presentation of the Library

General Information

Library Overview

The TeSys island library provides function blocks to support developing applications and to control avatar modules. Avatar modules are digital function objects managed by TeSys island system. It is used to interact with power devices and accessories such as AIM (Analog Interface Module). The avatar modules are configured on the TeSys island and the TeSys island bus coupler (System avatar) is managing the communication with the controller via fieldbus.

The function blocks are fieldbus independent and have no logic instructions. The inputs and outputs of the function blocks are linked to the process image (cyclic data) without any modifications. Inputs and outputs, which are not part of the process image are exchanged via acyclic communication methods.

The configuration of the TeSys island bus coupler and the avatar modules is managed by the TeSys island DTM (Device Type Manager), which is integrated in EcoStruxure Machine Expert environment. For further information, refer to *TeSys island DTM Library Online Help*.

System Requirements and Compatibility

Component	Type and version
Software	EcoStruxure Machine Expert V1.1
Controller	<ul style="list-style-type: none">● Modicon M241 Logic Controller EtherNet/IP and Modbus TCP● Modicon M251 Logic Controller EtherNet/IP and Modbus TCP● Modicon M262 Logic/Motion Controller EtherNet/IP and Modbus TCP

Characteristics of the Library

The table indicates the characteristics of the library:

Characteristic	Value
Library title	TeSys island
Company	Schneider Electric
Category	Devices
Component	TeSys - Devices and Libraries
Default namespace	TeSysI
Language model attribute	qualified-access-only (<i>see EcoStruxure Machine Expert, Functions and Libraries User Guide</i>)
Forward compatible library	Yes (FCL (<i>see EcoStruxure Machine Expert, Functions and Libraries User Guide</i>))

NOTE: For this library, qualified-access-only is set. This means, that the POUs, data structures, enumerations, and constants have to be accessed using the namespace of the library. The default namespace of the library is **TeSysI**.

General Considerations

Only IPv4 IP addresses are supported for the communication functions provided with this library.

Chapter 2

Common Inputs and Outputs

What Is in This Chapter?

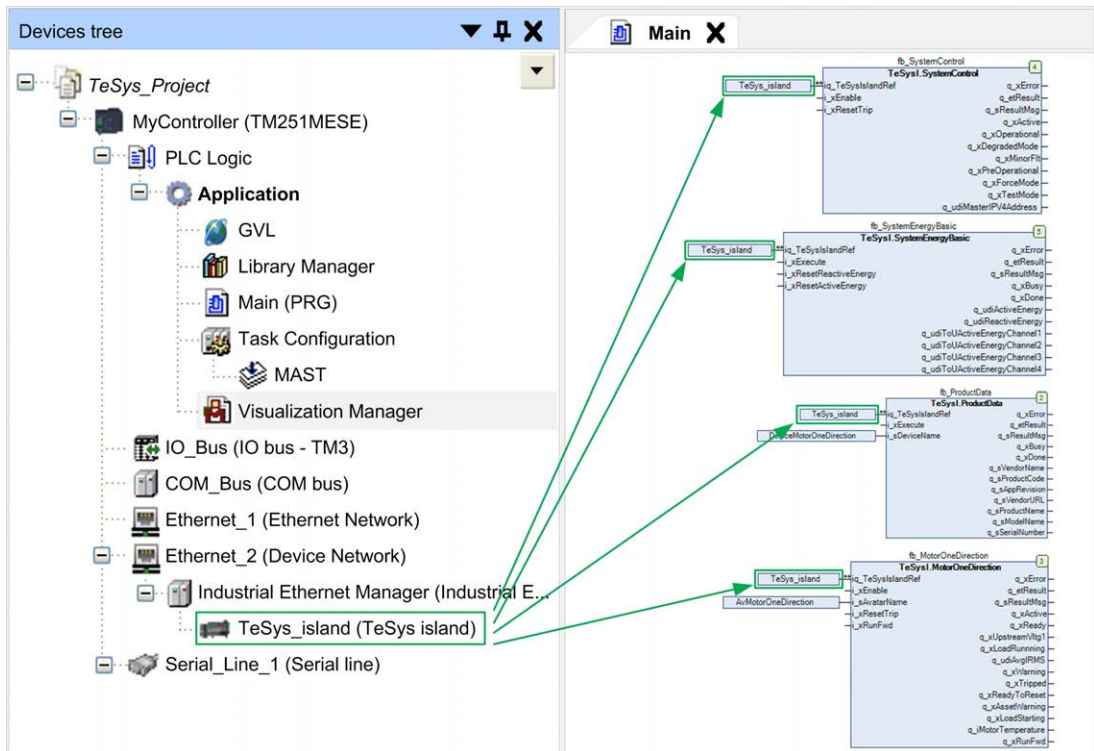
This chapter contains the following topics:

Topic	Page
Referencing Inputs to the TeSys island Library	22
Behavior of Function Blocks with the Input <code>i_xExecute</code>	27
Behavior of Function Blocks with the Input <code>i_xEnable</code>	28
Errorhandling of the Function Blocks	28

Referencing Inputs to the TeSys island Library

Referencing the TeSys island Bus Coupler from the Function Blocks

A reference to the TeSys island bus coupler is required by each function block of the TeSys island library. To achieve this, configure the name you assigned to the **TeSys_island** node in the **Devices tree** as input `iq_TeSysIslandRef` of the function blocks.



Referencing Avatars

The avatar name is used to select the avatar to be controlled by the function block. It is available at the avatar function blocks (see page 58), except the asset management function blocks. If the parameter **Avatar Name** is not configured or not correctly configured, the error message `ET_Result.AvatarNotAvailable` is returned. Modifying this name during the execution of the function block is ignored.

How to assign the avatar name to the function block input `i_sAvatarName`

Step	Action
1	Open the TeSys island configuration.
2	Select MY AVATAR .
3	Reference the avatar name (for example, <code>AvDigitalIO</code> or <code>AvMotorOneDirection</code>) to the function block input <code>i_sAvatarName</code> . NOTE: The avatar name can be found in the section AVATAR PARAMETERS .

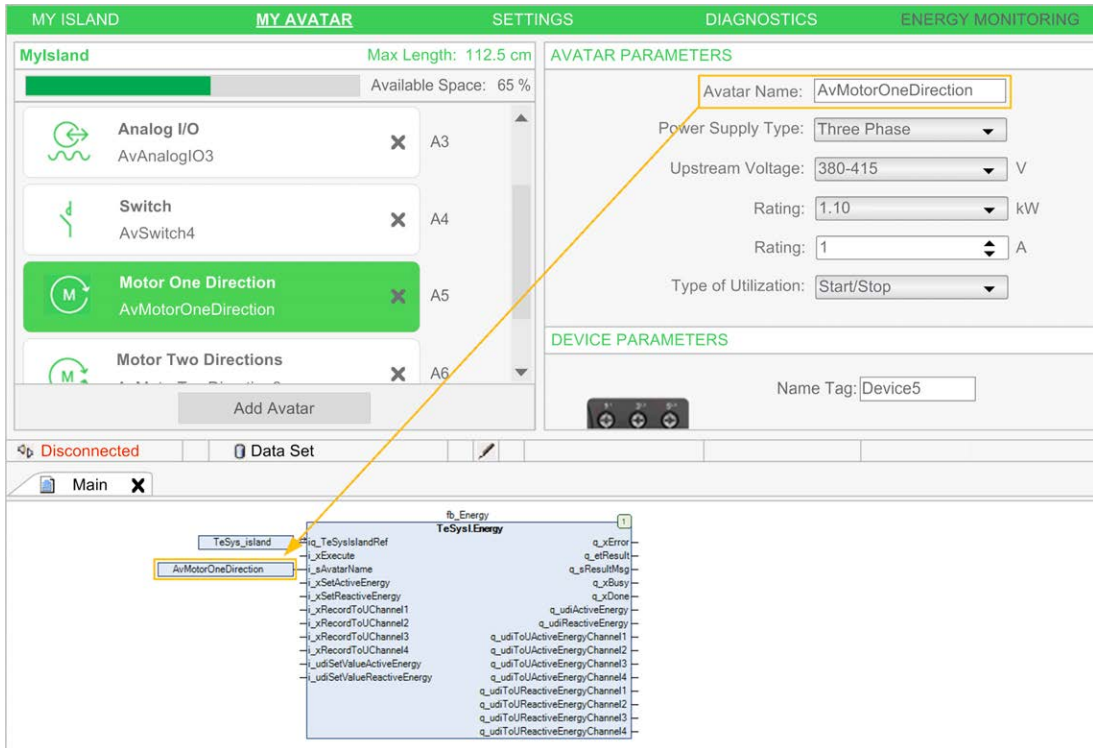
The following figure provides an example of the `DigitalIO` function block that is only available for **Digital I/O** avatars:

The screenshot displays the TeSys configuration interface for a 'MY AVATAR'. The interface is divided into several sections:

- MY ISLAND:** Shows the island configuration with a 'Max Length: 112.5 cm' and 'Available Space: 65 %'. A list of avatars is shown:
 - System (AvSystem1) - A1
 - Digital I/O (AvDigitalIO) - A2** (highlighted in green)
 - Analog I/O (AvAnalogIO3) - A3
 - Switch - A4
- AVATAR PARAMETERS:** Shows the 'Avatar Name' field set to 'AvDigitalIO'.
- DEVICE PARAMETERS:** Shows the 'Name Tag' set to 'Device2' and 'Power Supply Type' set to 'TPRDG4X2'.

At the bottom, a function block diagram shows the 'fb_DigitalIO' block. The input 'i_sAvatarName' is connected to the 'AvDigitalIO' avatar from the 'MY ISLAND' section. The block also shows various outputs such as 'q_xError', 'q_etResult', 'q_sResultMsg', 'q_xActive', 'q_xReady', 'q_xDI0', 'q_xDI1', 'q_xDI2', and 'q_xDI3'.

The following figure provides an example of the Energy function block that is available for all avatars, except for the **System** avatar:



Referencing Devices

The name is used to select the device at the function blocks for asset management. If the parameter **Name Tag** is not configured or not correctly configured, the error message `ET_Result.DeviceNotAvailable` is returned. Modifying this name during the execution of the function block is ignored.

How to assign the name tag to the function block input `i_sAvatarName`

Step	Action
1	Open the TeSys island configuration.
2	Select MY AVATAR .
3	Reference the name tag (for example, <code>Device1</code> , <code>Device2</code> , <code>Device3</code> , and so on) to the function block input <code>i_sDeviceName</code> . NOTE: The device name can be found in the section DEVICE PARAMETERS .

The following figure provides an example of the asset management ProductData function block that is available for all devices, except for the bus coupler (system device):

The screenshot displays the 'MY AVATAR' configuration window in the TeSys software. The window is divided into several sections:

- MY ISLAND:** Shows 'Max Length: 112.5 cm' and 'Available Space: 65 %'. A list of avatars is shown:
 - Analog I/O (AvAnalogIO3) - A3
 - Switch (AvSwitch4) - A4
 - Motor One Direction (AvMotorOneDirection) - A5** (highlighted in green)
 - Motor Two Directions (AvMotorTwoDirection6) - A6
- AVATAR PARAMETERS:**
 - Avatar Name: AvMotorOneDirection
 - Power Supply Type: Three Phase
 - Upstream Voltage: 380-415 V
 - Rating: 1.10 kW
 - Rating: 1 A
 - Type of Utilization: Start/Stop
- DEVICE PARAMETERS:**
 - Name Tag: Device5

Below the configuration window, a system diagram is visible. It shows a 'TeSys_Island' block connected to a 'TeSys_ProductData' block. The 'TeSys_ProductData' block has several input and output ports. The 'Device5' tag is highlighted in the diagram, and an arrow points from the 'Name Tag: Device5' field in the configuration window to this tag in the diagram.

System Function Blocks Automatically Referencing the Bus Coupler

In contrast to the above described function blocks, the system functions blocks do not require references to avatars or devices.

The SystemControl and SystemEnergyBasic function blocks, for example, do not have inputs referencing avatars or devices because they are directly linked to the bus coupler (system device):

The screenshot displays a control interface with several tabs: MY ISLAND, MY AVATAR, SETTINGS, DIAGNOSTICS, and ENERGY MONITORING. The MY AVATAR tab is active, showing a list of avatars on the left and configuration parameters on the right.

Avatar Configuration:

- MyIsland:** Max Length: 112.5 cm, Available Space: 65 %
- System (AvSystem1):** A1
- Digital I/O (AvDigitalIO):** A2
- Analog I/O (AvAnalogIO3):** A3
- Switch:** A4

Avatar Parameters (AvSystem1):

- Avatar Name: AvSystem1
- Fieldbus Protocol: EtherNet/IP

Device Parameters (Device1):

- Name Tag: Device1
- Power Supply Type: TPRBCEIP
- Firmware Version: 1.0

Below the configuration, a status bar shows "Disconnected" and "Data Set". A "Main" tab is also visible.

Function Block Diagrams:

The diagram shows two function blocks connected to a TeSys_island bus coupler:

- fb_SystemControl (TeSys1.SystemControl):**
 - Inputs: q_TeSysIslandRef, i_xEnable, i_xResetTrip
 - Outputs: q_xError, q_eResult, q_sResultMag, q_xActive, q_xOperational, q_xDegradedMode, q_xMinorFit, q_xPreOperational, q_xForceMode, q_xTestMode, q_udiMasterIPV4Address
- fb_SystemEnergyBasic (TeSys1.SystemEnergyBasic):**
 - Inputs: q_TeSysIslandRef, i_xExecute, i_xResetReactiveEnergy, i_xResetActiveEnergy
 - Outputs: q_xError, q_eResult, q_sResultMag, q_xBusy, q_xDone, q_udiActiveEnergy, q_udiReactiveEnergy, q_udiToUActiveEnergyChannel1, q_udiToUActiveEnergyChannel2, q_udiToUActiveEnergyChannel3, q_udiToUActiveEnergyChannel4

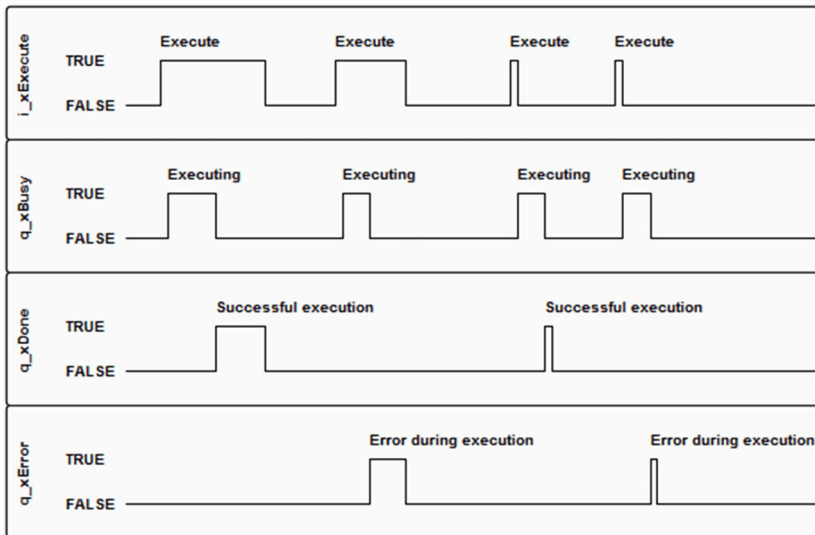
Behavior of Function Blocks with the Input `i_xExecute`

General Information

A rising edge of the input `i_xExecute` starts the execution of the function block and the output `q_xBusy` is set to TRUE. A rising edge at the input `i_xExecute` is ignored while the function block is being executed.

Once the execution is finished, the outputs `q_xDone` or `q_xError` are set to TRUE and remain TRUE until the input `i_xExecute` is set to FALSE. If the input is reset before the execution is finished, the outputs `q_xDone` or `q_xError` are set to TRUE for one controller cycle.

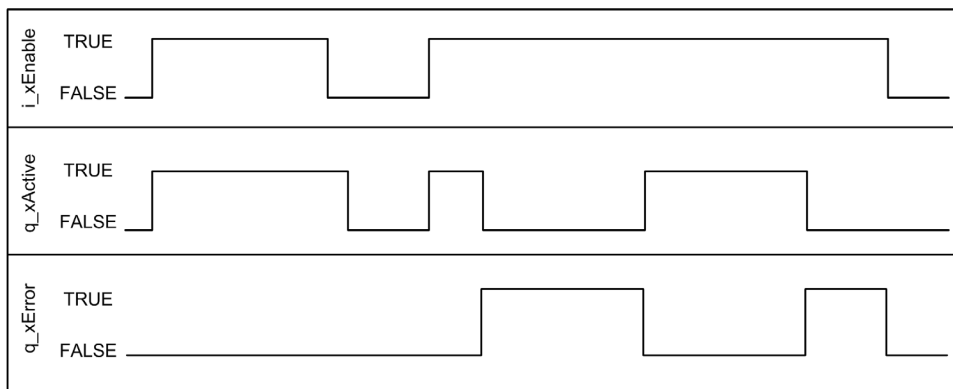
Example



Behavior of Function Blocks with the Input `i_xEnable`

General Information

A rising edge of the input `i_xEnable` starts the cyclic and acyclic data exchange between the function blocks and the selected avatar. As long as `i_xEnable` is TRUE, the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order they are defined in the function block. A falling edge at the input `i_xEnable` stops the data exchange and all active acyclic commands and the values of the outputs are set to zero.



Errorhandling of the Function Blocks

General Information

The function blocks have three outputs which are handling detected errors while executing the function block.

Output	Datatype	Description
<code>q_xError</code>	BOOL	A rising edge at the output <code>q_xError</code> provides information about an error occurred during the execution of the function block. The <code>q_etResult</code> and <code>q_sResultMsg</code> outputs are providing additional information on detected errors. The error outputs of the relevant function block are reset with a falling edge of the <code>i_xExecute</code> or <code>i_xEnable</code> input.
<code>q_etResult</code>	ET_Result	Provides diagnostic and status information. Only valid if the <code>q_xError</code> output is TRUE.
<code>q_sResultMsg</code>	STRING[80]	Provides additional diagnostic and status information. Only valid if the <code>q_xError</code> output is TRUE.

Part II

Avatar Description

Chapter 3

Avatar Description






Avatar Description









General Description

The TeSys island avatars comprise of three types:

- **System**
Represents the island as a system. This is used for the management and integration of the island.
- **Device**
Represents the modules used in the island, such as a starter or an input/output module, and describes the module role in the island. These avatars are used when modules are applied in a stand-alone manner.
- **Load**
Represents a module operation as it relates to a specific load, such as a forward-reverse motor. When load avatars are used, the island applies the appropriate number and type of modules, along with operating characteristics, to serve the load type. For example, a forward-reverse motor avatar includes a forward and reverse starter, along with pre-programmed forward-reverse operating logic.

List of Avatars

Name		Icon	Description
System Avatar			Enables a single point communication to the island.
Device Avatars	Switch		Establishes or breaks a power line in an electrical circuit.
	Switch Safe Stop Cat. 1&2		Establishes or breaks a power line in an electrical circuit with Safe Operating Stop function compliance for wiring category 1 and category 2.
	Digital I/O		Provides control of two digital outputs and status of four digital inputs.
	Analog I/O		Provides control of one analog output and status of two analog inputs.

Name		Icon	Description
Load Avatars	Power Interface without I/O (measure)		Monitors the current supplied to an external device, such as a solid-state relay, soft starter, or variable speed drive.
	Power Interface with I/O (control)		Monitors the current supplied to an external device and to control the external device, such as a solid-state relay, soft starter, or variable speed drive.
	Motor One Direction		Manages a motor in one direction.
	Motor One Direction Safe Stop 1&2		Manages a motor in one direction, with Safe Operating Stop function compliance for wiring category 1 and category 2.
	Motor Two Directions		Manages a motor in two directions (forward and reverse).
	Motor Two Directions Safe Stop 1&2		Manages a motor in two directions (forward and reverse), with Safe Operating Stop function compliance for wiring category 1 and category 2.
	Motor Y/D One Direction		Manages a wye/delta (star/delta) motor in one direction.
	Motor Y/D Two Directions		Manages a wye/delta (star/delta) motor in two directions (forward and reverse).
	Motor Two Speeds		Manages a two-speed motor.
	Motor Two Speeds Safe Stop 1&2		Manages a two-speed motor, with Safe Operating Stop function compliance for wiring category 1 and category 2.
	Motor Two Speeds Two Directions		Manages a two-speed motor in two directions (forward and reverse).
	Motor Two Speeds Two Directions Safe Stop 1&2		Manages a two-speed motor in two directions (forward and reverse), with Safe Operating Stop function compliance for wiring category 1 and category 2.
	Resistor		Manages a resistive load.
	Power Supply		Manages a power supply.
Transformer		Manages a transformer.	

Part III

Data Unit Types

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
4	Enumerations	35
5	Structures	45

Chapter 4

Enumerations

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
ET_Result	36
ET_SafeStop	42
ET_TripMessage	43
ET_WarningMessage	44

ET_Result

Overview

Type:	Enumeration
Available as of:	V1.0.0.0

Description

The enumeration `ET_Result` contains the possible values that indicate the result of operations executed by the POUs of this library.

Enumeration Elements

Name	Value (UINT)	Description
NoError	0	No error detected.
Done	1	The function block execution was completed successfully.
Working	2	The function block execution is still running.
OrderNotPresent	3	Internal communication interruption.
UnknownResponse	4	Internal communication interruption.
CommandError	5	Internal communication interruption.
BufferFull	6	Internal communication interruption.
CommandAborted	7	Internal status information.
NoParameterInterface	8	Internal interface error detected.
InititalizationFailed	10	The initialization of the library was not successful.
CommunicationError	11	The communication was interrupted.
AvatarNotAvailable	12	The referenced avatar via the input <code>i_sAvatarName</code> is not available in the DTM configuration list.
DeviceNotAvailable	13	The referenced device via the input <code>i_sDeviceName</code> is not available in the DTM configuration list.
IdentStructureEmpty	14	The identification structure provided by the DTM is empty.
AvatarNotSupported	15	The avatar is not supported by the function block.
SystemAvatarNotSupported	16	The system avatar is not supported by the function block.
SystemDeviceNotSupported	17	The system device is not supported by the asset management function blocks.
NoAvatarConfigured	18	No avatar configured.

Name	Value (UINT)	Description
FirmwareNotSupported	19	The firmware of the bus coupler is not supported by this library and an update of the library is required.
ConfigurationInvalid	20	The configuration of the bus coupler and the island configuration in the application is different.

NOTE: This enumeration inherits the enumerations `CommunicationErrorCodes`, `OperationErrorCodes`, and `CIPOperationErrorCodes` from the EtherNet/IP Scanner Library and from the PLCCommunication Library (see *EcoStruxure Machine Expert, Modbus and ASCII Read/Write Functions, PLCCommunication Library Guide*).

CIP (Common Industrial Protocol) Error Codes

The `CIPOperationErrorCodes` enumerated type contains information about operation diagnostics, such as interruptions and detected errors. It contains these values:

Element	Value	Description
CIPConnectionFailure	1025	A connection-related service was unsuccessful along the connection path.
CIPResourceUnavailable	1026	The resources required for the object to perform the requested service are not available.
CIPInvalidParameteValue	1027	To select the correct value for this condition, refer to the status code <code>TO_UINT(ET_ErrorCodeOffsets.CIPError + 16#20)</code> .
CIPPathSegmentError	1028	The path segment identifier or segment syntax was not interpreted by the processing node. Path processing is stopped if an error occurs in the path segment.
CIPPathDestinationUnknown	1029	The path references an object class, instance, or structural element that is not identified or contained in the processing node. Path processing must be stopped if an undefined error occurs in the path target.
CIPPartialTransfer	1030	Only part of the expected data was transferred.
CIPConnectionLost	1031	The messaging connection was interrupted.
CIPServiceNotSupported	1032	The requested service was not implemented or defined for this object class/instance.
CIPInvalidAttributeValue	1033	Invalid attribute data was detected.
CIPAttributeListError	1034	An attribute in the <code>Get_Attribute_List</code> or <code>Set_Attribute_List</code> response has a non-zero status.
CIPAlreadyInRequestedMode	1035	The object is already in the mode/state being requested by the service.
CIPObjectStateConflict	1036	The object cannot perform the requested service in the current mode/state.

Element	Value	Description
CIPObjectAlreadyExists	1037	The requested instance of the object to be created already exists.
CIPAttributeNotSettable	1038	A request to modify a nonmodifiable attribute was received.
CIPPrivilegeViolation	1039	A permission/privilege verification was unsuccessful.
CIPDeviceStateConflict	1040	The present mode/state of the device prohibits the execution of the requested service.
CIPReplyDataTooLarge	1041	The data to be transmitted in the response buffer is larger than the allocated response buffer.
CIPFragmentationPrimitiveValue	1042	The service specifies an operation that fragments a primitive data value that means half a REAL data type.
CIPNotEnoughData	1043	The service did not supply enough data to perform the specified operation.
CIPAttributeNotSupported	1044	The attribute specified in the request is not supported.
CIPTooMuchData	1045	The service supplied more data than was expected.
CIPObjectDoesNotExist	1046	The specified object does not exist in the device.
CIPFragmentationNotInProgress	1047	The fragmentation sequence for this service is not active for this data.
CIPNoStoredAttributeData	1048	The attribute data of this object was not stored before the requested service.
CIPStoreOperationFailure	1049	The attribute data of this object was not stored due to a detected error during the attempt.
CIPRoutingFailureRequestPacketTooLarge	1050	The service request packet was too large for transmission on a network. The routing device was forced to stop the service.
CIPRoutingFailureResponsePacketTooLarge	1051	The service response packet was too large for transmission on a network in the path from the destination. The routing device was forced to stop the service.
CIPMissingAttributeListEntryData	1052	The service did not provide an attribute from the attribute list required by the service to perform the requested behavior.
CIPInvalidAttributeValueList	1053	The service returns the list of attributes that contains status information about invalid attributes.
CIPEmbeddedServiceError	1054	An embedded service resulted in a detected error.
CIPVendorSpecificError	1055	A vendor-specific error was detected. The additional code field of the error response specifies the detected error. This general error code must only be used if none of the error codes displayed in this table or in an object class definition accurately represents the detected error.
CIPInvalidParameter	1056	A parameter associated with the request was invalid. This code is used if a parameter does not comply with the requirements of this specification and/or the requirements defined in an application object specification.

Element	Value	Description
CIPWriteOnceValueAlreadyWritten	1057	An attempt was made to write to a write-once medium (for example, WORM drive, PROM) that has already been written, or to modify a value that cannot be modified once established.
CIPInvalidReplyReceived	1058	An invalid response is received (for example, reply service code does not correspond to the request service code), or the response message is shorter than the minimum expected response size. This status code can be used for other purposes of invalid responses.
CIPBufferOverflow	1059	The message received is larger than the receiving buffer can handle. The entire message was discarded.
CIPMessageFormatError	1060	The format of the received message is not supported by the server.
CIPKeyFailureInPath	1061	The key segment that was included as the first segment in the path does not correspond to the target module. The object-specific status must specify which part of the key check was unsuccessful.
CIPPathSizeInvalid	1062	The size of the path sent with the service request is either not large enough to allow the request to be forwarded to an object, or too much routing data has been included.
CIPUnexpectedAttributeInList	1063	An attempt was made to set an attribute that cannot be set at this time.
CIPInvalidMemberID	1064	The member ID specified in the request does not exist in the specified class/instance/attribute.
CIPMemberNotSettable	1065	A request to modify a non-modifiable member was received.
CIPGroup2OnlyServerFailure	1066	Not used
CIPUnknownModbusError	1067	A CIP to Modbus translator has received an undefined Modbus exception code.
CIPAttributeNotGettable	1068	A request to read a non-readable attribute was received.
CIPInstanceNotDeletable	1069	The requested object instance cannot be deleted.
CIPServiceNotSupportedForAppPath	1070	The object supports the service, but not for the designated application path (for example, attribute).
CIPTimeout	1279	No response from the target.

Communication Error Codes

The `CommunicationErrorCodes` enumerated type contains information about communication diagnostics, such as interruptions and detected errors. It contains these values:

Element	Value	Description
<code>CommunicationOK</code>	256	The exchange is valid.

Element	Value	Description
TimedOut	257	The exchange stopped when the timeout expired.
Canceled	258	The exchange was stopped by a user request (the <code>Abort</code> command).
BadAddress	259	The address format is incorrect.
BadRemoteAddr	260	The remote address is incorrect.
BadMgtTable	261	The management table format is incorrect.
BadParameters	262	Specific parameters are incorrect.
ProblemSendingRq	263	There was an issue while sending the request to the destination.
RecvBufferTooSmall	265	The reception buffer size is too small.
SendBufferTooSmall	266	The send buffer size is too small.
SystemResourceMissingCommCode	267	A system resource is unavailable.
BadTransactionNb	268	The transaction number is incorrect.
BadLengthCommCode	270	The length is incorrect.
ProtocolSpecificError	510	The operation error code contains a protocol-specific code.
Refused	511	The message was refused.

Operation Error Codes

The `OperationErrorCodes` enumerated type contains information about operation diagnostics, such as interruptions and detected errors. It contains these values:

Element	Value	Description
NotProcessed_or_TargetResourceMissing	513	The request was not processed or the target system resource is unavailable.
BadResponse	514	The received response is incorrect.
BadLengthOperCode	517	The length is incorrect.
CommChannelErr	518	The communication channel is associated with a detected error.
BadAddr	519	The address is incorrect.
SystemResourceMissingOperCode	523	A system resource is unavailable.
TargetCommInactive	524	A target communication function is not active.
TargetMissing	525	The target is unavailable.
ChannelNotConfigured	527	The channel is not configured.

Used By

- AnalogIO
- DigitalIO
- Energy
- EnhancedStatus
- MotorOneDirection
- MotorOneDirectionSafeStopCatland2
- MotorTwoDirections
- MotorTwoDirectionsSafeStopCatland2
- MotorTwoSpeeds
- MotorTwoSpeedsSafeStopCatland2
- MotorTwoSpeedsTwoDirections
- MotorTwoSpeedsTwoDirectionsSafeStopCatland2
- MotorYDOneDirection
- MotorYDTwoDirections
- Power
- PowerInterface
- PowerInterfaceIO
- PowerSupply
- ProductData
- ReadDiagnostics
- ReadSystemDiagnostics
- ReadTripCounters
- ReadTripRegisters
- ReadWarningCounters
- Resistor
- Switch
- SwitchSafeStopCatland2
- SystemControl
- SystemEnergyBasic
- SystemMaintenanceData
- SystemPowerBasic
- SystemProductData
- SystemVoltageBasic
- SystemVoltageEnhanced
- Transformer

ET_SafeStop

Overview

Type:	Enumeration
Available as of:	V1.0.0.0

Description

The enumeration `ET_SafeStop` describes the status of the commands to the SIL (Safety Interface Module) group.

Enumeration Elements

Name	Value (USINT)	Description
<code>SILNotPresent</code>	0	The SIL group is not available in the system configuration.
<code>AvatarDeviceFault</code>	1	The SIL group is affected by a detected avatar device error.
<code>SILStartersNotOpen</code>	2	The safe stop command was received and the SIL Starters are not open yet.
<code>AllSILStartersOpen</code>	3	The safe stop command is successfully issued and the SIL Starters are open.
<code>CommandOnlyOneTerminal</code>	4	Safe command issued to only one terminal (jumper or wiring is causing an issue), but safe SIL Starters did successfully open.
<code>NormalOperation</code>	5	Normal operation, safe SIL Starters can be open or closed.

Used By

- `ReadSystemDiagnostics`

ET_TripMessage

Overview

Type:	Enumeration
Available as of:	V1.0.0.0

Description

The enumeration `ET_TripMessage` contains a value corresponding to the type of avatar protection trip detected.

Enumeration Elements

Name	Value (UINT)	Description
NoTrip	0	No trip has been detected.
ThermalOverload	1	The avatar thermal capacity has increased above 100%.
MotorOverheat	2	The motor temperature has increased above the motor overheat trip level.
Jam	3	The avatar has detected the conditions to cause a jam trip event.
Undercurrent	4	The avatar has detected the conditions to cause an undercurrent trip event.
LongStart	5	The avatar has detected the conditions to cause a long start trip event.
Overcurrent	6	The avatar has detected the conditions to cause an overcurrent trip event.
Stall	7	The avatar has detected the conditions to cause a stall trip event.
GroundCurrent	8	The avatar has detected the conditions to cause a ground current trip event.
CurrentPhaseReversal	9	The avatar has detected the conditions to cause a current phase reversal trip event.
PhaseConfiguration	10	The avatar has detected the conditions to cause a phase configuration trip event.
CurrentPhaseUnbalance	11	The avatar has detected the conditions to cause a current phase unbalance trip event.
CurrentPhaseLoss	12	The avatar has detected the conditions to cause a current phase loss trip event.

Used By

- EnhancedStatus

ET_WarningMessage

Overview

Type:	Enumeration
Available as of:	V1.0.0.0

Description

The enumeration `ET_WarningMessage` contains a value corresponding to the type of avatar protection advisory condition detected.

Enumeration Elements

Name	Value (UINT)	Description
NoWarning	0	No advisory condition has been detected.
ThermalOverload	1	The avatar thermal capacity has increased above the thermal overload level.
MotorOverheat	2	The motor temperature has increased above the motor overheat level.
Jam	3	The avatar has detected the conditions to cause a jam event.
Undercurrent	4	The avatar has detected the conditions to cause an undercurrent event.
Overcurrent	5	The avatar has detected the conditions to cause an overcurrent event.
GroundCurrent	6	The avatar has detected the conditions to cause a ground current event.
CurrentPhaseUnbalance	7	The avatar has detected the conditions to cause a current phase unbalance event.

Used By

- `EnhancedStatus`

Chapter 5

Structures

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
ST_MinorFltRegister	46
ST_TripRegister	47
ST_VoltageRegister	48

ST_MinorFltRegister

Overview

Type:	Structure
Available as of:	V1.0.0.0
Inherits from:	-

Description

The structure `ST_MinorFltRegister` contains date, time, and an identifier for a detected minor error.

Structure Elements

Name	Data type	Description
<code>RTS_SYSTIMEDATE</code>	<code>SysTimeRtc.RTS_SYSTIMEDATE</code>	Structure of date and time from the <code>SysTimeRtc</code> library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
<code>uiFaultId</code>	<code>UINT</code>	Identifier of the detected error. Description of the detected errors: <ul style="list-style-type: none"> ● 0000: No error detected ● 0001: No module in the island ● 0002: Number of physical devices detected in the island is beyond the limit allowed ● 0003: Modules mismatch ● 0004: Island control power supply voltage fluctuation

Used By

- `ReadDiagnostics`
- `SystemPowerBasic`
- `SystemVoltageBasic`
- `Power`

ST_TripRegister

Overview

Type:	Structure
Available as of:	V1.0.0.0
Inherits from:	-

Description

The structure `ST_TripRegister` contains date, time, and an identifier for the most recent detected trip event.

Structure Elements

Name	Data type	Description
RTS_SYSTIMEDATE	SysTimeRtc.RTS_SYSTIMEDATE	Structure of date and time from the SysTimeRtc library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
uiFaultId	UINT	Identifier of the detected trip event. Description of the detected trip events: <ul style="list-style-type: none"> ● 0000: No error detected ● 0001: Thermal overload ● 0002: Motor overheat ● 0003: Jam ● 0004: Undercurrent ● 0005: Long start ● 0006: Overcurrent ● 0007: Stall ● 0008: Ground current ● 0009: Current phase reversal ● 0010: Phase configuration ● 0011: Current phase unbalance ● 0012: Current phase interruption

Used By

- ReadTripRegisters

ST_VoltageRegister

Overview

Type:	Structure
Available as of:	V1.0.0.0
Inherits from:	-

Description

The structure `ST_VoltageRegister` contains the maximum voltage magnitude for a voltage dip or swell and the start and stop time stamp of the voltage dip or swell.

Structure Elements

Name	Data type	Description
<code>uiMagnitude</code>	UINT	Maximum voltage magnitude for a voltage dip or swell. (Unit: V)
<code>stStartDate</code>	SysTimeRtc.RTS_SY STIMEDATE	Start time stamp of the voltage dip or swell. Refer to SysTimeRtc library (<i>see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
<code>stStopDate</code>	SysTimeRtc.RTS_SY STIMEDATE	Stop time stamp of the voltage dip or swell. Refer to SysTimeRtc library (<i>see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).

Used By

- `SystemVoltageEnhanced`

Part IV

Global Variables

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
6	Global Constants List	51
7	Global Parameter List	53

Chapter 6

Global Constants List

Global Constants List (GCL)

Overview

Type:	Global constants
Available as of:	V1.0.0.0

Description

The global constants list contains the global constants of the TeSys island library.

Global Constants

Variable	Data type	Value	Description
Gc_sLibraryVersion	STRING[80]	Vx.x.x.0 ⁽¹⁾	Library version
(1) This value varies to indicate the version of the library.			

Chapter 7

Global Parameter List

Global Parameter List (GPL)

Overview

Type:	Global parameters
Available as of:	V1.0.0.0

Description

The global parameter list (GPL) contains global constants which are used by certain components of this library. The parameters can be edited individually for each application where the library is used. The modification must be done within the **Library Manager** of the project where the library is referenced.

Global Parameters

Variable	Data type	Default value	Description
Gc_uiMaxNumberOfAvatars	UINT	20	The value indicates the maximum number supported avatars.
Gc_uiLengthOfAvatarName	UINT	20	The value indicates the number of characters of the avatar name.
Gc_uiLengthOfDeviceName	UINT	20	The value indicates the number of characters of the device name.

Part V

Program Organization Units (POU)

Chapter 8

Function Block Description

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
8.1	General Function Block Description	58
8.2	System Control Function Blocks	60
8.3	System Diagnostic Function Blocks	62
8.4	System Energy Function Blocks	65
8.5	System Asset Management Function Blocks	75
8.6	Control Function Blocks	79
8.7	Energy Function Blocks	129
8.8	Diagnostic Function Blocks	135
8.9	Asset Management Function Blocks	144

Section 8.1

General Function Block Description

Avatar Function Block Description

Avatar Function Blocks - General Description

Avatars are a digital representation of the physical modules on the TeSys island.

The function blocks are fieldbus independent, only the `TeSysIslandRef` (TeSys island device in the devices tree) is fieldbus-dependent. One specific `TeSysIslandRef` is available for every supported fieldbus. The type of this reference is `FB_TeSys_island` and an instance of this type is created automatically, when a TeSys island device is added to the devices tree.

The avatar is referenced via the input `i_sAvatarName`. The function block verifies if the referenced avatar input type is supported by the function block while executing. If not, the function block execution is canceled and the diagnostic message `AvatarNotSupported` is displayed.

The function blocks have no logic operations and are not modifying or interpreting the avatar data. They copy the values of their inputs into the cyclic output data frame and copy the data of the cyclic input frame to their outputs. If the function block requires acyclic data exchange, the read and write requests are managed by the function block.

The system provides one acyclic connection per TeSys island bus coupler so that the acyclic communication requests must be handled sequentially. A new request can only be sent if the response to the previous request was received. If an error occurs during the execution, the function block stops and provides the error information. You cannot stop the function block by the application (for example, cancel input).

Some function blocks are providing inputs to reset or preset parameters of the function block. When executing the function block and one of these inputs is `TRUE`, the update of the outputs is delayed until the reset or preset command is executed in the avatar.

There are two basic types of function blocks:

- System avatar function blocks
- Standard avatar function blocks

System Avatars Function Blocks

The system avatar is unique in the TeSys island and supported by specific function block implementations, indicated by the string `System` in the function block name. These function blocks do not have the input `i_sAvatarName`.

Standard Avatar Function Blocks

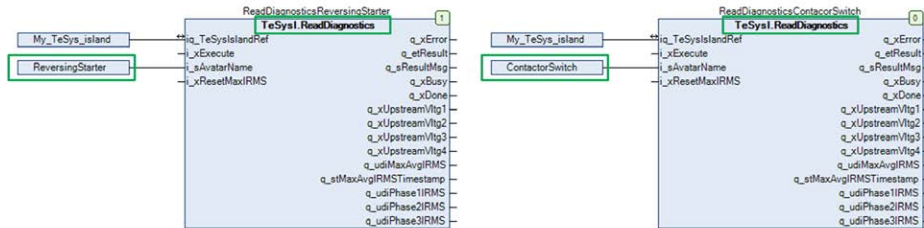
The standard avatars are supported by two types of function blocks:

- Control function blocks
- Read/write function blocks

Each control avatar is supported by its own function block implementation, which can be instantiated for multiple usage of the same avatar type in one TeSys island. Create one function block instance for each avatar in your application and link the input `i_sAvatarName` to the avatar.

There is only one implementation for the read/write function blocks `Asset`, `Diagnostic`, and `Energy` avatars available because nearly all avatars are supporting the same asset, diagnostic, and energy data. Exceptions are described in the dedicated function block descriptions. Executing the function block for a not supported avatar stops the function block with an error. For multiple usage of the same function block with different avatars, you have to create an instance of the function block for each avatar.

Example for two instances of the diagnostic function block used for two different avatars:



Section 8.2

System Control Function Blocks

SystemControl - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SystemControl` returns the status of the system avatar.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code> <i>(see page 58)</i>	Reference to the TeSys island device.
<code>i_xExecute</code>	<code>BOOL</code>	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_xResetTrip</code>	<code>BOOL</code>	Triggers the trip reset for all configured avatars.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xOperational	BOOL	If this output is set to TRUE, the system avatar is in Operational mode. For detailed information on machine states, refer to TeSys island Operating Guide (<i>see page 14</i>).
q_xDegradedMode	BOOL	If this output is set to TRUE, the system avatar is in Degraded mode.
q_xMinorFlt	BOOL	If this output is set to TRUE, the system avatar is in Minor Fault mode.
q_xPreoperational	BOOL	If this output is set to TRUE, the system avatar is in Preoperational mode.
q_xForceMode	BOOL	If this output is set to TRUE, the system avatar is in Force mode.
q_xTestMode	BOOL	If this output is set to TRUE, the system avatar is in Test mode.
q_udiIP4Address	STRING[15]	IP address of the bus coupler (read from the TeSys island device in EcoStruxure Machine Expert software during initialization).

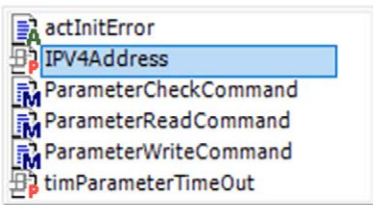
NOTE: The IP address is also provided as a property of the TeSys island device. The IP address is updated during the initialization of the application.

Example:

```

1  IPAddress := TeSys_island.
2
3
4
5
6
7
8

```



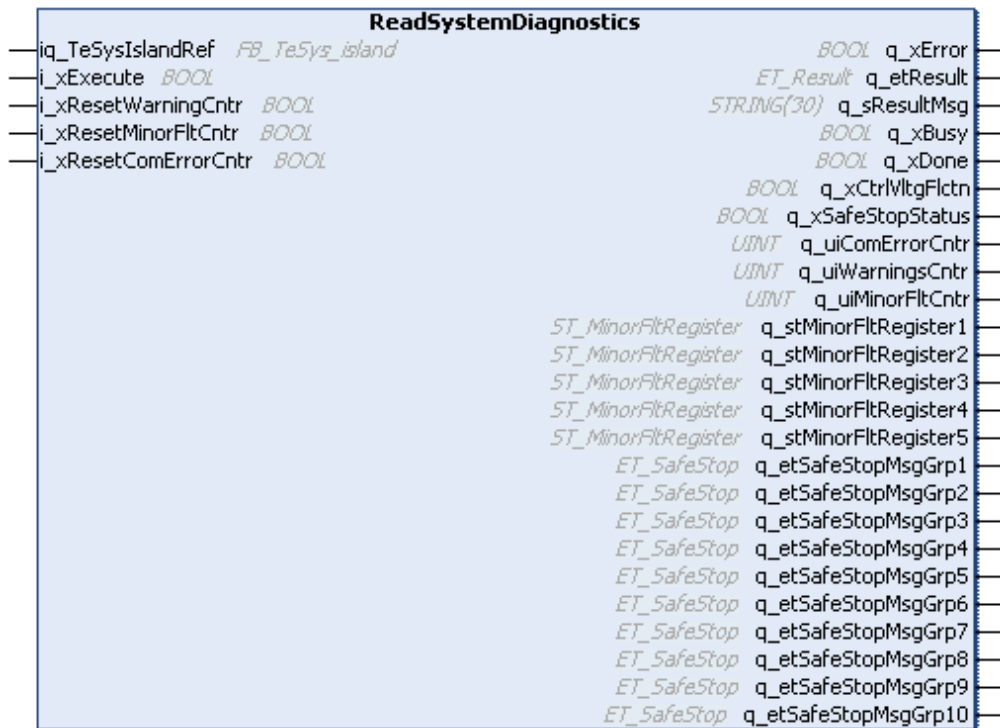
Section 8.3

System Diagnostic Function Blocks

ReadSystemDiagnostics - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `ReadSystemDiagnostics` returns and resets the diagnostic information of the system avatar.

Interface

Input	Data type	Description
<code>i_q_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_xResetWarningCntr</code>	BOOL	If this input is set to TRUE, the counter of detected advisories for the system is set to 0.
<code>i_xResetMinorFltCntr</code>	BOOL	If this input is set to TRUE, the counter of detected minor errors for the system is set to 0.
<code>i_xResetComErrorCntr</code>	BOOL	If this input is set to TRUE, the counter of detected errors for the fieldbus communication is set to 0.

Output	Data type	Description
<code>q_xError</code>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
<code>q_etResult</code>	<code>ET_Result</code> (see page 36)	Provides diagnostic and status information as a numeric value.
<code>q_sResultMsg</code>	STRING[30]	Provides additional diagnostic and status information as a text message.
<code>q_xBusy</code>	BOOL	If this output is set to TRUE, the function block execution is in progress.
<code>q_xDone</code>	BOOL	If this output is set to TRUE, the execution has been completed successfully.
<code>q_xCtrlVltgFlctn</code>	BOOL	If this output is set to TRUE, a control voltage fluctuation is detected.
<code>q_xSafeStopStatus</code>	BOOL	Status of Safe Stop 0 function. If this output is set to FALSE, no safety-related group has received a Safe Stop command.
<code>q_uiComErrorCntr</code>	UINT	Number of detected errors for the fieldbus communication.
<code>q_uiWarningsCntr</code>	UINT	Number of detected warnings for the system.
<code>q_uiMinorFltCntr</code>	UINT	Number of detected minor faults for the system.

Function Block Description

Output	Data type	Description
q_stMinorFltRegister1	ST_MinorFltRegister <i>(see page 46)</i>	Information on a detected minor fault. q_stMinorFltRegister1 = most recent
q_stMinorFltRegister2	ST_MinorFltRegister <i>(see page 46)</i>	Information on a detected minor fault.
...
q_stMinorFltRegister5	ST_MinorFltRegister	Information on a detected minor fault.
q_etSafeStopMsgGrp1	ET_SafeStop <i>(see page 42)</i>	SIL group 1: Information on the Safe Stop 0 function.
...
q_etSafeStopMsgGrp10	ET_SafeStop <i>(see page 42)</i>	SIL group 10: Information on the Safe Stop 0 function.

Section 8.4

System Energy Function Blocks

What Is in This Section?

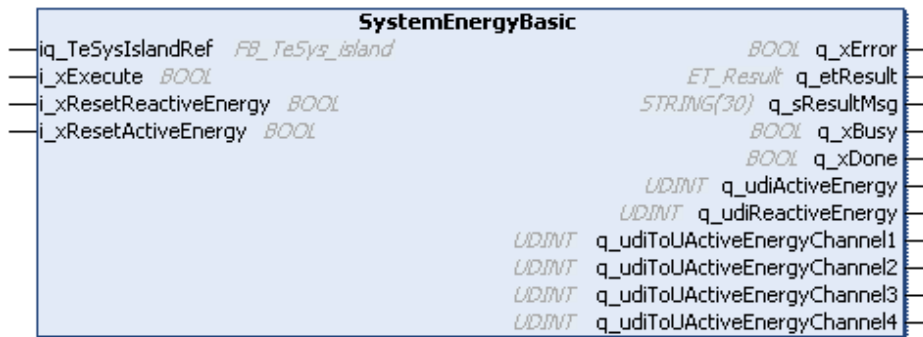
This section contains the following topics:

Topic	Page
SystemEnergyBasic - Functional Description	66
SystemPowerBasic - Functional Description	68
SystemVoltageBasic - Functional Description	71
SystemVoltageEnhanced - Functional Description	73

SystemEnergyBasic - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SystemEnergyBasic` returns the energy information of the system avatar. Resets the energy registers of the system avatar. Sets the energy preset values of the system avatar.

Interface

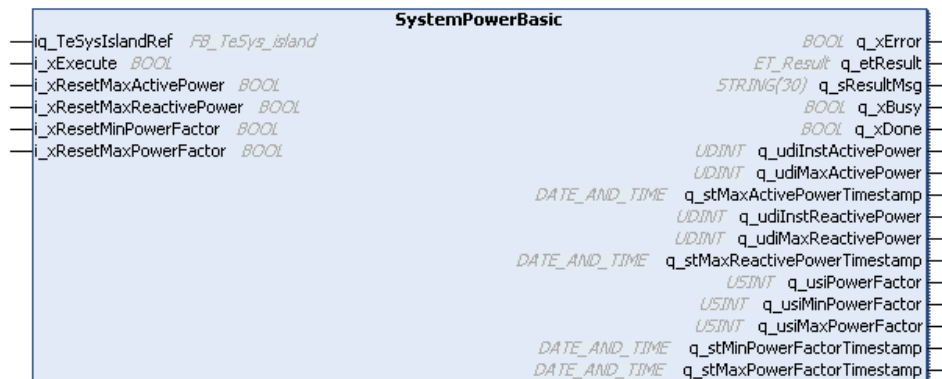
Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_xResetReactiveEnergy</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiReactiveEnergy</code> is reset.
<code>i_xResetActiveEnergy</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiActiveEnergy</code> is reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_udiTotalActiveEnergy	UDINT	Total active energy value for all avatars in the system with Energy Monitoring Enabled . (Unit: Watt-hours)
q_udiTotalReactiveEnergy	UDINT	Total reactive energy value for all avatars in the system with Energy Monitoring Enabled . (Unit: VAR-hours)
q_udiToUActiveEnergyChannel1	UDINT	Channel 1: ToU (Time of Use) total active energy value. NOTE: Not used in this version of function block (will return all 0 s).
...
q_udiToUActiveEnergyChannel4	UDINT	Channel 4: ToU (Time of Use) total active energy value. NOTE: Not used in this version of function block (will return all 0 s).

SystemPowerBasic - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SystemPowerBasic` returns power information and resets the power registers of the system avatar.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input i_xExecute</i> (see page 27).
<code>i_xResetMaxActivePower</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiMaxActivePower</code> and the associated time stamp <code>q_stMaxActivePowerTimestamp</code> are reset.
<code>i_xResetMaxReactivePower</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiMaxReactivePower</code> and the associated time stamp <code>q_stMaxReactivePowerTimestamp</code> are reset.

Input	Data type	Description
i_xResetMinPowerFactor	BOOL	If this input is set to TRUE: <ul style="list-style-type: none"> The true value of the parameter <code>q_usiMinPowerFactor</code> is reset to 1. The associated time stamp <code>q_stMinPowerFactorTimestamp</code> is reset.
i_xResetMaxPowerFactor	BOOL	If this input is set to TRUE: <ul style="list-style-type: none"> The true value of the parameter <code>q_usiMaxPowerFactor</code> is reset to 0. The associated time stamp <code>q_stMaxPowerFactorTimestamp</code> is reset.

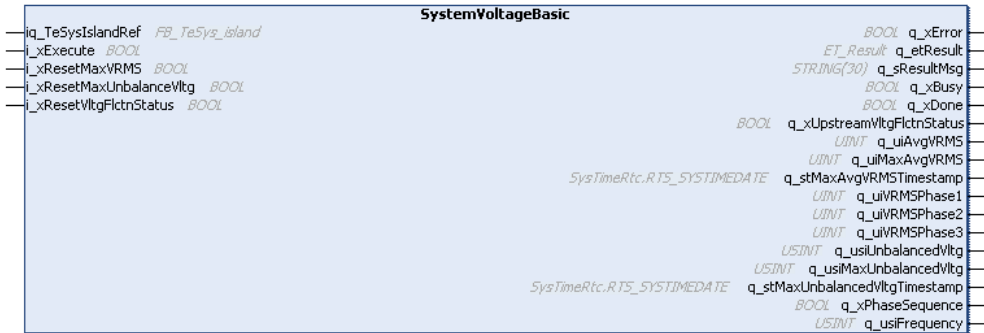
Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_udiInstActivePower	UDINT	Total active power for the avatar. (Unit: W)
q_udiMaxActivePower	UDINT	Maximum value of the active power for the avatar. (Unit: W)
q_stMaxActivePowerTimestamp	SysTimeRtc.RTS_SY STIMEDATE	Date and time when the maximum active power value was recorded. Refer to the SysTimeRtc library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
q_udiInstReactivePower	UDINT	Total reactive power for the avatar. (Unit: VAR)
q_udiMaxReactivePower	UDINT	Maximum value of the reactive power for the avatar. (Unit: VAR)

Output	Data type	Description
q_stMaxReactivePowerTimestamp	SysTimeRtc.RTS_SY STIMEDATE	Date and time when the maximum reactive power value was recorded. Refer to the SysTimeRtc library (<i>see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
q_usiPowerFactor	USINT	True power factor value.
q_usiMinPowerFactor	USINT	True minimum power factor value.
q_usiMaxPowerFactor	USINT	True maximum power factor value.
q_stMinPowerFactorTimestamp	SysTimeRtc.RTS_SY STIMEDATE	Date and time when the minimum power factor value was recorded. Refer to the SysTimeRtc library (<i>see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
q_stMaxPowerFactorTimestamp	SysTimeRtc.RTS_SY STIMEDATE	Date and time when the maximum power factor value was recorded. Refer to the SysTimeRtc library (<i>see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).

SystemVoltageBasic - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SystemVoltageBasic` returns and resets the voltage registers of the system avatar.

Interface

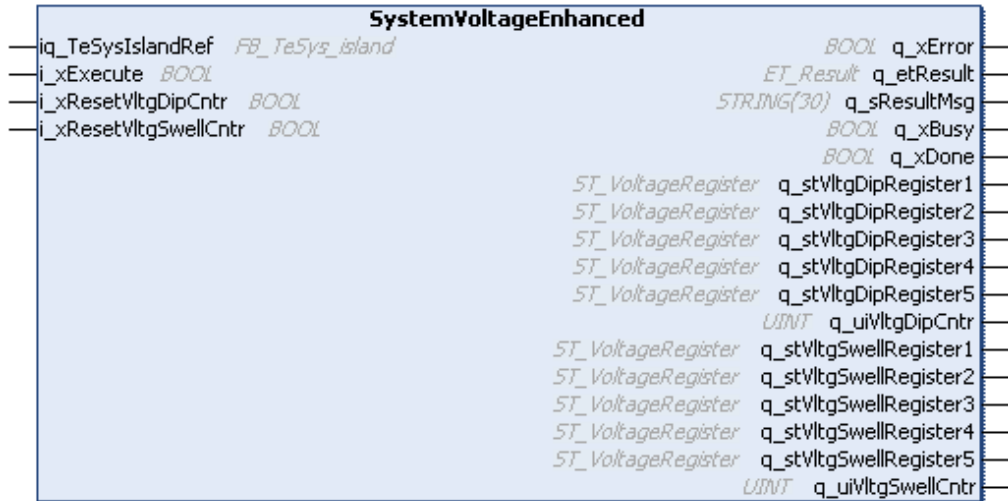
Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input i_xExecute</i> (see page 27).
<code>i_xResetMaxVRMS</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiResetMaxVRMS</code> is reset.
<code>i_xResetMaxUnbalanceVltg</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiResetMaxUnbalanceVltg</code> is reset.
<code>i_xResetVltgFlctnStatus</code>	BOOL	If this input is set to TRUE, the value of the parameter <code>q_udiResetVltgFlctnStatus</code> is reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_xUpstreamVltgFlctnStatus	BOOL	If this output is set to TRUE, a voltage dip or swell is detected. Can be reset with i_xResetVltgFlctnStatus.
q_uiAvgVRMS	UINT	Average IRMS voltage on three phases (Unit: mV).
q_uiMaxAvgVRMS	UINT	Maximum voltage the system measured (Unit: mV).
q_stMaxAvgVRMSTimestamp	SysTimeRtc.RTS_SYSTI MEDATE	Date and time when maximum average voltage value was recorded. Refer to the SysTimeRtc library (see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide).
q_uiVRMSPhase1	UINT	Average IRMS voltage between L1 and neutral. (Unit: V)
q_uiVRMSPhase2	UINT	Average IRMS voltage between L2 and neutral. (Unit: V)
q_uiVRMSPhase3	UINT	Average IRMS voltage between L3 and neutral. (Unit: V)
q_usiUnbalancedVltg	USINT	Unbalance voltage in percent (%).
q_usiMaxUnbalancedVltg	USINT	Maximum unbalance voltage in percent (%).
q_stMaxUnbalancedVltgTimestamp	SysTimeRtc.RTS_SYSTI MEDATE	Date and time of maximum unbalance voltage. Refer to the SysTimeRtc library (see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide).
q_xPhaseSequence	BOOL	If this output is set to TRUE, the phase order is ACB (FALSE = phase order ABC).
q_usiFrequency	USINT	Main power voltage frequency (line frequency as measured on phase 1). (Unit: Hertz)

SystemVoltageEnhanced - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SystemVoltageEnhanced` returns and resets the enhanced voltage registers of the system avatar.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xExecute</code>	<code>BOOL</code>	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_xResetVltgDipCntr</code>	<code>BOOL</code>	If this input is set to TRUE, the value of the parameter <code>q_udiResetVltgDipCntr</code> is reset.

Function Block Description

Input	Data type	Description
i_xResetVltgSwellCntr	BOOL	If this input is set to TRUE, the value of the parameter q_udiResetVltgSwellCntr is reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_stVltgDipRegister1	ST_VoltageRegister <i>(see page 48)</i>	Maximum voltage magnitude for a voltage dip. q_stVltgDipRegister1 = most recent (Unit: V)
q_stVltgDipRegister2		Maximum voltage magnitude for a voltage dip.
...		
q_stVltgDipRegister5		
q_uiVltgDipCntr		UINT
q_stVltgSwellRegister1	ST_VoltageRegister <i>(see page 48)</i>	Maximum voltage magnitude for a voltage swell (Units of Volts). q_stVltgSwellRegister1 = most recent
q_stVltgSwellRegister2		Maximum voltage magnitude for a voltage swell.
...		
q_stVltgSwellRegister5		
q_uiVltgSwellCntr		UINT

Section 8.5

System Asset Management Function Blocks

What Is in This Section?

This section contains the following topics:

Topic	Page
SystemMaintenanceData - Functional Description	76
SystemProductData - Functional Description	77

SystemMaintenanceData - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SystemMaintenanceData` returns the maintenance information of the system device.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).

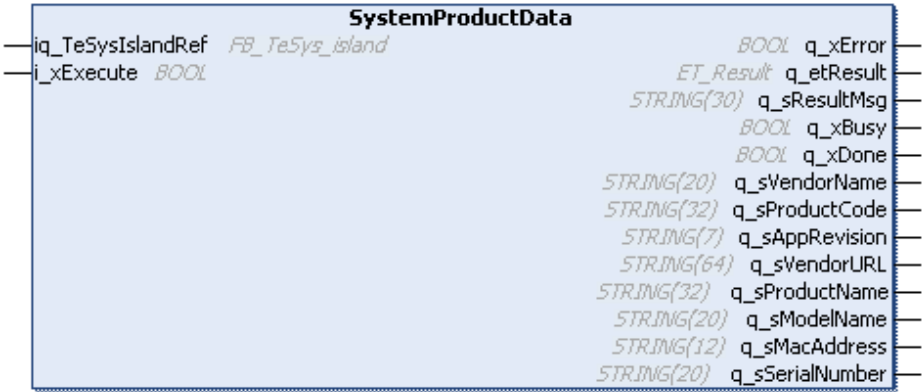
Output	Data type	Description
<code>q_xError</code>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
<code>q_etResult</code>	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
<code>q_sResultMsg</code>	STRING[30]	Provides additional diagnostic and status information as a text message.
<code>q_xBusy</code>	BOOL	If this output is set to TRUE, the function block execution is in progress.
<code>q_xDone</code>	BOOL	If this output is set to TRUE, the execution has been completed successfully.

Output	Data type	Description
q_udiTimeModuleOn	UDINT	This register indicates the time that the module has been powered on in its lifetime (Unit: h).
q_uiFaultCntr	UINT	This register indicates number of times this module has experienced a detected device error (Unit: h). This value does not include detected device errors which prevent the saving or corruption of the non-volatile memory.

SystemProductData - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block SystemProductData returns the product-specific information of the system device.

Interface

Input	Data type	Description
i_q_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xExecute	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs q_xDone, q_xError, q_etResult, and q_etResultMsg are reset with the falling edge of i_xExecute. Refer to <i>Behavior of Function Blocks with the Input i_xExecute (see page 27)</i> .

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (<i>see page 36</i>)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_sVendorName	STRING[20]	Name of the vendor.
q_sProductCode	STRING[32]	Name of the product.
q_sAppRevision	STRING[7]	Revision number of the application firmware presently in the device.
q_sVendorURL	STRING[30]	URL of the vendor.
q_sProductName	STRING[32]	Name of the product.
q_sModelName	STRING[20]	Name of the model.
q_sMacAddress	STRING[12]	MAC address of the device.
q_sSerialNumber	STRING[20]	Serial number.

Section 8.6

Control Function Blocks

What Is in This Section?

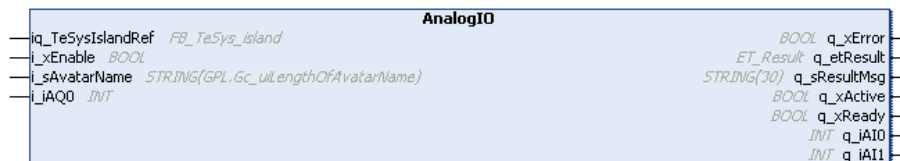
This section contains the following topics:

Topic	Page
AnalogIO - Functional Description	80
DigitalIO - Functional Description	82
EnhancedStatus - Functional Description	84
MotorOneDirection - Functional Description	86
MotorOneDirectionSafeStopWCat1and2 - Functional Description	88
MotorTwoDirections - Functional Description	90
MotorTwoDirectionsSafeStopWCat1and2 - Functional Description	92
MotorTwoSpeeds - Functional Description	95
MotorTwoSpeedsSafeStopWCat1and2 - Functional Description	98
MotorTwoSpeedsTwoDirections - Functional Description	101
MotorTwoSpeedsTwoDirectionsSafeStopWCat1and2 - Functional Description	105
MotorYDOneDirection - Functional Description	108
MotorYDTwoDirections - Functional Description	111
PowerInterface - Functional Description	114
PowerInterfaceIO - Functional Description	116
PowerSupply - Functional Description	119
Resistor - Functional Description	121
Switch - Functional Description	123
SwitchSafeStopWCat1and2 - Functional Description	125
Transformer - Functional Description	127

AnalogIO - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block AnalogIO provides information about the analog I/O avatar with two inputs and one output.

Interface

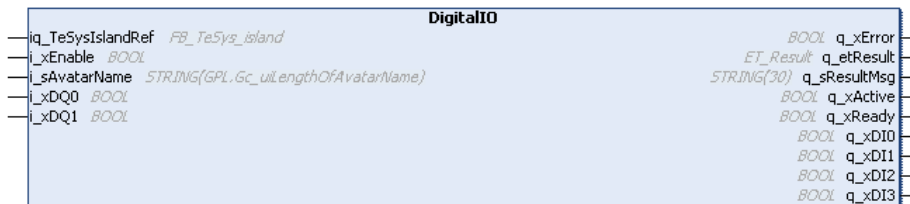
Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input i_xEnable starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as i_xEnable is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input i_xEnable stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable</i> (see page 28).
i_sAvatarName	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_iAQ0	INT	Value to be written to analog output 0. Unit and scaling depends on the analog output type configured. <ul style="list-style-type: none"> • Type 0 (Unit: mV) • Type 1 (Unit: mV) • Type 2 (Unit: μA) • Type 3 (Unit: μA)

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (<i>see page 36</i>)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_iAI0	INT	Indicates the value read from the analog input 0. Unit and scaling depends on the analog output type configured. <ul style="list-style-type: none"> ● Type 0 to 12 (Unit: 0.1 deg C) ● Type 13 (Unit: mV) ● Type 14 (Unit: mV) ● Type 15 (Unit: μA) ● Type 16 (Unit: μA)
q_iAI1	INT	Indicates the value read from the analog input 1.

DigitalIO - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block DigitalIO provides information about the digital I/O avatar with four inputs and two outputs.

Interface

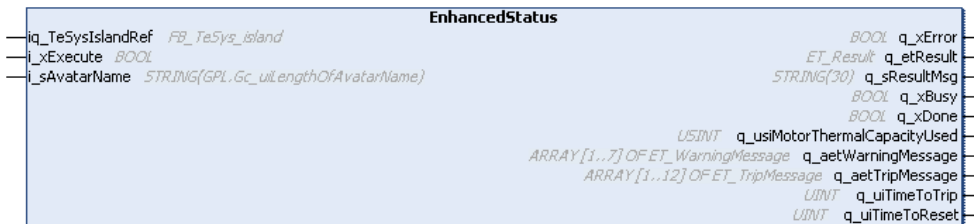
Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input i_xEnable starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as i_xEnable is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input i_xEnable stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable</i> (see page 28).
i_sAvatarName	STRING(GPL_Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xDQ0	BOOL	If this input is set to TRUE, the digital output 0 is set to TRUE.
i_xDQ1	BOOL	If this input is set to TRUE, the digital output 1 is set to TRUE.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (<i>see page 36</i>)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xDI0	BOOL	If this output is set to TRUE, the digital input 0 of the Digital I/O avatar is set to TRUE.
q_xDI1	BOOL	If this output is set to TRUE, the digital input 1 of the Digital I/O avatar is set to TRUE.
q_xDI2	BOOL	If this output is set to TRUE, the digital input 2 of the Digital I/O avatar is set to TRUE.
q_xDI3	BOOL	If this output is set to TRUE, the digital input 3 of the Digital I/O avatar is set to TRUE.

EnhancedStatus - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block EnhancedStatus provides thermal information of the motor.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input i_xExecute (see page 27)</i> .
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.

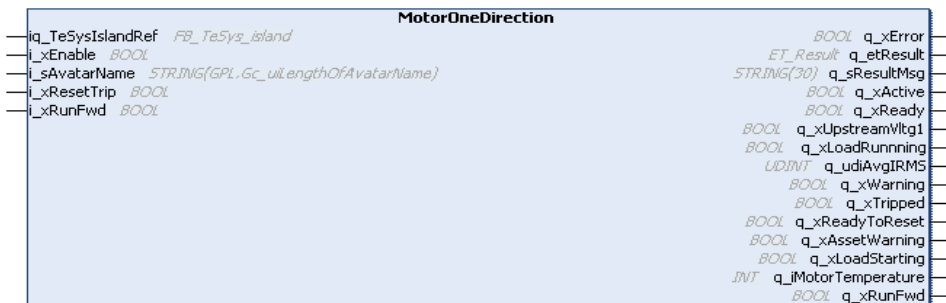
Output	Data type	Description
<code>q_xDone</code>	BOOL	If this output is set to TRUE, the execution has been completed successfully.
<code>q_xBusy</code>	BOOL	If this output is set to TRUE, the function block execution is in progress.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_usiThermalCapacity	USINT	Provides the percentage (%) of the thermal capacity of the motor which has been used.
q_aetWarningMsg	ARRAY OF ET_WarningMessage <i>(see page 44)</i>	A list of warning messages based on the active protection functions.
q_aetTripMsg	ARRAY OF ET_TripMessage <i>(see page 43)</i>	A list of trip messages based on the active protection functions.
q_uiTimeToTrip	UINT	Estimated time before a thermal overload trip will occur given the current conditions. (Unit: s)
q_uiTimeToReset	UINT	Estimated time to wait before being able to reset a thermal overload trip. (Unit: s)

MotorOneDirection - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block MotorOneDirection is used to manage a motor in one direction.

Interface

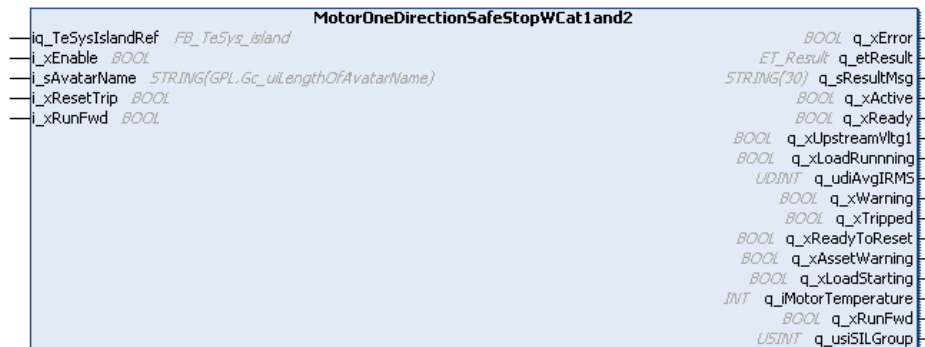
Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input i_xEnable starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as i_xEnable is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input i_xEnable stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable (see page 28)</i> .
i_sAvatarName	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwd	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> • -200...850°C (-328...1562°F) (for PT100) • -200...600°C (-328...1112°F) (for PT1000) • -60...180°C (-76...356°F) (for NI 100/1000)
q_xRunFwd	BOOL	If this output is set to TRUE, the avatar forward switch is closed.

MotorOneDirectionSafeStopWCat1and2 - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorOneDirectionSafeStopWCat1and2` is used to manage a motor in one direction with Safe Stop function compliance for wiring category 1 and category 2.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xEnable</code>	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	STRING(GPL.Gc_u_iLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	BOOL	If this input is set to TRUE, a detected trip error is reset.

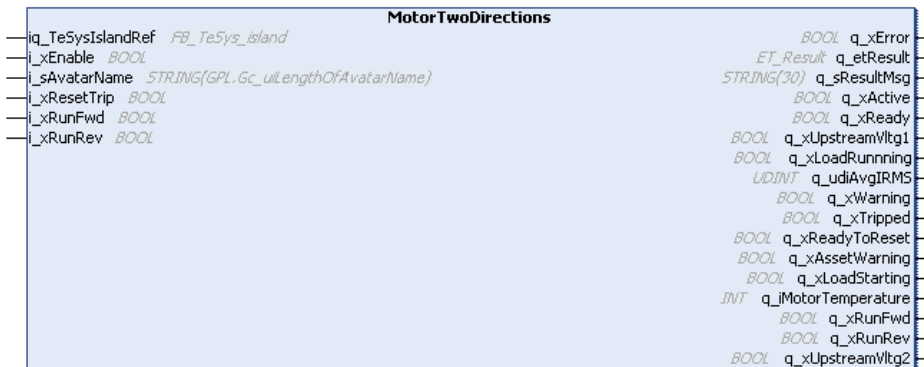
Input	Data type	Description
i_xRunFwd	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltgl	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)
q_xRunFwd	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_usiSILGroup	USINT	Indicates the number of the SIL group.

MotorTwoDirections - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorTwoDirections` is used to manage a motor in two directions (forward and reverse).

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
i_sAvatarName	STRING(GPL.Gc_uLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.

Input	Data type	Description
i_xRunFwd	BOOL	If this input is set to TRUE, the avatar forward switch is closed.
i_xRunRev	BOOL	If this input is set to TRUE, the avatar reverse switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> • -200...850°C (-328...1562°F) (for PT100) • -200...600°C (-328...1112°F) (for PT1000) • -60...180°C (-76...356°F) (for NI 100/1000)
q_xRunFwd	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xRunRev	BOOL	If this output is set to TRUE, the avatar reverse switch is closed.
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.

MotorTwoDirectionsSafeStopWCat1and2 - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorTwoDirectionsSafeStopWCat1and2` is used to manage a motor in two directions (forward and reverse) with Safe Stop function compliance for wiring category 1 and category 2.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xEnable</code>	<code>BOOL</code>	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).

Input	Data type	Description
i_sAvatarName	STRING (GPL.Gc_uiLengthOfAvatarName) <i>(see page 53)</i>	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwd	BOOL	If this input is set to TRUE, the avatar forward switch is closed.
i_xRunRev	BOOL	If this input is set to TRUE, the avatar reverse switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.

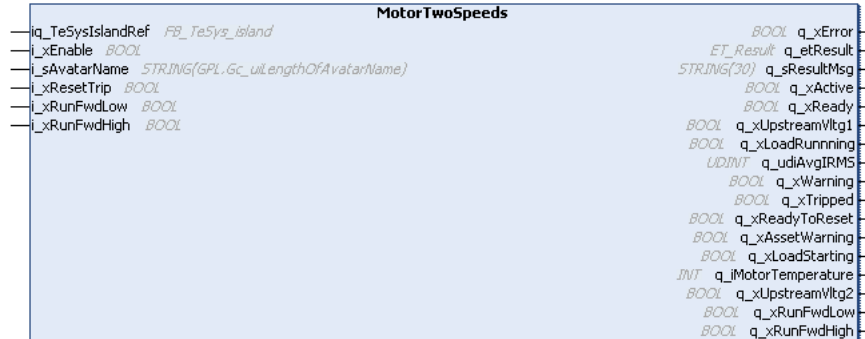
Function Block Description

Output	Data type	Description
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)
q_xRunFwd	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xRunRev	BOOL	If this output is set to TRUE, the avatar reverse switch is closed.
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_usiSILGroup	USINT	Indicates the number of the SIL group.

MotorTwoSpeeds - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorTwoSpeeds` is used to manage a two speed motor.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xEnable</code>	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	BOOL	If this input is set to TRUE, a detected trip error is reset.
<code>i_xRunFwdLow</code>	BOOL	If this input is set to TRUE, the motor starts in forward direction with low speed.

Function Block Description

Input	Data type	Description
i_xRunFwdHigh	BOOL	If this input is set to TRUE, the motor starts in forward direction with high speed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.

Output	Data type	Description
q_xRunFwdLow	BOOL	If this output is set to TRUE, the motor is running in low speed. If this output is set to FALSE, the motor is stopped or is running in high speed.
q_xRunFwdHigh	BOOL	If this output is set to TRUE, the motor is running in high speed. If this output is set to FALSE, the motor is stopped or is running in low speed.

MotorTwoSpeedsSafeStopWCat1and2 - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorTwoSpeedsSafeStopWCat1and2` is used to manage a two speed motor with Safe Stop function compliance for wiring category 1 and category 2.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xEnable</code>	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).

Input	Data type	Description
i_sAvatarName	STRING (GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwdLow	BOOL	If this input is set to TRUE, the motor starts in forward direction with low speed.
i_xRunFwdHigh	BOOL	If this input is set to TRUE, the motor starts in forward direction with high speed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).

Output	Data type	Description
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none">● -200...850°C (-328...1562°F) (for PT100)● -200...600°C (-328...1112°F) (for PT1000)● -60...180°C (-76...356°F) (for NI 100/1000)
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_xRunFwdLow	BOOL	If this output is set to TRUE, the motor is running in low speed. If this output is set to FALSE, the motor is stopped or is running in high speed.
q_xRunFwdHigh	BOOL	If this output is set to TRUE, the motor is running in high speed. If this output is set to FALSE, the motor is stopped or is running in low speed.
q_usiSILGroup	USINT	Indicates the number of the SIL group.

MotorTwoSpeedsTwoDirections - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorTwoSpeedsTwoDirections` is used to manage a two speed motor in two directions (forward and reverse).

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.

Input	Data type	Description
i_xEnable	BOOL	A rising edge at the input i_xEnable starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as i_xEnable is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input i_xEnable stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable</i> (see page 28).
i_sAvatarName	STRING (GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwdLow	BOOL	If this input is set to TRUE, the motor starts in forward direction with low speed.
i_xRunFwdHigh	BOOL	If this input is set to TRUE, the motor starts in forward direction with high speed.
i_xRunRevLow	BOOL	If this input is set to TRUE, the motor starts in reverse direction with low speed.
i_xRunRevHigh	BOOL	If this input is set to TRUE, the motor starts in reverse direction with high speed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltgl	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).

Output	Data type	Description
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_xRunFwdLow	BOOL	If this output is set to TRUE, the motor is running in low speed. If this output is set to FALSE, the motor is stopped or is running in high speed.
q_xRunFwdHigh	BOOL	If this output is set to TRUE, the motor is running in high speed. If this output is set to FALSE, the motor is stopped or is running in low speed.
q_xRunRevLow	BOOL	If this output is set to TRUE, the low speed reverser switch is closed.

Function Block Description

Output	Data type	Description
q_xRunRevHigh	BOOL	If this output is set to TRUE, the high speed reverser switch is closed.
q_xUpstreamVltg3	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the third device (if available) is present.
q_xUpstreamVltg4	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the fourth device (if available) is present.

MotorTwoSpeedsTwoDirectionsSafeStopWCat1and2 - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorTwoSpeedsTwoDirectionsSafeStopWCat1and2` is used to manage a two speed motor in two directions (forward and reverse) with Safe Stop function compliance for wiring category 1 and category 2.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.

Input	Data type	Description
i_xEnable	BOOL	A rising edge at the input i_xEnable starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as i_xEnable is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input i_xEnable stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable (see page 28)</i> .
i_sAvatarName	STRING (GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwdLow	BOOL	If this input is set to TRUE, the motor starts in forward direction with low speed.
i_xRunFwdHigh	BOOL	If this input is set to TRUE, the motor starts in forward direction with high speed.
i_xRunRevLow	BOOL	If this input is set to TRUE, the motor starts in reverse direction with low speed.
i_xRunRevHigh	BOOL	If this input is set to TRUE, the motor starts in reverse direction with high speed.

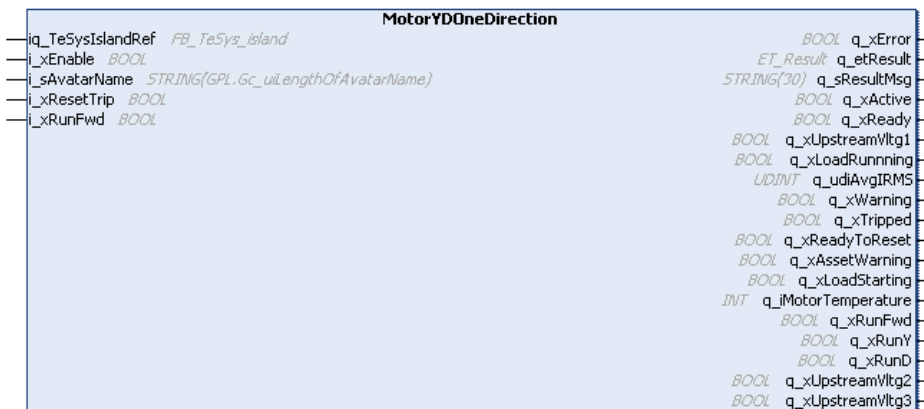
Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)

Output	Data type	Description
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> • -200...850°C (-328...1562°F) (for PT100) • -200...600°C (-328...1112°F) (for PT1000) • -60...180°C (-76...356°F) (for NI 100/1000)
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_xRunFwdLow	BOOL	If this output is set to TRUE, the motor is running in low speed. If this output is set to FALSE, the motor is stopped or is running in high speed.
q_xRunFwdHigh	BOOL	If this output is set to TRUE, the motor is running in high speed. If this output is set to FALSE, the motor is stopped or is running in low speed.
q_xRunRevLow	BOOL	If this output is set to TRUE, the low speed reverser switch is closed.
q_xRunRevHigh	BOOL	If this output is set to TRUE, the high speed reverser switch is closed.
q_xUpstreamVltg3	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the third device (if available) is present.
q_xUpstreamVltg4	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the fourth device (if available) is present.
q_usiSILGroup	USINT	Indicates the number of the SIL group.

MotorYDOneDirection - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `MotorYDOneDirection` is used to manage a wye/delta (star/triangle) motor in one direction.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xEnable</code>	<code>BOOL</code>	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).

Input	Data type	Description
i_sAvatarName	STRING (GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwd	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltgl	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).

Output	Data type	Description
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)
q_xRunFwd	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xRunY	BOOL	If this output is set to TRUE, the Y switch for Motor Y/D avatars is closed.
q_xRunD	BOOL	If this output is set to TRUE, the D switch for Motor Y/D avatars is closed.
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_xUpstreamVltg3	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the third device (if available) is present.

MotorYDTwoDirections - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block **MotorYDTwoDirections** is used to manage a wye/delta (star/triangle) motor in two directions (forward and reverse).

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input <i>i_xEnable</i> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <i>i_xEnable</i> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <i>i_xEnable</i> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable</i> (see page 28).

Function Block Description

Input	Data type	Description
i_sAvatarName	STRING (GPL.Gc_uiLengthOfAvatarName) <i>(see page 53)</i>	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRunFwd	BOOL	If this input is set to TRUE, the avatar forward switch is closed.
i_xRunRev	BOOL	If this input is set to TRUE, the avatar reverse switch is closed.

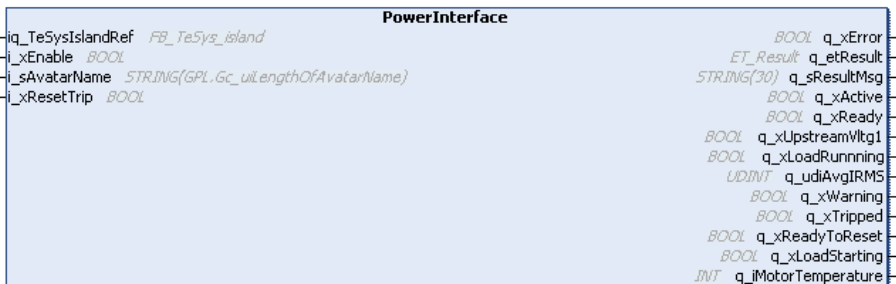
Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltgl	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.

Output	Data type	Description
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)
q_xRunFwd	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xRunY	BOOL	If this output is set to TRUE, the Y switch for Motor Y/D avatars is closed.
q_xRunD	BOOL	If this output is set to TRUE, the D switch for Motor Y/D avatars is closed.
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_xUpstreamVltg3	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the third device (if available) is present.
q_xRunRev	BOOL	If this output is set to TRUE, the avatar reverse switch is closed.
q_xUpstreamVltg4	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the fourth device (if available) is present.

PowerInterface - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `PowerInterface` is used to monitor current on an external power device, such as a solid-state relay, soft starter, or variable speed drive.

Interface

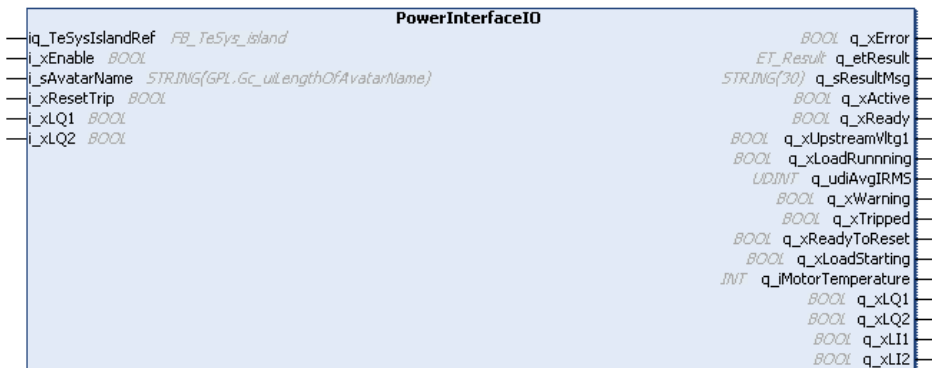
Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xEnable</code>	<code>BOOL</code>	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is <code>TRUE</code> the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	<code>STRING(GPL.Gc_uiLengthOfAvatarName)</code> (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	<code>BOOL</code>	If this input is set to <code>TRUE</code> , a detected trip error is reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltgl	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none"> ● -200...850°C (-328...1562°F) (for PT100) ● -200...600°C (-328...1112°F) (for PT1000) ● -60...180°C (-76...356°F) (for NI 100/1000)

PowerInterfaceIO - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block **PowerInterfaceIO** is used to monitor current and control an external power device, such as a solid-state relay, soft starter, or variable speed drive.

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input <i>i_xEnable</i> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <i>i_xEnable</i> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <i>i_xEnable</i> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable</i> (see page 28).
i_sAvatarName	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.

Input	Data type	Description
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xLQ1	BOOL	If this input is set to TRUE, the logical output 1 is set to TRUE.
i_xLQ2	BOOL	If this input is set to TRUE, the logical output 2 is set to TRUE.

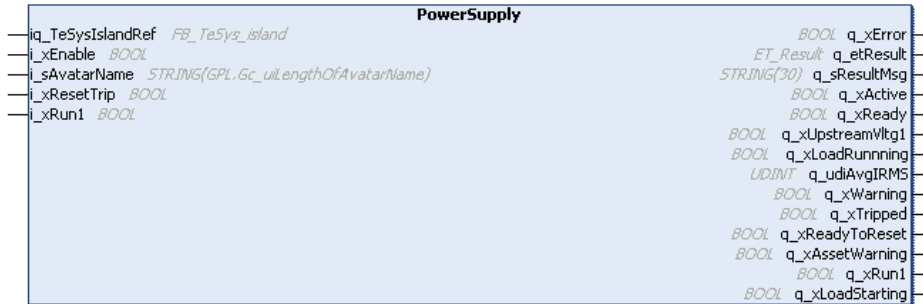
Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a <code>Run</code> or <code>Close</code> command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.

Output	Data type	Description
q_iMotorTemperature	INT	Indicates the temperature of the motor in 0.1 °C scaling unit. Depending on temperature sensor type, the range is: <ul style="list-style-type: none">● -200...850°C (-328...1562°F) (for PT100)● -200...600°C (-328...1112°F) (for PT1000)● -60...180°C (-76...356°F) (for NI 100/1000)
q_xLQ1	BOOL	If this output is set to TRUE, the logical output 1 is set to TRUE.
q_xLQ2	BOOL	If this output is set to TRUE, the logical output 2 is set to TRUE.
q_xLI1	BOOL	If this output is set to TRUE, the logical input 1 of the avatar is set to TRUE.
q_xLI2	BOOL	If this output is set to TRUE, the logical input 2 of the avatar is set to TRUE.

PowerSupply - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `PowerSupply` is used to manage a power supply.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xEnable</code>	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	BOOL	If this input is set to TRUE, a detected trip error is reset.
<code>i_xRun1</code>	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xRun1	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.

Resistor - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `Resistor` is used to manage a resistive load.

Interface

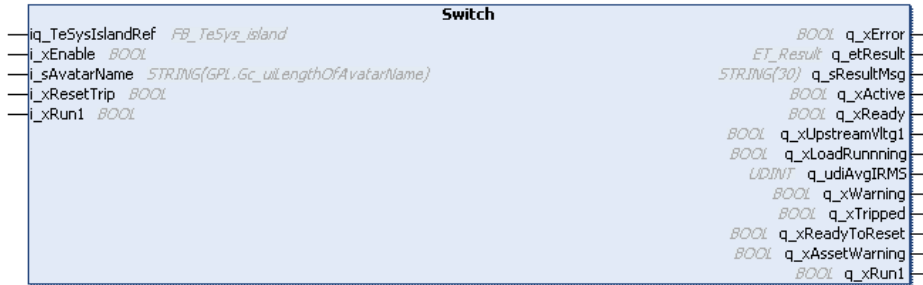
Input	Data type	Description
<code>iq_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xEnable</code>	<code>BOOL</code>	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is <code>TRUE</code> the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	<code>STRING(GPL.Gc_uiLengthOfAvatarName)</code> (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	<code>BOOL</code>	If this input is set to <code>TRUE</code> , a detected trip error is reset.
<code>i_xRun1</code>	<code>BOOL</code>	If this input is set to <code>TRUE</code> , the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xRun1	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.

Switch - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `Switch` establishes or interrupts a power line in an electric circuit.

Interface

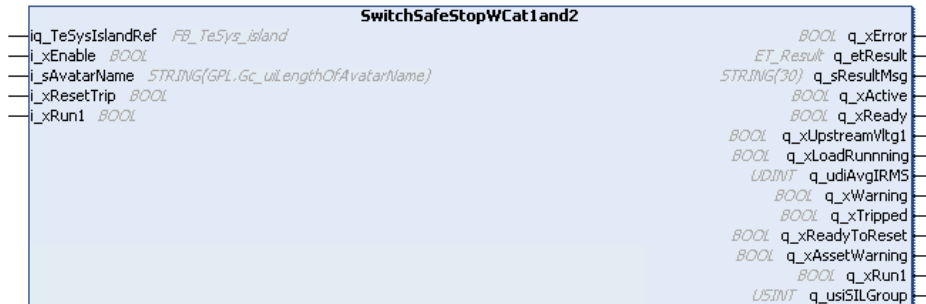
Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xEnable</code>	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	STRING (GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	BOOL	Triggers the trip reset for all configured avatars.
<code>i_xRun1</code>	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (<i>see page 36</i>)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xRun1	BOOL	If this output is set to TRUE, the avatar forward switch is closed.

SwitchSafeStopWCat1and2 - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `SwitchSafeStopWCat1and2` establishes or interrupts a power line in an electric circuit with Safe Stop function compliance for wiring category 1 and category 2.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xEnable</code>	BOOL	A rising edge at the input <code>i_xEnable</code> starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as <code>i_xEnable</code> is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input <code>i_xEnable</code> stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input <code>i_xEnable</code></i> (see page 28).
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTrip</code>	BOOL	Triggers the trip reset for all configured avatars.

Function Block Description

Input	Data type	Description
i_xRun1	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (<i>see page 36</i>)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xRun1	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_usiSILGroup	USINT	Indicates the number of the SIL group.

Transformer - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block Transformer is used to manage a transformer.

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xEnable	BOOL	A rising edge at the input i_xEnable starts the cyclic and acyclic data exchange between the function block and the selected avatar. As long as i_xEnable is TRUE the cyclic data is exchanged every controller cycle and the acyclic parameters are updated sequential in the order, they are defined in the function block. A falling edge at the input i_xEnable stops the data exchange and active acyclic commands. Refer to <i>Behavior of the Function Blocks with the Input i_xEnable</i> (see page 28).
i_sAvatarName	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetTrip	BOOL	If this input is set to TRUE, a detected trip error is reset.
i_xRun1	BOOL	If this input is set to TRUE, the avatar forward switch is closed.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xActive	BOOL	If the function block is active, this output is set to TRUE.
q_xReady	BOOL	Indicates TRUE if the function block is ready to receive an execute command.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xLoadRunning	BOOL	If this output is set to TRUE, a Run or Close command has been executed and current is flowing in the poles (equivalent to motor running but also for non-motor avatars). (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_xWarning	BOOL	If this output is set to TRUE, a protection event has been detected by the avatar.
q_xTripped	BOOL	If this output is set to TRUE, a protection trip event has been detected by the avatar.
q_xReadyToReset	BOOL	If this output is set to TRUE, the avatar meets the trip reset conditions and can be reset with a trip reset command.
q_xAssetWarning	BOOL	If this output is set to TRUE, a power device or SIL Interface Module within the avatar have reached or exceeded 90% of the rated number of operations (per avatar parameter).
q_xRun1	BOOL	If this output is set to TRUE, the avatar forward switch is closed.
q_xLoadStarting	BOOL	If this output is set to TRUE, the motor is in start phase.

Section 8.7

Energy Function Blocks

What Is in This Section?

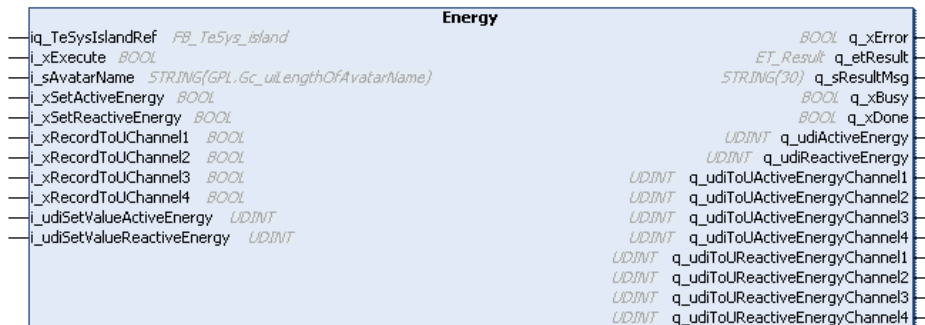
This section contains the following topics:

Topic	Page
Energy - Functional Description	130
Power - Functional Description	133

Energy - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `Energy` returns the energy information of the selected avatar. Resets the energy registers of the selected avatar. Sets the energy preset values of the selected avatar.

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xExecute	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_sResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input i_xExecute</i> (see page 27).
i_sAvatarName	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xSetActiveEnergy	BOOL	Command to set the <code>TotalActiveEnergy</code> value to <code>TotalActiveEnergyPreset</code> value.

Input	Data type	Description
i_xSetReactiveEnergy	BOOL	Command to set the TotalReactiveEnergy value to TotalReactiveEnergyPreset value.
i_xRecordToUChannel1	BOOL	NOTE: Not used in this version of function block (will return all 0 s).
...
i_xRecordToUChannel4	BOOL	NOTE: Not used in this version of function block (will return all 0 s).
i_udiSetValueActiveEnergy	UDINT	Value to preset the TotalActiveEnergy, will take effect on a TRUE command to i_xSetActiveEnergy (Unit: Watt-hours)
i_udiSetValueReactiveEnergy	UDINT	Value to preset the TotalReactiveEnergy, will take effect on a TRUE command to i_xSetReactiveEnergy (Unit: VAR-hours)

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_udiTotalActiveEnergy	UDINT	Total active energy value. value for all avatars in the system with Energy Monitoring Enabled . (Unit: Watt-hous)
q_udiTotalReactiveEnergy	UDINT	Total reactive energy value. value for all avatars in the system with Energy Monitoring Enabled . (Unit: VAR-hours)
q_udiToUActiveEnergyChannel1	UDINT	Channel 1: ToU (Time of Use) total active energy value.
...

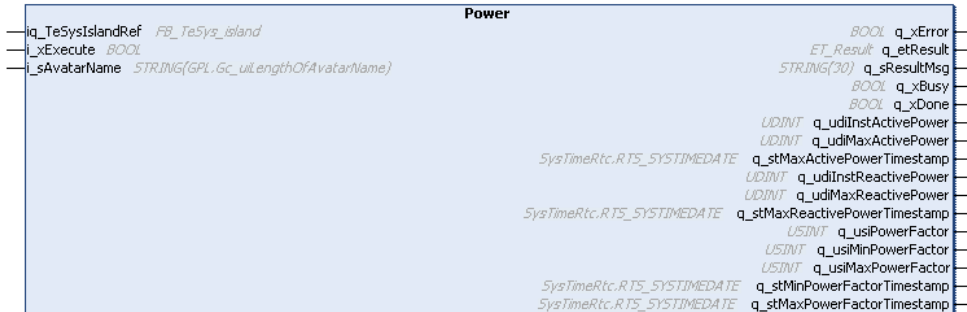
Function Block Description

Output	Data type	Description
q_udiToUActiveEnergyChannel4	UDINT	Channel 4: ToU (Time of Use) total active energy value.
q_udiToUReactiveEnergyChannel1	UDINT	Channel 1: ToU (Time of Use) total reactive energy value. NOTE: Not used in this version of function block (will return all 0 s).
...
q_udiToUReactiveEnergyChannel4	UDINT	Channel 4: ToU (Time of Use) total reactive energy value. NOTE: Not used in this version of function block (will return all 0 s).

Power - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `Power` returns power information of the selected avatar.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.

Output	Data type	Description
<code>q_xError</code>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .

Output	Data type	Description
q_etResult	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_udiInstActivePower	UDINT	Total active power for the avatar. (Unit: W).
q_stMaxActivePowerTimestamp	SysTimeRtc.RTS_ SYSTIMEDATE	Date and time when the maximum active power value was recorded. Refer to the SysTimeRtc library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
q_udiInstReactivePower	UDINT	Total reactive power for the avatar. (Unit: VAR).
q_udiMaxReactivePower	UDINT	Maximum value of the reactive power for the avatar.
q_stMaxReactivePowerTimestamp	SysTimeRtc.RTS_ SYSTIMEDATE	Date and time when the maximum reactive power value was recorded. Refer to the SysTimeRtc library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
q_usiPowerFactor	USINT	True power factor value.
q_usiMinPowerFactor	USINT	Minimum true power factor value.
q_usiMaxPowerFactor	USINT	Maximum true power factor value.
q_stMinPowerFactorTimestamp	SysTimeRtc.RTS_ SYSTIMEDATE	Date and time when the minimum power factor value was recorded. Refer to the SysTimeRtc library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).
q_stMaxPowerFactorTimestamp	SysTimeRtc.RTS_ SYSTIMEDATE	Date and time when the maximum power factor value was recorded. Refer to the SysTimeRtc library (see <i>EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide</i>).

Section 8.8

Diagnostic Function Blocks

What Is in This Section?

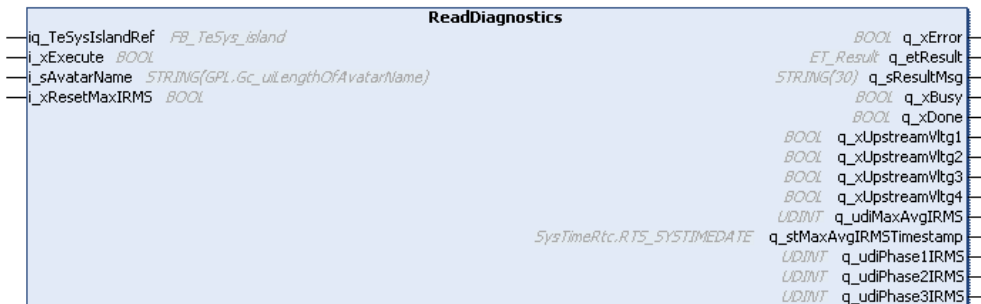
This section contains the following topics:

Topic	Page
ReadDiagnostics - Functional Description	136
ReadTripCounters - Functional Description	138
ReadTripRegisters - Functional Description	140
ReadWarningCounters - Functional Description	142

ReadDiagnostics - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block ReadDiagnostics returns the diagnostic information and resets the max IRMS register of the selected avatar.

Interface

Input	Data type	Description
iq_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xExecute	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs q_xDone, q_xError, q_etResult, and q_etResultMsg are reset with the falling edge of i_xExecute. Refer to <i>Behavior of Function Blocks with the Input i_xExecute (see page 27)</i> .
i_sAvatarName	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
i_xResetMaxIRMS	BOOL	If this input is set to TRUE, the maximum average IRMS current value and the timestamp are reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_xUpstreamVltg1	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the first device is present (breaker closed).
q_xUpstreamVltg2	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the second device (if available) is present.
q_xUpstreamVltg3	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the third device (if available) is present.
q_xUpstreamVltg4	BOOL	If this output is set to TRUE, the avatar has detected that upstream main power of the fourth device (if available) is present.
q_udiMaxAvg1IRMS	UDINT	Maximum average IRMS current value.
q_stMaxAvg1IRMSTimestamp	SysTimeRtc.RTS_SYSTIMEDATE	Date and time when the maximum average IRMS current value was recorded. Refer to the SysTimeRtc library <i>(see EcoStruxure Machine Expert, Getting & Setting Real Time Clock, SysTimeRtc and SysTimeCore Library Guide)</i> .
q_udiPhase1IRMS	UDINT	Value of the phase L1 IRMS. (Unit: mA)
q_udiPhase2IRMS	UDINT	Value of the phase L2 IRMS. (Unit: mA)
q_udiPhase3IRMS	UDINT	Value of the phase L3 IRMS. (Unit: mA)

ReadTripCounters - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `ReadTripCounters` returns the the trip counters and resets all trip counters of the selected avatar.

Interface

Input	Data type	Description
<code>i_q_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetTripCntns</code>	BOOL	If this input is set to TRUE, all trip counters are reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_uiOverloadCntr	UINT	Counter of trips related to thermal overload protection.
q_uiJamCntr	UINT	Counter of trips related to jam protection.
q_uiUndercurrentCntr	UINT	Counter of trips related to undercurrent protection.
q_uiLongStartCntr	UINT	Counter of trips related to long start protection.
q_uiOvercurrentCntr	UINT	Counter of trips related to overcurrent protection.
q_uiMotorOverheatCntr	UINT	Counter of motor overheat trip events.
q_uiStallCntr	UINT	Counter of trips related to stall protection.
q_uiPhaseUnbalancedCntr	UINT	Counter of trips related to phase unbalance protection.
q_uiPhaseConfigCntr	UINT	Counter of trips related to phase configuration protection.
q_uiGroundCurrentCntr	UINT	Counter of trips related to ground current protection.
q_uiPhaseReversalCntr	UINT	Counter of trips related to phase reversal protection.
q_uiPhaseLossCntr	UINT	Counter of trips related to phase loss protection.
q_uiTripsCntr	UINT	Counter of trips related to protections.

ReadTripRegisters - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `ReadTripRegisters` returns the the trip registers of the selected avatar.

Interface

Input	Data type	Description
<code>i_q_TeSysIslandRef</code>	<code>FB_TeSys_island</code>	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input <code>i_xExecute</code></i> (see page 27).
<code>i_sAvatarName</code>	STRING(<code>GPL.Gc_uiLengthOfAvatarName</code>) (see page 53)	Reference to the avatar defined in the TeSys island configuration.

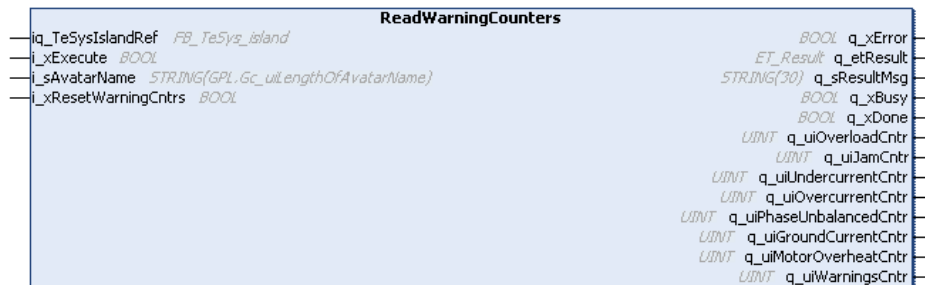
Output	Data type	Description
<code>q_xError</code>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
<code>q_etResult</code>	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.
<code>q_sResultMsg</code>	STRING[30]	Provides additional diagnostic and status information as a text message.

Output	Data type	Description
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_stTripRegister1	ST_TripRegister <i>(see page 47)</i>	Date and trip reason register 1.
...
q_stTripRegister5	ST_TripRegister <i>(see page 47)</i>	Date and trip reason register 5.

ReadWarningCounters - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `ReadWarningCounters` returns the warning counters and resets all warning counters of the selected avatar.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input i_xExecute</i> (see page 27).
<code>i_sAvatarName</code>	STRING(GPL.Gc_uiLengthOfAvatarName) (see page 53)	Reference to the avatar defined in the TeSys island configuration.
<code>i_xResetWarningCntrs</code>	BOOL	If this input is set to TRUE, all warning counters are reset.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result <i>(see page 36)</i>	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_uiOverloadCntr	UINT	Counter of advisories related to thermal overload protection.
q_uiJamCntr	UINT	Counter of advisories related to jam protection.
q_uiUndercurrentCntr	UINT	Counter of advisories related to undercurrent protection.
q_uiOvercurrentCntr	UINT	Counter of advisories related to overcurrent protection.
q_uiPhaseUnbalancedCntr	UINT	Counter of advisories related to phase unbalance protection.
q_uiGroundCurrentCntr	UINT	Counter of advisories related to ground current protection.
q_uiMotorOverheatCntr	UINT	Counter of motor overheat events.
q_uiWarningsCntr	UINT	Counter of advisories related to protections.

Section 8.9

Asset Management Function Blocks

What Is in This Section?

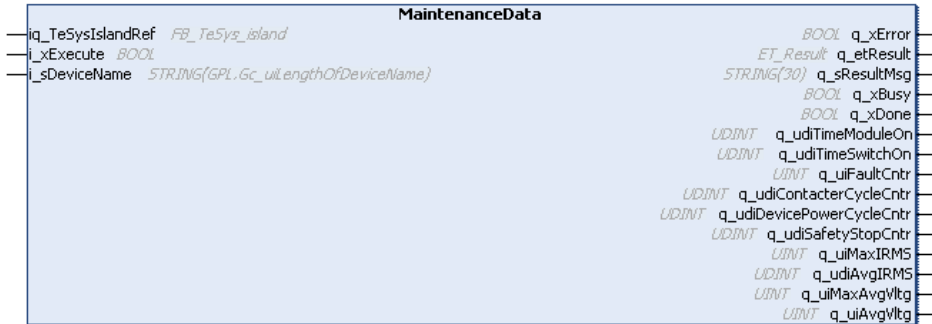
This section contains the following topics:

Topic	Page
MaintenanceData - Functional Description	145
ProductData - Functional Description	147

MaintenanceData - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block MaintenanceData returns the maintenance information of the devices.

Interface

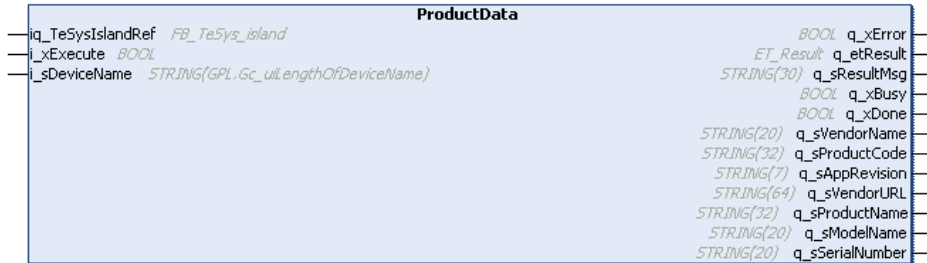
Input	Data type	Description
i_q_TeSysIslandRef	FB_TeSys_island	Reference to the TeSys island device.
i_xExecute	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs q_xDone, q_xError, q_etResult, and q_etResultMsg are reset with the falling edge of i_xExecute. Refer to <i>Behavior of Function Blocks with the Input i_xExecute</i> (see page 27).
i_sDeviceName	STRING(GPL.Gc_uiLengthOfDeviceName) (see page 53)	Reference to the device defined in the TeSys island configuration.

Output	Data type	Description
q_xError	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to q_etResult and q_etResultMsg.
q_etResult	ET_Result (<i>see page 36</i>)	Provides diagnostic and status information as a numeric value.
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_udiTimeModuleOn	UDINT	This register indicates the time that the module has been powered on in its lifetime (Unit: h).
q_udiTimeSwitchOn	UDINT	This register indicates the time the contactor has been in the closed state (Units in Hours).
q_uiFaultCntr	UINT	This register indicates number of times this module has experienced a detected device error (Unit: h). This value does not include detected device errors which prevent the saving or corruption of the non-volatile memory.
q_udiContacterCycleCntr	UDINT	This register indicates number of times the contactor has been commanded to the closed state from the open state.
q_udiContacterCycleCntr	UDINT	This register indicates number of times the device has been powered on.
q_uiSafetyStopCntr	UDINT	This value indicates the number of mirror relay operations.
q_uiMaxIRMS	UDINT	This register indicates that maximum current the device has measured in its lifetime. (Unit: mA)
q_udiAvgIRMS	UDINT	Indicates the average of the most recent phase current IRMS values. (Unit: mA)
q_uiMaxAvgVRMS	UDINT	This register indicates maximum voltage the device has measured in its lifetime. (Unit: mV)
q_uiAvgVRMS	UDINT	Average IRMS voltage on three phases (Unit: mV).

ProductData - Functional Description

Overview

Type:	Function block
Available as of:	V1.0.0.0



Functional Description

The function block `ProductData` returns the product-specific information of the devices.

Interface

Input	Data type	Description
<code>iq_TeSysIslandRef</code>	FB_TeSys_island	Reference to the TeSys island device.
<code>i_xExecute</code>	BOOL	Upon a rising edge of this input, the function block starts the execution. The outputs <code>q_xDone</code> , <code>q_xError</code> , <code>q_etResult</code> , and <code>q_etResultMsg</code> are reset with the falling edge of <code>i_xExecute</code> . Refer to <i>Behavior of Function Blocks with the Input i_xExecute</i> (see page 27).
<code>i_sDeviceName</code>	STRING(GPL.Gc_uiLengthOfDeviceName) (see page 53)	Reference to the device defined in the TeSys island configuration.

Output	Data type	Description
<code>q_xError</code>	BOOL	If this output is set to TRUE, an error has been detected. For details, refer to <code>q_etResult</code> and <code>q_etResultMsg</code> .
<code>q_etResult</code>	ET_Result (see page 36)	Provides diagnostic and status information as a numeric value.

Function Block Description

Output	Data type	Description
q_sResultMsg	STRING[30]	Provides additional diagnostic and status information as a text message.
q_xBusy	BOOL	If this output is set to TRUE, the function block execution is in progress.
q_xDone	BOOL	If this output is set to TRUE, the execution has been completed successfully.
q_sVendorName	STRING[20]	Name of the vendor.
q_sProductCode	STRING[32]	Name of the product.
q_sAppRevision	STRING[7]	Revision number of the application firmware presently in the device.
q_sVendorURL	STRING[64]	Name of the vendor.
q_sProductName	STRING[32]	Name of the product.
q_sModelName	STRING[20]	Name of the model.
q_sSerialNumber	STRING[20]	Serial number.



D

DTM

(*device type manager*) Classified into 2 categories:

- Device DTMs connect to the field device configuration components.
- CommDTMs connect to the software communication components.

The DTM provides a unified structure for accessing device parameters and configuring, operating, and diagnosing the devices. DTMs can range from a simple graphical user interface for setting device parameters to a highly sophisticated application capable of performing complex real-time calculations for diagnosis and maintenance purposes.

E

EtherNet/IP

(*Ethernet industrial protocol*) An open communications protocol for manufacturing automation solutions in industrial systems. EtherNet/IP is in a family of networks that implement the common industrial protocol at its upper layers. The supporting organization (ODVA) specifies EtherNet/IP to accomplish global adaptability and media independence.

F

function block

A programming unit that has 1 or more inputs and returns 1 or more outputs. FBs are called through an instance (function block copy with dedicated name and variables) and each instance has a persistent state (outputs and internal variables) from 1 call to the other.

Examples: timers, counters

M

Modbus

The protocol that allows communications between many devices connected to the same network.

P

POU

(program organization unit) A variable declaration in source code and a corresponding instruction set. POUs facilitate the modular re-use of software programs, functions, and function blocks. Once declared, POUs are available to one another.

T

TCP

(transmission control protocol) A connection-based transport layer protocol that provides a simultaneous bi-directional transmission of data. TCP is part of the TCP/IP protocol suite.



A

AnalogIO, *80*
avatar description, *31*

C

common inputs and outputs
 behavior of function blocks with the input
 i_xExecute, *27*

D

DigitalIO, *82*

E

Energy, *130*
EnhancedStatus, *84*
ET_Result, *36*
 AvatarNotAvailable, *36*
 AvatarNotSupported, *36*
 BufferFull, *36*
 CommandAborted, *36*
 CommandError, *36*
 CommunicationError, *36*
 ConfigurationInvalid, *37*
 DeviceNotAvailable, *36*
 Done, *36*
 FirmwareNotSupported, *37*
 IdentStructureEmpty, *36*
 InitializationFailed, *36*
 NoAvatarConfigured, *36*
 NoError, *36*
 NoParameterInterface, *36*
 OrderNotPresent, *36*
 SystemAvatarNotSupported, *36*
 SystemDeviceNotSupported, *36*
 UnknownResponse, *36*
 Working, *36*

ET_SafeStop, *42*
 AllSILStartersOpen, *42*
 AvatarDeviceFault, *42*
 CommandOnlyOneTerminal, *42*
 NormalOperation, *42*
 SILNotPresent, *42*
 SILStartersNotOpen, *42*
ET_TripMessage, *43*
 CurrentPhaseLoss, *43*
 CurrentPhaseReversal, *43*
 CurrentPhaseUnbalance, *43*
 GroundCurrent, *43*
 Jam, *43*
 LongStart, *43*
 MotorOverheat, *43*
 NoTrip, *43*
 Overcurrent, *43*
 PhaseConfiguration, *43*
 Stall, *43*
 ThermalOverload, *43*
 Undercurrent, *43*
ET_WarningMessage, *44*
 CurrentPhaseUnbalance, *44*
 GroundCurrent, *44*
 Jam, *44*
 MotorOverheat, *44*
 NoWarning, *44*
 Overcurrent, *44*
 ThermalOverload, *44*
 Undercurrent, *44*

F

function block

- AnalogIO, *80*
- DigitalIO, *82*
- Energy, *130*
- EnhancedStatus, *84*
- MaintenanceData, *145*
- MotorOneDirection, *86*
- MotorOneDirectionSafeStopWCat1and2, *88*
- MotorTwoDirections, *90*
- MotorTwoDirectionsSafeStopWCat1and2, *92*
- MotorTwoSpeeds, *95*
- MotorTwoSpeedsSafeStopWCat1and2, *98*
- MotorTwoSpeedsTwoDirections, *101*
- MotorTwoSpeedsTwoDirections-SafeStopWCat1and2, *105*
- MotorYDOneDirection, *108*
- MotorYDTwoDirections, *111*
- Power, *133*
- PowerInterface, *114*
- PowerInterfaceIO, *116*
- PowerSupply, *119*
- ProductData, *147*
- ReadDiagnostics, *136*
- ReadSystemDiagnostics, *62*
- ReadTripCounters, *138*
- ReadTripRegisters, *140*
- ReadWarningCounters, *142*
- Resistor, *121*
- Switch, *123*
- SwitchSafeStopWCat1and2, *125*
- SystemControl, *60*
- SystemEnergyBasic, *66*
- SystemMaintenanceData, *76*
- SystemPowerBasic, *68*
- SystemProductData, *77*
- SystemVoltageBasic, *71*
- SystemVoltageEnhanced, *73*
- Transformer, *127*

G

- GCL (Global Constants List)
 - TeSys island, *51*
- GPL
 - TeSys island, *53*

I

- i_xEnable, *28*

M

- MaintenanceData, *145*
- MotorOneDirection, *86*
- MotorOneDirectionSafeStopWCat1and2, *88*
- MotorTwoDirections, *90*
- MotorTwoDirectionsSafeStopWCat1and2, *92*
- MotorTwoSpeeds, *95*
- MotorTwoSpeedsSafeStopWCat1and2, *98*
- MotorTwoSpeedsTwoDirections, *101*
- MotorTwoSpeedsTwoDirections-SafeStopWCat1and2, *105*
- MotorYDOneDirection, *108*
- MotorYDTwoDirections, *111*

P

- Power, *133*
- PowerInterface, *114*
- PowerInterfaceIO, *116*
- PowerSupply, *119*
- ProductData, *147*

R

- ReadDiagnostics, *136*
- ReadSystemDiagnostics, *62*
- ReadTripCounters, *138*
- ReadTripRegisters, *140*
- ReadWarningCounters, *142*
- Resistor, *121*

S

ST_MinorFltRegister, *46*
ST_TripRegister, *47*
ST_VoltageRegister, *48*
Switch, *123*
SwitchSafeStopWCat1and2, *125*
SystemControl, *60*
SystemEnergyBasic, *66*
SystemMaintenanceData, *76*
SystemPowerBasic, *68*
SystemProductData, *77*
SystemVoltageBasic, *71*
SystemVoltageEnhanced, *73*

T

TeSys island, *19*

- GCL (Global Constants List), *51*
- GPL, *53*
 - referencing avatars and devices, *22*

Transformer, *127*