



Process Expert - General Purpose Library Classic

Device Control Services Reference Manual

EIO0000001309.16
03/2023



Legal Information

The Schneider Electric brand and any trademarks of Schneider Electric SE and its subsidiaries referred to in this guide are the property of Schneider Electric SE or its subsidiaries. All other brands may be trademarks of their respective owners.

This guide and its content are protected under applicable copyright laws and furnished for informational use only. No part of this guide may be reproduced or transmitted in any form or by any means (electronic, mechanical, photocopying, recording, or otherwise), for any purpose, without the prior written permission of Schneider Electric.

Schneider Electric does not grant any right or license for commercial use of the guide or its content, except for a non-exclusive and personal license to consult it on an "as is" basis. Schneider Electric products and equipment should be installed, operated, serviced, and maintained only by qualified personnel.

As standards, specifications, and designs change from time to time, information contained in this guide may be subject to change without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this material or consequences arising out of or resulting from the use of the information contained herein.

As part of a group of responsible, inclusive companies, we are updating our communications that contain non-inclusive terminology. Until we complete this process, however, our content may still contain standardized industry terms that may be deemed inappropriate by our customers.

© 2023 – Schneider Electric. All rights reserved.

Table of Contents

Safety Information	7
Qualification of Personnel	7
Proper Use	8
Before You Begin	8
Start-up and Test	9
Operation and Adjustments	9
About the Book	11
General	15
General Overview of Device Library	16
Introduction	16
Overview of Device Resources	17
List of Function Blocks	17
Overview of Supervision Services	20
Function Blocks (DFB) Interface	20
Set-Point Management	22
Monitoring	22
Circuit Breakers	23
COMPACT Profile	24
Description	24
DFB Representation	24
Inputs	26
Outputs	28
Inputs/Outputs	33
Diagnostics Information Management	36
MASTERPACT Profile	39
Description	39
DFB Representation	39
Inputs	41
Outputs	43
Inputs/Outputs	50
Diagnostics Information Management	54
HWCIRCUITBREAKER	56
Description	56
DFB Representation	56
Inputs	57
Outputs	58
Inputs/Outputs	58
Diagnostics Information Management	60
OperationLimit	61
Description	61
DFB Representation	61
Inputs	62
Outputs	62
Digital Protection Relays	63
SEPAM Profile	64
Description	64
DFB Representation	65
Inputs	66

Outputs	68
Inputs/Outputs	80
Diagnostics Information Management	82
Motor Controllers and Starters	84
TeSys T Profile	85
Description	85
DFB Representation	86
Inputs	87
Outputs	95
Inputs/Outputs	101
Diagnostics Information Management	104
Tesys U Profile	107
Description	107
DFB Representation	108
Inputs	110
Outputs	112
Inputs/Outputs	118
Diagnostics Information Management	120
Power Monitoring Devices	123
PM Profile - Power Monitoring Profile	124
Description	124
DFB Representation	125
Inputs	127
Outputs	129
Inputs/Outputs	135
Diagnostic Information Management	138
Smart UPS Profile	140
Description	140
DFB Representation	141
Inputs	141
Outputs	143
Inputs/Outputs	146
Diagnostic Information Management	148
ACCUSINE PCS Profile	150
Description	150
DFB Representation	151
Inputs	151
Outputs	153
Inputs/Outputs	157
Public Variables	160
Diagnostic Information Management	160
Communication Needs And/Or Requirements	161
Progressive Starters	163
ATS Progressive Starter Profile	164
Description	164
DFB Representation	165
Inputs	166
Outputs	169
Inputs/Outputs	173
Diagnostic Information Management	177

Variable Speed Drives.....	179
ATV Profile.....	180
Description.....	180
DFB Representation	182
Inputs.....	186
Outputs.....	194
Inputs/Outputs.....	205
Public Variables	211
Diagnostics Information Management.....	212
Weighing Module	215
Weighing Module Profile	216
Description.....	216
DFB Representation	217
Inputs.....	217
Outputs.....	219
Inputs/Outputs.....	220
Public Variables	224
Diagnostic Information Management	225
Configuration of PMESWT Module.....	226
PMESWT DFB Work Flow.....	231
Safety Module.....	234
MBXPSMC - XPSMC Safety Module (Modbus)	235
DFB Representation	235
Inputs.....	236
Outputs.....	237
Inputs/Outputs.....	242
Communication Technologies.....	247
Ethernet Technology	248
CommunicationOK Variable.....	248
Description.....	248
ATV71/61 Control Block on Ethernet with IO Scanning.....	249
Introduction	249
Preparing the Device	250
ATV 71/61 IO Scanner Configuration	251
Modifying the ATV7161 IOScanner Configuration	253
Configuring the IO Scanning of the Controller	255
Tsys T Control Block on Ethernet.....	256
Tsys T Hardware Configuration.....	257
Preparing the Device	257
Configuring the IO Scanning of the Controller	258
Modbus Devices Used Through Gateway	259
Modbus Devices Used Through Gateway.....	260
TSXETG100 Gateway Configuration Description.....	261
Control Program Example.....	262
Modbus TCPIP - Implicit with NOE	264
Creating an Application using Modbus TCPIP Communication Technology (Implicit with NOE).....	264
Modbus TCPIP - Implicit with NOC	268
Modbus TCPIP - Implicit with NOC	269
Modbus TCPIP - Explicit.....	273

Creating an Application using Modbus TCPIP Communication Technology (Explicit)	274
Modbus TCPIP - Explicit with Gateway	278
Creating an Application using Modbus Communication Technology (Explicit) with Gateway	278
Modbus over ULP Technology	282
Creating a Project with \$MASTERPACTMTZMBU for IFM connected to Masterpact MTZ and Modbus Serial RJ45 Ports	282
Creating a Project with \$MasterpactMTZMBU for IFE or IFE stacked with IFM connected to Masterpact MTZ and Modbus Ethernet RJ45 Ports	284
Modbus Technology	287
General	287
Modbus Technology	287
Determining Timeout Times	288
Configuring the Devices	290
ATV 31, ATV 312, and ATV 12	290
ATV 71 and ATV 61	290
Tsys U	291
Tsys T	294
ATS 22 and ATS 48	295
SEPAM	296
PM	297
Compact and Masterpact	297
Modbus Serial - Explicit	298
Creating an Application using Modbus Communication Technology (Explicit)	298
Creating an Application using Modbus Communication Technology (Explicit) with M580/M340 Controller	302
Creating an Application using Modbus Communication Technology (Explicit) with M580 controller on x80 drop	304
Profibus Technology	308
Steps to Create a Working Project with Profibus Templates	308
Configuring PRM Device	308
Creating a Project with Profibus Template	310
CANOpen Technology	312
Advantys Configuration in EcoStruxure Process Expert	312
Advantys Configuration in EcoStruxure Process Expert	312
Index	319

Safety Information

Important Information

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 and experience in industrial control programming.
- 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 automation control systems and is intended solely for the purposes 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 concept of this overall system (for example, machine concept).

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 are 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:

(In case of divergence or contradiction between any translation and the English original, the original text in the English language will prevail.)

- 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 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

Document Scope

This document describes the DFBs for devices that are included in the General Purpose library of the EcoStruxure Process Expert software as well as the Control services that they provide.

This document does not cover any development procedures and internal functionality details.

This document is for users with good knowledge in the employment of the Control participant.

NOTE: The terms circuit breaker, short-circuit, and disconnect as used in this document, are defined by the standards EN ISO 13849-1, EN 15011, and EN 14439.

Validity Note

This document has been updated for the release of EcoStruxure™ Process Expert 2023.

Related Documents

The characteristics that are described in the present document, as well as those described in the documents included in the Related Documents section below, can be found online. To access the information online, go to the Schneider Electric home page www.se.com/ww/en/download/.

The characteristics that are described 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.

Title of Documentation	Reference Number
EcoStruxure™ Process Expert - General Purpose Library Classic Device Templates Reference Manual	EIO0000001308
EcoStruxure™ Process Expert - General Purpose Library Classic Device Supervision Services Reference Manual	EIO0000001310
EcoStruxure Process Expert User Guide	EIO0000001114

Technical Support

Visit <https://www.se.com/myschneider/> for support, software updates, and latest information.

Product Related Information

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA), or equivalent risk analysis, of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate them.
- Review the implications of communication link interruptions and take actions to mitigate them.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and error conditions) according to your risk assessment, and applicable codes and regulations.
- Apply local accident prevention and safety regulations and guidelines.¹
- Test each implementation of a system for proper operation before placing it 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.

Examples described in this manual are provided for information only.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Adapt examples that are given in this manual to the specific functions and requirements of your industrial application before you implement them.

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

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.

Standard	Description
	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.

General

What's in This Part

General Overview of Device Library	16
Overview of Device Resources.....	17
Overview of Supervision Services.....	20

Overview

This part explains general overview of EcoStruxure Process Expert device library and concepts implemented for controlling devices.

General Overview of Device Library

What's in This Chapter

Introduction..... 16

Overview

This chapter provides overview of Process Expert General Purpose library for devices.

Introduction

Overview of Device Templates

The EcoStruxure Process Expert automation system provides resources that have been pre-configured and tested by Schneider Electric and that are designed specifically for automating the control of Schneider Electric devices such as variable-speed drives, starters, and power meters.

The control resources for Control (functional blocks, components, and libraries) and the monitoring resources (components for Supervision) provide common required functions, facilitating the development of control systems.

The system provides Control function blocks (DFB) for control and also provides dynamic representations (Genies) and faceplates (implemented through windows with SuperGenies syntax) for monitoring.

To automate and simplify the implementation of control systems of the electrical devices, you can use these resources with the tools for massive code generation and for the synchronization of control and monitoring subsystems.

You can link the resources that are specifically designed for device control purposes with other generic resources for discrete, continuous, and/or batch process management included in the EcoStruxure Process Expert General Purpose library. You can also use EcoStruxure Process Expert General Purpose library to link device control functions with extra data monitoring and for Modbus communications.

This document describes the basic concepts and details of each function blocks (DFBs) to implement the device control functions.

Overview of Device Resources

What's in This Chapter

List of Function Blocks 17

List of Function Blocks

Overview

This chapter lists the resources designed for device control automation purposes.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

Only authorized and suitably trained personnel have to implement an industrial control scheme or application using these function blocks.

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

Function Blocks for Devices

The DFBs have been classified based on the family being used.

The following table lists the function blocks and their description:

Family name	Profile name	Function blocks	Description
Circuit breakers	Compact	MBCOMPACTNSX, page 24	Compact NSX management
		MBUCOMPACTNSX, page 24	Compact NSX management.
	Masterpact	MBMASTERPACT, page 39	Masterpact management.
		MBMASTERPACTC, page 39	Masterpact chassis management.
		MBUMASTERPACTMTZ, page 39	Masterpact MTZ circuit breaker without chassis management.
		MBUMASTERPACTMTZC, page 39	Masterpact MTZ circuit breaker with chassis management.
		MBUMASTERPACTNx, page 39	Masterpact Nx circuit breaker without chassis management.
		MBUMASTERPACTNxC, page 39	Masterpact Nx circuit breaker with chassis management.
Digital protective relays	Sepam	MBSEPAM20, page 64	Sepam 20 management
		MBSEPAM40, page 64	Sepam 40 management
		MBSEPAM80, page 64	Sepam 80 management (Modbus serial)
		ESEPAM80C, page 64	Sepam 80 management (I/O scanning)
Motor controllers and starters	TeSys T	EIOSTESYST, page 85	TeSys T management (normal I/O scanning)
		ETESYST, page 85	TeSys T management (fast I/O scanning)
		EMESTESYST, page 85	TeSys T management (Ethernet explicit messaging)
		MBTESYST, page 85	TeSys T management (Modbus serial)
		TESYSTCTL, page 85	TeSys T management (Advantys STB)

Family name	Profile name	Function blocks	Description
		TESYSTPB, page 85	TeSys T management (Profibus).
	TeSys U	MBTESYSUSCST, page 107	TeSys U standard starter management (Modbus serial)
		MBTESYSUSC, page 107	TeSys U advanced and multifunction starter (Modbus serial)
		MBTESYSUC, page 107	TeSys U advanced and multifunction controller (Modbus serial)
		TESYSUSCST, page 107	TeSys U standard starter management (Advantys STB)
		TESYSUCTL, page 107	TeSys U advanced starter management (Advantys STB)
		TESYSUSC, page 107	TeSys U multifunction starter management (Advantys STB)
		TESYSUC, page 107	TeSys U advanced and multifunction controller management (Advantys STB)
Power monitoring devices	PM	MBPM700, page 124	Power Meter 700 management
		MBPM800, page 124	Power Meter 800 management (Modbus serial)
		MBPM1200, page 124	Power Meter 1200 management
		MBPM9C, page 124	Power Meter 9C management
		EPM800, page 124	Power Meter 800 management (I/O scanning)
		EMPM800, page 124	Power Meter 800 management
		MBPM5350, page 124	Power Meter 5350 management
		EMPM53xx, page 124	Power Meter 53xx management (Modbus TCPIP).
		EMPM82xx, page 124	Power Meter 82xx management (Modbus TCPIP).
	Smart UPS	MBSMARTUPS, page 140	Smart UPS management
	Accusine	EACCUSINE, page 150	Accusine management
Progressive starters	ATS	MBATS22, page 164	Altistart 22 management
		MBATS48, page 164	Altistart 48 management
Variable speed drives	ATV	ATV7161, page 180	Altivar 61 and Altivar 71 management (I/O scanning)
		EATV32, page 180	Altivar 32 management (I/O scanning)
		EMESATV7161, page 180	Altivar 61 and Altivar 71 management (Ethernet explicit messaging)
		MBATV, page 180	Altivar 12, Altivar 312 and Altivar 31 management (Modbus)
		MBATV212, page 180	Altivar 212 management (Modbus)
		MBATV7161, page 180	Altivar 61 and Altivar 71 management (Modbus serial)
		ASATV31, page 180	Altivar 31 management (Advantys STB)
		ASATV7161, page 180	Altivar 61 and Altivar 71 management (Advantys STB)
		PBATV7161, page 180	Altivar 61 and Altivar 71 management (Profibus)
		ATV6xx, page 180	Altivar 6xx management
		ATV9xx, page 180	Altivar 9xx management
		ATV6xxx, page 180	Altivar 6xxx management

Family name	Profile name	Function blocks	Description
Weighing Module	Weighing module Profile	EIPMPMESWT, page 216	Weighing module
Safety modules	Safety module	MBXPSMC, page 235	XPSMC safety module (Modbus)

NOTE: EcoStruxure Process Expert device templates do not support device specific DTMs, it supports only Generic DTM.

Overview of Supervision Services

What's in This Chapter

Function Blocks (DFB) Interface	20
Set-Point Management	22
Monitoring	22

Overview

This chapter describes the concepts implemented by the resources used for device Supervision purposes.

Function Blocks (DFB) Interface

Overview

The Control function blocks for devices provide an interface that allows the DFBs to be configured, monitored, and controlled both from the monitoring subsystem and from the control subsystem itself.

The following interfaces are provided:

- Basic Configuration
- Control
- States and Monitoring
- Variables

Basic Configuration

DFB public variables for static and identified data are in engineering time (for example, range for an input channel, refreshing communication variables and so on).

Control

DFB input and output pins:

- enable issuing commands from other program blocks or sections.
- provide the DFB status to allow implementing switching operations (for example, device controlled from HMI/SCADA system, `ExtControlled` output pin), detected failures (for example, communication interruption), alarm (for example, thermal trip alarm), and so on.

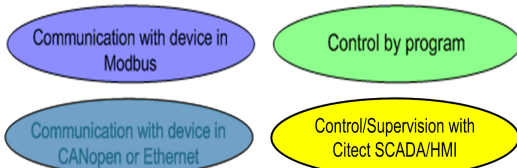
States and Monitoring

Depending on their types, the DFBs have several input/output pins that need to be connected to variables used to hold either the states of the pins or data. In addition, these variables allow commands and parameters received from the monitoring subsystem to be managed.

The following variables are considered:

21

Function Block Illustration



Set-Point Management

Overview

The blocks used for device control purposes manage the set-points coming from the following sources according to the owner:

- Operator
- Program

To avoid unwanted transitions in the set-point, continuously set the local set-points (Operator or Program) to match the selected set-point.

In principle, the owner (Operator or Program) of the function block is not modified from the control subsystem (through the program in the PLC) because the control subsystem is mechanism that stops the Program from affecting the block. This is to confirm that the commands issued towards the block are those generated from the monitoring subsystem.

NOTE: The dynamic symbols and the faceplates corresponding to the General Purpose library devices are used to implement the control function for block resetting (as long as the block is in Operator mode). If you want to perform additional control actions (Run, QuickStop, and so on) from faceplates, use the control blocks included in the General Purpose library for process.

Operator (Operator/Local set-point)

Receives commands from the monitoring subsystem (Supervision/HMI). The DFBs receive these commands through `***_ST` and `***_CFG` structures. The `Owner` variable is implemented in the devices in the `***_ST.CFGW.1` and is set to 1 to specify operator-based control.

Program

Generates commands from the control program (Control). The DFBs receive these commands through the input and output pins of the DFBs. The `Owner` variable is implemented in the devices in the `***_ST.CFGW.1` and is set to 0 to specify program-based control.

Monitoring

Overview

The function blocks in the General Purpose library have Monitoring mode (which you can select by setting the `ControlCommand` input to FALSE) that allows the device to be monitored in a communications-based manner independently of the state of the control inputs. This way, the control block only performs read operations on the device and updates the output variables of the block without sending any commands.

You can use the operating mode if switching operations are carried out on the device locally through a physical screw terminal.

NOTE: If the block is in Monitoring mode (`ControlCommand` set to FALSE), it resets communication interruptions through the program or HMI/SCADA system.

If you configure the device to be controlled and monitored completely by the PLC in a communications-based manner, set this variable to TRUE.

Circuit Breakers


What's in This Part

COMPACT Profile	24
MASTERPACT Profile	39
HWCIRCUITBREAKER	56
OperationLimit	61

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Circuit Breakers.

These function blocks do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines.¹
- Test each implementation of this library for proper operation before placing it 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.

COMPACT Profile

What's in This Chapter

Description	24
DFB Representation	24
Inputs	26
Outputs	28
Inputs/Outputs	33
Diagnostics Information Management	36

Overview

This chapter describes the DFBs of Compact profile.

Description

General

The COMPACTNSX profile is the control function to manage Compact NSX electrical protection circuit breakers.

NOTE: EMCOMPACTNSX and MBCOMPACTNSX are deprecated control functions.

Function Description

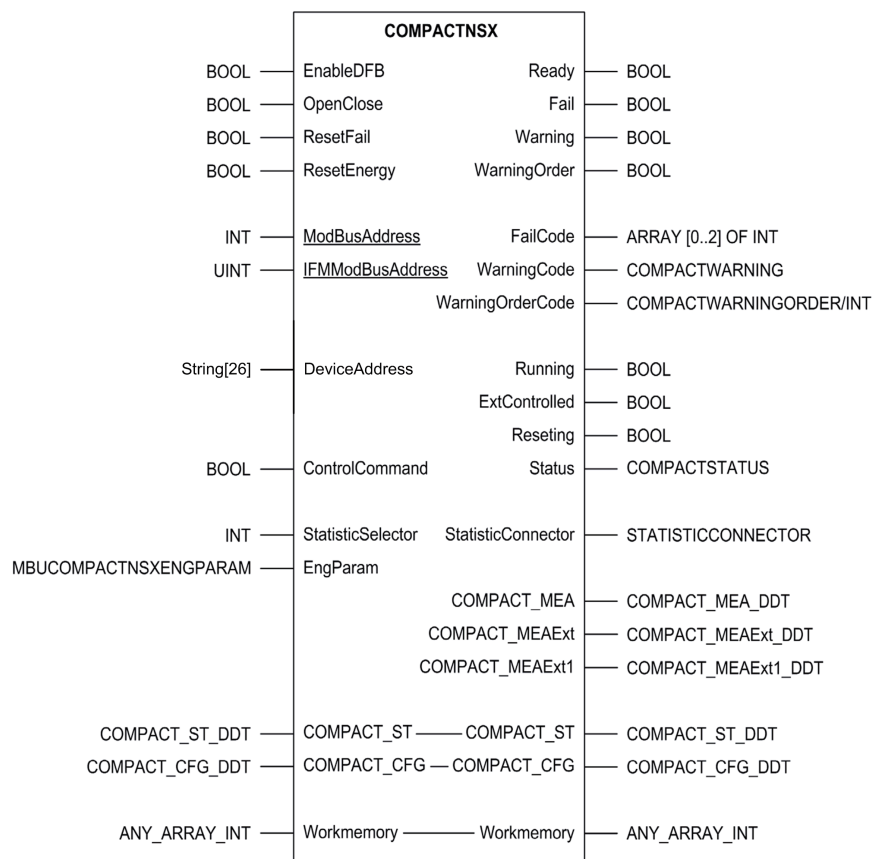
The main functions of the DFB are described in the following table:

Function	Description
Remote Resetting	Allows resetting of the device.
Open/Close	Allows circuit breaker to open and close.
Monitoring	Allows the device measurement data to be monitored.

DFB Representation

Representation

The following figure represents the function module of COMPACTNSX profile:

**NOTE:**

- The underlined parameters are specific for some components.
- Data type of `WarningOrderCode` output pin for `MBUCOMPACTNSX` is `COMPACTWARNINGORDER`. Similarly, for `MBUCOMPACTNSX` the data type is `INT`

The table shows the parameters available for specific DFBs:

Parameters		Components		
		Modbus	Modbus over ULP	Ethernet
		<i>MBCOMPACTNSX</i> (Deprecated)	<i>MBUCOMPACTNSX</i>	<i>EMCOMPACTNSX</i> (Deprecated)
Inputs	ModBusAddress	X	–	–
	IFModBusAddress	–	X	–
	DeviceAddress	–	–	X
X : Parameter is available				
– : Parameter is not available				

NOTE:

- If `IFModbusAddress` is not configured then default value of 255 is considered.
- For how to create project using CompactNSX circuit breakers, refer Communication technologies (see Modicon Libraries General Purpose, Devices Components User Guide).
- IFM v2 and above hardware supports these control functions.

Inputs

Input Parameter Description

Name	Data type	Description
EnableDFB	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. <p>Public variable values are loaded during the first enabling cycle.</p>
OpenClose	BOOL	<p>This input controls the circuit breaker.</p> <ul style="list-style-type: none"> Rising edge on pin = Closes the circuit breaker. Falling edge on pin = Opens the circuit breaker.
ResetFail	BOOL	<p>1 = Resets the <code>Fail</code> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <code>ControlCommand</code> is 1.</p>
ModbusAddress ¹	INT	<p>Device address within the Modbus network.</p> <p>You can find this variable in Modbus communications.</p>
IFMModbusAddress ²	UINT	Modbus address of the IFM Module.
DeviceAddress ³	Platform	IP Addressing DeviceAddress (variable)
	M340	'{IP}ID'
	M580	'{IP}ID'
	Quantum	'{IP}ID'
	NOTE: ID is 255	
ControlCommand	BOOL	<p>Indicates to the DFB whether the Compact NSX is being controlled locally or from a source external to the DFB.</p> <ul style="list-style-type: none"> 0 = Performs only read operations to monitor the status of the device and does not perform any control functions. 1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. <p>NOTE: This pin configuration does not impact the Compact unit.</p>
StatisticSelector	INT	<p>Variable used to obtain statistics for the Modbus network (requests carried out, time between requests, and so on). This data provides information for using <code>StatisticConnector</code> pin within the <code>StatisticCounter</code> DFB in General Purpose library for communication.</p> <p>The following table displays the <code>StatisticSelector</code> value:</p>
	Variable Value	Description
	1	Read statistics, client.
	2	Write statistics, client.

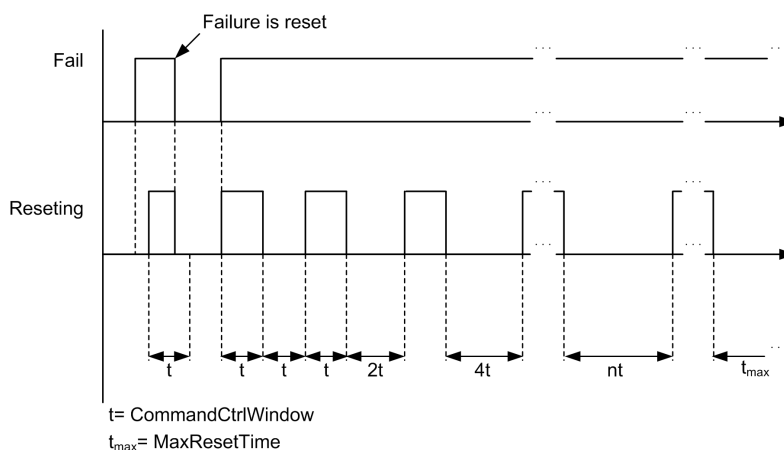
Name	Data type	Description
EngParam ²	MBUCOMPACTNSX- XENGPARAM, page 27	Engineering parameters.
¹ : Parameter available only for <i>MBCompactNSX</i> . ² : Parameter available only for <i>MBUCompactNSX</i> . ³ : Parameter available only for <i>EMCompactNSX</i> .		

MBUCOMPACTNSXENGPARAM

Name	Data type	Description
Refresh	TIME	Refresh time for device data on serial Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when required and have maximum priority.
CommandCtrlWindow	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is issued. The commands controlled are <code>Open</code> and <code>Close</code> . In the event of a <code>ResetFail</code> , this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the <code>Reseting</code> output (to <code>FALSE</code>).
ScanTime	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
Password	STRING	Password is made up of 4 ASCII bytes and is sent internally before executing certain commands. If the password is incorrect, it returns <code>WarningOrder</code> to point this out.
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in <code>CommandCtrlWindow</code> is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in <code>CommandCtrlWindow</code> elapses. The second reset is carried out after <code>CommandCtrlWindow * 2</code> elapses, so on. If the value of <code>CommandCtrlWindow</code> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable Value	Description
	FALSE	Communications are/the device is reset with the <code>ResetFail</code> variable.
	TRUE	Communications are/the device is reset automatically.
ResetMinMax	INT	Resets registers to its minimum/maximum values.

Name	Data type	Description
MaxResetTime	TIME	When in automatic ResetMode, this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.
Scalefactor	REAL	Scale factor for Masterpact NT/NW and Compact NS Micrologix 5.0, 6.0 Or 7.0 trip units is 1. Scale factor for Compact NSX Micrologix 5.2, 5.3, 6.2, 7.2 or 7.3 trip units is 10.

Timing diagram:



Outputs

Output Parameter Description

Parameter	Type	Description
Ready	BOOL	1 = The device is enabled and free of detected errors. This variable is TRUE as long as there are no communication interruptions and the device has not tripped.
Fail	BOOL	1 = A detected failure in the control block or in the device or communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode. NOTE: If a communication interruption occurs, the variables being read from the device ceases to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.
Warning	BOOL	1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset. This signal remains active until the cause of the alarm disappears.
WarningOrder	BOOL	1 = The device has returned an alarm condition as the result of having executed a command. It does not affect the block operation and does not need to be reset. This signal remains active until a different command that ends without an alarm condition is executed.
FailCode	ARRAY [0..2] OF INT	When the Fail output is 1, it holds the code for the detected error. if the Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3-level structure. Refer to Diagnostic Information Management, page 36 for more details.
WarningCode	COMPACTNSXWARNING	Variable holds the alarm code.

Parameter	Type	Description	
		The following table describes the <code>WarningCode</code> information:	
	Parameter	Type	Description
	Alarm201	BOOL	1 = User-defined alarm 201.
	Alarm202	BOOL	1 = User-defined alarm 202.
	Alarm203	BOOL	1 = User-defined alarm 203.
	Alarm204	BOOL	1 = User-defined alarm 204.
	Alarm205	BOOL	1 = User-defined alarm 205.
	Alarm206	BOOL	1 = User-defined alarm 206.
	Alarm207	BOOL	1 = User-defined alarm 207.
	Alarm208	BOOL	1 = User-defined alarm 208.
	Alarm209	BOOL	1 = User-defined alarm 209.
	Alarm210	BOOL	1 = User-defined alarm 210.
	LongTimeProtection	BOOL	1 = Long delay protection prealarm (PAL Ir).
	EarthLeakage	BOOL	1 = Differential protection prealarm (Vigi) (PAL lΔn).
	GroundFault	BOOL	1 = Ground fault protection prealarm (PAL Ig).
	LongTimePickup	BOOL	1 = Protection prealarm.
WarningOrderCode ¹	COMPACTNSXWARNINGORDER	Variable holds the alarm code returned as the result of executing a command. Commands can generate the following diagnostic codes. These codes are returned in word 8020. The following table describes the <code>WarningOrderCode</code> information:	
	Parameter	Type	Description
	Order	BOOL	1 = A follow-up alarm. The device is not responding to the control command within the time specified in <code>CommandCtrlWindow</code> .
	WrongPassword	BOOL	1 = Insufficient user permissions (incorrect password).
WarningOrderCode ¹	MbPadLocked	BOOL	1 = Access violation (the padlock for the Modbus communications interface module is locked).
	InternalWarning	BOOL	1 = Any other positive detected error code represents an internal detected error.
	OutOfOrder	BOOL	1 = The BSCM module is out of service.
	NeedReset	BOOL	1 = The circuit breaker has tripped. It has to reset before the command.
	AlreadyClosed	BOOL	1 = The circuit breaker is already closed.
	AlreadyOpen	BOOL	1 = The circuit breaker is already open.
	AlreadyReset	BOOL	1 = The circuit breaker is already reset.
	ManualMode	BOOL	1 = The control is in Manual mode. Remote commands are not permitted.
	NotPresent	BOOL	1 = No control present.
	InProgress	BOOL	1 = A previous command is still being executed.
	ResetForbidden	BOOL	1 = The reset command is prohibited when SDE has been established.
WarningOrderCode ²	INT	Warning order register for Compact circuit breakers.	
	Parameter	Value	Description
	OrderUnsuccessful	1	Order unsuccessful
	WrongPassword	2	Insufficient user rights (incorrect password)

Parameter	Type	Description	
	ManualMode	4	Manual mode
	AlreadyClosed	13	Already closed
	AlreadyOpen	14	Already open
	AlreadyReset	15	Already reset
	ResourceNotPresent	16	Actuator not present
	InhibitModeOn	17	Inhibit mode on
	OutOfOrder	18	Resource is out of order
	NeedReset	19	Need reset
	InProgress	20	In progress
	ResetForbidden	21	Reset forbidden
	IFMPadlocked	22	IFM locking pad is locked
Running	BOOL	1 = The Compact unit is closed.	
ExtControlled	BOOL	<p>1 = The device is being controlled from a source external (for example, from the console, from a push button panel, or from the monitoring system) to the system.</p> <p>NOTE: To calculate the state of this signal, use the <code>ControlCommand</code> signal and the <code>Owner</code> variable. You cannot use this signal as a <code>ControlCommand</code> input.</p>	
Reseting	BOOL	<p>1 = A reset is being carried out.</p> <p>The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected error.</p> <p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected error within the time period defined in <code>CommandCtrlWindow</code>.</p> <p>If the detected error is reset, the <code>Fail</code> and <code>Reseting</code> output variables are reset (set to FALSE). On the other hand, if the detected error is not reset, the <code>Reseting</code> variable is set to FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Refer to the Timing diagram below.</p>	
Status	COMPACTSTATUS	<p>The structure holds data containing the information that the block extracts from the status variable of the circuit breaker and communicating electrical drive.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	Open	BOOL	<p>OFF state.</p> <p>1 = The circuit breaker is open.</p>
	Closed	BOOL	<p>ON state.</p> <p>1 = The circuit breaker is closed.</p>
	Trip	BOOL	<p>SD trip signal.</p> <p>0 = Not tripped.</p> <p>1 = The circuit breaker has tripped due to an inappropriate electrical condition or a leakage-caused trip.</p>
	TripElec	BOOL	<p>Detected fault trip signal - SDE.</p> <p>0 = Not tripped.</p> <p>1 = The circuit breaker has tripped due to an inappropriate electrical condition.</p>
	NotAvailable	BOOL	<p>Availability of device status value.</p> <p>0 = Status available</p> <p>1 = Status not available due to device internal detected error.</p>

Parameter	Type	Description		
	Info	INT	Device status. Refer to Info table.	
	MotorAvailable	BOOL	Availability of motor mechanism. 0 = Not available 1 = Available	
	MotorMode	BOOL	Motor operating mode. 0 = Manual 1 = Auto	
StatisticConnector	STATISTICCONNECTOR	Information used with Modbus communication to obtain statistics on the Modbus network (requests carried out, time between requests, and so on). This structure has been created to use together with the StatisticCounter DFB in General Purpose library for communication. The following table describes the StatisticConnector:		
	Parameter	Type	Description	
	Start	BOOL	1 = Operation has started.	
	EndOk	BOOL	1 = Operation has ended correctly.	
	EndNOk	BOOL	1 = Operation has ended with a detected error.	
	PartialTime	DINT	Partial time.	
COMPACT_MEA	COMPACT_MEA_DDT	Data structure with measurement information. Whether this data is available depending on the specific Micrologic model: A= Ammeter, E=Energy, M=Motor, 30/31/40/41=Type of system (W3313; refer to the Communication Manual for the Compact unit). The following table describes the COMPACT_MEA_DDT type:		
	Parameter	Availability	Type	Description
	CurrentI1	A/E/M	REAL	Current, instantaneous, phase 1.
	CurrentI2	A/E/M	REAL	Current, instantaneous, phase 2.
	CurrentI3	A/E/M	REAL	Current, instantaneous, phase 3.
	ResidualCurrentIN	A/E, 30/41	REAL	Current, instantaneous, N.
	TrippingCause	–	INT	–
	OverrunAlarms	–	INT	–
	PreAlarms	–	INT	–
	UserAlarms	–	INT	–
	TripCounter	–	INT	–
	ResetDevice	–	INT	–
COMPACT_MEExt	COMPACT_MEExt_DDT	Data structure with extended measurement information. Whether this data is available depending on the specific Micrologic model: E, P, H. The following table describes the COMPACT_MEExt_DDT type:		
	VoltageU21	E/M	REAL	Voltage, phases 1-2.
	VoltageU32	E/M	REAL	Voltage, phases 2-3.
	VoltageU13	E/M	REAL	Voltage, phases 1-3.
	VoltageV1	E, 40/41	REAL	Voltage, phase 1-N.
	VoltageV2	E, 40/41	REAL	Voltage, phase 2-N.
	VoltageV3	E, 40/41	REAL	Voltage, phase 3-N.
	Frequency	E/M	REAL	Frequency (derived from phase 1).
	TotalActivePower	E/M	REAL	Total real power.

Parameter	Type	Description		
	TotalReactivePower	E/M	REAL	Total reactive power.
	TotalApparentPower	E/M	REAL	Total apparent power.
	ActivePowerL1	E, 30/41	REAL	Real power, phase 1.
	ActivePowerL2	E, 30/41	REAL	Real power, phase 2.
	ActivePowerL3	E, 30/41	REAL	Real power, phase 3.
	ReactivePowerL1	E, 30/41	REAL	Reactive power, phase 1.
	ReactivePowerL2	E, 30/41	REAL	Reactive power, phase 2.
	ReactivePowerL3	E, 30/41	REAL	Reactive power, phase 3.
	ApparentPowerL1	E, 30/41	REAL	Apparent power, phase 1.
	ApparentPowerL2	E, 30/41	REAL	Apparent power, phase 2.
	ApparentPowerL3	E, 30/41	REAL	Apparent power, phase 3.
	CosPhi	E/M	REAL	Cos phi power factor.
	ActiveEnergy	E/M	REAL	Real power consumption.
	ReactiveEnergy	E/M	REAL	Reactive power consumption.
	ApparentEnergyTotal	E/M	REAL	Total apparent power consumption.
	PositiveActiveEnergy	E/M	REAL	Positive real power Ea+ (kWh).
	NegativeActiveEnergy	E/M	REAL	Negative real power Ea- (kWh).
	PositiveReactiveEnergy	E/M	REAL	Positive reactive power Er+ (kVARh).
	NegativeReactiveEnergy	E/M	REAL	Negative reactive power Er- (kVARh).
COMPACT_MEAEExt1	COMPACT_MEAEExt1_DDT	Data structure with extended measurement information. Whether this data is available depending on the specific Micrologic model: P, H. The following table describes the COMPACT_MEAEExt1_DDT type:		
	THDVoltageL1ToL2	E/M	REAL	THD, voltage, 1-2.
	THDVoltageL2ToL3	E/M	REAL	THD, voltage, 2-3.
	THDVoltageL1ToL3	E/M	REAL	THD, voltage, 1-3.
	THDVoltageL1ToNeutral	E, 40/41	REAL	THD, voltage, 1-N.
	THDVoltageL2ToNeutral	E, 40/41	REAL	THD, voltage, 2-N.
	THDVoltageL3ToNeutral	E, 40/41	REAL	THD, voltage, 3-N.
	THDCurrentL1	E/M	REAL	THD, current, phase 1.
	THDCurrentL2	E/M	REAL	THD, current, phase 2.
	THDCurrentL3	E/M	REAL	THD, current, phase 3.
1: These parameters are available only with MBCompactNSX.				
2: These parameters are available only with MBUCompactNSX.				

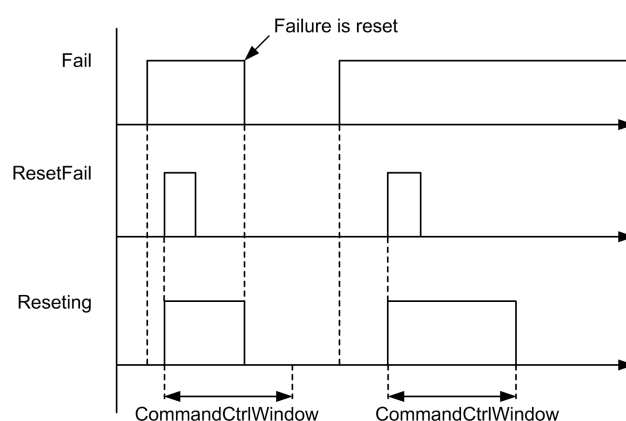
Info Structure

Code with the information shown on the Control operator screen. The following table describes the Info structure:

Variable value	Status
2	Waiting for Ready signal.
6	Compact unit open.

Variable value	Status
7	Compact unit closed.
10	Waiting <code>ControlCommand</code> . It needs to be set to 1.
11	Missing <code>EnabledDFB</code> .
12	Missing <code>CommunicationOK</code> . communication interruption.
13	The status register is not available.
24	Remove <code>ResetFail</code> . Needs to be reset again.
81	Missing <code>ResetFail</code> . Inoperable device.
82	A reset is required.

Timing diagram:



Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
COMPACT_ST	COMPACT_ST_DDT	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is usable from the HMI/SCADA system.
COMPACT_CFG	COMPACT_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
WorkMemory	ANY_ARRAY_INT	Array used for Modbus communications. This variable is to be used with a Modbus port that serializes Modbus requests in an optimum manner.

COMPACT_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	<p>Device control.</p> <p>Enables to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0).</p> <p>If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.</p>

COMPACT_CFG_DDT Type

Name	Type	Description
DataStatus	WORD	Information on the device status. Information on the <code>Status</code> output structure.
Info	INT	Code with information on statuses and required actions.
WarningCode	WORD	Information on the alarm code. Takes the values from the <code>WarningCode</code> output.
WarningCodeOrder	WORD	Information on the command alarm code. Takes the values from the <code>WarningCodeOrder</code> output.
FailCode0	INT	Code of last level 0 detected error. Indicates which detected error has occurred, <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates which detected error has occurred, <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates which detected error has occurred, <code>FailCode[2]</code> .

COMPACT_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
2	Refer to the <code>Closed</code> status in the <code>Status</code> output pin, page 28.
3	Inoperable device.
4	Alarm on device or repetitive detected fault alarm requires resetting.
5	communication interruption.
6	Requires resetting. A <code>ResetFail</code> is required.
7	Externally controlled.
8	Refer to the <code>Resetting</code> output pin, page 28.
9	Refer to the <code>EnableDFB</code> input pin (see Modicon Libraries General Purpose, Devices Components User Guide).

COMPACT_ST.CFG Word Structure

Bit	Description
0	Refer to the <code>ResetFail</code> input pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	<code>Owner</code> .
3	Refer to the <code>Closed</code> status in the <code>Status</code> output pin, page 28.
6	Refer to the <code>Open</code> status in the <code>Status</code> output pin, page 28.
7	Refer to the <code>ControlCommand</code> input pin (see Modicon Libraries General Purpose, Devices Components User Guide).

COMPACT_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the <code>Open</code> status in the <code>Status</code> output pin, page 28.
1	Refer to the <code>Closed</code> status in the <code>Status</code> output pin, page 28.
2	Refer to the <code>Trip</code> status in the <code>Status</code> output pin, page 28.
3	Refer to the <code>TripElec</code> status in the <code>Status</code> output pin, page 28.
4	Refer to the <code>NotAvailable</code> status in the <code>Status</code> output pin, page 28.
5	Refer to the <code>MotorAvailable</code> status in the <code>Status</code> output pin, page 28.
6	Refer to the <code>MotorMode</code> status in the <code>Status</code> output pin, page 28.

COMPACT_CFG.WarningCode Word Structure

Bit	Description
0	Refer to the <code>Alarm201</code> code in the <code>WarningCode</code> output pin, page 28.
1	Refer to the <code>Alarm202</code> code in the <code>WarningCode</code> output pin, page 28.
2	Refer to the <code>Alarm203</code> code in the <code>WarningCode</code> output pin, page 28.
3	Refer to the <code>Alarm204</code> code in the <code>WarningCode</code> output pin, page 28.
4	Refer to the <code>Alarm205</code> code in the <code>WarningCode</code> output pin, page 28.
5	Refer to the <code>Alarm206</code> code in the <code>WarningCode</code> output pin, page 28.
6	Refer to the <code>Alarm207</code> code in the <code>WarningCode</code> output pin, page 28.
7	Refer to the <code>Alarm208</code> code in the <code>WarningCode</code> output pin, page 28.
8	Refer to the <code>Alarm209</code> code in the <code>WarningCode</code> output pin, page 28.
9	Refer to the <code>Alarm210</code> code in the <code>WarningCode</code> output pin, page 28.
10	Refer to the <code>LongTimeProtection</code> code in the <code>WarningCode</code> output pin, page 28.
11	Refer to the <code>EarthLeakage</code> code in the <code>WarningCode</code> output pin, page 28.
12	Refer to the <code>GroundFault</code> code in the <code>WarningCode</code> output pin, page 28.

COMPACT_CFG.WarningOrderCode Word Structure for MBCompactNSX

Bit	Description
0	Refer to the <code>Order</code> code in the <code>WarningOrderCode</code> output pin, page 28.
1	Refer to the <code>WrongPassword</code> code in the <code>WarningOrderCode</code> output pin, page 28.
2	Refer to the <code>MbPadLocked</code> code in the <code>WarningOrderCode</code> output pin, page 28.
3	Refer to the <code>InternalWarning</code> code in the <code>WarningOrderCode</code> output pin, page 28.
4	Refer to the <code>OutOfOrder</code> code in the <code>WarningOrderCode</code> output pin, page 28.
5	Refer to the <code>NeedReset</code> code in the <code>WarningOrderCode</code> output pin, page 28.
6	Refer to the <code>AlreadyClosed</code> code in the <code>WarningOrderCode</code> output pin, page 28.
7	Refer to the <code>AlreadyOpen</code> code in the <code>WarningOrderCode</code> output pin, page 28.
8	Refer to the <code>AlreadyReset</code> code in the <code>WarningOrderCode</code> output pin, page 28.
9	Refer to the <code>ManualMode</code> code in the <code>WarningOrderCode</code> output pin, page 28.
10	Refer to the <code>NotPresent</code> code in the <code>WarningOrderCode</code> output pin, page 28.
11	Refer to the <code>InProgress</code> code in the <code>WarningOrderCode</code> output pin, page 28.
12	Refer to the <code>ResetForbidden</code> code in the <code>WarningOrderCode</code> output pin, page 28.

COMPACT_CFG.WarningOrderCode Word Structure for MBUCompactNSX

Bit	Description
0	Refer to the <code>OrderUnsuccessful</code> code in the <code>WarningOrderCode²</code> output pin, page 28.
1	Refer to the <code>WrongPassword</code> code in the <code>WarningOrderCode</code> output pin, page 28.
2	Refer to the <code>ManualMode</code> code in the <code>WarningOrderCode</code> output pin, page 28.
3	Refer to the <code>AlreadyClosed</code> code in the <code>WarningOrderCode</code> output pin, page 28.
4	Refer to the <code>AlreadyOpen</code> code in the <code>WarningOrderCode</code> output pin, page 28.
5	Refer to the <code>AlreadyReset</code> code in the <code>WarningOrderCode</code> output pin, page 28.
6	Refer to the <code>ResourceNotPresent</code> code in the <code>WarningOrderCode</code> output pin, page 28.
7	Refer to the <code>InhibitModeOn</code> code in the <code>WarningOrderCode</code> output pin, page 28.
8	Refer to the <code>OutOfOrder</code> code in the <code>WarningOrderCode</code> output pin, page 28.
9	Refer to the <code>NeedReset</code> code in the <code>WarningOrderCode</code> output pin, page 28.
10	Refer to the <code>InProgress</code> code in the <code>WarningOrderCode</code> output pin, page 28.
11	Refer to the <code>ResetForbidden</code> code in the <code>WarningOrderCode</code> output pin, page 28.
12	Refer to the <code>IFMPadlocked</code> code in the <code>WarningOrderCode</code> output pin, page 28.

Diagnostics Information Management

Overview

The diagnostic codes that the device can return are read on the `FailCode` output variable.

Modbus Communication Diagnostics Codes

This code indicates that communications have not been established and can be reset:

- FailCode[0]: 16#0002
- FailCode[1]: 16#0000
- FailCode[2]: 16#0004

This code indicates that communications have not been established between IFE and TRIP UNIT and can be reset:

- FailCode[0]: 16#0004
- FailCode[1]: 16#0000
- FailCode[2]: 16#0004

or

- FailCode[0]: 16#0007
- FailCode[1]: 16#0200
- FailCode[2]: 16#0001

or

- FailCode[0]: 16#01FF
- FailCode[1]: 16#0200
- FailCode[2]: 16#0001

After the communications have been established, check Modbus client diagnostic codes for FailCode [0] and FailCode [1]. The components make a distinction between detected read request problems and write request problems:

- FailCode[2]: 16#0001 Read
- FailCode[2]: 16#0002 Write

Diagnostics Code Example

For a detected error, the code is:

- FailCode[1]: 16#0000
- FailCode[2]: 16#0005

The field status can be reset.

The FailCode[0] can have one of the following codes:

Bit	Value	FailCode[0]—Cause of Trip Operation
0	1	Long-time protection I _r .
1	2	Short-time protection I _{sd} .
2	4	Instantaneous protection I _i .
3	8	Ground-fault protection I _g .
4	16	Earth leakage (Vigi) protection I _{Δn} .
5	32	Integrated instantaneous protection.
6	64	Internal detected failure (Stop).
9	512	Instantaneous protection with earth leakage (Vigi) trip unit.
10	1024	Unbalanced motor protection.
11	2048	Motor jam protection.
12	4096	Underload motor protection.

Bit	Value	FailCode[0]—Cause of Trip Operation
13	8192	Longstart motor protection.
14	16384	Reflex tripping protection.

MASTERPACT Profile

What's in This Chapter

Description	39
DFB Representation	39
Inputs	41
Outputs	43
Inputs/Outputs	50
Diagnostics Information Management	54

Overview

This chapter describes the DFBs of the MASTERPACT profile.

Description

General

The MASTERPACT profile is used to manage the Masterpact and Masterpact draw-out/chassis electrical protection circuit breakers.

NOTE: MBMASTERPACT, MBMASTERPACTC, EMMASTERPACT and EMMASTERPACTC are deprecated functions.

Function Description

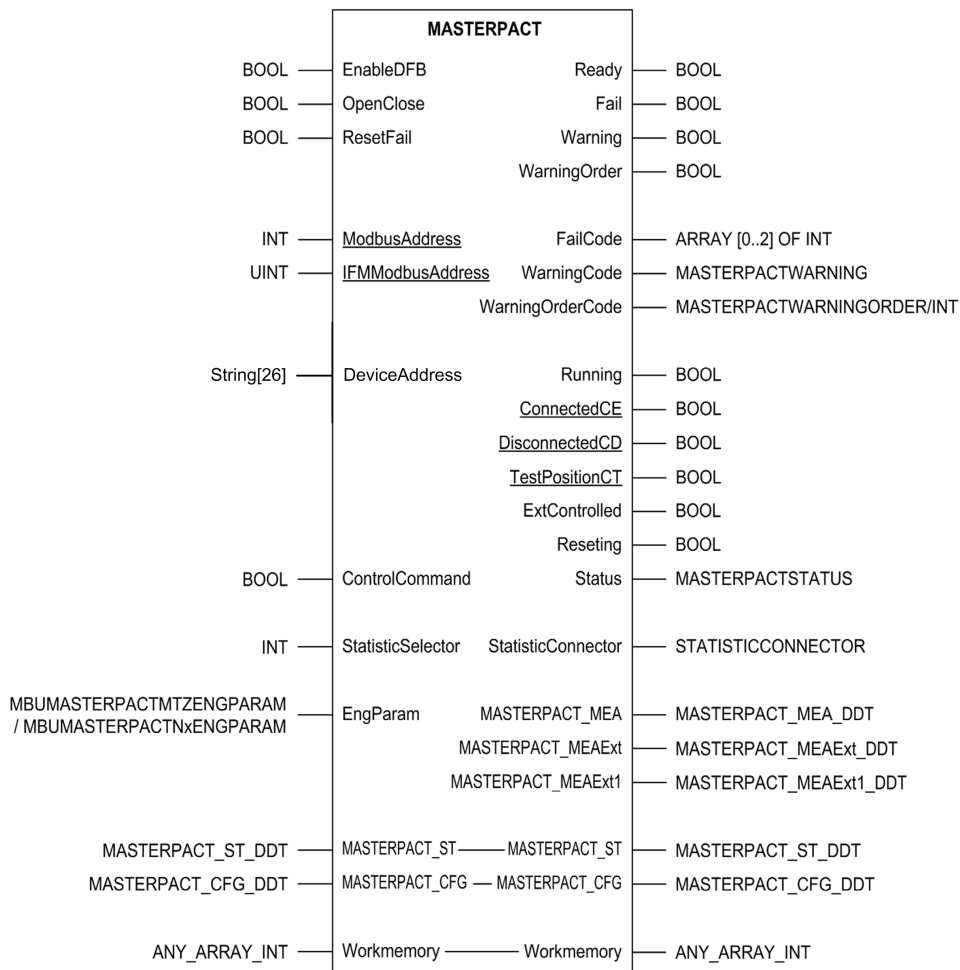
The main functions of the DFB are described in the following table:

Function	Description
Remote resetting	Allows resetting of the device.
Open/Close	Allows circuit breaker to open and close.
Monitoring	Allows the device measurement data to be monitored.

DFB Representation

Representation

The following figure represents the function module of Masterpact profile:

**NOTE:**

- The underlined parameters are specific to profiles supported by DFB.
- Datatype of WarningOrderCode output pin for MBMASTERPACT and MBMASTERPACTC is MASTERPACTSTATUS. Similarly, for MBUMASTERPACTMTZ, MBUMASTERPACTMTZC, MBUMASTERPACTNx and MBUMASTERPACTNxC the datatype is INT

The table shows the parameters available for specific DFBs:

Parameters		Components							
		Modbus		Modbus over ULP				Ethernet	
		MBMASTERPACT (Deprecated)	MBMASTERPACTC (Deprecated)	MBUMASTERPACTMTZ	MBUMASTERPACTMTZC	MBUMASTERPACTNx	MBUMASTERPACTNxC	EMMASTERPACT (Deprecated)	EMMASTERPACTC (Deprecated)
In-puts	ModBusAddress	X	X	—	—	—	—	—	—
	IFMModBusAddress	—	—	X	X	X	X	—	—
	DeviceAddress	—	—	—	—	—	—	X	X
Out-puts	ConnectedCE	—	X	—	X	—	X	—	X
	DisconnectedCD	—	X	—	X	—	X	—	X
	TestPositionCT	—	X	—	X	—	X	—	X
X : Parameter is available									
— : Parameter is not available									

NOTICE

COMMUNICATION INTERRUPTION

Provide possible combination of communication interruptions between Masterpact chassis and PLC. Otherwise, it leads to loss of data.

Failure to follow these instructions can result in equipment damage.

The IO module rotary position has to be set to have Cradle Management for *Masterpact BCM-ULP*.

Function	User Defined Applications	Predefined Application Selected									
		1	2	3	4	5	6	7	8	9(IO1)	9(IO2)
Monitoring	Cradle management	X	–	–	–	–	–	–	–	–	X

X : User defined application available
 – : User defined application not available

NOTE:

- If a configuration of IO module is changed the *EnableDFB* pin has to be reset.
- *EMMasterPactC* and *MBMasterPactC* has to be used only if IO module is configured for cradle management application.
- *MBMasterPactC*, *MBUMasterPactMTZC* and *MBUMasterPactNxC* has to be used only if IO module is configured for cradle management application.
- Cradle management is always IO1 for EIFE.
- For how to create project using Masterpact MTZ/Masterpact Nx circuit breakers, refer Communication technologies (see Modicon Libraries General Purpose, Devices Components User Guide).
- If *IFMModbusAddress* is not configured then default value of 255 is considered.
- IFM v2 and above hardware supports these control functions.

Inputs

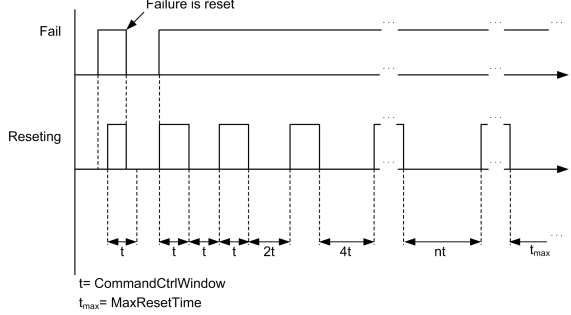
Input Parameter Description

Name	Data type	Description
EnableDFB	BOOL	This input enables the normal execution of the control block. <ul style="list-style-type: none"> • 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. • 1 = Enables communications with the devices for their operation. Public variable values are loaded during the first enabling cycle.
OpenClose	BOOL	This input controls the circuit breaker. <ul style="list-style-type: none"> • Rising edge on pin = Closes the circuit breaker. • Falling edge on pin = Opens the circuit breaker.
ResetFail	BOOL	1 = Resets the <i>Fail</i> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <i>ControlCommand</i> is 1.
ModbusAddress ¹	INT	Device address within the Modbus network. You can find this variable in Modbus communications.

Name	Data type	Description
IFModbusAddress ²	UINT	Modbus address of the IFM Module.
DeviceAddress ³	String[26]	Device address within the Ethernet network. Depending on the platform, the following definitions apply:
	Platform	IP Addressing DeviceAddress (variable)
	M340	'{IP}ID'
	M580	'{IP}ID'
	Quantum	'{IP}ID'
	NOTE: ID is 255.	
ControlCommand	BOOL	<p>Indicates to the DFB whether the Masterpact is being controlled locally or from a source external to the DFB.</p> <ul style="list-style-type: none"> 0 = Performs only read operations to monitor the status of the device and does not perform any control functions. 1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. <p>NOTE: This pin configuration does not impact the Masterpact unit.</p>
StatisticSelector	INT	<p>Variable used to obtain statistics for the Modbus network (requests carried out, time between requests, and so on). This data provides information for using <code>StatisticConnector</code> pin within the <code>StatisticCounter</code> DFB in General Purpose library for communication.</p> <p>The following list displays the <code>StatisticSelector</code> value:</p> <ul style="list-style-type: none"> 1 = Statistics of read request. 2 = Statistics of write request.
EngParam ²	MBUMASTERPACTMTZENGPARAM/ MBUMASTERPACTNxENGPARAM, page 42	Engineering parameters
¹ : Parameter available only for <i>MBMASTERPACT</i> and <i>MBMASTERPACTC</i> . ² : Parameter available only for <i>MBUMASTERPACTMTZ</i> , <i>MBUMASTERPACTMTZC</i> , <i>MBUMASTERPACTNx</i> and <i>MBUMASTERPACTNxC</i> . .		

MBUMASTERPACTMTZENGPARAM/ MBUMASTERPACTNxENGPARAM

Name	Data type	Description
Refresh	TIME	<p>Refresh time for device data on serial Modbus communications.</p> <p>NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed.</p>
CommandCtrlWindow	TIME	<p>Control time for operations. This is the time that the block waits for the operations to be carried out by the device.</p> <p>If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is issued.</p> <p>The commands controlled are <code>Open</code> and <code>Close</code>. In the event of a <code>ResetFail</code>, this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the <code>Reseting</code> output (to FALSE).</p>
ScanTime	TIME	<p>Allows you to configure the time for which the alarm signals are kept active.</p> <p>Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.</p>
Password	STRING	Password is made up of 4 ASCII bytes. This password is sent internally before executing certain commands. If the

Name	Data type	Description
		password is incorrect, it returns <code>WarningOrder</code> to point this out.
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in <code>CommandCtrlWindow</code> is used to define the interval after which a reset has to be carried out. The first reset is carried out after the time defined in <code>CommandCtrlWindow</code> elapses. The second reset is carried out after <code>CommandCtrlWindow * 2</code> elapses, so on. If the value of <code>CommandCtrlWindow</code> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications are/the device is reset with the <code>ResetFail</code> variable.
	TRUE	Communications are/the device is reset automatically.
MaxResetTime	TIME	When in automatic <code>ResetMode</code> , this variable is used to define the maximum time that can elapse between 2 consecutive resets. Timing diagram:  $t = \text{CommandCtrlWindow}$ $t_{\max} = \text{MaxResetTime}$
Scalefactor	REAL	Scale factor for Masterpact NT/NW and Compact NSX Micrologix 5.0, 6.0 or 7.0 trip units is 1. Scale factor for Compact NSX Micrologix 5.2, 5.3, 6.2, 7.2 or 7.3 trip units is 10.

Outputs

Output Parameter Description

Parameter	Type	Description
Ready	BOOL	1 = The device is enabled and free of detected errors. This variable is TRUE as long as there is chassis connected (CE), no communication interruption, no device resetting, and the device is not tripped.
Fail	BOOL	1 = A detected failure in the control block or in the device or communication interruption. To reset the <code>Fail</code> output pin, the <code>ResetFail</code> input has to be activated. The last detected error code is shown on <code>FailCode</code> . NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.
Warning	BOOL	1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset. This signal remains active until the configured scan time.
WarningOrder	BOOL	1 = The device has returned an alarm condition as the result of executing a command. It does not affect the block operation and does not need to be reset.

Parameter	Type	Description	
		This signal remains active until the configured scan time.	
FailCode	ARRAY [0..2] OF INT	<p>When the Fail output is 1, it holds the code for the detected error.</p> <p>If the Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3-level structure.</p> <p>Refer to Diagnostics Information Management, page 54 for more details.</p>	
WarningCode	MASTERPACTWARNING	<p>Variable holds the alarm code.</p> <p>The following table describes the WarningCode information:</p>	
	Parameter	Type	Description
	LongTimePickUp	BOOL	1 = Long-delay trip threshold.
	CurrentUnbalance	BOOL	1 = Current unbalance.
	MaxCurrentPhase1	BOOL	1 = Maximum current on phase 1.
	MaxCurrentPhase2	BOOL	1 = Maximum current on phase 2.
	MaxCurrentPhase3	BOOL	1 = Maximum current on phase 3.
	MaxCurrentNeutral	BOOL	1 = Maximum current on neutral.
	MinVoltage	BOOL	1 = Minimum voltage.
	MaxVoltage	BOOL	1 = Maximum voltage.
	VoltageUnbalance	BOOL	1 = Voltage unbalance.
	MaxPower	BOOL	1 = Maximum power.
	ReversePower	BOOL	1 = Reverse power.
	MinFrequency	BOOL	1 = Minimum frequency.
	MaxFrequency	BOOL	1 = Maximum frequency.
	PhaseRotation	BOOL	1 = Phase rotation.
	LoadSheddingCurrent	BOOL	1 = Current-based load shedding.
	LoadSheddingPower	BOOL	1 = Power-based load shedding.
	GroundFault	BOOL	1 = Ground fault notification.
	EarthLeakage	BOOL	1 = Differential alarm (Vigi).
	Discordance	BOOL	1 = Chassis status discordance.
	UnknownIOConfigura- tion	BOOL	1 = Unknown IO configuration.
WarningOrderCode ¹	MASTERPACTWARNING- ORDER	<p>Variable holds the alarm code returned as the result of executing a command. Commands can generate the following diagnostic codes.</p> <p>The following table describes the WarningOrderCode information for MBMASTERPACT and MBMASTERPACTC:</p>	
	Parameter	Type	Description
	Order	BOOL	1 = Follow-up alarm. The device is not responding to the control command within the time specified in CommandCtrlWindow.
	WrongPassword	BOOL	1 = Insufficient user permissions (incorrect password).
	InternalWarning	BOOL	1 = Any other positive diagnostic codes represent an internal detected error.
	ManualMode	BOOL	1 = The control is in Manual mode. Remote commands are not permitted.
	IncorrectCoilValue	BOOL	1 = The DFB has an internal detected error. An invalid number of windings are requested during a write operation.
	IncorrectNbrOfParam	BOOL	1 = The DFB has an internal detected error. An invalid number of parameters are sent for a 57400 command.
WarningOrderCode ²	INT	Variable holds the alarm code returned as the result of executing a command. Commands can generate the following diagnostic codes.	

Parameter	Type	Description	
			The following table describes the <code>WarningOrderCode</code> information for <code>MBUMASTERPACTMTZ</code> , <code>MBUMASTERPACTMTZC</code> , <code>MBUMASTERPACTNx</code> and <code>MBUMASTERPACTNxC</code> :
	Parameter	Value	Description
	<code>Order</code>	1	1 = Follow-up alarm. The device is not responding to the control command within the time specified in <code>CommandCtrlWindow</code> .
	<code>WrongPassword</code>	2	1 = Insufficient user permissions (incorrect password).
	<code>InternalWarning</code>	3	1 = Any other positive diagnostic codes represent an internal detected error.
	<code>ManualMode</code>	4	1 = The control is in Manual mode. Remote commands are not permitted.
	<code>IncorrectCoilValue</code>	5	1 = The DFB has an internal detected error. An invalid number of windings are requested during a write operation.
	<code>IncorrectNumberOfParam</code>	6	1 = The DFB has an internal detected error. An invalid number of parameters are sent for a 57400 command.
	<code>InProgress</code>	7	Command in progress.
	<code>IFEPadlocked</code>	8	Access violation (IFE locking pad locked or EIFE intrusive command mode is locked).
	<code>IFMPadlocked</code>	9	Unable execute service (IFM locking pad locked).
	<code>ResourceNotExist</code>	10	Resource does not exist.
	<code>TimeoutDuringCommand</code>	11	Timeout during command.
	<code>ResetBeforeCommand</code>	12	Circuit breaker tripped, rests before commands.
	<code>AlreadyClosed</code>	13	Circuit breaker already closed.
	<code>AlreadyOpen</code>	14	Circuit breaker already open.
	<code>AlreadyReset</code>	15	Circuit breaker already reset.
	<code>NotPresent</code>	16	Actuator not present.
	<code>InhibitModeOn</code>	17	Inhibit mode on.
<code>Running</code>	BOOL	1 = The Masterpact unit is closed.	
<code>ConnectedCE³</code>	BOOL	1 = In case of Masterpact units with draw-out/chassis enclosures, a value of 1 signal that the Masterpact unit is mounted and connected for normal operation.	
<code>DisconnectedCD³</code>	BOOL	1 = In case of Masterpact units with draw-out/chassis enclosures, a value of 1 signal that the Masterpact unit is disconnected and cannot be actuated.	
<code>TestPositionCT³</code>	BOOL	1 = In case of Masterpact units with draw-out/chassis enclosures, a value of 1 signal that the Masterpact unit is in its Test position and can be actuated. Although, it does not actually close the contact.	
<code>ExtControlled</code>	BOOL	1 = The device is being controlled from a source external (for example, from the console, from a push-button panel, or from the monitoring system) to the program. NOTE: To calculate the state of this signal, use the <code>ControlCommand</code> signal and the <code>Owner</code> variable. You cannot use this signal as a <code>ControlCommand</code> input.	
<code>Resetting</code>	BOOL	1 = A reset is being carried out. The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected error. When a device or communication reset is carried out with <code>ResetFail</code> , the DFB tries to reset the detected error within the time period defined in <code>CommandCtrlWindow</code> . If the detected error is reset, the <code>Fail</code> and <code>Resetting</code> output variables are reset (set to FALSE). If the detected error is not reset, the <code>Resetting</code> variable is FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based. Refer to the Timing diagram below.	
<code>Status</code>	MASTERPACTSTATUS	The structure holds data containing the information that the block extracts from the status variable of the circuit breaker and communicating electrical drive.	

Parameter	Type	Description		
		The following table describes the status information:		
	Parameter	Type	Description	
	Open	BOOL	OFF state. 0 = The circuit breaker is open.	
	Closed	BOOL	ON state. 1 = The circuit breaker is closed.	
	TripSD	BOOL	SD trip indication contact 0 = Not tripped. 1 = Circuit breaker is tripped due to electrical fault, shunt trip, or push-to-trip.	
	TripSDE	BOOL	SDE fault trip indication contact 0 = Circuit breaker is not tripped on electrical fault. 1 = Circuit breaker is tripped due to electrical fault (including ground-fault test and earth-leakage test).	
	Discharged	BOOL	1 = Discharged springs in motor-equipped devices.	
	Charged	BOOL	1 = Charged springs in motor-equipped devices.	
	NotReadyToClose	BOOL	1 = Not ready to close.	
	ReadyToClose	BOOL	1 = Ready to close.	
	NotAvailable	BOOL	1 = The status of the circuit breaker is unavailable.	
	Info	INT	Indicates device status.	
	TrippingCause	INT	Cause of trip operation.	
	TrippingCauseExt	INT	Cause of trip operation extended.	
StatisticConnector	STATISTICCONNECTOR	Information used with Modbus communication to obtain statistics on the Modbus network (requests carried out, time between requests, and so on). This structure has been created to use together with the <i>StatisticCounter</i> DFB in General Purpose library for communication. The following table describes the <i>StatisticConnector</i> :		
	Parameter	Type	Description	
	Start	BOOL	1 = Operation has started.	
	EndOk	BOOL	1 = Operation has ended correctly.	
	EndNok	BOOL	1 = Operation has ended with a detected error.	
	PartialTime	DINT	Partial time.	
MASTERPACT_MEA	MASTERPACT_MEA_DDT	Data structure with measurement information. Whether this data is available depends on the specific Micrologic model: A=Ammeter, P=Power meter, H=Harmonics, 30/31/40/41=Type of system. The following table describes the <i>MASTERPACT_MEA_DDT</i> type:		
	Parameter	Availability	Type	Description
	CurrentI1	A/P/H	REAL	Current, instantaneous, phase 1.
	CurrentI2	A/P/H	REAL	Current, instantaneous, phase 2.
	CurrentI3	A/P/H	REAL	Current, instantaneous, phase 3.
	ResidualCurrentIN	A/P/H, 41	REAL	Current, instantaneous, N.
	OverrunAlarms	–	INT	Overrun alarms.
	OverrunAlarmsExt	–	INT	Overrun alarms extended.
	OverrunAlarmsExt2	–	INT	Overrun alarms extended 2.
	TripCounter	–	DINT	Trip counter.

Parameter	Type	Description		
MASTERPACT_MEAEExt	MASTERPACT_MEAEExt_DDT	Data structure with measurement information. Whether this data is available depends on the specific Micrologic model: A, E, P, H. The following table describes the MASTERPACT_MEAEExt_DDT type:		
	Parameter	Availability	Type	Description
	VoltageU21	P/H	REAL	Voltage, phases 1-2.
	VoltageU32	P/H	REAL	Voltage, phases 2-3.
	VoltageU13	P/H	REAL	Voltage, phases 1-3.
	VoltageV1	P/H, 40/41	REAL	Voltage, phase 1-N.
	VoltageV2	P/H, 40/41	REAL	Voltage, phase 2-N.
	VoltageV3	P/H, 40/41	REAL	Voltage, phase 3-N.
	Frequency	P/H	REAL	Frequency (derived from phase 1).
	TotalActivePower	P/H	REAL	Total real power.
	TotalReactivePower	P/H	REAL	Total reactive power.
	TotalApparentPower	P/H	REAL	Total apparent power.
	ActivePowerL1	P/H, 40/41	REAL	Real power, phase 1.
	ActivePowerL2	P/H, 40/41	REAL	Real power, phase 2.
	ActivePowerL3	P/H, 40/41	REAL	Real power, phase 3.
	ReactivePowerL1	P/H, 40/41	REAL	Reactive power, phase 1.
	ReactivePowerL2	P/H, 40/41	REAL	Reactive power, phase 2.
	ReactivePowerL3	P/H, 40/41	REAL	Reactive power, phase 3.
	ApparentPowerL1	P/H, 40/41	REAL	Apparent power, phase 1.
	ApparentPowerL2	P/H, 40/41	REAL	Apparent power, phase 2.
	ApparentPowerL3	P/H, 40/41	REAL	Apparent power, phase 3.
	CosPhi	H	REAL	Cos phi power factor.
	ActiveEnergy	P/H	REAL	Real power consumption.
	ReactiveEnergy	P/H	REAL	Reactive power consumption.
	ApparentEnergyTotal	P/H	REAL	Total apparent power consumption.
	PositiveActiveEnergy	P/H	REAL	Positive real power Ea+ (kWh).
	NegativeActiveEnergy	P/H	REAL	Negative real power Ea- (kWh).
	PositiveReactiveEnergy	P/H	REAL	Positive reactive power Er+ (kVarh).
	NegativeReactiveEnergy	P/H	REAL	Negative reactive power Er- (kVarh).
MASTERPACT_MEAEExt1	MASTERPACT_MEAEExt1_DDT	Data structure with measurement information. Whether this data is available depends on the specific Micrologic model: P, H. The following table describes the MASTERPACT_MEAEExt1_DDT type:		
	Parameter	Availability	Type	Description
	THDVoltageL1ToL2	H	REAL	THD, voltage, 1-2.
	THDVoltageL2ToL3	H	REAL	THD, voltage, 2-3.
	THDVoltageL1ToL3	H	REAL	THD, voltage, 1-3.
	THDVoltageL1ToNeutral	H	REAL	THD, voltage, 1-N.
	THDVoltageL2ToNeutral	H	REAL	THD, voltage, 2-N.

Parameter	Type	Description		
	THDVoltageL3ToNeutral	H	REAL	THD, voltage, 3-N.
	THDCurrentL1	H	REAL	THD, current, phase 1.
	THDCurrentL2	H	REAL	THD, current, phase 2.
	THDCurrentL3	H	REAL	THD, current, phase 3.
	THDCurrentAvg	H	REAL	Average of 3 phase current THD.
<p>¹: These parameters are available only with <i>MBMASTERPACT</i> and <i>MBMASTERPACTC</i>.</p> <p>²: These parameters are available only with <i>MBUMASTERPACTMTZ</i>, <i>MBUMASTERPACTMTZC</i>, <i>MBUMASTERPACTNx</i> and <i>MBUMASTERPACTNxC</i></p> <p>³: These parameters are available only with <i>MBMASTERPACTC</i>, <i>MBUMASTERPACTNxC</i>, <i>MBUMASTERPACTMTZC</i> and <i>EMMASTERPACTC</i>.</p>				

Info

Code with the information shown on the Control operator screen. The following table describes the `Info` structure:

Variable value	Status
2	Waiting for ready signal.
6	Masterpact unit open.
7	Masterpact unit closed.
10	Waiting <code>ControlCommand</code> . It has to be set to 1.
11	Missing <code>EnabledDFB</code> .
12	Missing <code>CommunicationOK</code> . Communication interruption.
13	The status register is not available.
24	Remove <code>ResetFail</code> . Needs to be reset again.
81	Missing <code>ResetFail</code> . Inoperable device.
82	A reset is required.

TrippingCause

The register used to record the cause of a trip operation gives information regarding the cause of a trip operation for basic protective functions. If a bit is set in this register, it means that the device has tripped. The following table describes the reasons for tripping:

Bit	Variable Value	TrippingCause
0	1	Long-time protection <code>I_r</code> .
1	2	Short-time protection <code>I_{sd}</code> .
2	4	Instantaneous protection <code>I_i</code> .
3	8	Ground-fault protection <code>I_g</code> .
4	16	Earth leakage (Vigi) protection <code>I_{Δn}</code> .
5	32	Integrated instantaneous protection.
6	64	Internal detected failure (Stop).
7	128	Excessively high temperature.
8	256	Other protection (check <code>TrippingCauseExt</code>).

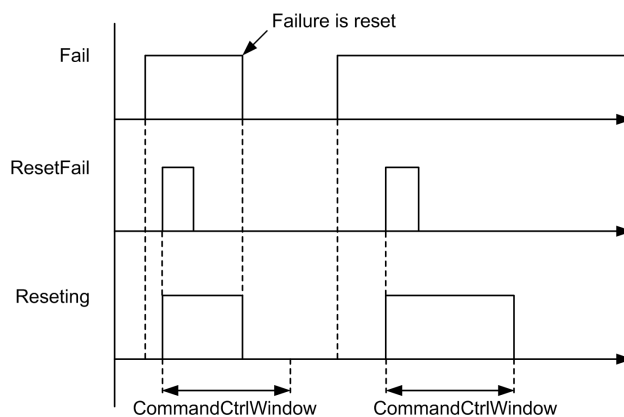
Bit	Variable Value	TrippingCause
9	512	Instantaneous with earth-leakage protection (Vigi module) on the trip unit.
10	1024	Unbalance motor protection.
11	2048	Jam motor protection.
12	4096	Underload motor protection.
13	8192	Long-start motor protection.
14	16384	Reflex tripping protection.

TrippingCauseExt

If bit 8 of `TrippingCause` is active, the following table describes the reasons for trip operation:

Bit	TrippingCauseExt
0	Current imbalance.
1	Overcurrent, phase 1.
2	Overcurrent, phase 2.
3	Overcurrent, phase 3.
4	Overcurrent on neutral.
5	Undervoltage.
6	Overvoltage.
7	Voltage imbalance.
8	Excessive power.
9	Reverse power.
10	Underfrequency.
11	Overfrequency.
12	Phase rotation.
13	Load shedding based on current.
14	Load shedding based on power.

Timing diagram:



Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
MASTERPACT_ST	MASTERPSACT_ST_DDT	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is usable from the HMI/SCADA system.
MASTERPACT_CFG	MASTERPACT_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
WorkMemory	ANY_ARRAY_INT	Array used for Modbus communications. This variable is used with a Modbus port that serializes Modbus requests in an optimum manner.

MASTERPACT_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	Device control. Provides the means to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.

MASTERPACT_CFG_DDT Type

Name	Type	Description
DataStatus	WORD	Device status
Info	INT	Code with information on statuses and required actions.
WarningCode	WORD	Information on the alarm code. Takes the values from the <code>WarningCode</code> output.
WarningCodeExt	WORD	Extra information on the alarm code. Takes the values from the <code>WarningCode</code> output.
WarningOrder-Code	WORD	Information on the command alarm code. Takes the values from the <code>WarningOrderCode</code> output.
FailCode0	INT	Code of last level 0 detected error. Indicates which detected error has occurred, <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates which detected error has occurred, <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates which detected error has occurred, <code>FailCode[2]</code> .
TrippingCause	INT	Information about what caused the device trip.
TrippingCauseExt	INT	Extended information about what caused the device trip. Active when <code>TrippingCause</code> bit 8 is active.

MASTERPACT_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
2	Refer to the <code>Closed</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
3	Inoperable device.
4	Alarm on the device or repetitive detected fault alarm requires resetting.
5	Communication interruption.
6	Requires resetting. A <code>ResetFail</code> is required.
7	Externally controlled.
8	Refer to the <code>Resetting</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
9	Refer to the <code>EnableDFB</code> input pin (see Modicon Libraries General Purpose, Devices Components User Guide).
10	<code>TestPositionCT</code> .
11	<code>Charged</code> .
12	Refer to the <code>ReadyToClose</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
13	<code>ConnectedCE</code> .
14	Refer to the <code>Open</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
15	Refer to the <code>Trip</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

MASTERPACT_ST.CFGW Word Structure

Bit	Description
0	Refer to the <code>ResetFail</code> input pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	<code>Owner</code> .
3	Refer to the <code>Closed</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
6	Refer to the <code>Open</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
7	Refer to the <code>ControlCommand</code> input pin (see Modicon Libraries General Purpose, Devices Components User Guide).

MASTERPACT_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the <code>Open</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	Refer to the <code>Closed</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
2*	Refer to the <code>TestPositionCT</code> status in the output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

Bit	Description
3	Refer to the <code>TripSD</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
4	Refer to the <code>TripSDE</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
5	Refer to the <code>Discharged</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
6	Refer to the <code>Charged</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
7	Refer to the <code>NotReadyToClose</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
8	Refer to the <code>ReadyToClose</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
9	Refer to the <code>NotAvailable</code> status in the <code>Status</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
10*	Refer to the <code>ConnectedCE</code> status in the output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
11*	Refer to the <code>DisconnectedCD</code> status in the output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
*: Available only for <i>MBMASTERPACTC</i> , <i>MBUMASTERPACTMTZC</i> , <i>MBUMASTERPACTNx</i> and <i>MBUMASTERPACTNxC</i> .	

MASTERPACT_CFG.WarningCode Word Structure

Bit	Description
0	Refer to the <code>LongTimePickUp</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	Refer to the <code>CurrentUnbalance</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
2	Refer to the <code>MaxCurrentPhase1</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
3	Refer to the <code>MaxCurrentPhase2</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
4	Refer to the <code>MaxCurrentPhase3</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
5	Refer to the <code>MaxCurrentNeutral</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
6	Refer to the <code>MinVoltage</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
7	Refer to the <code>MaxVoltage</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
8	Refer to the <code>VoltageUnbalance</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
9	Refer to the <code>MaxPower</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
10	Refer to the <code>ReversePower</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
11	Refer to the <code>MinFrequency</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
12	Refer to the <code>MaxFrequency</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
13	Refer to the <code>PhaseRotation</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

Bit	Description
14	Refer to the <code>LoadSheddingCurrent</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
15	Refer to the <code>LoadSheddingPower</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

MASTERPACT_CFG.WarningCodeExt Word Structure

Bit	Description
0	Refer to the <code>GroundFault</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	Refer to the <code>EarthLeakage</code> code in the <code>WarningCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

MASTERPACT_CFG.WarningOrderCode Word Structure

This table describes the `WarningOrderCode` for MBMASTERPACT and MBMASTERPACTC only.

Bit	Description
0	Refer to the <code>Order</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	Refer to the <code>WrongPassword</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
2	Refer to the <code>InternalWarning</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
3	Refer to the <code>ManualMode</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
4	Refer to the <code>IncorrectCoilValue</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
5	Refer to the <code>IncorrectNbrOfParam</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

MASTERPACT_CFG.WarningOrderCode Word Structure

This table describes the `WarningOrderCode` for MBUMASTERPACTMTZ, MBUMASTERPACTMTZC, MBUMASTERPACTNx and MBUMASTERPACTNx C only.

Bit	Description
0	Refer to the <code>Order</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
1	Refer to the <code>WrongPassword</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
2	Refer to the <code>IFEPadLocked</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
3	Refer to the <code>IFMPadLocked</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
4	Refer to the <code>ResourceNotExist</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
5	Refer to the <code>TimeoutDuringCommand</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

Bit	Description
6	Refer to the <code>ResetBeforeCommand</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
7	Refer to the <code>InProgress</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
8	Refer to the <code>AlreadyOpen</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
9	Refer to the <code>AlreadyReset</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
10	Refer to the <code>ManualMode</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
11	Refer to the <code>NotPresent</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).
12	Refer to the <code>InhibitModeOn</code> code in the <code>WarningOrderCode</code> output pin (see Modicon Libraries General Purpose, Devices Components User Guide).

Diagnostics Information Management

Overview

The diagnostic codes that the device can return are read on the `FailCode` output variable.

Modbus Communication Diagnostics Codes

This code indicates that communications have not been established and can be reset:

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

This code indicates that communications have not been established between IFE and TRIP UNIT and can be reset:

- `FailCode[0]: 16#0004`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

or

- `FailCode[0]: 16#0007`
- `FailCode[1]: 16#0200`
- `FailCode[2]: 16#0001`

or

- `FailCode[0]: 16#01FF`
- `FailCode[1]: 16#0200`
- `FailCode[2]: 16#0001`

This code indicates that communications have not been established between IFE and IO modules and can be reset:

- `FailCode[0]: 16#0005`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

After the communications have been established, check Modbus client diagnostic codes for `FailCode [0]` and `FailCode [1]`. Components make a distinction between detected read request problem and write request problem:

- `FailCode[2]: 16#0001 Read`
- `FailCode[2]: 16#0002 Write`

Diagnostics Code Example

For a detected error, the code is:

- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

The diagnostic code is found in the `TrippingCause` status or `TrippingCauseExt` field.

The `FailCode[0]` can have one of the following codes:

Bit	Value	FailCode[0]—Cause of Trip Operation
0	1	Long-time protection I_r .
1	2	Short-time protection I_{sd} .
2	4	Instantaneous protection I_i .
3	8	Ground-fault protection I_g .
4	16	Earth leakage (Vigi) protection $I_{\Delta n}$.
5	32	Integrated instantaneous protection.
6	64	Internal detected failure (Stop).
9	512	Instantaneous protection with earth leakage (Vigi) trip unit.
10	1024	Unbalanced motor protection.
11	2048	Motor jam protection.
12	4096	Underload motor protection.
13	8192	Longstart motor protection.
14	16384	Reflex tripping protection.

HWCIRCUITBREAKER

What's in This Chapter

Description	56
DFB Representation	56
Inputs	57
Outputs	58
Inputs/Outputs	58
Diagnostics Information Management	60

Overview

This chapter describes the DFB of the Hardwired Circuit Breaker.

Description

General

The `HWCIRCUITBREAKER` control function is used to manage hardwired circuit breakers such as `MASTERPACT`, `COMPACT` and other third-party circuit breakers.

Function Description

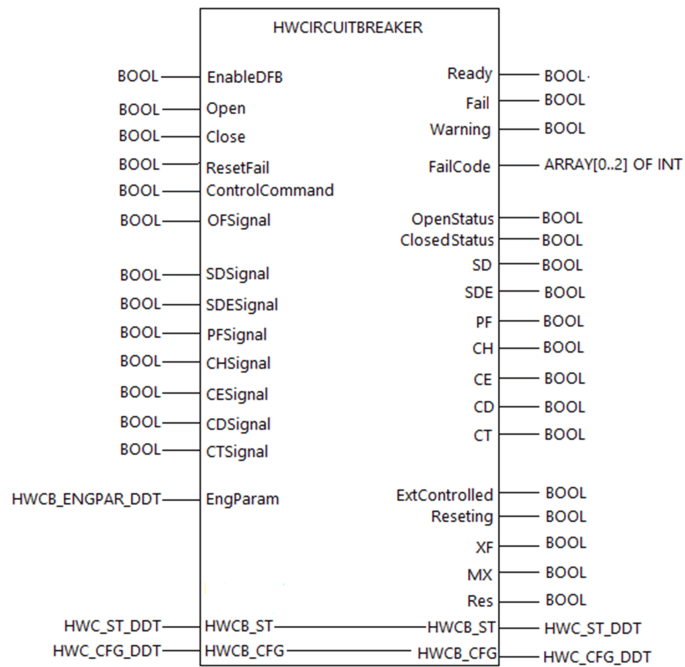
The main functions of the DFB are described in the following table:

Function	Description
Remote resetting	Allows resetting of the device.
Open/Close	Allows circuit breaker to open and close.
Monitoring	Allows the device status to be monitored.

DFB Representation

Representation

The following figure represents the function module of Hardwired circuit breaker.



Inputs

Input Parameter Description

Parameter	Type		Description
EnableDFB	BOOL		Enables the control function.
Open	BOOL		Command to open the circuit breaker (Rising edge opens the breaker).
Close	BOOL		Command to close the circuit breaker (Rising edge closes the breaker).
ResetFail	BOOL		Reset the detected failure of the DFB and the circuit breaker.
ControlCommand	BOOL		Enables or disables the command sending to the circuit breaker.
OFSignal	BOOL		Open or closed status. <ul style="list-style-type: none"> 1 = Close 0 = Open
SDSignal	BOOL		1=Tripped (Push to trip).
SDESignal	BOOL		1= Electrically tripped.
PFSignal	BOOL		1= Breaker is ready to close.
CHSignal	BOOL		1= Spring charged.
CESignal	BOOL		1= Chassis connected.
CDSignal	BOOL		1= Chassis disconnected.
CTSignal	BOOL		1= Chassis in test position.
EngParam	HWC_B_ENGPARG_DDT		Engineering parameters
	Parameter	Type	Description
	CommandCtrlWindow	TIME	Time window for the device to execute orders.
	ScanTime	TIME	Minimum time to maintain detected warning signals.
	MaxResetTime	TIME	Maximum time between two resets.

Parameter	Type		Description
	ResetMode	BOOL	<ul style="list-style-type: none"> 0 = Manual Reset 1 = Automatic reset

Outputs

Output Parameter Description

Parameter	Type	Description
Ready	BOOL	Device ready for operation.
Fail	BOOL	Inoperable device.
Warning	BOOL	Device alert
FailCode	ARRAY [0..2] OF INT	Stores the last active FailCode.
OpenStatus	BOOL	Circuit breaker in open position.
ClosedStatus	BOOL	Circuit breaker in closed position.
SD	BOOL	Tripped (for example, push to trip).
SDE	BOOL	Electrically tripped.
PF	BOOL	Ready to close.
CH	BOOL	Spring charged.
CE	BOOL	Chassis connected.
CD	BOOL	Chassis disconnected.
CT	BOOL	Chassis in test position.
ExtControlled	BOOL	Circuit breaker is not controlled by the DFB.
Reseting	BOOL	Resetting the detected failures.
XF	BOOL	Close command pulse for 500 milliseconds.
MX	BOOL	Open Command pulse for 500 milliseconds.
Res	BOOL	Reset Command pulse for 500 milliseconds.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
HWCB_ST	HWCB_ST_DDT	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is usable from the HMI/SCADA system.
HWCB_CFG	HWCB_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.

HWCB_ST_DDT

Parameter	Type	Description
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	Provides the data status and FailCode.

HWCB_CFG_DDT

Parameter	Type	Description
DataStatus	WORD	Device status
FailCode0	INT	Code of last level 1 detected error. Indicates which detected error has occurred, FailCode [0].
FailCode1	INT	Code of last level 2 detected error. Indicates which detected error has occurred, FailCode [1].
FailCode2	INT	Code of last level 3 detected error. Indicates which detected error has occurred, FailCode [2].
Info	INT	Provides device status information.

HWCB_ST.STW Word Structure

Bit	Description
1	Control Function is not ready to send command.
2	Circuit breaker is in closed state.
3	Control Function is in <code>Fail</code> state due to trip.
4	Control Function is in <code>Warning</code> state due to feedback not obtained after command is sent.
6	Reset is required as control function is in <code>Fail</code> state.
7	Control function is externally controlled.
8	Resetting is in progress.
9	<code>EnableDFB</code> pin status.
10	Test position.
11	Spring is charged.
12	Breaker is ready to close.
13	Chassis is connected to breaker.
14	Circuit breaker is in opened state.
15	Trip status (SD as well as SDE).

HWCB_ST.CFGW Word Structure

Bit	Description
0	<code>ResetFail</code> command obtained from HMI.
1	<code>Owner</code>

Bit	Description
3	Close command
6	Open command
7	Control command

HWCB_CFG.DataStatus Word Structure

Bit	Description
0	Circuit breaker is in opened state.
1	Circuit breaker is in closed state.
2	SD trip status.
3	SDE trip status.
4	Chassis connected status.
5	Chassis disconnected status.
6	Chassis test position status
7	Spring charged status.
8	Ready to close status

Diagnostics Information Management

Overview

Diagnostic information describes the type of detected fail, by displaying diagnostic codes on the `FailCode` output variable.

Modbus Communication Diagnostics Codes

This code indicates that the device has SD Electrical Trip:

- `FailCode[0]: 16#0001`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

This code indicates that the device has SDE Electrical Trip:

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

OperationLimit

What's in This Chapter

Description	61
DFB Representation	61
Inputs	62
Outputs	62

Overview

This chapter describes the Operation Limit DFB which helps you limit the number of commands issued to `HWCIRCUITBREAKER` control function within the configured time.

Description

General

The OperationLimit control function is used to limit the number of commands to manage opening and closing operation of hardwired circuit breaker.

Function Description

The main functions of the DFB are described in the following table:

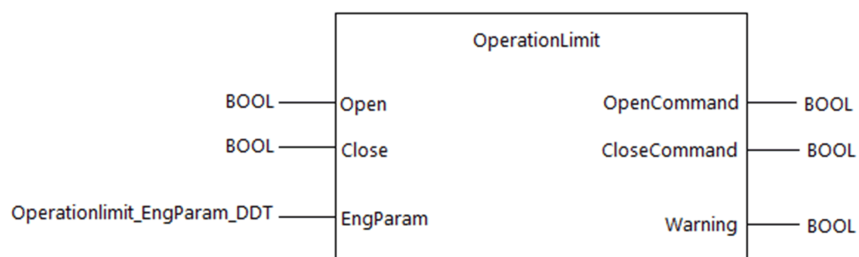
Function	Description
Open/Close	Limits the open and close commands issued to circuit breaker control function.
Retries	Number of command Retries.
Alert	An alert is issued if the number of commands exceed the configured Retries.

For example, if you configure the `WatchTime` as 4 seconds and `Retries` as 3 seconds, then you can issue a total of 7 commands considering both open and close within 4 seconds.

DFB Representation

Representation

The following figure represents the function module of Operation limit.



Inputs

Input Parameter Description

Parameter	Type	Description
Open	BOOL	Open command.
Close	BOOL	Close command.
EngParam	OperationLimit_ENGPAR_DDT	Engineering parameters
	Parameter	Type
	WatchTime	TIME
	Retries	TIME
		Within this time interval, the maximum number of commands possible is less than or equal to Retries configured.
		Number of command retries.

NOTE:

- If EngParam is not connected then a default value of 60 seconds as WatchTime and 3 Retries are considered.
- If EngParam is connected and WatchTime is 0 seconds, then there are no restrictions on commands.

Outputs

Output Parameter Description

Parameter	Type	Description
OpenCommand	BOOL	Open command within retry.
CloseCommand	BOOL	Close command within retry.
Warning	BOOL	Retries exceeded for open and close command combined. NOTE: Warning is reset automatically after the WatchTime is expired.

Digital Protection Relays


What's in This Part

SEPAM Profile 64

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Digital Protection Relays family.

These function blocks do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it 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.

SEPAM Profile

What's in This Chapter

Description	64
DFB Representation	65
Inputs	66
Outputs	68
Inputs/Outputs	80
Diagnostics Information Management	82

Overview

This chapter describes the DFBs of SEPAM profile.

Description

General

The SEPAM profile is used to manage Sepam 20, 40, and 80 digital protection devices in a communications-based manner.

The Sepam20MB is classified into two variants:

- The **MBSEPAM20CSTM** DFB is the control block to manage the Sepam 20 digital protection devices on a Modbus network for S (Substation), T (Transformer), M (Motor) variants of the product.
- The **MBSEPAM20CB** DFB is the control block to manage the Sepam 20 digital protection devices on a Modbus network for B (Busbar) variant of the product.

Function Description

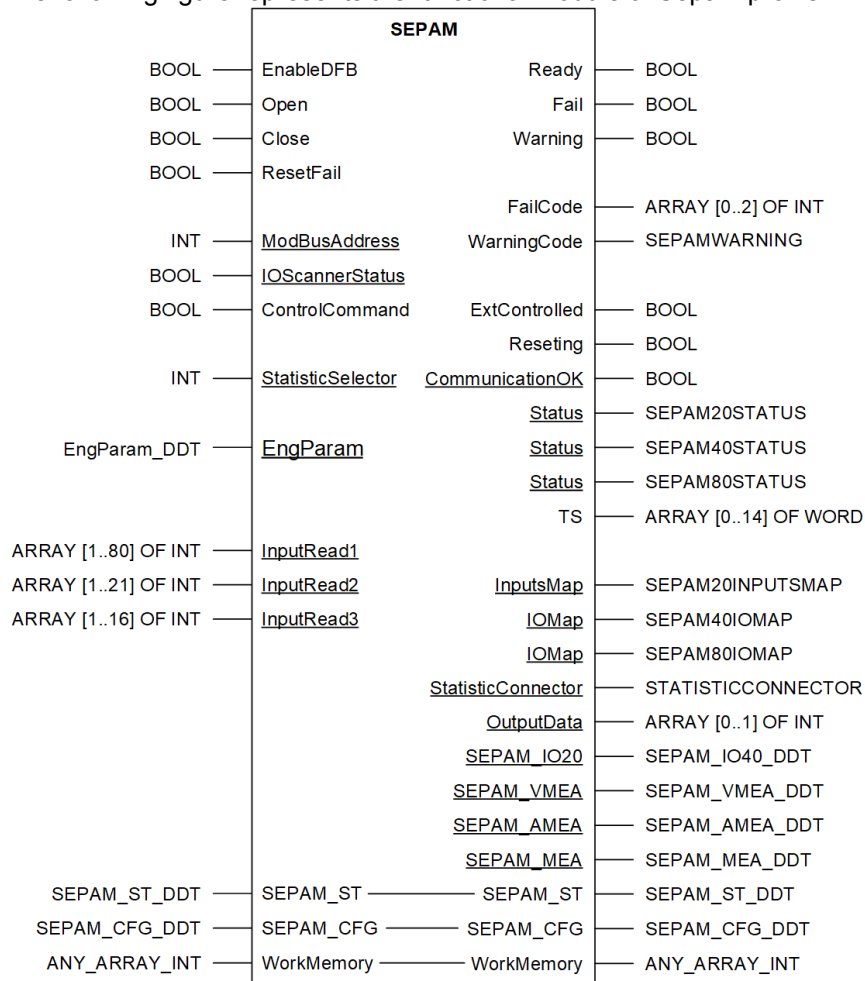
The main functions of the DFB are described in the following table:

Function	Description
Remote resetting	Allows resetting of the device.
Opening/Closing	Allows the element controlled by the Sepam to be opened or closed.
Monitoring	Allows the required parameter devices to be monitored.

DFB Representation

Representation

The following figure represents the functional module of Sepam profile:



NOTE: The underlined parameters are specific for some components.

For DFBs communicating by I/O scanning, variables read from the device retain their last value when a communication interruption occurs. For details, refer to the description of the corresponding output parameter, page 68.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

When you configure the *Last value* parameter of the I/O scanner line, take into consideration the behavior of the DFB when a communication interruption occurs.

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

The table shows the parameters available for specific components:

Parameters		Components				
		SEPAM 20		SEPAM 40	SEPAM 80	
		MBSE-PAM20CB	MBSE-PAM20CST-M	MBSEPAM40C	MBSE-PAM80C	ESEPAM80C
Inputs	ModBusAddress	X	X	X	X	–
	IOScannerStatus	–	–	–	–	X
	StatisticSelector	X	X	X	X	–
	EngParam	X	X	X	X	–
	InputRead1	–	–	–	–	X
	InputRead2	–	–	–	–	X
	InputRead3	–	–	–	–	X
Out-puts	Status (SEPAM20STATUS)	X	X	–	–	–
	Status (SEPAM40STATUS)	–	–	X	–	–
	Status (SEPAM80STATUS)	–	–	–	X	X
	InputsMap	X	X	–	–	–
	IOMap (SEPAM40IOMAP)	–	–	X	–	–
	IOMap (SEPAM80IOMAP)	–	–	–	X	X
	StatisticConnector	X	X	X	X	–
	SEPAM_IO20	X	X	–	–	–
	SEPAM_IO40	–	–	X	–	–
	SEPAM_IO80	–	–	–	X	X
	SEPAM_AMEA	–	X	–	–	–
	SEPAM_VMEA	X	–	–	–	–
	SEPAM_MEAS	–	–	X	X	X
	CommunicationOK	–	–	–	–	X
	OutputData	–	–	–	–	X
X: Parameter is available. –: Parameter is not available.						

Inputs

Input Parameter Description

Name	Data type	Description
EnableDFB	BOOL	This input enables the normal execution of the control block. <ul style="list-style-type: none"> 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. Public variable values are loaded during the first enabling cycle.
Open	BOOL	1 = Opens the element controlled by the Sepam.
Close	BOOL	1 = Closes the element controlled by the Sepam.
ResetFail	BOOL	1 = Resets the Fail output parameter to 0 or in case of inoperable device, sends a reset

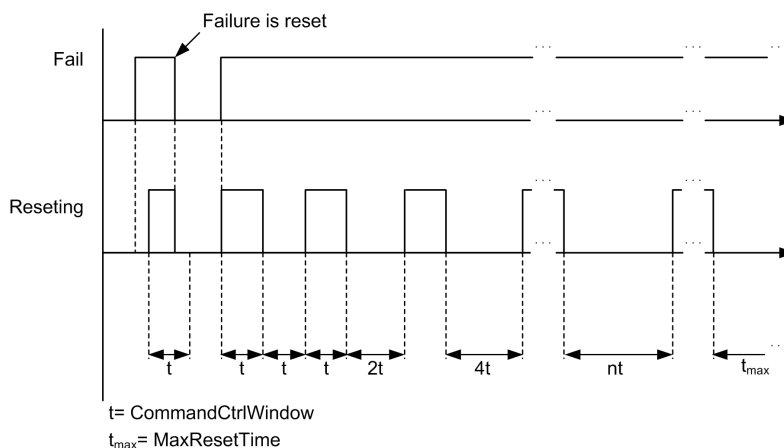
Name	Data type	Description
		command to the device if <code>ControlCommand</code> is 1.
<code>ModBusAddress*</code>	INT	Device address within the Modbus network.
<code>IOScannerStatus*</code>	BOOL	1 = The node is present on the bus. You can find this variable in Ethernet communications.
<code>ControlCommand</code>	BOOL	Indicates to the DFB whether the Sepam is being controlled locally or from a source external to the DFB. <ul style="list-style-type: none"> 0 = Performs only read operations to monitor the status of the device and does not perform any control functions. 1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. NOTE: This input does not configure the Sepam.
<code>InputRead1*</code>	ARRAY [1..80] of INT	An array contains the read information obtained from the SEPAM80 unit through the IOScan. For the block to work properly, allocate the structure (%MWx).
<code>InputRead2*</code>	ARRAY [1..21] of INT	
<code>InputRead3*</code>	ARRAY [1..16] of INT	
<code>StatisticSelector*</code>	INT	This variable is used to obtain statistics for the Modbus network (requests carried out, time between requests, so on). This data provides information for using <code>StatisticConnector</code> pin within the <code>StatisticCounter DFB</code> in General Purpose library for communication. The following table displays the <code>StatisticSelector</code> value:
	Variable value	Description
	1	Read statistics, client.
	2	Write statistics, client.
<code>EngParam</code>	<code>EngParam_DDT</code> , page 67 <code>EngParamSEPAM20CB/</code> <code>EngParamSEPAM40C/</code> <code>EngParamSEPAM80C/</code> <code>EngParamSEPAM20CS</code>	Engineering parameters.
*: Parameter is available for specific components. 1: The data type <code>EngParamSEPAM20CB</code> is applicable for MBSEPAM20CB DFB , <code>EngParamSEPAM40C</code> for MBSEPAM40C DFB , <code>EngParamSEPAM80C</code> for MBSEPAM80C DFB , and <code>EngParamSEPAM20CS</code> for MBSEPAM20CSTM DFB .		

EngParam_DDT

Name	Data type	Description
<code>Refresh</code>	TIME	Refresh time for device data on serial Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed.
<code>CommandCtrlWindow</code>	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is issued.

Name	Data type	Description
		The commands controlled are <code>Open</code> and <code>Close</code> . In the event of a <code>ResetFail</code> , this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the <code>Resetting</code> output (to <code>FALSE</code>).
<code>ScanTime</code>	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
<code>ResetMode</code>	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in <code>CommandCtrlWindow</code> is used to define the interval after which a reset has to be carried out. The first reset is carried out after the time defined in <code>CommandCtrlWindow</code> elapses. The second reset is carried out after <code>CommandCtrlWindow * 2</code> elapses, so on. If the value of <code>CommandCtrlWindow</code> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications are/the device is reset with the <code>ResetFail</code> variable.
	TRUE	Communications are/the device is reset automatically.
<code>MaxResetTime</code>	TIME	When in automatic <code>ResetMode</code> , this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.

Timing diagram:



Outputs

Output Parameter Description

Parameter	Type	Description
<code>Ready</code>	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out or carrying out any Open or Close command.
<code>Fail</code>	BOOL	1 = A detected failure in the control block or in the device or communication interruption. To reset the <code>Fail</code> output pin, the

Parameter	Type	Description	
		ResetFail input has to be activated. The last detected error code is shown on FailCode. NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables retain their last value.	
Warning	BOOL	1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset. This signal remains active until the cause of the alarm disappears.	
FailCode	ARRAY [0..2] OF INT	When the Fail output is 1, it holds the code for the detected error. If the Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3-level structure. Refer to the Diagnostics Information Management, page 82 for more details.	
WarningCode	SEPAMWARNING	Holds a data structure with information on the alarm currently on the Sepam unit. The following table describes the code for the WarningCode:	
	Parameter	Type	Description
	Order	BOOL	1 = Follow-up alarm. The device is not responding to the control command within the time specified in CommandCtrlWindow.
ExtControlled	BOOL	1 = The Sepam is being controlled from a source external (from the HMI/SCADA monitoring system) to the system. It does not consider local control through inputs because the parameters of the local forcing input can be configured with positive or negative logic.	
Resetting	BOOL	1 = A reset is being carried out. The CommandCtrlWindow variable indicates the maximum time for resetting the detected error. When a device or communication reset is carried out with ResetFail, the DFB tries to reset the detected error within the time period defined in CommandCtrlWindow. If the detected error is reset, the Fail and Resetting output variables are reset (set to FALSE). If the detected error is not reset, the Resetting variable is FALSE and the Fail variable remains TRUE. The ResetFail is edge-based. Refer to the Timing diagram below.	
CommunicationOK	BOOL	1 = Communication is OK.	
Status*	SEPAM20STATUS	The structure holds data containing the information that the block extracts from the status variable of the Sepam 20C unit (W16#0100). NOTE: You can use the LocalRemote status variable as an input for ControlCommand with positive or negative logic depending on how parameters are configured on the Sepam unit. The following table describes the status information:	
	Parameter	Type	Description
	Opened	BOOL	1 = The element controlled by the Sepam unit is opened.
	Closed	BOOL	1 = The element controlled by the Sepam unit is closed.
	Trip	BOOL	1 = The Sepam unit is in tripped status.
	Fault	BOOL	1 = The Sepam unit requires resetting after detected fault.
	LocalRemote	BOOL	1 = Remote control (to be configured through an input).
	SettingAActive	BOOL	1 = Setting group A active.
	SettingBActive	BOOL	1 = Setting group B active.

Parameter	Type	Description	
	TimeOutdated	BOOL	1 = The Sepam unit does not have the correct time.
	PartialFault	BOOL	1 = Partial detected error on the Sepam unit.
	MajorFault	BOOL	1 = Major detected error on the Sepam unit.
	SettingMode	BOOL	1 = The Sepam unit is in Local settings mode.
	SynchronizationLost	BOOL	1 = The Sepam unit is not synchronized.
	DataLoss1Event	BOOL	1 = Event detected in first zone.
	Event1Zone	BOOL	1 = Event occurrence in the first event zone.
	Info	INT	Numerical code with the information on statuses and required actions. Refer to the <code>Info</code> table below.
Status*	SEPAM40STATUS	<p>The structure holds data containing the information that the block extracts from the status variable of the Sepam 40C unit (W16#0100).</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	Opened	BOOL	1 = The element controlled by the Sepam unit is opened.
	Closed	BOOL	1 = The element controlled by the Sepam unit is closed.
	Trip	BOOL	1 = The Sepam unit is in tripped status.
	Fault	BOOL	1 = The Sepam unit requires resetting after detected fault.
	LocalRemote	BOOL	1 = Remote control (to be configured through an input).
	DataLoss2Event	BOOL	1 = Event detected in second zone.
	SettingAActive	BOOL	1 = Setting group A active.
	Event2Zone	BOOL	1 = Event occurrence in second event zone.
	SettingBActive	BOOL	1 = Setting group A active.
	TimeOutdated	BOOL	1 = The Sepam unit does not have the correct time.
	PartialFault	BOOL	1 = Partial detected error on the Sepam unit.
	MajorFault	BOOL	1 = Major detected error on the Sepam unit.
	SettingMode	BOOL	1 = The Sepam unit is in Local settings mode.
	SynchronizationLost	BOOL	1 = The Sepam unit is not synchronized.
	DataLoss1Event	BOOL	1 = Event detected in first zone.
	Event1Zone	BOOL	1 = Event occurrence in the first event zone.
	Info	INT	Numerical code with the information on statuses and required actions. Refer to the <code>Info</code> table below.
Status*	SEPAM80STATUS	<p>The structure holds data containing the information that the block extracts from the status variable of the Sepam 80 unit.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	Opened	BOOL	1 = The element controlled by the Sepam unit is opened.
	Closed	BOOL	1 = The element controlled by the Sepam unit is closed.
	Trip	BOOL	1 = The Sepam unit in tripped status.

Parameter	Type	Description	
	Fault	BOOL	1 = The Sepam unit requires resetting after detected fault.
	LocalRemote	BOOL	1 = Remote control (to be configured through an input.
	ModbusSecurity	BOOL	1 = The function is enabled.
	DataLoss2Event	BOOL	1 = Event detected in second zone.
	SettingAActive	BOOL	1 = Setting group A active.
	Event2Zone	BOOL	1 = Event occurrence in second event zone.
	SettingBActive	BOOL	1 = Setting group B active.
	TimeOutdated	BOOL	1 = The Sepam unit does not have the correct time.
	PartialFault	BOOL	1 = Partial detected error on the Sepam unit.
	MajorFault	BOOL	1 = Major detected error on the Sepam unit.
	SettingMode	BOOL	1 = The Sepam unit is in Local settings mode.
	RemoteSetting	BOOL	1 = Remote setting is disabled.
	CapacitiveInductive	BOOL	<ul style="list-style-type: none"> 0 = Capacitive network. 1 = Inductive network.
	SynchronizationLost	BOOL	1 = The Sepam unit is not synchronized.
	DataLoss1Event	BOOL	1 = Event detected in first zone.
	Event1Zone	BOOL	1 = Event occurrence in the first event zone.
	Info	INT	Numerical code with the information on statuses and required actions. Refer to the <code>Info</code> table below.
TS ¹	ARRAY OF WORD		Array that holds remote indications (TS). Remote indications are pre-assigned to protection or control functions depending on the type of the Sepam unit being used. The following table describes the TS:
	Parameter	Type	Description
	TS0	WORD	TS1–TS16. Remote indications (TS) from TS1 to 16.
	TS1	WORD	TS17–TS32. Remote indications (TS) from TS17 to 32.
	TS2	WORD	TS33–TS48. Remote indications (TS) from TS33 to 48.
	TS3	WORD	TS49–TS64. Remote indications (TS) from TS49 to 64.
	TS4	WORD	TS65–TS80. Remote indications (TS) from TS65 to 80.
	TS5	WORD	TS81–TS96. Remote indications (TS) from TS81 to 96.
	TS6	WORD	TS97–TS112. Remote indications (TS) from TS97 to 112.
	TS7	WORD	TS113–TS128. Remote indications (TS) from TS113 to 128.
	TS8	WORD	TS129–TS144. Remote indications (TS) from TS129 to 144.
	TS9	WORD	TS145–TS160. Remote indications (TS) from TS145 to 160.
	TS10	WORD	TS161–TS176. Remote indications (TS) from TS161 to 176.
	TS11	WORD	TS177–TS192. Remote indications (TS) from TS177 to 192.
	TS12	WORD	TS193–TS208. Remote indications (TS) from TS193 to 208.

Parameter	Type	Description	
	TS13	WORD	TS209–TS224. Remote indications (TS) from TS209 to 224.
	TS14	WORD	TS225–TS240. Remote indications (TS) from TS225 to 240.
InputsMap*	SEPAM20INPUTSMAP	Holds a data structure with the information on the state of the inputs of the Sepam 20 unit. The following table describes the InputsMap:	
	Parameter	Type	Description
	I11	BOOL	1 = The state of the input I11.
	I12	BOOL	1 = The state of the input I12.
	I13	BOOL	1 = The state of the input I13.
	I14	BOOL	1 = The state of the input I14.
	I21	BOOL	1 = The state of the input I21.
	I22	BOOL	1 = The state of the input I22.
	I23	BOOL	1 = The state of the input I23.
	I24	BOOL	1 = The state of the input I24.
	I25	BOOL	1 = The state of the input I25.
	I26	BOOL	1 = The state of the input I26.
IOMap*	SEPAM40IOMAP	Holds a data structure with the information on the state of the inputs and outputs of the Sepam 40 unit. The following table describes the IOMap:	
	Parameter	Type	Description
	I11	BOOL	1 = The state of the input I11.
	I12	BOOL	1 = The state of the input I12.
	I13	BOOL	1 = The state of the input I13.
	I14	BOOL	1 = The state of the input I14.
	I21	BOOL	1 = The state of the input I21.
	I22	BOOL	1 = The state of the input I22.
	I23	BOOL	1 = The state of the input I23.
	I24	BOOL	1 = The state of the input I24.
	I25	BOOL	1 = The state of the input I25.
	I26	BOOL	1 = The state of the input I26.
	O1	BOOL	1 = The state of the output O1.
	O2	BOOL	1 = The state of the output O2.
	O3	BOOL	1 = The state of the output O3.
	O4	BOOL	1 = The state of the output O4.
	O11	BOOL	1 = The state of the output O11.
	O12	BOOL	1 = The state of the output O12.
	O13	BOOL	1 = The state of the output O13.
	O14	BOOL	1 = The state of the output O14.
	Led1	BOOL	1 = The state of the LED 1.
	Led2	BOOL	1 = The state of the LED 2.
	Led3	BOOL	1 = The state of the LED 3.
	Led4	BOOL	1 = The state of the LED 4.
	Led5	BOOL	1 = The state of the LED 5.

Parameter	Type	Description	
	Led6	BOOL	1 = The state of the LED 6.
	Led7	BOOL	1 = The state of the LED 7.
	Led8	BOOL	1 = The state of the LED 8.
	Led9	BOOL	1 = The state of the LED 9.
IOMap*	SEPAM80IOMAP	Holds a data structure with the information on the state of the inputs and outputs of the Sepam 80 unit. The following table describes the IOMap:	
	Parameter	Type	Description
	I101	BOOL	1 = The state of the input I101.
	I102	BOOL	1 = The state of the input I102.
	I103	BOOL	1 = The state of the input I103.
	I104	BOOL	1 = The state of the input I104.
	I105	BOOL	1 = The state of the input I105.
	I106	BOOL	1 = The state of the input I106.
	I107	BOOL	1 = The state of the input I107.
	I108	BOOL	1 = The state of the input I108.
	I109	BOOL	1 = The state of the input I109.
	I110	BOOL	1 = The state of the input I110.
	I111	BOOL	1 = The state of the input I111.
	I112	BOOL	1 = The state of the input I112.
	I113	BOOL	1 = The state of the input I113.
	I114	BOOL	1 = The state of the input I114.
	O1	BOOL	1 = The state of the output O1.
	O2	BOOL	1 = The state of the output O2.
	O3	BOOL	1 = The state of the output O3.
	O4	BOOL	1 = The state of the output O4.
	O5	BOOL	1 = The state of the output O5.
	O101	BOOL	1 = The state of the output O101.
	O102	BOOL	1 = The state of the output O102.
	O103	BOOL	1 = The state of the output O103.
	O104	BOOL	1 = The state of the output O104.
	O105	BOOL	1 = The state of the output O105.
	O106	BOOL	1 = The state of the output O106.
	Led1	BOOL	1 = The state of the LED 1.
	Led2	BOOL	1 = The state of the LED 2.
	Led3	BOOL	1 = The state of the LED 3.
	Led4	BOOL	1 = The state of the LED 4.
StatisticConnector*	STATISTICCONNECTOR	Information data is used with Modbus communication to obtain statistics on the Modbus network (requests carried out, time between requests, so on). This structure has been created for its use together with the StatisticCounter DFB in General Purpose library for communication. The following table describes the StatisticConnector:	
	Parameter	Type	Description
	Start	BOOL	1 = Operation has started.

Parameter	Type	Description	
	EndOk	BOOL	1 = Operation has ended correctly.
	EndNok	BOOL	1 = Operation has ended with a detected error.
	PartialTime	DINT	Partial time.
OutputData*	ARRAY [0...2] OF INT	Remote control order.	
SEPAM_IO20*	SEPAM_IO20_DDT	Device data structure holds the information for performing monitoring. The information used by operator screen is usable from the HMI/SCADA system. The following table describes the SEPAM_IO20_DDT type:	
	Parameter	Type	Description
	InputsMap	WORD	Data with inputs. Provides the states for the Sepam 20 inputs. Refer to the InputsMap table below.
	TS0	WORD	TS1–TS16. Remote indications (TS) from TS1 to 16.
	TS1	WORD	TS17–TS32. Remote indications (TS) from TS17 to 32.
	TS2	WORD	TS33–TS48. Remote indications (TS) from TS33 to 48.
	TS3	WORD	TS49–TS64. Remote indications (TS) from TS49 to 64.
SEPAM_IO40*	SEPAM_IO40_DDT	Device data structure holds the information for performing monitoring. The information used by operator screen is usable from the HMI/SCADA system. The following table describes the SEPAM_IO40_DDT type:	
	Parameter	Type	Description
	InputsMap	WORD	Data with inputs. Provides the status for the Sepam 40 inputs. Refer to the InputsMap table below.
	OutputsMap	WORD	Data with outputs. Provides the status for the Sepam 40 outputs. Refer to the OutputsMap table below.
	LedStatus	WORD	Data with LEDs. Provides the status for the Sepam 40 LEDs. Refer to the LedStatus table below.
	TS0	WORD	TS1–TS16. Remote indications (TS) from TS1 to 16.
	TS1	WORD	TS17–TS32. Remote indications (TS) from TS17 to 32.
	TS2	WORD	TS33–TS48. Remote indications (TS) from TS33 to 48.
	TS3	WORD	TS49–TS64. Remote indications (TS) from TS49 to 64.
	TS4	WORD	TS65–TS80. Remote indications (TS) from TS65 to 80.
	TS5	WORD	TS81–TS96. Remote indications (TS) from TS81 to 96.
	TS6	WORD	TS97–TS112. Remote indications (TS) from TS97 to 112.
	TS7	WORD	TS113–TS128. Remote indications (TS) from TS113 to 128.
	TS8	WORD	TS129–TS144. Remote indications (TS) from TS129 to 144.
SEPAM_IO80*	SEPAM_IO80_DDT	Device data structure holds the information for performing functions. The information used by operator screen is usable from the HMI/SCADA system.	

Parameter	Type	Description	
		The following table describes the SEPAM_IO80_DDT type:	
	Parameter	Type	Description
	DataStatusExt	WORD	Device status extended for the Sepam 80. Refer to the DataStatusExt table below.
	InputsMap	WORD	Digital inputs. Provides the status for the Sepam 80 inputs. Refer to the InputsMap table below.
	OutputsMap	WORD	Digital outputs. Provides the status for the Sepam 80 outputs. Refer to the OutputsMap table below.
	LedStatus	WORD	Data with LEDs. Provides the status for the Sepam 80 LEDs. Refer to the LedStatus table below.
	TS0	WORD	TS1–TS16. Remote indications (TS) from TS1 to 16.
	TS1	WORD	TS17–TS32. Remote indications (TS) from TS17 to 32.
	TS2	WORD	TS33–TS48. Remote indications (TS) from TS33 to 48.
	TS3	WORD	TS49–TS64. Remote indications (TS) from TS49 to 64.
	TS4	WORD	TS65–TS80. Remote indications (TS) from TS65 to 80.
	TS5	WORD	TS81–TS96. Remote indications (TS) from TS81 to 96.
	TS6	WORD	TS97–TS112. Remote indications (TS) from TS97 to 112.
	TS7	WORD	TS113–TS128. Remote indications (TS) from TS113 to 128.
	TS8	WORD	TS129–TS144. Remote indications (TS) from TS129 to 144.
	TS9	WORD	TS145–TS160. Remote indications (TS) from TS145 to 160.
	TS10	WORD	TS161–TS176. Remote indications (TS) from TS161 to 176.
	TS11	WORD	TS177–TS192. Remote indications (TS) from TS177 to 192.
	TS12	WORD	TS193–TS208. Remote indications (TS) from TS193 to 208.
	TS13	WORD	TS209–TS224. Remote indications (TS) from TS209 to 224.
	TS14	WORD	TS225–TS240. Remote indications (TS) from TS225 to 240.
	Reserved	INT	Reserved.
SEPAM_MEA*	SEPAM_MEA_DDT	Data structure with the device measurement information. The following table describes the SEPAM_MEA_DDT:	
	Parameter	Type	Description
	CurrentI1	REAL	Phase current I1 (A).
	CurrentI2	REAL	Phase current I2 (A).
	CurrentI3	REAL	Phase current I3 (A).
	ResidualCurrentSum	REAL	Total residual current I0 (A).
	ResidualCurrentMeasured	REAL	Measured residual current (A).

Parameter	Type	Description	
	AverageCurrentIm1	REAL	Average phase current Im1 (A).
	AverageCurrentIm2	REAL	Average phase current Im2 (A).
	AverageCurrentIm3	REAL	Average phase current Im3 (A).
	PeakCurrentIM1	REAL	Peak demand phase current IM1 (A).
	PeakCurrentIM2	REAL	Peak demand phase current IM2 (A).
	PeakCurrentIM3	REAL	Peak demand phase current IM3 (A).
	VoltageU21	REAL	Line-to-line voltage U21 (V).
	VoltageU32	REAL	Line-to-line voltage U32 (V).
	VoltageU13	REAL	Line-to-line voltage U13 (V).
	VoltageV1	REAL	Line-to-neutral voltage V1 (V).
	VoltageV2	REAL	Line-to-neutral voltage V2 (V).
	VoltageV3	REAL	Line-to-neutral voltage V3 (V).
	ResidualVoltage	REAL	Residual voltage v0 (V).
	PositiveVoltage	REAL	Forward voltage Vd (V).
	NegativeVoltage	REAL	Inverse voltage Vi (V).
	Frequency	REAL	Frequency (Hz).
	ActivePower	REAL	Real power p(kW).
	ReactivePower	REAL	Reactive power Q(kVar).
	AparentPower	REAL	Apparent power S(kVA).
	PeakActivePower	REAL	Peak demand real power Pm(kW).
	CosPhi	REAL	Cos Phi power factor.
	PositiveActiveEnergy	DINT	Positive real power Ea+(kWh).
	NegativeActiveEnergy	DINT	Negative real power Ea-(kWh).
	PositiveReactiveEnergy	DINT	Positive reactive power Er+(kVarh).
	NegativeReactiveEnergy	DINT	Negative reactive power Er-(kVarh).

*: Parameter is available only for specific components.

1: From TS0 to TS8 is applicable for MBSEPAM40C. From TS0 to TS14 is applicable for MBSEPAM80C and ESEPAM80C.

SEPAM_VMEA_DDT for MBSEPAM20CB DFB

Parameter	Type	Description	
SEPAM_VMEA*	SEPAM_VMEA_DDT	Voltage measurements for the Sepam B20, B21, and B22.	
	Parameter	Type	Description
	VoltageU13	REAL	Phase-to-phase voltage U13 (in Volts).
	VoltageU21	REAL	Phase-to-phase voltage U21 (in Volts).
	VoltageU32	REAL	Phase-to-phase voltage U32 (in Volts).
	VoltageV1	REAL	Phase-to-neutral voltage V1 (in Volts).
	VoltageV2	REAL	Phase-to-neutral voltage V2 (in Volts).

Parameter	Type	Description	
	<i>VoltageV3</i>	REAL	Phase-to-neutral voltage V3 (in Volts).
	<i>ResidualVoltage</i>	REAL	Residual voltage V0 (in Volts).
	<i>PositiveVoltage</i>	REAL	Positive sequence voltage Vd (in Volts).
	<i>Frequency</i>	REAL	Frequency (in Hertz).

SEPAM_AMEA_DDT for MBSEPAM20CSTM DFB

Parameter	Type	Description	
SEPAM_AMEA*	SEPAM_AMEA_DDT	Current measurements for the Sepam B20, B21, and B22.	
	Parameter	Type	Description
	<i>CurrentI1</i>	REAL	Phase current I1 (in Amps).
	<i>CurrentI2</i>	REAL	Phase current I2 (in Amps).
	<i>CurrentI3</i>	REAL	Phase current I3 (in Amps).
	<i>ResidualCurrent</i>	REAL	Residual current I0 (in Amps).
	<i>ResidualCurrent-Mea</i>	REAL	Residual current measured (in Amps).
	<i>AverageCurrentI1</i>	REAL	Average phase current I1 (in Amps).
	<i>AverageCurrentI2</i>	REAL	Average phase current I2 (in Amps).
	<i>AverageCurrentI3</i>	REAL	Average phase current I3 (in Amps).
	<i>PeakCurrentIM1</i>	REAL	Peak demand phase current IM1 (in Amps).
	<i>PeakCurrentIM2</i>	REAL	Peak demand phase current IM2 (in Amps).
	<i>PeakCurrentIM3</i>	REAL	Peak demand phase current IM3 (in Amps).

Info

The code with the information is shown on the Control Expert operator screen. The following table describes the Info:

Variable value	Description
1	Incorrect configuration of DFB parameter.
2	Waiting for Ready.
6	The Sepam unit is open.
7	The Sepam unit is closed.
10	Waiting <code>ControlCommand</code> . It has to be 1.
11	Missing <code>EnabledDFB</code> .
12	Missing <code>Communication OK</code> . Communication interruption.
13	Status register value is 0.
23	Remove <code>Open</code> . It has to be 0.

Variable value	Description
24	Remove <code>ResetFail</code> . Reset again.
81	Missing <code>ResetFail</code> . Inoperable device.
82	A reset is needed.

DataStatusExt Word Structure

The following table describes the `DataStatusExt` Word structure:

Bit	Description
0	Refer to the <code>ModbusSecurity</code> status in the <code>Status</code> output pin.
1	Refer to the <code>RemoteSettings</code> status in the <code>Status</code> output pin.
2	Refer to the <code>CapacitiveInductive</code> status in the <code>Status</code> output pin.

OutputsMap Word Structure

The following table describes the `OutputsMap` Word structure:

Bit	Description
0	Shows the state of the output 01.
1	Shows the state of the output 02.
2	Shows the state of the output 03.
3	Shows the state of the output 04.
4	Shows the state of the output 05.
5	Shows the state of the output 0101.
6	Shows the state of the output 0102.
7	Shows the state of the output 0103.
8	Shows the state of the output 0104.
9	Shows the state of the output 0105.
10	Shows the state of the output 0106.

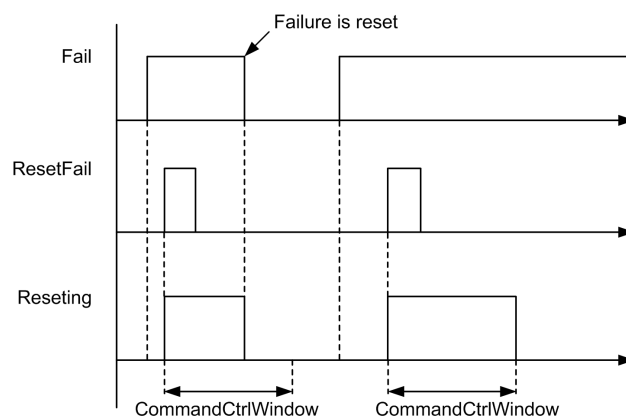
InputsMap Word Structure (SEPAM 20 and SEPAM 40)

The following table describes the `InputsMap` Word structure:

Bit	Description
0	Shows the state of the digital input 11.
1	Shows the state of the digital input 12.
2	Shows the state of the digital input 13.
3	Shows the state of the digital input 14.
4	Shows the state of the digital input 21.
5	Shows the state of the digital input 22.
6	Shows the state of the digital input 23.
7	Shows the state of the digital input 24.

Bit	Description
8	Shows the state of the digital input 25.
9	Shows the state of the digital input 26.

Timing diagram:



InputsMap Word Structure (SEPAM 80)

The following table describes the `InputsMap` Word structure:

Bit	Description
0	Shows the state of the digital input I101.
1	Shows the state of the digital input I102.
2	Shows the state of the digital input I103.
3	Shows the state of the digital input I104.
4	Shows the state of the digital input I105.
5	Shows the state of the digital input I106.
6	Shows the state of the digital input I107.
7	Shows the state of the digital input I108.
8	Shows the state of the digital input I109.
9	Shows the state of the digital input I110.
10	Shows the state of the digital input I111.
11	Shows the state of the digital input I112.
12	Shows the state of the digital input I113.
13	Shows the state of the digital input I114.

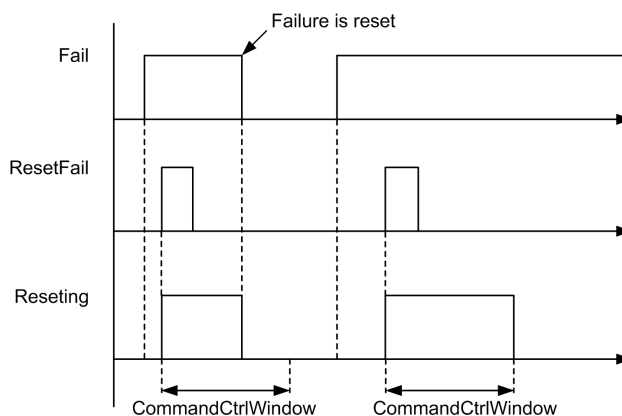
LEDStatus Word Structure

The following table describes the `LEDStatus` Word structure:

Bit	Description
0	Shows the state of the LED 1.
1	Shows the state of the LED 2.
2	Shows the state of the LED 3.
3	Shows the state of the LED 4.

Bit	Description
4	Shows the state of the LED 5.
5	Shows the state of the LED 6.
6	Shows the state of the LED 7.
7	Shows the state of the LED 8.
8	Shows the state of the LED 9.
9	Shows the state of the LED 10.

Timing diagram:



Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
SEPAM_ST	SEPAM_ST_DDT	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is readable/writable from the HMI/SCADA system.
SEPAM_CFG	SEPAM_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
WorkMemory	ANY_ARRAY_INT	Array is used for Modbus communications. This variable is meant to be used with a Modbus port that serializes Modbus requests in an optimum manner.

SEPAM_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	Device control. Enables to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.

SEPAM_CFG_DDT Type

Name	Type	Description
DataStatus	WORD	Information on the device status. Information on the Status output structure.
Info	INT	Device information. Its value is Info status.
WarningCode	WORD	Sepam alarm code information. Takes the values from the WarningCode output pin.
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred FailCode[0].
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred FailCode[1].
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred FailCode[2].

SEPAM_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
2	Module is running.
3	Inoperable device problem.
4	Alarm on the device or DFB (follow-up or screw terminal-based control).
5	Communication interruption.
6	Requires resetting. ResetFail is required.
8	Refer to the Reseting output pin, page 68.
9	Refer to the EnableDFB input pin, page 66.

SEPAM_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 66.
1	Owner.
3	Refer to the Close input pin, page 66.
6	Refer to the Open input pin, page 66.
7	Refer to the ControlCommand input pin, page 66.

SEPAM_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the Fault status in the Status output pin, page 68.
1	Refer to the Opened status in the Status output pin, page 68.
2	Refer to the Closed status in the Status output pin, page 68.
3	Refer to the Trip status in the Status output pin, page 68.

Bit	Description
4	Refer to the <code>LocalRemote</code> status in the <code>Status</code> output pin, page 68.
5	Refer to the <code>SettingAActive</code> status in the <code>Status</code> output pin, page 68.
6	Refer to the <code>SettingBActive</code> status in the <code>Status</code> output pin, page 68.
7	Refer to the <code>TimeOutdated</code> status in the <code>Status</code> output pin, page 68.
8	Refer to the <code>SettingMode</code> status in the <code>Status</code> output pin, page 68.
9	Refer to the <code>SynchronizationLost</code> status in the <code>Status</code> output pin, page 68.
10	Refer to the <code>DataLoss1Event</code> status in the <code>Status</code> output pin, page 68.
11	Refer to the <code>Event1Zone</code> status in the <code>Status</code> output pin, page 68.
12	Refer to the <code>PartialFault</code> status in the <code>Status</code> output pin, page 68.
13	Refer to the <code>MajorFault</code> status in the <code>Status</code> output pin, page 68.
14	Refer to the <code>DataLoss2Event</code> status in the <code>Status</code> output pin, page 68.
15	Refer to the <code>Event2Zone</code> status in the <code>Status</code> output pin, page 68.

SEPAM_CFG.WarningCode Word Structure

Bit	Description
0	Refer to the <code>Order</code> in the <code>WarningCode</code> output pin, page 68.

Diagnostics Information Management

Overview

The diagnostic codes that the device can return are read on the `FailCode` output variable.

Parameter Configuration Diagnostics Codes

This detected error indicates that a public variable parameter contains a value that is not allowed.

To load new values, a rising edge is required on the `EnableDFB` input:

- `FailCode[0]`: 16#0003
- `FailCode[1]`: 16#0000
- `FailCode[2]`: 16#0004

Modbus Communication Diagnostics Codes

This code indicates that communications have not been established and can be reset:

- `FailCode[0]`: 16#0002
- `FailCode[1]`: 16#0000
- `FailCode[2]`: 16#0004

After the communications have been established, check Modbus client diagnostic codes for `FailCode [0]` and `FailCode [1]`. The components make a distinction between detected read request problem and write request problem:

- `FailCode [2]: 16#0001` Read
- `FailCode [2]: 16#0002` Write

Diagnostics Code Example

For a detected error, the code is:

- `FailCode [1]: 16#0000`
- `FailCode [2]: 16#0005`

It can be reset.

Motor Controllers and Starters


What's in This Part

TeSys T Profile.....	85
Tesys U Profile	107

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Motor Controllers and Starters family.

These function blocks do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it 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.

TeSys T Profile

What's in This Chapter

Description	85
DFB Representation	86
Inputs	87
Outputs	95
Inputs/Outputs	101
Diagnostics Information Management	104

Overview

This chapter describes the DFBs of TeSys T profile.

Description

General

The TeSys T profile is used to manage TeSys T motor management system on different communication networks (Ethernet network based on IO scanning, Ethernet network using fast IO scanning, Ethernet network using messaging, Modbus network, Advantys STB I/O island or Profibus using PRM gateway/PRM master).

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Control	Device operation.
Address	Forward/reverse direction of rotation.
Speed	This function enables the change of speed (fast or slow) if you are working with a 2-speed motor start (double winding).
Device status indication	Displays the status of the device.
Remote resetting	Allows resetting of the device.
Control or Monitoring	Enables you to control or monitor the device.
Owner	Manages the control system which is the owner (Operator or Program). Therefore, it is responsible for setting the control.

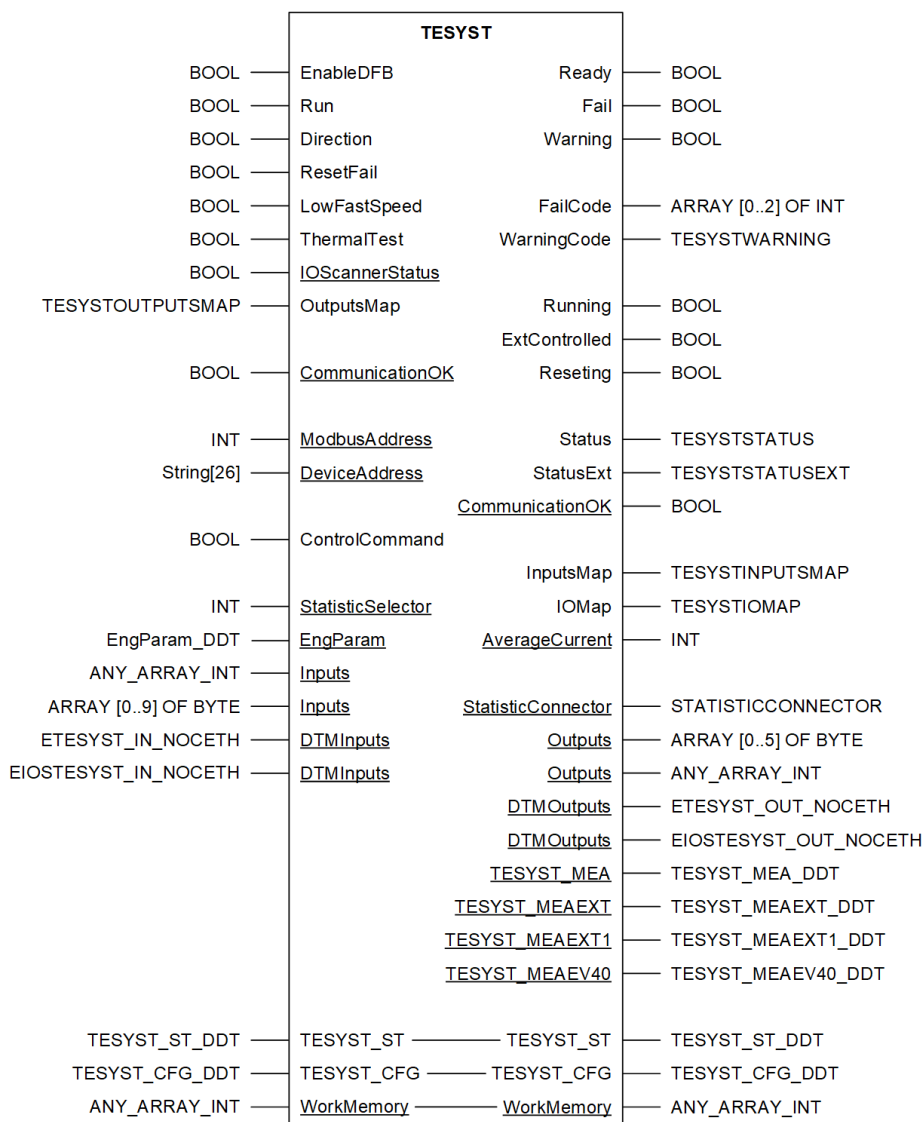
NOTE:

- DTM Inputs and DTM Outputs pins are not supported in EcoStruxure Process Expert.
- For establishing the communication with TesysT devices through IOScanning or by using DTM, the device `Unit ID` for TesysT varies for different firmware versions. Hence default value is set to 1, as it supports most of the firmware versions. In case of unsuccessful communication, you may need to configure the correct device `Unit ID` in the configuration parameter.

DFB Representation

Representation

The following figure represents the functional module of TeSys T profile:



NOTE: The underlined parameters are specific for some components.

For DFBs communicating by I/O scanning, variables read from the device retain their last value when a communication interruption occurs. For details, refer to the description of the corresponding output parameter, page 68.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

When you configure the *Last value* parameter of the I/O scanner line, take into consideration the behavior of the DFB when a communication interruption occurs.

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

The table shows the parameters available for specific components:

Parameters		Components					
		Ethernet			Modbus (MBTE-SYST)	Advantys (TE-SYSTCTL)	Profibus (TESYSTPB)
		Normal IO scanning (EIOSTESYST)	Fast IO scanning (ETESYST)	Messaging (EMESTE-SYST)			
Inputs	<i>ModbusAddress</i>	–	–	–	X	–	–
	<i>DeviceAddress</i>	–	–	X	–	–	–
	<i>IOScannerStatus</i>	X	X	–	–	–	X
	<i>CommunicationOK</i>	–	–	–	–	X	–
	<i>StatisticSelector</i>	–	–	X	X	–	–
	<i>EngParam</i>	–	–	X	X	–	–
	<i>Inputs</i>	X	X	–	–	X	X
	<i>DTMInputs</i>	X	X	–	–	–	–
Outputs	<i>CommunicationOK</i>	X	X	–	–	–	X
	<i>StatusExt</i> (TESYSTSTATUSEXT)	X	X	X	X	X	X
	<i>AverageCurrent</i>	X	–	X	X	–	X
	<i>TESYST_MEA</i>	X	–	X	X	–	–
	<i>TESYST_MEAEXT</i>	X	–	X	X	–	–
	<i>TESYST_MEAEXT1</i>	–	–	–	X	–	–
	<i>TESYST_MEADEV40</i>	X	–	X	X	–	–
	<i>StatisticConnector</i>	–	–	X	X	–	–
	<i>Outputs</i>	X	X	–	–	X	X
	<i>DTMOutputs</i>	X	X	–	–	–	–
Inputs/ Outputs	<i>WorkMemory</i>	–	–	X	X	–	–
X: Parameter is available. –: Parameter is not available.							

Inputs

Input Parameter Description

Name	Data type	Description
EnableDFB	BOOL	This input enables the normal execution of the control block. <ul style="list-style-type: none"> 0 = The entire DFB is restarted (states, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. Public variable values are loaded during the first enabling cycle.

If the device is reset and *Run* pin is active, then the device will auto start. If manual start of the device is required, then reset the *Run* pin followed by device.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Reset the `Run` variable before resuming operation.

Failure to follow these instructions can result in equipment damage.

Name	Data type	Description	
Run	BOOL	1 = Starts the motor run in the direction selected with the <code>Direction</code> input variable.	
Direction	BOOL	<p>Direction of rotation of the motor.</p> <ul style="list-style-type: none">0 = Activates the forward direction drive.1 = Activates the reverse direction drive. <p>Only used in 2/3 Wire Reverser mode configuration in TeSys T device. The device causes incorrect operation in other operating modes.</p> <div><div>Operating Mode</div><div>2 Wire Reverser</div></div>	
ResetFail	BOOL	1 = Resets the <code>Fail</code> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <code>ControlCommand</code> is 1.	
LowFastSpeed	BOOL	<p>Slow/fast speed of rotation. For 2-speed motor start configuration in device, refer to the <i>TesysT user manual</i>.</p> <p>Only used in the 2/3 Wire two Speed mode configuration in TeSys T device. The device causes incorrect operation in other operating modes.</p> <div><div>Operating Mode</div><div>2 Wire Two Speed</div></div>	
ThermalTest	BOOL	1 = Checks the forced thermal detected error input on the TeSys T.	
OutputsMap	TESYSTOUTPUTSMAP	Holds a structure to control the Tesys outputs.	
		These outputs hold an image of this structure values. (Memory in Modbus 700).	
	Name	Type	Description
	Output1	BOOL	1 = Controls the state of output 1.
	Output2	BOOL	1 = Controls the state of output 2.
	Output3	BOOL	1 = Controls the state of output 3.
Output4	BOOL	1 = Controls the state of output 4.	
ModbusAddress*	INT	Device address within the Modbus network.	
		You can find this variable in Modbus communications.	
DeviceAddress*	STRING[26]	Device address within the Ethernet network.	
		Refer to the <i>Ethernet Technology</i> , page 248.	
		Depending on the platform, the following definitions apply:	
	Platform	IP Addressing <code>DeviceAddress</code> (variable)	
	M340	'{IP}ID'	
	Quantum	'{IP}ID'	
M580	'{IP}ID'		
	NOTE: ID is 0.		
IOScannerStatus*	BOOL	1 = Indicates that the node is present on the bus.	
		You can find this variable in Ethernet communications.	
		NOTE: <i>UnitID</i> for <i>EIOSTesysT</i> and <i>ETesysT</i> profile is 1.	

Name	Data type	Description
CommunicationOK*	BOOL	1 = The communication ok for the device present in the IO scanner.
StatisticSelector*	INT	Refer to <i>StatisticSelector</i> .
EngParam*	EngParam_DDT, page 94 EngParamTESYST/EngParamEMESTESYST	Engineering parameters.
ControlCommand	BOOL	Indicates to the DFB whether the motor is being controlled locally or from a source external to the DFB. <ul style="list-style-type: none"> 0 = Performs only read operations to monitor the status of the device and does not perform any control functions. 1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. NOTE: This input does not configure the motor.
INPUTS ¹	ANY_ARRAY_INT	<p>Holds an array structure with data obtained from the device. You can control the starter/controller with this input variable. This input is reserved for the DFB, and you cannot use this input directly. For the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247.</p> <p>The Inputs pin should not be connected when DTMInputs pin is connected, otherwise the function detects an incorrect configuration.</p> <p>The following table describes the INPUTSstructure:</p>
	Parameter	Type Description
	Inputs [0]	INT Reserved.
	Inputs [1]	INT Reserved.
	Inputs [2]	INT Reserved.
	Inputs [3]	INT Reserved.
	Inputs [4]	INT Status register 1.
INPUTS ¹	Inputs [5]	INT Status register 2.
	Inputs [6]	INT Logic input status.
	Inputs [7]	INT Logic output status.
	Inputs [8]	INT Reserved.
	Inputs [9]	INT Reserved.
	Inputs [10]	INT Notification register.
	Inputs [11]	INT Reserved.
	Inputs [12]	INT Reserved.
	Inputs [13]	INT Reserved.
	Inputs [14]	INT Thermal capacity.
	Inputs [15]	INT Average current.
	Inputs [16]	INT FLC current L1.
	Inputs [17]	INT FLC current L2.
	Inputs [18]	INT FLC current L3.
	Inputs [19]	INT FLC ground current.
	Inputs [20]	INT Current phase.
	Inputs [21]	INT Temperature.
	Inputs [22]	INT Reserved.
	Inputs [23]	INT Frequency.
	Inputs [24]	INT Motor temperature.

Name	Data type	Description	
	Inputs [25]	INT	Average voltage.
	Inputs [26]	INT	L3 to L1 voltage.
	Inputs [27]	INT	L1 to L2 voltage.
	Inputs [28]	INT	L2 to L3 voltage.
	Inputs [29]	INT	Voltage imbalance.
	Inputs [30]	INT	Power factor.
	Inputs [31]	INT	Real power.
	Inputs [32]	INT	Reactive power.
INPUTS ¹	Inputs [33]	INT	Reserved.
	Inputs [34]	INT	Reserved.
	Inputs [35]	INT	Reserved.
	Inputs [36]	INT	Reserved.
	Inputs [37]	INT	Reserved.
	Inputs [38]	INT	Reserved.
	Inputs [39]	INT	Reserved.
	Inputs [40]	INT	Reserved.
	Inputs [41]	INT	Reserved.
	Inputs [42]	INT	Reserved.
	Inputs [43]	INT	Reserved.
	Inputs [44]	INT	Reserved.
	Inputs [45]	INT	Reserved.
	Inputs [46]	INT	Reserved.
	Inputs [47]	INT	Reserved.
	Inputs [48]	INT	Reserved.
	Inputs [49]	INT	Average current.
	Inputs [50]	INT	Average current.
	Inputs [51]	INT	L1 current.
	Inputs [52]	INT	L1 current.
	Inputs [53]	INT	L2 current.
	Inputs [54]	INT	L2 current.
	Inputs [55]	INT	L3 current
	Inputs [56]	INT	L3 current.
	Inputs [57]	INT	Ground current.
	Inputs [58]	INT	Ground current.
	Inputs [59]	INT	Reserved.
	Inputs [60]	INT	Time trip.
	Inputs [61]	INT	Last start current.
	Inputs [62]	INT	Last start duration.
	Inputs [63]	INT	Number of starts.
INPUTS [*]	ANY_ARRAY_INT	<p>Holds an array structure with data obtained from the device. You can control the starter/controller with this input variable. This input is reserved for the DFB, and you cannot use this input directly. For the control block to work properly, allocate the structure (%MW×). Refer to the Communications Technologies, page 247.</p>	

Name	Data type	Description	
		<p>The <code>Inputs</code> pin should not be connected when <code>DTMInputs</code> pin is connected, otherwise the function detects an incorrect configuration.</p> <p>The following table describes the <code>INPUTS</code> structure:</p>	
	Parameter	Type	Description
	<code>Inputs [0]</code>	INT	Status register 1.
	<code>Inputs [1]</code>	INT	Status register 2.
	<code>Inputs [2]</code>	INT	Logic input status.
	<code>Inputs [3]</code>	INT	Logic output status.
INPUTS ⁴	ARRAY [0..9] OF BYTE	<p>Holds an array structure with data coming from the acutuator. This variable is reserved for the DFB, and You cannot use this variable directly.</p> <p>The following table describes the <code>INPUTS</code> structure:</p>	
	Parameter	Type	Description
	<code>Inputs [0]</code>	BYTE	System status.
	<code>Inputs [1]</code>	BYTE	System status.
	<code>Inputs [2]</code>	BYTE	Average current MSB.
	<code>Inputs [3]</code>	BYTE	Average current LSB.
	<code>Inputs [4]</code>	BYTE	Logic input status MSB.
	<code>Inputs [5]</code>	BYTE	Logic input status LSB.
	<code>Inputs [6]</code>	BYTE	Logic output status MSB.
	<code>Inputs [7]</code>	BYTE	Logic output status LSB.
	<code>Inputs [8]</code>	BYTE	System status register 2 - MSB.
	<code>Inputs [9]</code>	BYTE	System status register 2 - LSB.
DTMINPUTS ¹	EIOSTESYST_IN_NOCETH	<p>Holds an input structure for the data to be obtained from device DTM. You can read device monitoring information with this input variables. This input pin has to be used with the device DTM.</p> <p>The <code>DTMInputs</code> pin should not be connected when <code>Inputs</code> pin is connected, otherwise the function detects an incorrect configuration</p> <p>The information available for <code>EIOSTESYST_IN_NOCETH</code> on Ethernet is shown in the following table:</p>	
	Parameter	Type	Description
	<code>Fault_code</code>	UINT	Code of detected fault
	<code>Fault_register_1</code>	WORD	Register 1 of detected fault
	<code>Fault_register_2</code>	WORD	Register 2 of detected fault
DTMINPUTS ¹	<code>Fault_register_3</code>	WORD	Register 3 of detected fault
	<code>System_status_register_1</code>	WORD	System status register 1
	<code>System_status_register_2</code>	WORD	System status register 2
	<code>Logic_inputs_status</code>	WORD	Logic input status
	<code>Logic_outputs_status</code>	WORD	Logic output status
	<code>IO_status</code>	WORD	I/O status
	<code>Warning_code</code>	UINT	Alert code
	<code>Warning_register_1</code>	WORD	Alert register 1
	<code>Warning_register_2</code>	WORD	Alert register 2
	<code>Warning_register_3</code>	WORD	Alert register 3

Name	Data type	Description
	Motor_temperature_sensor_degree	UINT Motor temperature sensor
	Thermal_capacity_level	UINT Thermal capacity level (% trip level)
	Average_current_ratio	UINT Average current ratio (% FLC)
	L1_current_ratio	UINT L1 current ratio (% FLC)
	L2_current_ratio	UINT L2 current ratio (% FLC)
	L3_current_ratio	UINT L3 current ratio (% FLC)
	Ground_current_ratio	UINT Ground current ratio (x0.1% FLC Min)
	Current_phase_imbalance	UINT Current phase imbalance (%)
	Controller_internal_temperature	INT Controller Internal Temperature
	Controller_config_checksum	UINT Controller configuration checksum
	Frequency	UINT Frequency (x0.01Hz)
	Motor_temperature_sensor	UINT Motor temperature sensor (x0.1 ohms)
	Average_voltage	UINT Average voltage
	L3L1_voltage	UINT L3 to L1 voltage
	L1L2_voltage	UINT L1 to L2 voltage
	L2L3_voltage	UINT L2 to L3 voltage
	Voltage_phase_imbalance	UINT Voltage phase imbalance (%)
	Power_factor	UINT Power factor (x0.01)
DTMINPUTS ¹	Active_power	UINT Active power (x0.01kW)
	Reactive_power	UINT Reactive power (x0.0kVAR)
	Auto_restart_status_register	WORD Auto restart status register
	Controller_last_power_off_durati	WORD Last power Off duration
	Not_significant	UINT Not significant
	Not_significantA	UINT Not significant
	Not_significantB	UINT Not significant
	Not_significantC	UINT Not significant
	Network_port_monitoring	WORD Network port monitoring
	Network_port_baud_rate	UINT Network port baud rate
	Not_significantD	UINT Not significant
	Network_port_parity	UINT Network port parity
	Not_significantE	UINT Not significant
	Not_significantF	UINT Not significant
	Not_significantG	UINT Not significant
	Not_significantH	UINT Not significant
	Not_significantI	UINT Not significant
	Not_significantJ	UINT Not significant
	Average_current	UINT Average current (x0.01A)
	Average_current_1	UINT Average current (x0.01A)
	L1_current	UINT L1 current (x0.001A)
	L1_current_1	UINT L1 current (x0.001A)
	L2_current	UINT L2 current (x0.001A)
	L2_current_1	UINT L2 current (x0.001A)

Name	Data type	Description	
	L3_current	UINT	L3 current (x0.001A)
	L3_current_1	UINT	L3 current (x0.001A)
	Ground_current	UINT	Ground current (mA)
	Ground_current_1	UINT	Ground current (mA)
DTMINPUTS ¹	Controller_port_ID	UINT	Controller port ID
	Time_to_trip	UINT	Time to trip (x1s)
	Motor_last_start_current_ratio	UINT	Motor last start current ratio (%FLC)
	Motor_last_duration	UINT	Motor last duration (x0.1% FLC Min)
	Motor_last_duration	UINT	Motor starts per hour count
DTMINPUTS ²	ETESYST_IN_NOCETH	<p>Holds an input structure for the data to be obtained from device DTM. You can read device monitoring information with this input variables. This input pin has to be used with the device DTM.</p> <p>The <i>DTMInputs</i> pin should not be connected when <i>Inputs</i> pin is connected, otherwise the function detects an incorrect configuration</p> <p>The information available for ETESYST_IN_NOCETH on Ethernet is shown in the following table:</p>	
	Parameter	Type	Description
	Mirror_status_register	WORD	Mirror status register
	Reserved	WORD	Reserved
	Mirrors_System_Status_Register_1	WORD	Mirrors system status register 1
	Mirrors_System_Status_Register_2t	WORD	Mirrors system status register 2
	Mirrors_logic_inputs_status	WORD	Mirrors logic inputs status
	Logic_outputs_status	WORD	Logic outputs status
	<p>*: Parameter is available for specific components.</p> <p>1: Parameter is available only for Ethernet normal IO scanning.</p> <p>2: Parameter is available for Ethernet fast IO scanning.</p> <p>3: Parameter is available for Advantys components.</p> <p>4: Parameter is available only for Profibus components.</p>		

StatisticSelector

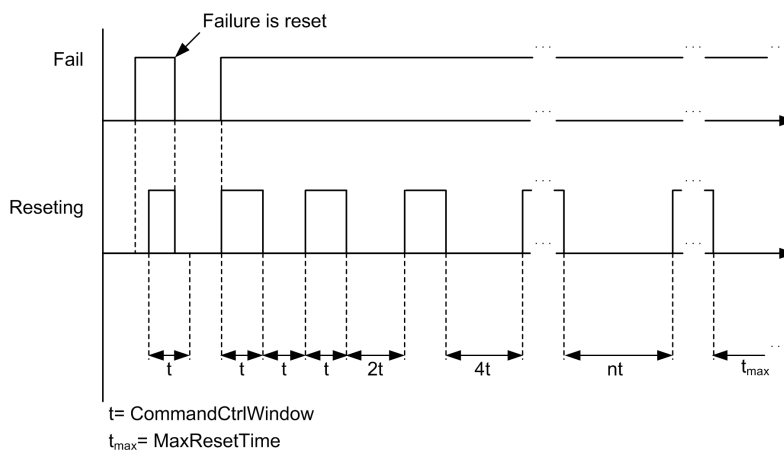
Variable is used to obtain statistics for the Modbus network (requests carried out, time between requests, so on). This data provides information for using `StatisticConnector` pin within the `StatisticCounter` DFB in General Purpose library for communication.

Variable value	Description
1	Read statistics, client.
2	Write statistics, client.

EngParam_DDT

Name	Data type	Description
Refresh	TIME	Refresh time for device data on serial Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed and have maximum priority.
CommandCtrlWindow	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is issued. The command that is controlled is Run. In the event of ResetFail, this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the Resetting output.
ScanTime	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in CommandCtrlWindow is used to define the interval after which a reset has to be carried out. The first reset is carried out after the time defined in CommandCtrlWindow elapses. The second reset is carried out after CommandCtrlWindow * 2 elapses, and so on. If the value of CommandCtrlWindow is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications are/the device is reset with the ResetFail variable.
	TRUE	Communications are/the device is reset automatically.
MaxResetTime	TIME	When in automatic ResetMode, this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.

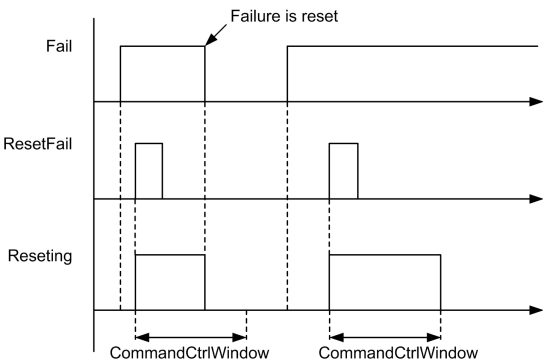
Timing diagram:



Outputs

Output Parameter Description

Output	Type	Description	
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out or carrying out any Run or Stop command.	
Fail	BOOL	1 = A detected failure in the control block or in the device or communication interruption.To reset the <code>Fail</code> output pin, the <code>ResetFail</code> input has to be activated. The last detected error code is shown on <code>FailCode</code> . NOTE: If communication interruption occurs, the variables being read from the device ceases to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.	
Warning	BOOL	1 = An alarm has been activated for the device. It cannot be reset because the signal remains active until the cause of the alarm is removed or until the maximum set <code>ScanTime</code> is reached.	
FailCode	ARRAY [0..2] OF INT	When <code>Fail</code> output is 1, it holds the code for the detected error. If <code>Fail</code> output is 0, it indicates the last detected error that occurred. The detected error source is specified by a 3-level structure. Refer to the Diagnostics Information Management , page 104 for more details.	
WarningCode	TESYSTWARNING	Holds a data structure with information about the alarm currently on the device.	
	Name	Type	Description
	Ground	BOOL	1 = Ground alarm.
	Thermal	BOOL	1 = Thermal alarm on the device.
	Jam	BOOL	1 = Mechanical jam alarm.
	Phase	BOOL	1 = Phase alarm.
	UnderCurrent	BOOL	1 = An undercurrent alarm.
	HMIPort	BOOL	1 = An HMI port communication interruption alarm.
	InternalTmp	BOOL	1 = Internal temperature alarm.
	Internal	BOOL	1 = An alarm for internal detected error.
	Network	BOOL	1 = Network port communication interruption alarm.
	Order	BOOL	1 = Follow-up alarm. The device is not responding to the control command within the time specified in <code>CommandCtrlWindow</code> .
	ForcedLocalMode	BOOL	1 = The device is forced locally in the <code>ForcedLocalMode</code> status. Is controlled through the screw terminal.
Running	BOOL	1 = The TeSys T is running. Current is higher than 10% of FLC.	
ExtControlled	BOOL	1 = The device is being controlled from an external source (for example, from the console, from a push-button panel, or from the monitoring system) to the system. Provides information for programming. NOTE: The <code>ControlCommand</code> signal, the <code>Owner</code> variable, and the <code>ForcedLocalMode</code> status are used to activate this signal. You cannot use this signal as a <code>ControlCommand</code> input.	
Reseting	BOOL	1 = A reset is being carried out. The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected failure. When a device or communication reset is carried out with <code>ResetFail</code> , the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code> . If the detected failure is reset, the <code>Fail</code> and <code>Reseting</code> output variables are reset (set to FALSE). If the detected failure is not reset, the <code>Reseting</code>	

Output	Type	Description	
		<p>variable is set to FALSE and the Fail variable remains TRUE. The ResetFail is edge-based.</p> <p>Timing diagram:</p>  <p>CommandCtrlWindow</p>	
CommunicationOk*	BOOL	1 = The communication is ok for the device present in the IO scanner.	
Status	TESYSTSTATUS	<p>The structure holds data structure with the information that the module extracts from the status variable of the device.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	Ready	BOOL	1 = Switch in ON position. Device is operational.
	Activated	BOOL	1 = The contact is activated. System on.
	Fail	BOOL	1 = TesysT is not operational.
	Warning	BOOL	1 = Alarm present on device.
	Trip	BOOL	1 = The Tesys protection has tripped.
	ResetAuth	BOOL	1 = Reset authorized detected error.
	ControllerPower	BOOL	1 = The system is powered.
	MotorRunning	BOOL	1 = Motor running with a current higher than 10% of full-load amperage.
	HMIControlled	BOOL	1 = HMI port is controlled.
	MotorStarting	BOOL	1 = The motor is starting.
	State, page 100	INT	Numerical code corresponding to the device status. Refer to the State table below.
Info, page 100	INT	Numerical code with the information on statuses and required actions. Refer to the Info table below.	
StatusExt*	TESYSTSTATUSEXT	<p>Holds a data structure with information that the module extracts from the state variable of the device.</p> <p>The following table describes the StatusExt:</p>	
	Parameter	Type	Description
	AutoResetActive	BOOL	1 = Auto-reset is active.
	FaultRequested	BOOL	1 = Detected error, restart (off/on) request.
	RestartTimeUnd	BOOL	1 = Motor restart time undefined.
	RapidCicleLockout	BOOL	1 = Rapid cycle lockout.
	LoadSheding	BOOL	1 = Load shedding.
	MotorSpeed	BOOL	1 = Motor speed.
	HMILostComms	BOOL	1 = Communication interruption with the HMI port.
	LostComms	BOOL	1 = Communication interruption with the network port.
	MotorTransitionLockout	BOOL	1 = Motor transition lockout.

Output	Type	Description	
InputsMap	TESYSTINPUTSMAP	Holds a data structure with information on the state of the device inputs. The following table describes the <i>InputsMap</i> :	
	Parameter	Type	Description
	Input1	BOOL	1 = The state of the logic input 1.
	Input2	BOOL	1 = The state of the logic input 2.
	Input3	BOOL	1 = The state of the logic input 3.
	Input4	BOOL	1 = The state of the logic input 4.
	Input5	BOOL	1 = The state of the logic input 5.
	Input6	BOOL	1 = The state of the logic input 6.
	Input7	BOOL	1 = The state of the logic input 7.
	Input8	BOOL	1 = The state of the logic input 8.
	Input9	BOOL	1 = The state of the logic input 9.
	Input10	BOOL	1 = The state of the logic input 10.
IOMap	TESYSTIOMAP	Holds a data structure with information on the state of the device inputs and outputs. The following table describes the <i>IOMap</i> :	
	Parameter	Type	Description
	Output1	BOOL	1 = The state of the digital output 1.
	Output2	BOOL	1 = The state of the digital output 2.
	Output3	BOOL	1 = The state of the digital output 3.
	Output4	BOOL	1 = The state of the digital output 4.
AverageCurrent*	INT	Average value of current ratio in %FLC.	
TESYST_MEA*	TESYST_MEA_DDT	Refer to <i>TESYST_MEA_DDT</i> Type.	
TESYST_MEAEXT*	TESYST_MEAEXT_DDT	Refer to <i>TESYST_MEAEXT_DDT</i> Type.	
TESYST_MEAEV40*	TESYST_MEAEV40_DDT	Refer to <i>TESYST_MEAEV40_DDT</i> Type.	
TESYST_MEAEXT1*	TESYST_MEAEXT1_DDT	Refer to <i>TESYST_MEAEXT1_DDT</i> Type.	
StatisticConnector	STATISTICCONNECTOR	Refer to <i>STATISTICCONNECTOR</i> .	
Outputs*	ANY_ARRAY_INT	Holds an array structure with data sent to the device. You can control the starter/controller with this output variable. This variable is reserved for the DFB, and You cannot use this variable directly. For the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247. The following table describes the <i>OUTPUTS</i> structure:	
	Parameter	Type	Description
	Outputs [0]	INT	Output control.
	Outputs [1]	INT	System control.
	Outputs [2]	INT	Reserved.
Outputs ³	ANY_ARRAY_INT	Data sent to the actuator.	
Outputs ⁴	ARRAY [0..5] OF BYTE	Holds an array structure with data sent to the acuator. This variable is reserved for the DFB, and You cannot use this variable directly. The following table describes the <i>OUTPUTS</i> structure:	
	Parameter	Type	Description
	Outputs [0]	BYTE	Control command.
	Outputs [1]	BYTE	Control command.
	Outputs [2]	BYTE	Reserved.

Output	Type	Description	
	Outputs [3]	BYTE	Reserved.
	Outputs [4]	BYTE	Logic output command.
	Outputs [5]	BYTE	Logic output command.
DTMOutputs ¹	EIOSTESYST_OUT_NOCETH	Holds an output structure for the data to be written from controller to the device via Device DTM. This output pin has to be used when the device DTM is used. The following table describes the EIOSTESYST_OUT_NOCETH information available for TeSysT on Ethernet networks:	
	Parameter	Type	Description
	Logic_outputs_command_register	WORD	Logic outputs command register of Tesys T
	Reserved	WORD	Reserved
	Reserved_1	WORD	Reserved 1
	Reserved_2	WORD	Reserved 2
	Control_register_1	WORD	Control register 1
DTMOutputs ²	ETESYST_OUT_NOCETH	Holds an output structure for the data to be written from controller to the device via Device DTM. This output pin has to be used when the device DTM is used. The following table describes the ETESYST_OUT_NOCETH information available for TeSysTFast on Ethernet networks:	
	Parameter	Type	Description
	Logic_Outputs_Command_Register	WORD	Logic outputs command register
	Control_Register_1	WORD	Control register 1
	Analog_Output_1_Command	WORD	Analog output 1 command
<p>*: Parameter is available only for specific components.</p> <p>1: Parameter is available only for Ethernet normal IO scanning.</p> <p>2: Parameter is available for Ethernet fast IO scanning.</p> <p>3: Parameter is available for Advantys components.</p> <p>4: Parameter is available only for Profibus components.</p>			

TESYST_MEA_DDT Type

TESYST_MEA is a data structure with device information. The following table describes the TESIYST_MEA_DDT:

Parameter	Type	Description
FLCGroundCurrent	REAL	Ground current ratio (%FLC minimum).
Frequency	REAL	Frequency (Hz).
ThermalCapacity	INT	Thermal capacity level (% trip level).
CurrentPhase	INT	Phase current imbalance.
StartsCount	INT	Number of starts per hour.
Reserved	INT	Reserved.

TESYST_MEAEXT_DDT Type

TESYST_MEAEXT is a data structure with extended information about the device on Modbus communications. The following table describes the *TESYST_MEAEXT_DDT*:

Parameter	Type	Description
FLCCurrentL1	INT	L1 current (%FLC).
FLCCurrentL2	INT	L2 current (%FLC).
FLCCurrentL3	INT	L3 current (%FLC).
Temperature	INT	Internal controller temperature (°C).
Current	REAL	Average current (A).
CurrentL1	REAL	L1 current (A).
CurrentL2	REAL	L2 current (A).
CurrentL3	REAL	L3 current (A).
GroundCurrent	REAL	Ground current (A).
MotorTemperature	INT	Motor temperature sensor (%).
TimeTrip	INT	Time to trip (x 1 s).
LastStartCurrent	INT	Motor last start current ratio (%FLC).
LastStartDuration	INT	Motor last start duration (s).

TESYST_MEAEV40_DDT Type

TESYST_MEAEV40 is a data structure with information about the device with an EV40 expansion module on Modbus communications. The following table describes the *TESYST_MEAEV40_DDT*:

Parameter	Type	Description
PowerFactor	REAL	Power factor.
AverageVoltage	INT	Average voltage (V).
L3L1Voltage	INT	L3-L1 voltage (V).
L1L2Voltage	INT	L1-L2 voltage (V).
L2L3Voltage	INT	L2-L3 voltage (V).
ActivePower	REAL	Active power consumption (kW).
ReactivePower	REAL	Active power consumption (kVAR).
VoltageImbalance	INT	Phase voltage imbalance (%).
Reserved	INT	Reserved.

TESYST_MEAEXT1_DDT Type

TESYST_MEAEXT1 is a data structure with extended information on the device with Modbus communications.

Parameter	Type	Description
FltReg1	WORD	Detected fault register 1.
FltReg2	WORD	Detected fault register 2.
FltReg3	WORD	Detected fault register 3.

Parameter	Type	Description
WarReg1	WORD	Notification register 1.
WarReg2	WORD	Notification register 2.
WarReg3	WORD	Notification register 3.
ActivePwrRecord	REAL	Active power consumption (kW).
ReactivePwrRecord	REAL	Reactive power consumption (kVAR).

STATISTICCONNECTOR

`StatisticConnector` is information data used with Ethernet communication to obtain statistics on the Ethernet network (requests carried out, time between requests, and so on). This structure has been created to be used with the `StatisticCounter` DFB in General Purpose for communication.

Parameter	Type	Description
Start	BOOL	1= The operation has started.
EndOk	BOOL	1 = The operation has ended correctly.
EndNOk	BOOL	1 = The operation has ended with a detected error.
PartialTime	DINT	1 = Partial time.

State

The following table describes the `State` variable:

Variable value	Description
-2	Device has detected error.
-1	Not initialized. Waiting for data.
0	Disabled.
5	Ready.
8	Inoperable device.
9	Device notification.

Info

The following table describes the `Info` variable:

Variable value	Description
1	Incorrect configuration of DFB parameter.
2	Waiting Ready.
8	Tesys working OK.
10	Waiting for device information.
11	Missing <code>EnableDFB</code> .
12	Missing <code>Communication OK</code> Communication interruption.
13	Status word value is 0.
14	Local forcing has to be 0.
23	Remove <code>Run</code> has to be 0.

Variable value	Description
24	Remove <code>ResetFail</code> . Reset again.
25	Tesys T stopped.
81	Missing <code>ResetFail</code> . Inoperable device.
82	A reset is needed.
99	Unknown status.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
<code>TESYST_ST</code>	<code>TESYST_ST_DDT</code>	Refer to <i>TESYST_ST_DDT</i> Type.
<code>TESYST_CFG</code>	<code>TESYST_CFG_DDT</code>	Refer to <i>TESYST_CFG_DDT</i> Type.
<code>WorkMemory*</code>	<code>ANY_ARRAY_INT</code>	Array is used for Ethernet communications. This variable is meant for use with an Ethernet port that serializes Ethernet requests in an optimum manner.
*: Parameter is available only for specific components.		

TESYST_ST_DDT Type

TESYST_ST is a device data structure that holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is readable/writable from the HMI/SCADA system.

Name	Type	Description
<code>STW</code> , page 101	WORD	Provides the device status. Access to the data held in this bit word is read-only.
<code>CFGW</code> , page 102	WORD	Device control. Provides the means to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
<code>AverageCurrent</code>	REAL	Actual current value in FLC%.

TESYST_ST.STW Word Structure

Bit	Description
0	Unknown technological module status. No variable refreshing.
1	Not ready.
2	Technological module is running.
3	Inoperable device.
4	Alarm on the device or repetitive detected fault alarm requires resetting.
5	Communication interruption.

Bit	Description
6	Requires resetting. <code>ResetFail</code> is required.
7	Refer to the <code>ExtControlled</code> output pin, page 95.
8	Refer to the <code>Reseting</code> output pin, page 95.
9	Refer to the <code>EnableDFB</code> input pin, page 87.

TESYST_ST.CFGW Word Structure

Bit	Description
0	Refer to the <code>ResetFail</code> input pin, page 87.
1	Owner.
3	Refer to the <code>Direction</code> input pin, page 87.
6	Refer to the <code>Run</code> input pin, page 87.
7	Refer to the <code>ControlCommand</code> input pin, page 87.
11	Refer to the <code>LowFastSpeed</code> input pin, page 87.
12	Refer to the <code>ThermalTest</code> input pin, page 87.

NOTE: The `Owner` bit enables to control the block from the `TESYST_ST_DDT` input/output structure ignoring the input signals of the block. It enables control from a monitoring system (HMI, SCADA, operator screen) in the Manual mode without using the programmed switching operation.

TESYST_CFG_DDT Type

TESYST_CFG is a data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.

Name	Type	Description
<code>DataStatus</code> , page 103	WORD	Information on the device status.
<code>InputsMap</code> , page 103	WORD	Information on the digital input status.
<code>IOMap</code> , page 104	WORD	Information on the outputs command and status.
<code>Info</code>	INT	TeSys T information. Its value is <code>Info</code> status.
<code>Warning-Code</code> , page 103	WORD	TeSys T alarm code information. Takes the values from the <code>WarningCode</code> output pin.
<code>FailCode0</code>	INT	Code of last level 0 detected error. Indicates which detected error has occurred, <code>FailCode[0]</code> .
<code>FailCode1</code>	INT	Code of last level 1 detected error. Indicates which detected error has occurred, <code>FailCode[1]</code> .
<code>FailCode2</code>	INT	Code of last level 2 detected error. Indicates which detected error has occurred, <code>FailCode[2]</code> .

TESYST_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the Ready status in the Status output pin, page 95.
1	Refer to the Activated status in the Status output pin, page 95.
2	Refer to the Fail status in the Status output pin, page 95.
3	Refer to the Warning status in the Status output pin, page 95.
4	Refer to the Trip status in the Status output pin, page 95.
5	Refer to the ResetAuth in the Status output pin, page 95.
6	Refer to the ControlledPower status in the Status output pin, page 95.
7	Refer to the MotorRunning status in the Status output pin, page 95.
8	Refer to the HMIControlled status in the Status output pin, page 95.
9	Refer to the MotorStarting status in the Status output pin, page 95.
10	Refer to the AutoResetActive status in the StatusExt output pin, page 95.
11	Refer to the FaultRequested status in the StatusExt output pin, page 95.
12	Refer to the RapidCycleLockout in the StatusExt output pin, page 95.
13	Refer to the LoadSheding in the StatusExt output pin, page 95.
14	Refer to the HMILostComms in the StatusExt output pin, page 95.
15	Refer to the MotorTransitionLockout in the StatusExt output pin, page 95.

TESYST_CFG.WarningCode Word Structure

Bit	Description
2	Refer to the Ground in the WarningCode output pin, page 95.
3	Refer to the Thermal in the WarningCode output pin, page 95.
5	Refer to the Jam in the WarningCode output pin, page 95.
6	Refer to the Phase in the WarningCode output pin, page 95.
7	Refer to the UnderCurrent in the WarningCode output pin, page 95.
8	Refer to the Order in the WarningCode output pin, page 95.
9	Refer to the ForcedLocalMode in the WarningCode output pin, page 95.
10	Refer to the HMIPort in the WarningCode output pin, page 95.
11	Refer to the InternalTmp in the WarningCode output pin, page 95.
12	Refer to the Internal in the WarningCode output pin, page 95.
15	Refer to the Network in the WarningCode output pin, page 95.

TESYST_CFG.InputsMap Word Structure

Bit	Description
0	Status of the digital input I1 on a controller base.
1	Status of the digital input I2 on a controller base.
2	Status of the digital input I3 on a controller base.
3	Status of the digital input I4 on a controller base.

Bit	Description
4	Status of the digital input I5 on a controller base.
5	Status of the digital input I6 on a controller base.
6	Status of the digital input I7 on a controller base.
7	Status of the digital input I8 on a controller base.
8	Status of the digital input I9 on a controller base.
9	Status of the digital input I10 on a controller base.

TESYST_CFG.IOMap Word Structure

Bit	Description
0	Status and command of the output O1 on a controller base.
1	Status and command of the output O2 on a controller base.
2	Status and command of the output O3 on a controller base.
3	Status and command of the output O4 on a controller base.

Diagnostics Information Management

Overview

The diagnostics codes the device can return are read from the `FailCode` output variable.

Parameter Configuration Diagnostic Codes

This diagnostic code indicates that the function block has incorrect configuration.

- `FailCode[0]`: 16#0003
- `FailCode[1]`: 16#0000
- `FailCode[2]`: 16#0004

This diagnostic code can occur for any of the below conditions:

- Wrong array size at `Inputs` or `Outputs` pins of the function block.
- Variables connected to both `DTMInputs` and `Inputs` pins of the function Block.

During the above detected `FailCode` the function block does not process any inputs and the function blocks output displays the last processed state.

This detected `FailCode` can be reset by a rising edge to the `EnableDFB` input pin after correcting the configuration of the function block.

Modbus Communications Diagnostics Codes

For Modbus communications, this code is used to indicate that communications have not been established. It can be reset.

- `FailCode[0]`: 16#0002
- `FailCode[1]`: 16#0000
- `FailCode[2]`: 16#0004

After Modbus communications have been established, check Modbus client diagnostic codes for `FailCode [0]` and `FailCode [1]`. Components make a distinction between detected read request problems and write request problems:

- `FailCode[2]: 16#0001` - Read
- `FailCode[2]: 16#0002` - Write

Diagnostics Code Example

For a detected error, the code is:

- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

The `Failcode[0]` variable can have the following codes:

Diagnostics codes	Meaning
0	No detected error.
3	Ground current.
4	Thermal overload.
5	Extended start.
6	Mechanical jam.
7	Phase current imbalance.
8	Undercurrent.
10	Test.
11	HMI port detected error.
12	Interruption of HMI port communications.
13	Internal network port detected error
16	External detected error assigned by PCODE.
18	Diagnosis.
19	Wiring.
20	Overcurrent.
21	Phase current dropout.
22	Phase currents inverted.
23	Motor temperature sensor.
24	Phase voltage imbalance.
25	Phase voltage dropout.
26	Control over voltage.
26	Phase voltages inverted.
27	Undervoltage.
28	Overvoltage.
29	Insufficient power.
30	Excessive power.
31	Insufficient power factor.
32	Excessive power factor.
33	Load shedding.
34	Incorrect temperature reading.
35	Open circuit in temperature sensor.
36	Inverted TC.

Diagnostics codes	Meaning
46	Startup test.
47	Run repeat test.
48	Stop test.
49	Stop repeat test.
51	Internal controller temperature detected error.
55	Internal controller detected error (stack overflow).
56	Internal controller detected error (RAM).
57	Internal controller detected error (RAM checksum).
58	Internal controller detected error (hardware monitoring problem).
60	L2 current detected in single-phase mode.
64	Inoperable non-volatile memory.
65	Communication interruption in expansion module.
66	Reset button jammed.
67	Logical function detected error.
100–104	Internal network port detected error.
109	Communication interruption in network port.
555	Incorrect network port configuration.

The inoperable device reset is sent to the device.

Tesys U Profile

What's in This Chapter

Description	107
DFB Representation	108
Inputs	110
Outputs	112
Inputs/Outputs	118
Diagnostics Information Management	120

Overview

This chapter describes the DFBs of Tesys U profile.

Description

General

The Tesys U profile is used to manage the TesysU family of devices on different communications networks (Modbus network or Advantys STB I/O island).

Function Description

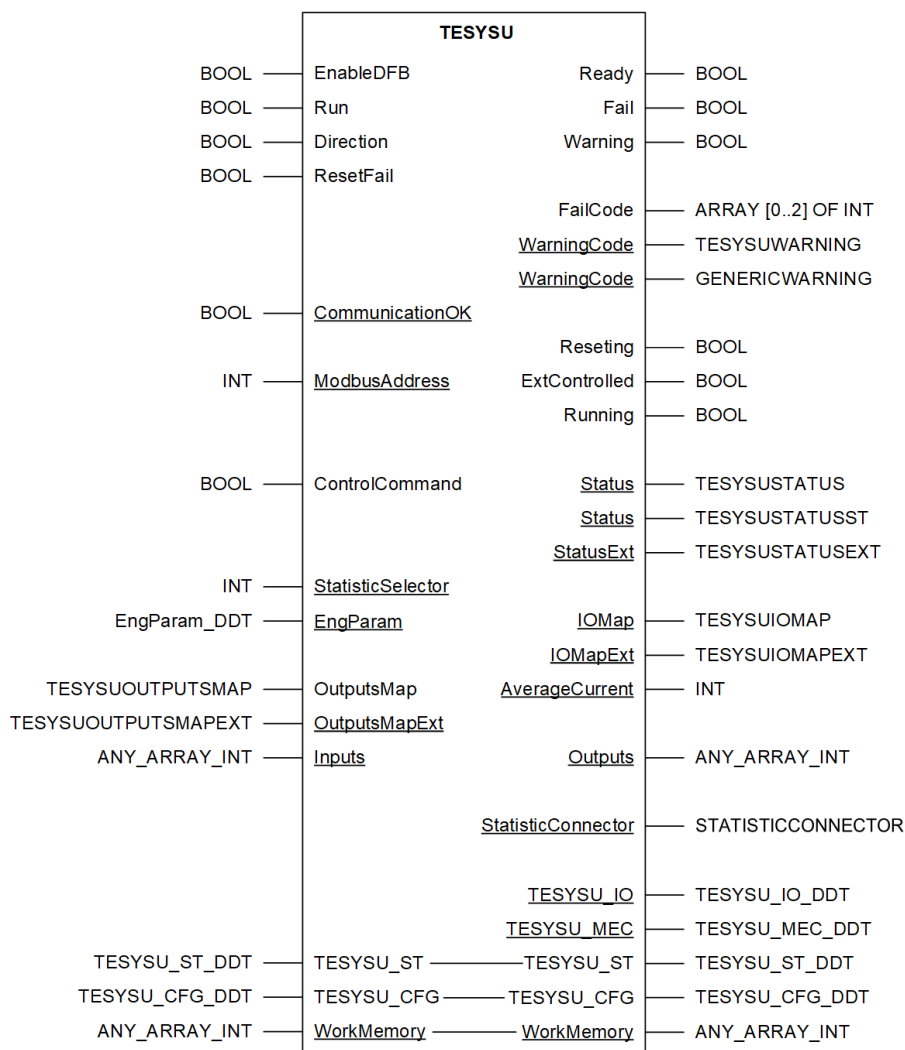
The main functions of the template are described in the following table:

Function	Description
Control	Forward/reverse direction of rotation.
Device status indication	Displays the status of the device.
Remote resetting	Allows resetting of the device.
Control or Monitoring	Enables you to control and monitor the device.
Owner	Manages the control system which is the owner (Operator or Program). Therefore, it is responsible for setting the control.

DFB Representation

Representation

The following figure represents the functional module of TESYS U profile:



NOTE: The underlined parameters are specific for some components.

The table shows the parameters available for specific components:

Parameters		Starter components				
		Modbus		Advantys		
		Standard (MBTESYSUSCST)	Advanced and multifunction (MBTESYSUSC)	Standard (TESYSUSCST)	Advanced (TESYSUCTL)	Multifunction (TESYSUSC)
Inputs	ModbusAddress	X	X	–	–	–
	CommunicationOK	–	–	X	X	X
	StatisticSelector	X	X	–	–	–
	EngParam	X	X	–	–	–
	Inputs	–	–	X	X	X
Outputs	WarningCode (GENERICWARNING)	X	–	X	–	–
	WarningCode (TESYSUWARNING)	–	X	–	X	X

Parameters		Starter components				
		Modbus		Advantys		
		Standard (MBTESYSUSCST)	Advanced and multifunction (MBTESYSUSC)	Standard (TESYSUSCST)	Advanced (TESYSUCTL)	Multifunction (TESYSUSC)
	Status (TESYSUSTATUSST)	X	–	X	–	–
	Status (TESYSUSTATUS)	–	X	–	X	X
	StatusExt (TESYSUSTATUSEXT)	–	X	–	–	X
	AverageCurrent	–	X	–	X	X
	StatisticConnector	X	X	–	–	–
	Outputs	–	–	X	X	X
	TESYSU_MEC	–	X	–	–	X
Input/Output	WorkMemory	X	X	–	–	–
X: Parameter is available. –: Parameter is not available.						

The table shows the parameters available for specific components:

Parameters		Controller components	
		Modbus	Advantys
		Advanced and multifunction (MBTESYSUC)	Advanced and multifunction (TESYSUC)
Inputs	ModbusAddress	X	–
	CommunicationOK	–	X
	OutputsMapExt	X	X
	StatisticSelector	X	–
	EngParam	X	–
	Inputs	–	X
Outputs	WarningCode (GENERICWARNING)	–	–
	WarningCode (TESYSUWARNING)	X	X
	Status (TESYSUSTATUSST)	–	–
	Status (TESYSUSTATUS)	X	X
	StatusExt (TESYSUSTATUSEXT)	X	–
	AverageCurrent	X	X
	IOMapExt	X	X
	StatisticConnector	X	–
	Outputs	–	X
	TESYSU_IO	X	X
Input/Output	WorkMemory	X	–
X: Parameter is available. –: Parameter is not available.			

Inputs

Input Parameter Description

Name	Data type	Description
EnabledDFB	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = The entire DFB is restarted (states, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. <p>Public variable values are loaded during the first enabling cycle.</p>

If the device is reset and *Run* pin is active, then the device will auto start. If manual start of the device is required, then reset the *Run* pin followed by device.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Reset the *Run* variable before resuming operation.

Failure to follow these instructions can result in equipment damage.

Name	Data type	Description	
Run	BOOL	1 = Starts the starter run in the direction selected with the <code>Direction</code> input variable. If the device is being enabled or reset, reset the inputs so that unexpected start does not occur. In these cases, first reset the input to resume the operation.	
Direction	BOOL	Direction of rotation of the starter. <ul style="list-style-type: none">0 = Activates the forward direction drive.1 = Activates the reverse direction drive.	
ResetFail	BOOL	1 = Resets the <code>Fail</code> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <code>ControlCommand</code> is 1.	
CommunicationOk*	BOOL	1 = The node is present on the bus. You can find this variable in Ethernet communications.	
ModbusAddress*	INT	Device address within the Modbus network. You can find this variable in Modbus communications.	
OutputsMapExt*	TESYSUOUTPUTS-MAPEXT	Standard outputs of the controller. The following table describes <code>TESYSUOUTPUTSMAPEXT</code> :	
	Name	Type	Description
	O13	BOOL	1 = Enables O13 digital output (if 687 LSB = 2).
	O23	BOOL	1 = Enables O13 digital output (if 687 LSB = 2).
ControlCommand	BOOL	Indicates to the DFB whether the motor is being controlled locally or from a source external to the DFB. <ul style="list-style-type: none">0 = Performs only read operations to monitor the status of the device and does not perform any control functions.1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. NOTE: This input does not configure the starter.	
StatisticSelector*	INT	Variable obtains statistics for the Modbus network (requests carried out, time between requests, so on). This data provides information for using <code>StatisticConnector</code> pin within the <code>StatisticCounter</code> DFB in <code>General Purpose</code> library for communication. The following table displays the <code>StatisticSelector</code> value:	
	Variable value	Description	
	1	Read statistics, client.	

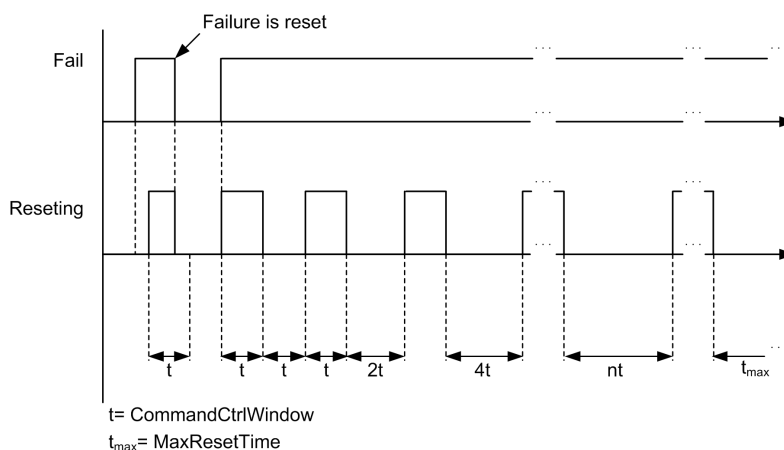
Name	Data type	Description
	2	Write statistics, client.
EngParam*	EngParam_DDT, page 111 EngParamMBTESYSUC/ EngParamMBTESYSUSC/ EngParamMBSTESYSUSCST	Engineering parameters.
OutputsMap	TESYSUOUTPUTS-MAP	Holds a structure used to control the Tesys outputs. If one of the variables is unavailable, a comment stating that the information cannot be accessed appears.
	Name	Type
	OA1	BOOL
	OA3	BOOL
	LO1	BOOL
Inputs ¹	ANY_ARRAY_INT	Holds an array structure with data obtained from the device. You can control the starter/controller with this input variable. This variable is reserved for the DFB, and you cannot use this variable directly. For the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247. Based on the device and the relevant communications, one of the structures is used. The information available for starters with a standard control unit on an Advantys STB is shown in the following table:
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
		Description
Inputs ²	ANY_ARRAY_INT	Holds an array structure with data obtained from the device. You can control the starter/controller with this input variable. This input is reserved for the DFB, and you cannot use this input directly. For the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247. Based on the device and the relevant communications, the following structure is used. The information available for starters with an advanced control unit on an Advantys STB is shown in the following table:
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
	Inputs [2]	INT
	Inputs [3]	INT
		Description
<p>*: Parameter is available for specific components.</p> <p>1: Parameter is available only for <i>TESYSUSCST</i>.</p> <p>2: Parameter is available only for <i>TESYSUCTL</i> and <i>TESYSUSC</i>.</p>		

EngParam_DDT

Name	Data type	Description
Refresh	TIME	Refresh time for device data on serial Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed and have maximum priority.
CommandCtrl-Window	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is issued.

Name	Data type	Description
		The command that is controlled is Run . In case of ResetFail , this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the Resetting output.
ScanTime	TIME	Allows the time that the alarm signals are kept active to be configured. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in CommandCtrlWindow is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in CommandCtrlWindow elapses. The second reset is carried out after CommandCtrlWindow * 2 elapses, so on. If the value of CommandCtrlWindow is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications are/the device is reset with the ResetFail variable.
	TRUE	Communications are/the device is reset automatically.
MaxResetTime	TIME	When in automatic ResetMode , this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.

Timing diagram:

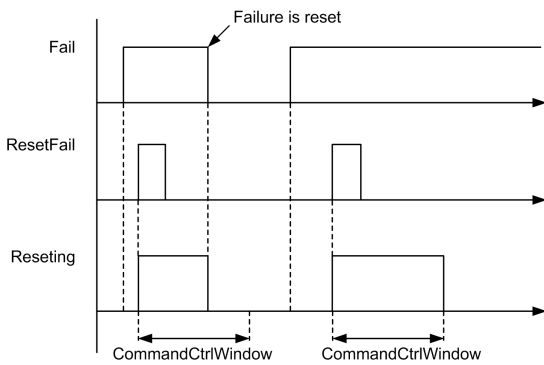


Outputs

Output Parameter Description

Output	Type	Description
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out or carrying out any Run or Stop command.
Fail	BOOL	1 = A detected failure in the control block or in the device or communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last diagnostic code is shown on FailCode .

Output	Type	Description	
		NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables retain their last value.	
Warning	BOOL	1 = An alarm has been activated for the device. It does not affect the blocks operation and does not need to be reset. The signal remains active until the cause for the alarm disappears.	
FailCode	ARRAY [0..2] OF INT	When Fail output is 1, it holds the code for the detected error. If Fail output is 0, it indicates the last detected error that occurred. The detected error source is specified by a 3-level structure. Refer to the <i>Diagnostics Information Management</i> , page 120 for more details.	
WarningCode*	GENERICWARNING	Holds a data structure with information about the alarm currently present on the starter with a standard control block.	
	Name	Type	Description
	Order	BOOL	1 = A follow-up alarm. The device is not responding to the control command (Run , so on) within the time specified in CommandCtrlWindow .
	Device	BOOL	1 = An alarm on the device.
WarningCode*	TESYSUWARNING	Holds a data structure with information about the alarm currently on the starter.	
	Name	Type	Description
	Ground	BOOL	1 = A ground alarm.
	Thermal	BOOL	1 = A thermal alarm on the device.
	LongStart	BOOL	1 = The start has been delayed.
	Jam	BOOL	1 = A mechanical jam alarm.
	Phase	BOOL	1 = A phase alarm.
	UnderCurrent	BOOL	1 = An undercurrent alarm.
	Communication	BOOL	1 = Communication interruption alarm.
	InternalTmp	BOOL	1 = An internal temperature alarm.
	Internal	BOOL	1 = An alarm for internal detected error.
	Module	BOOL	1 = An alarm is present on device.
	Order	BOOL	1 = A follow-up alarm. The device is not responding to the control command within the time specified in CommandCtrlWindow .
	ForcedLocalMode	BOOL	1 = The device is forced locally; indicated by the device having the ForcedLocalMode status. Is controlled through the screw terminal.
Running	BOOL	1 = The Tesys is running. The current is higher than 10% of FLA for advanced and multifunction starters and controllers.	
ExtControlled	BOOL	1 = The device is being controlled from an external source (for example, from the console, from a push-button panel, or from the monitoring system) to the system. Provides information for programming. NOTE: The ControlCommand signal, the Owner variable, and the ForcedLocalMode status are used to activate this signal. You cannot use this signal as a ControlCommand input.	
Reseting	BOOL	1 = A reset is being carried out. The CommandCtrlWindow variable indicates the maximum time for resetting the detected failure.	

Output	Type	Description																																	
		<p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code>.</p> <p>If the detected failure is reset, the <code>Fail</code> and <code>Resetting</code> output variables are reset (when set to FALSE). If the detected failure is not reset, the <code>Resetting</code> variable is FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Timing diagram:</p> 																																	
Status*	TESYSSTATUSST	<p>Holds data containing the information that the module extracts from the status variable of starters or controllers with advanced or multifunction control blocks.</p> <p>The following table describes the status information:</p> <table><tr><th>Parameter</th><th>Type</th><th>Description</th></tr><tr><td>Ready</td><td>BOOL</td><td>1 = Switch in ON position. Device is operational.</td></tr><tr><td>PoleClosed</td><td>BOOL</td><td>1 = Contacts are closed on power base.</td></tr><tr><td>Fault</td><td>BOOL</td><td>1 = An inoperable device.</td></tr><tr><td>Warning</td><td>BOOL</td><td>1 = An alarm is present on device.</td></tr><tr><td>Trip</td><td>BOOL</td><td>1 = The Tesys protection has tripped.</td></tr><tr><td>State</td><td>INT</td><td>Numerical code corresponding to the state of the starter.</td></tr><tr><td>Info</td><td>INT</td><td>Numerical code with the information on statuses and required actions.</td></tr></table>	Parameter	Type	Description	Ready	BOOL	1 = Switch in ON position. Device is operational.	PoleClosed	BOOL	1 = Contacts are closed on power base.	Fault	BOOL	1 = An inoperable device.	Warning	BOOL	1 = An alarm is present on device.	Trip	BOOL	1 = The Tesys protection has tripped.	State	INT	Numerical code corresponding to the state of the starter.	Info	INT	Numerical code with the information on statuses and required actions.									
	Parameter	Type	Description																																
	Ready	BOOL	1 = Switch in ON position. Device is operational.																																
	PoleClosed	BOOL	1 = Contacts are closed on power base.																																
	Fault	BOOL	1 = An inoperable device.																																
	Warning	BOOL	1 = An alarm is present on device.																																
	Trip	BOOL	1 = The Tesys protection has tripped.																																
	State	INT	Numerical code corresponding to the state of the starter.																																
	Info	INT	Numerical code with the information on statuses and required actions.																																
Status*	TESYSSTATUS	<p>Holds data structure containing the information that the module extracts from the status variable of controllers with advanced and multifunction control modules.</p> <p>The following table describes the status information:</p> <table><tr><th>Parameter</th><th>Type</th><th>Description</th></tr><tr><td>Ready</td><td>BOOL</td><td>1 = Switch in ON position. Device is operational.</td></tr><tr><td>PoleClosed</td><td>BOOL</td><td>1 = The contacts are closed on power base.</td></tr><tr><td>Fault</td><td>BOOL</td><td>1 = An inoperable device.</td></tr><tr><td>Warning</td><td>BOOL</td><td>1 = An alarm is present on device.</td></tr><tr><td>Trip</td><td>BOOL</td><td>1 = The Tesys protection has tripped.</td></tr><tr><td>ResetAuth</td><td>BOOL</td><td>1 = Reset authorized detected error.</td></tr><tr><td>PoweredUp</td><td>BOOL</td><td>1 = The A1/A2 contacts are powered with power base or powered I.1 or I.2 inputs.</td></tr><tr><td>Running</td><td>BOOL</td><td>1 = The device is active with a current higher than 10% of full-load amperage or pole closed.</td></tr><tr><td>ForcedLocalMode</td><td>BOOL</td><td>1 = Forced local mode on control unit.</td></tr><tr><td>Start</td><td>BOOL</td><td><ul style="list-style-type: none">1 = Rising current is higher than 10% of FLA.0 = Falling current is lower than 10% of FLA.</td></tr></table>	Parameter	Type	Description	Ready	BOOL	1 = Switch in ON position. Device is operational.	PoleClosed	BOOL	1 = The contacts are closed on power base.	Fault	BOOL	1 = An inoperable device.	Warning	BOOL	1 = An alarm is present on device.	Trip	BOOL	1 = The Tesys protection has tripped.	ResetAuth	BOOL	1 = Reset authorized detected error.	PoweredUp	BOOL	1 = The A1/A2 contacts are powered with power base or powered I.1 or I.2 inputs.	Running	BOOL	1 = The device is active with a current higher than 10% of full-load amperage or pole closed.	ForcedLocalMode	BOOL	1 = Forced local mode on control unit.	Start	BOOL	<ul style="list-style-type: none">1 = Rising current is higher than 10% of FLA.0 = Falling current is lower than 10% of FLA.
	Parameter	Type	Description																																
	Ready	BOOL	1 = Switch in ON position. Device is operational.																																
	PoleClosed	BOOL	1 = The contacts are closed on power base.																																
	Fault	BOOL	1 = An inoperable device.																																
	Warning	BOOL	1 = An alarm is present on device.																																
	Trip	BOOL	1 = The Tesys protection has tripped.																																
	ResetAuth	BOOL	1 = Reset authorized detected error.																																
	PoweredUp	BOOL	1 = The A1/A2 contacts are powered with power base or powered I.1 or I.2 inputs.																																
	Running	BOOL	1 = The device is active with a current higher than 10% of full-load amperage or pole closed.																																
	ForcedLocalMode	BOOL	1 = Forced local mode on control unit.																																
	Start	BOOL	<ul style="list-style-type: none">1 = Rising current is higher than 10% of FLA.0 = Falling current is lower than 10% of FLA.																																

Output	Type	Description	
	State, page 116	INT	Numerical code corresponding to the device status.
	Info, page 117	INT	Numerical code with the information on statuses and required actions.
StatusExt*	TESYSUSTATUSEXT	Holds a data structure with information on the mechanical status of advanced or multifunction starters on a Modbus network. The following table describes the StatusExt:	
	Parameter	Type	Description
	ButtonOn	BOOL	Indicates the button position: <ul style="list-style-type: none"> 0 = Off 1 = On
	ButtonTrip	BOOL	Indicates the button position: <ul style="list-style-type: none"> 0 = Not tripped 1 = Tripped
	VoltagePresent	BOOL	1 = 24 V dc power supply voltage is present on the outputs.
IOMap	TESYSUIOMAP	Holds a data structure that contains information on the state of the starter inputs and outputs. The following table describes the IOMap:	
	Parameter	Type	Description
	OA1	BOOL	1 = The state of the digital output OA1.
	OA3	BOOL	1 = The state of the digital output OA3.
	LO1	BOOL	1 = The state of the digital output LO1.
	LI1	BOOL	1 = The state of the digital input LI1.
	LI2	BOOL	1 = The state of the digital input LI2.
IOMapExt*	TESYSUIOMAPEXT	Standard inputs/outputs information of the controllers. The following table describes the IOMapExt:	
	Parameter	Type	Description
	I1	BOOL	1 = Output 13 locally controlled.
	I2	BOOL	1 = Output 23 locally controlled.
	I3	BOOL	1 = Contactor status healthy for output 13.
	I4	BOOL	1 = Contactor status healthy for output 23.
	I5	BOOL	1 = Input status (reset).
	I6	BOOL	1 = Input status (external detected fault)..
	I7	BOOL	1 = Input status (system ready).
	I8	BOOL	1 = Input status (free).
	I9	BOOL	1 = Input status (free).
	I10	BOOL	1 = Device is in local/remote mixed mode if 683=2.
	O13	BOOL	1 = Output 13 is closed.
	O23	BOOL	1 = Output 23 is closed.
	O95	BOOL	1 = Output 95, output 96 is closed and output 97 and output 98 is opened.
	O5	BOOL	1 = Output 5 and output 6 is closed.
AverageCurrent*	INT	Average value of device current in FLA%.	
StatisticConnector*	STATISTICCONNECTOR	Information data is used with Modbus communications to obtain statistics on the Modbus network.	

Output	Type	Description	
		This data provides information for using the <code>StatisticConnector</code> pin within the <code>StatisticCounter</code> DFB in the General Purpose library for communication.	
	Parameter	Type	Description
	Start	BOOL	1 = The operation has started.
	EndOk	BOOL	1 = The operation has ended correctly.
	EndNok	BOOL	1 = The operation has ended with a detected error.
	PartialTime	DINT	Partial time.
Outputs*	ANY_ARRAY_INT	<p>Holds an array structure with data sent to the device. You can control the starter/controller with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. For the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247.</p> <p>Based on the device and the relevant communications, the following structure is used.</p> <p>The information available for starters with a standard control unit on Advantys STB is shown in the following table:</p>	
	Parameter	Type	Description
	Outputs [0]	INT	System control.
	Outputs [1]	INT	Module control.
	Outputs [2]	INT	Output control.
TESYSU_MEC*	TESYSU_MEC_DDT	Information on the mechanical status of advanced or multifunction starters. Information for SCADA/HMI system.	
	Parameter	Type	Description
	MecStatus	WORD	<p>Provides information on the mechanical status of the device. Access to the data held in this bit word is read-only.</p> <p>Refer to the <code>TESYSU_MEC</code> table below.</p>
	Reserved	INT	Reserved.
TESYSU_IO	TESYSU_IO_DDT	Standard inputs/outputs information of the controllers for the controller base (used by HMI/SCADA). The following table describes the <code>TESYSU_IO</code> :	
	Parameter	Type	Description
	IOMap	WORD	<p>Input/output status on the controller base.</p> <p>Refer to the <code>IOMap</code> table below.</p>
	Reserved	INT	Reserved.
*: Parameter is available for specific components.			

State

The following table describes the `State` variable:

Variable value	Description
-2	Device detected error.
-1	Not initialized. Waiting for data.
0	Disabled.
5	Ready.
8	Inoperable device.
9	Device notification.

Info

The following table describes the `Info` variable:

Variable value	Description
1	Incorrect configuration of DFB parameter.
2	Waiting Ready.
8	Tesy working OK.
10	Waiting for device information.
11	Missing <code>EnabledDFB</code> .
12	Missing <code>CommunicationOK</code> . Communication interruption.
13	Device not ready or status word value is 0.
23	Remove <code>Run</code> has to be 0.
24	Remove <code>ResetFail</code> . Reset again.
81	Missing <code>ResetFail</code> . Inoperable device.
82	A reset is required.
99	Unknown status.

TESYSU_MEC_DDT Word

The following table describes the `TESYSU_MEC_DDT` word structure:

Variable value	Description
0	<code>ButtonOn</code>
1	<code>ButtonTrip</code>
2	<code>ContactoVoltagePresent</code>

MecStatus Word

The following table describes the `MecStatus` word structure:

Variable value	Description
0	Switch: ON.
1	Switch: Trip.
2	Auxiliary voltage on current communicator.

IOMap

The following table describes the `IOMap` variable:

Variable value	Description
0	Status of the input <i>I1</i> on a controller base.
1	Status of the input <i>I2</i> on a controller base.
2	Status of the input <i>I3</i> on a controller base.
3	Status of the input <i>I4</i> on a controller base.

Variable value	Description
4	Status of the input <i>I</i> 5 on a controller base.
5	Status of the input <i>I</i> 6 on a controller base.
6	Status of the input <i>I</i> 7 on a controller base.
7	Status of the input <i>I</i> 8 on a controller base.
8	Status of the input <i>I</i> 9 on a controller base.
9	Status of the input <i>I</i> 10 on a controller base.
10	Status of the output <i>O</i> 13 on a controller base.
11	Status of the output <i>O</i> 95 on a controller base.
12	Status of the output <i>O</i> 13 on a controller base.
13	Status of the output <i>O</i> 5 on a controller base.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
TESYSU_ST	TESYSU_ST_DDT, page 118	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is readable/writable from the HMI/SCADA system.
TESYSU_CFG	TESYSU_CFG_DDT, page 119	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
WorkMemory*	ANY_ARRAY_INT	Array is used for Modbus communications. This variable is used with a Modbus port that serializes Modbus requests in an optimum manner.
*: Parameter is available for specific components.		

TESYSU_ST_DDT Type

Name	Type	Description
STW, page 119	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW, page 119	WORD	Device control. Enables to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
AverageCurrent	REAL	Actual current value in FLA%.

TESYSU_CFG_DDT Type

Name	Type	Description
DataStatus, page 120	WORD	Provides the device status. Information on the Status output structure.
Info	INT	Tesys information. Its value is Info status in the Status output pin.
WarningCode, page 120	WORD	Tesys U alarm code information. It takes the value from the WarningCode output pin.
FailCode0	INT	Code of last level 0 detected error. Indicates which detected error has occurred.
FailCode1	INT	Code of last level 1 detected error. Indicates which detected error has occurred.
FailCode2	INT	Code of last level 2 detected error. Indicates which detected error has occurred.

TESYSU_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
2	Technological module is running.
3	Inoperable device.
4	Alarm on the device or repetitive detected fault alarm requires resetting.
5	Communication interruption.
6	Requires resetting. ResetFail is required.
7	Refer to the ExtControlled output pin, page 112.
8	Refer to the Reseting output pin, page 112.
9	Refer to the EnableDFB input pin, page 110.

TESYSU_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 110.
1	Owner.
3	Refer to the Direction input pin, page 110.
6	Refer to the Run input pin, page 110.
7	Refer to the ControlCommand input pin, page 110.

NOTE: The Owner bit enables to control the block from the TESYSU_ST_DDT input/output structure ignoring the input signals of the block. It enables control from a monitoring system (HMI, SCADA, operator screen) in the Manual mode without using the programmed switching operation.

TESYSU_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the <code>Ready</code> status in the <code>Status</code> output pin, page 112.
1	Refer to the <code>PoleClosed</code> status in the <code>Status</code> output pin, page 112.
2	Refer to the <code>Fault</code> status in the <code>Status</code> output pin, page 112.
3	Refer to the <code>Warning</code> status in the <code>Status</code> output pin, page 112.
4	Refer to the <code>Trip</code> status in the <code>Status</code> output pin, page 112.
10	Refer to the <code>OA1</code> in the <code>IOMap</code> output pin, page 112.
11	Refer to the <code>OA3</code> in the <code>IOMap</code> output pin, page 112.
12	Refer to the <code>LO1</code> in the <code>IOMap</code> output pin, page 112.
13	Refer to the <code>LI1</code> in the <code>IOMap</code> output pin, page 112.
14	Refer to the <code>LI2</code> in the <code>IOMap</code> output pin, page 112.

TESYSU_CFG.WarningCode Word Structure

Bit	Description
2	Refer to the <code>Ground</code> in the <code>WarningCode</code> output pin, page 112.
3	Refer to the <code>Thermal</code> in the <code>WarningCode</code> output pin, page 112.
4	Refer to the <code>LongStart</code> in the <code>WarningCode</code> output pin, page 112.
5	Refer to the <code>Jam</code> in the <code>WarningCode</code> output pin, page 112.
6	Refer to the <code>Phase</code> in the <code>WarningCode</code> output pin, page 112.
7	Refer to the <code>UnderCurrent</code> in the <code>WarningCode</code> output pin, page 112.
8	Refer to the <code>Order</code> in the <code>WarningCode</code> output pin, page 112.
9	Refer to the <code>ForcedLocalMode</code> in the <code>WarningCode</code> output pin, page 112.
10	Refer to the <code>Communication</code> in the <code>WarningCode</code> output pin, page 112.
11	Refer to the <code>InternalTmp</code> in the <code>WarningCode</code> output pin, page 112.
12	Refer to the <code>Internal</code> in the <code>WarningCode</code> output pin, page 112.
15	Refer to the <code>Module</code> in the <code>WarningCode</code> output pin, page 112.

NOTE: All control blocks cannot show alarm codes. Check the device manual and the `WarningCode` output variable.

Diagnostics Information Management

Overview

The detected error codes the device can return are read from the `FailCode` output variable.

Parameter Configuration Diagnostics Codes

This detected error indicates that a public variable parameter contains a value that is not allowed.

To reload new values, a rising edge is required on the `EnableDFB` input:

- `FailCode[0]: 16#0003`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

This diagnostic code can occur for any of the below conditions:

- Wrong array size at `Inputs` or `Outputs` pins of the function block.
- Variables connected to both `DTMInputs` and `Inputs` pins of the function Block.

During the above detected `FailCode` the function block does not process any inputs and the function blocks output displays the last processed state.

This detected `FailCode` can be reset by a rising edge to the `EnableDFB` input pin after correcting the configuration of the function block.

Modbus Communications Diagnostics Codes

For Modbus communications, this code is used to indicate that communications have not been established. It can be reset.

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

After Modbus communications have been established, check Modbus client diagnostic codes for `FailCode [0]` and `FailCode [1]`. Components make a distinction between detected read and write request problem:

- `FailCode[2]: 16#0001 Read`
- `FailCode[2]: 16#0002 Write`

Diagnostics Code Example

For a detected error, the code is:

- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

The `Failcode[0]` variable can have the following codes:

Diagnostic code	Meaning
0	No detected failure.
1	Trip due to short-circuit (Trip).
2	Magnetic trip (Trip)
3	Ground detected error trip (Trip).
4	Thermal overload.
5	Starting time too long.
6	Mechanical jam.
7	Phase imbalance.
8	Detected failure due to load condition.
9	Detected failure due to switching operation (Shunt trip).
10	Test trip (TRIP. Overheating simulation).
11	Interruption of communications on LUCM. Detected error in Modbus port (dropped out).
12	Interruption of communications on LUCM. Detected error in Modbus port (tripped).

Diagnostic code	Meaning
13	Reserved.
14	Module identification detected error.
15	Module not installed or missing 24 V supply.
51	Internal detected error in LUCM; temperature or broken sensor.
52 - 58	Internal detected error.
59	Current flow with unpowered winding.
60	L2 current detected in single-phase mode.
61	Trip base not detected.
62	Control cabling.
63	Control over voltage.
100	Detected internal problem in communications module.
101	Communication interruption with LUCM multifunction control unit.
102	Detected internal problem in communications module.
104	Detected internal problem in communications module.
105	Communication interruption with LUCM control base.
300	Communication interruption.

The inoperable device can be reset as long as `ControlCommand` is `TRUE`. Otherwise, it can be reset only on the controllers through the screw terminal with the appropriate parameter configuration.

NOTE: The diagnostics codes of device are available for Tesys, except for those that communicate through the Advantys STB.

Power Monitoring Devices

What's in This Part

PM Profile - Power Monitoring Profile	124
Smart UPS Profile	140
ACCUSINE PCS Profile.....	150

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Power Monitoring Devices family.

These function blocks do not reflect any specific installation.

⚠ WARNING

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines.¹
- Test each implementation of this library for proper operation before placing it 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.

PM Profile - Power Monitoring Profile

What's in This Chapter

Description	124
DFB Representation	125
Inputs	127
Outputs	129
Inputs/Outputs	135
Diagnostic Information Management.....	138

Overview

This chapter describes the DFBs of Power Monitoring profile.

Description

General

The PM profile is used to manage the Power logic series power meters on different communication networks (Modbus or Ethernet).

The power meter is a multifunction, digital instrumentation and data acquisition device. It can replace various meters, relays, transducers, and other components. You can install the power meter at multiple locations within a facility.

NOTE: *MBPM710*, *MBPM800*, *EPM800*, *EMPM800* and *MBPM9C* are deprecated control functions.

Function Description

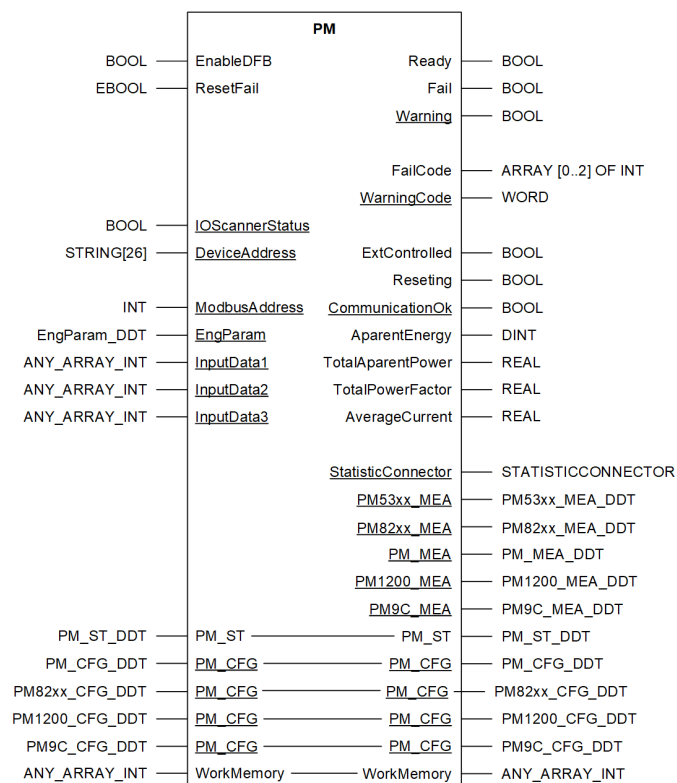
The main functions of the DFB are described in the following table:

Function	Description
Remote resetting	Allows resetting of the device.
Monitoring	Allows the device parameters to be monitored.

DFB Representation

Representation

The following figure represents the functional module of PM profile:



NOTE: The underlined parameters are specific for some components.

For DFBs communicating by I/O scanning, variables read from the device retain their last value when a communication interruption occurs. For details, refer to the description of the corresponding [output parameter](#), page 68.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

When you configure the *Last value* parameter of the I/O scanner line, take into consideration the behavior of the DFB when a communication interruption occurs.

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

The table shows the parameters available for specific components:

Parameters		Components								
		Modbus					Ethernet			
		MBP-M700 (Deprecated)	MBP-M800 (Deprecated)	MBP-M1200 (Deprecated)	MBP-M9C (Deprecated)	MBP-M53-50	EMP-M800 (Deprecated)	EMP-M53-xx	EMP-M82-xx	
In-puts	ModbusAddress	X	X	X	X	X	-	-	-	-

Parameters		Components								
		Modbus					Ethernet			
		<i>MBP-M700</i> (Deprecated)	<i>MB-P-M80-0</i> (Deprecated)	<i>MB-P-M12-00</i>	<i>MBP-M9C</i> (Deprecated)	<i>MBP-M53-50</i>	<i>E-P-M-80-0</i> (Deprecated)	<i>EMP-M800</i> (Deprecated)	<i>EMP-M53-xx</i>	<i>EMPM82-xx</i>
	<i>DEVI-CEAD-DRESS</i>	–	–	–	–	–	–	X	X	X
	<i>EngParam</i>	–	–	X	–	X	–	–	X	X
	<i>IOSCAN-NERSTATUS</i>	–	–	–	–	–	X	–	–	–
	<i>INPUT-DATA1</i>	–	–	–	–	–	X	–	–	–
	<i>INPUT-DATA2</i>	–	–	–	–	–	X	–	–	–
	<i>INPUT-DATA3</i>	–	–	–	–	–	X	–	–	–
Out-puts	<i>CommunicationOK</i>	–	–	–	–	–	X	–	–	–
	<i>Statistic-Connector</i>	X	X	X	X	X	–	X	X	X
	<i>Warning</i>	X	X	–	–	X	X	X	X	–
	<i>Warning-Code</i>	X	X	–	–	X	X	X	X	–
	<i>PM_MEA</i>	X	X	–	–	X	X	X	–	–
	<i>PM9C_MEA</i>	–	–	–	X	–	–	–	–	–
	<i>PM1200_MEA</i>	–	–	X	X	–	–	–	–	–
	<i>PM53xx_MEA</i>	–	–	–	–	–	–	–	X	–
	<i>PM82xx_MEA</i>	–	–	–	–	–	–	–	–	X

Parameters		Components								
		Modbus					Ethernet			
		MBP-M700 (Deprecated)	MB-P-M80-0 (Deprecated)	MB-P-M12-00	MBP-M9C (Deprecated)	MBP-M53-50	E-P-M-80-0 (Deprecated)	EMP-M800 (Deprecated)	EMP-M53-xx	EMPM82-xx
In-puts/ Out-puts	WorkMemory	X	X	X	X	X	–	X	X	X
	PM_ST_DDT	X	X	X	X	X	X	X	X	X
	PM_CFG (PM_CFG_DDT)	–	–	–	–	–	–	–	X	–
	PM_CFG (PM82xx_CFG_DDT)	–	–	–	–	–	–	–	–	X
	PM_CFG (P-M1200_CFG_DDT)	–	–	X	–	–	–	–	–	–
	PM_CFG (PM9C_CFG_DDT)	–	–	–	X	–	–	–	–	–
X: Parameter is available.										
–: Parameter is not available.										

Inputs

Input Parameter Description

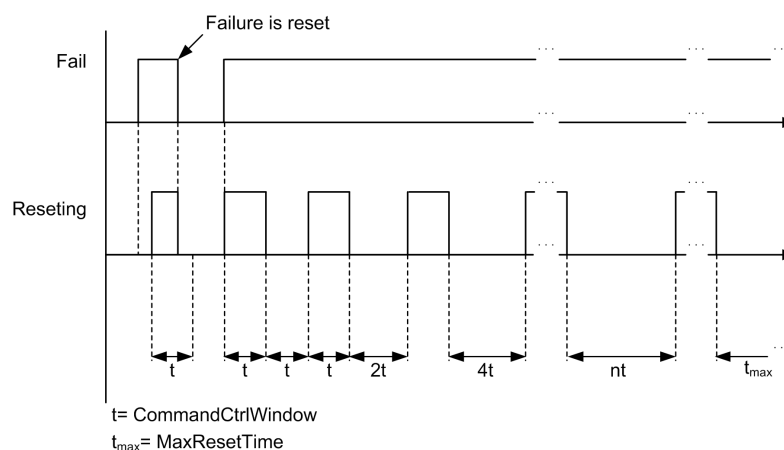
Name	Data type	Description
EnabledDFB	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. <p>Public variable values are loaded during the first enabling cycle.</p>
ResetFail	EBOOL	1 = Resets the Fail output parameter to 0 or in case of inoperable device, sends a reset command to the device if ControlCommand is 1.
ModbusAddress ¹	INT	Device address within the Modbus network.
DeviceAddress ²	STRING[26]	PM Ethernet address.
	Platform	IP Addressing DeviceAddress (variable)
	M340	'{IP}ID'
	Quantum	'{IP}ID'
	M580	'{IP}ID'
NOTE: ID is 255.		

Name	Data type	Description
EngParam	EngParam_DDT, page 128 EngParamEMPM53XX/ EngParamMBPM5350/ EngParamEMPM82XX/ EngParamMBPM1200	Engineering parameters.
InputData1 ³	ANY_ARRAY_INT	These arrays contain the read information obtained from the PM 800 unit through the IOScan. For the block to work properly, allocate the structure (%MWx)
InputData2 ³	ANY_ARRAY_INT	
InputData3 ³	ANY_ARRAY_INT	
<p>¹ Parameter available only for MBPM800, MBPM5350,MBPM9C, MBPM1200,MBPM710.</p> <p>²Parameter available only for EMPM800, EMPM53xx (PM5320,PM5340) and EMPM82xx (PM8240,PM8244)</p> <p>³Parameter available only for EPM800</p> <p>* The data type EngParamEMPM53XX is applicable for EMPM53XX DFB, EngParamMBPM5350 for MBPM5350 EngParamEMPM82XX for EMPM82XX and EngParamMBPM1200 for MBPM1200 DFB.</p>		

EngParam_DDT

Name	Data type	Description
Refresh	TIME	Refresh time for device data on Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed.
CommandCtrl-Window	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. The PM unit only allows using <code>ResetFail</code> for communications. If the detected failure continues after the period specified in <code>CommandCtrlWindow</code> has elapsed, <code>Fail</code> remains TRUE and the <code>Reseting</code> output resets (when set to FALSE).
ScanTime*	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset. (Minimum time to maintain alarm signals).
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in <code>CommandCtrlWindow</code> is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in <code>CommandCtrlWindow</code> elapses. The second reset is carried out after <code>CommandCtrlWindow * 2</code> elapses, so on. If the value of <code>CommandCtrlWindow</code> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications are/the device is reset with the <code>ResetFail</code> variable.
	TRUE	Communications are/the device is reset automatically.
MaxResetTime	TIME	When in automatic <code>ResetMode</code> , this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.
*: These parameters are available only with specific components.		

Timing diagram:



Outputs

Output Parameter Description

Parameter	Type	Description
Ready	BOOL	1 = The device is active and free of detected errors. The device is ready to carry out or is carrying out any read operation.
Fail	BOOL	<p>1 = A detected failure in the control block or in the device or communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode.</p> <p>NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.</p>
Warning*	BOOL	<p>1 = An alarm has been activated for the device. It does not affect the block operation, and does not need to be reset.</p> <p>This signal remains active until the cause of the alarm disappears.</p>
FailCode	ARRAY [0..2] OF INT	<p>When the Fail output is 1, it holds the code for the detected error.</p> <p>If the Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3-level structure. Refer to the Diagnostic Information Management, page 138 for more details.</p>
WarningCode*	WORD	<p>Variable holds the alarm code.</p> <p>The following table describes the WarningCode for PM700 block:</p>
	Bit	Description for PM700 (W1112)
	0	Phase 1 voltage out of range.
	1	Phase 2 voltage out of range.
	2	Phase 3 voltage out of range.
	3	Phase 1 current out of range.
	4	Phase 2 current out of range.
	5	Phase 3 current out of range.
	6	Frequency out of range, or phase 1 voltage insufficient to determine frequency.
	Bit	Description for MBPM800, EPM800, EMPM800, and MBPM5350 (W3254), EMPM53xx.

Parameter	Type	Description
	0	Summary bit (activated if any other bit is activated).
	1	Incorrect configuration.
	2	Incorrect scaling.
	3	Phase dropout.
	4	Incorrect cabling.
	5	Incremental energy could be incorrect due to the reset operation of the meter.
	6	External demand synchronous waiting time.
ExtControlled	BOOL	1 = Function block is controlled from the Supervision system (Operator mode).
Reseting	BOOL	<p>1 = A reset is being carried out.</p> <p>The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected failure.</p> <p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code>.</p> <p>If the detected failure is reset, the <code>Fail</code> and <code>Reseting</code> output variables are reset (when set to FALSE). If the detected failure is not reset, the <code>Reseting</code> variable is FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Refer to the Timing diagram below.</p>
CommunicationOK*	BOOL	1 = The communication is OK.
ApparentEnergy	DINT	Apparent power consumption (kVAh).
TotalApparentPower	REAL	Total apparent power (kVA).
TotalPowerFactor	REAL	Total power factor.
AverageCurrent	REAL	Average current between the three phases (A).
StatisticConnector*	STATISTICCONNECTOR	
	<p>The information data is used with Ethernet communication to obtain statistics on the Modbus network (requests carried out, time between requests, so on). This structure has been created for its use together with the <code>StatisticCounter</code> DFB in General Purpose updated for communication.</p> <p>The following table describes the <code>StatisticConnector</code>:</p>	
	Parameter	Type
	Start	BOOL
	EndOk	BOOL
	EndNOk	BOOL
	PartialTime	DINT
PM_MEA*	PM_MEA_DDT	
	<p>Data structure with the device measurement information.</p> <p>The following table describes the <code>PM_MEA_DDT</code>:</p>	
	Parameter	Type
	ActiveEnergy	REAL
	ReactiveEnergy	REAL
	TotalActivePower	REAL
	TotalReactivePower	REAL
	AverageLineToLineVoltage	REAL
	AverageLineToNeutralVoltage	REAL

Parameter	Type	Description	
	Frequency	REAL	Frequency (derived from phase 1) (Hz).
	ActualCurrentL1	REAL	Instantaneous current phase 1 (A).
	ActualCurrentL2	REAL	Instantaneous current phase 2 (A).
	ActualCurrentL3	REAL	Instantaneous current phase 3 (A).
	VoltageL1ToL2	REAL	Voltage, phases 1-2 (V).
	VoltageL2ToL3	REAL	Voltage, phases 2-3 (V).
	VoltageL1ToL3	REAL	Voltage, phases 1-3 (V).
	VoltageL1ToNeutral	REAL	Voltage, phase 1-N (V).
	VoltageL2ToNeutral	REAL	Voltage, phase 2-N (V).
	VoltageL3ToNeutral	REAL	Voltage, phase 3-N (V).
	ActivePowerL1	REAL	Real power, phase 1 (kW).
	ActivePowerL2	REAL	Real power, phase 2 (kW).
	ActivePowerL3	REAL	Real power, phase 3 (kW).
	ApparentPowerL1	REAL	Apparent power, phase 1 (kVA).
	ApparentPowerL2	REAL	Apparent power, phase 2 (kVA).
	ApparentPowerL3	REAL	Apparent power, phase 3 (kVA).
	ReactivePowerL1	REAL	Reactive power, phase 1 (kVAR).
	ReactivePowerL2	REAL	Reactive power, phase 2 (kVAR).
	ReactivePowerL3	REAL	Reactive power, phase 3 (kVAR).
	THDCurrentL1	REAL	THD, current, phase 1 (%).
	THDCurrentL2	REAL	THD, current, phase 2 (%).
	THDCurrentL3	REAL	THD, current, phase 3 (%).
	THDVoltageL1ToNeutral	REAL	THD, voltage, phase 1-N (%).
	THDVoltageL2ToNeutral	REAL	THD, voltage, phase 2-N (%).
	THDVoltageL3ToNeutral	REAL	THD, voltage, phase 3-N (%).
	THDVoltageL1ToL2	REAL	THD, voltage, phases 1-2 (%).
	THDVoltageL2ToL3	REAL	THD, voltage, phases 2-3 (%).
	THDVoltageL1ToL3	REAL	THD, voltage, phases 1-3 (%).
PM9C_MEA*	PM9C_MEA_DDT	Data structure with the device measurement information. The following table describes the PM1200_MEA_DDT:	
	Parameter	Type	Description
	ActiveEnergy	REAL	Active energy consumption (kWh).
	ReactiveEnergy	REAL	Reactive energy consumption (kVARh).
	TotalActivePower	REAL	Total real power (kW).
	TotalReactivePower	REAL	Total reactive power (kVAR).
	AverageLineToLineVoltage	REAL	Voltage, L-L, average between the 3 phases (V).
	AverageLineToNeutralVoltage	REAL	Voltage, L-N, average between the 3 phases (V).
	Frequency	REAL	Frequency (derived from phase 1) (Hz).
	ActualCurrentL1	REAL	Instantaneous current phase 1(A).

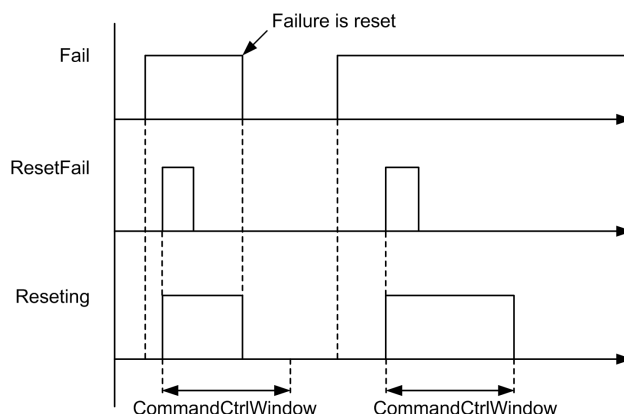
Parameter	Type	Description	
	ActualCurrentL2	REAL	Instantaneous current phase 2 (A).
	ActualCurrentL3	REAL	Instantaneous current phase 3 (A).
	VoltageL1ToL2	REAL	Voltage, phases 1-2 (V).
	VoltageL2ToL3	REAL	Voltage, phases 2-3 (V).
	VoltageL1ToL3	REAL	Voltage, phases 1-3 (V).
	VoltageL1ToNeutral	REAL	Voltage, phase 1-N (V).
	VoltageL2ToNeutral	REAL	Voltage, phase 2-N (V).
	VoltageL3ToNeutral	REAL	Voltage, phase 3-N (V).
	ActivePowerL1	REAL	Real power, phase 1 (kW).
	ActivePowerL2	REAL	Real power, phase 2 (kW).
	ActivePowerL3	REAL	Real power, phase 3 (kW).
	ReactivePowerL1	REAL	Reactive power, phase 1 (kVAR).
	ReactivePowerL2	REAL	Reactive power, phase 2 (kVAR).
	ReactivePowerL3	REAL	Reactive power, phase 3 (kVAR).
PM1200_MEA*	PM1200_MEA_DDT	Data structure with the device measurement information. The following table describes the PM1200_MEA_DDT:	
	Parameter	Type	Description
	ActiveEnergy	REAL	Active energy consumption (kWh).
	ReactiveInductiveEnergy	REAL	Inductive reactive energy (kVARh)
	ReactiveCapacitiveEnergy	REAL	Capacitive reactive energy (kVARh)
	TotalActivePower	REAL	Total real power (kW).
	TotalReactivePower	REAL	Total reactive power (kVAR).
	AverageLineToLineVoltage	REAL	Voltage, L-L, average between the 3 phases (V).
	AverageLineToNeutralVoltage	REAL	Voltage, L-N, average between the 3 phases (V).
	Frequency	REAL	Frequency (derived from phase 1) (Hz).
	ActualCurrentL1	REAL	Instantaneous current phase 1 (A).
	ActualCurrentL2	REAL	Instantaneous current phase 2 (A).
	ActualCurrentL3	REAL	Instantaneous current phase 3 (A).
	VoltageL1ToL2	REAL	Voltage, phases 1-2 (V).
	VoltageL2ToL3	REAL	Voltage, phases 2-3 (V).
	VoltageL1ToL3	REAL	Voltage, phases 1-3 (V).
	VoltageL1ToNeutral	REAL	Voltage, phase 1-N (V).
PM1200_MEA	VoltageL2ToNeutral	REAL	Voltage, phase 2-N (V)
	VoltageL3ToNeutral	REAL	Voltage, phase 3-N (V).
	ActivePowerL1	REAL	Real power, phase 1 (kW).
	ActivePowerL2	REAL	Real power, phase 2 (kW).
	ActivePowerL3	REAL	Real power, phase 3 (kW).
	AparentPowerL1	REAL	Apparent power, phase 1 (kVA).
	AparentPowerL2	REAL	Apparent power, phase 2 (kVA).
	AparentPowerL3	REAL	Apparent power, phase 3 (kVA).

Parameter	Type	Description	
	ReactivePowerL1	REAL	Reactive power, phase 1 (kVAR).
	ReactivePowerL2	REAL	Reactive power, phase 2 (kVAR).
	ReactivePowerL3	REAL	Reactive power, phase 3 (kVAR).
	THDCurrentL1	REAL	THD, current, phase 1 (%).
	THDCurrentL2	REAL	THD, current, phase 2 (%).
	THDCurrentL3	REAL	THD, current, phase 3 (%).
	THDVoltageL1ToNeutral	REAL	THD, voltage, phase 1-N (%).
	THDVoltageL2ToNeutral	REAL	THD, voltage, phase 2-N (%).
	THDVoltageL3ToNeutral	REAL	THD, voltage, phase 3-N (%).
PM53xx_MEA*	PM53xx_MEA_DDT	Data structure with the device measurement information. The following table describes the PM53xx_MEA_DDT:	
	Parameter	Type	Description
	ActiveEnergy	REAL	Active energy consumption (kWh).
	Reactive Energy	REAL	Reactive energy consumption (kVARh).
	TotalActivePower	REAL	Total active power detected (kW).
	TotalReactivePower	REAL	Total reactive power detected (kVAR).
	AverageLineToLineVoltage	REAL	Average voltage between lines (V).
	AverageLineToNeutralVoltage	REAL	Average voltage between line and ground (V).
	Frequency	REAL	Detected frequency from line 1 (Hz).
	ActualCurrentL1	REAL	Instant current from line 1(A).
	ActualCurrentL2	REAL	Instant current from line 2(A).
	ActualCurrentL3	REAL	Instant current from line 3(A).
	VoltageL1ToL2	REAL	Voltage between line 1 and 2(V).
	VoltageL2ToL3	REAL	Voltage between line 2 and 3(V).
	VoltageL1ToL3	REAL	Voltage between line 1 and 3(V).
	VoltageL1ToNeutral	REAL	Voltage between line 1 and ground (V).
	VoltageL2ToNeutral	REAL	Voltage between line 2 and ground (V).
	VoltageL3ToNeutral	REAL	Voltage between line 3 and ground (V).
	ActivePowerL1	REAL	Active power at line 1 (kW).
	ActivePowerL2	REAL	Active power at line 2 (kW).
	ActivePowerL3	REAL	Active power at line 3 (kW).
	ApparentPowerL1	REAL	Apparent power at line 1 (kVA).
	ApparentPowerL2	REAL	Apparent power at line 2 (kVA).
	ApparentPowerL3	REAL	Apparent power at line 3 (kVA).
	ReactivePowerL1	REAL	Reactive power at line 1 (kW).
	ReactivePowerL2	REAL	Reactive power at line 2 (kW).
	ReactivePowerL3	REAL	Reactive power at line 3 (kW).
	UnbalVoltageLinetoLine-Worst	REAL	Unbalanced voltage line to line worst (%).

Parameter	Type	Description	
	ActivePowerDemand	REAL	Active power demand - Present (kW)
	ReactivePowerDemand	REAL	Reactive power demand - Present (kVAR)
	ApparentPowerDemand	REAL	Apparent power demand - Present (kVA)
PM82xx_MEA*	PM82xx_MEA_DDT	Data structure with the device measurement information. The following table describes the PM82xx_MEA_DDT:	
	Parameter	Type	Description
	ActiveEnergy	REAL	Active energy consumption (kWh).
	ReactiveEnergy	REAL	Reactive energy consumption (kVARh).
	TotalActivePower	REAL	Total Active power (kW)
	TotalReactivePower	REAL	Total reactive power (kVAR).
	AverageLineToLineVoltage	REAL	Average L-L Voltage, between the 3 phases (V).
	AverageLineToNeutral Voltage	REAL	Average L-N Voltage, between the 3 phases (V).
	Frequency	REAL	Frequency (Hz)
	ActualCurrentL1	REAL	Instantaneous current from Phase 1 (A).
	ActualCurrentL2	REAL	Instantaneous current from Phase 2 (A).
	ActualCurrentL3	REAL	Instantaneous current from Phase 3 (A).
	VoltageL1ToL2	REAL	Voltage between line 1 and 2 (V).
	VoltageL2ToL3	REAL	Voltage between line 2 and 3 (V).
	VoltageL1ToL3	REAL	Voltage between line 1 and 3 (V).
	VoltageL1ToNeutral	REAL	Voltage between line 1 and ground (V).
	VoltageL2ToNeutral	REAL	Voltage between line 2 and ground (V).
	VoltageL3ToNeutral	REAL	Voltage between line 3 and ground (V).
	ActivePowerL1	REAL	Active power at line 1(kW).
	ActivePowerL2	REAL	Active power at line 2 (kW).
	ActivePowerL3	REAL	Active power at line 3 (kW).
	ApparentPowerL1	REAL	Apparent power at line 1 (kVA).
	ApparentPowerL2	REAL	Apparent power at line 2 (kVA).
	ApparentPowerL3	REAL	Apparent power at line 3 (kVA).
	ReactivePowerL1	REAL	Reactive Power at Line 1 (kVAR).
	ReactivePowerL2	REAL	Reactive Power at Line 2 (kVAR).
	ReactivePowerL3	REAL	Reactive Power at Line 3 (kVAR).
	UnbalancedVoltageLtoNWorst	REAL	Unbalanced voltage Line to Neutral Worst(%).
	ActivePowerLastDemand	REAL	Active power last demand (kW). (kW = Active power demand delivered - Active power demand received)

Parameter	Type	Description	
	<i>ReactivePowerTotalLastDemand</i>	REAL	Reactive power total last demand (kVAR). (kVAR = Reactive power demand delivered + Reactive power demand received)
	<i>ApparentPowerLastDemand</i>	REAL	Apparent power last demand (kVA). (kVA = Apparent power demand delivered - Apparent power demand received)
* These parameters are available only with specific components.			

Timing diagram:



Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PM_ST	PM_ST_DDT	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is usable from the HMI/SCADA system.
PM_CFG	PM_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
<i>PM_CFG*</i>	PM82xx_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
<i>PM_CFG*</i>	PM1200_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
<i>PM_CFG*</i>	PM9C_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.
WorkMemory	ANY_ARRAY_INT	Array is used for Modbus communications. This variable is to be used with a Modbus port that serializes Modbus requests in an optimum manner.
* These parameters are available only with specific components.		

PM_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	Device control. Enables to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
AparentEnergy	DINT	Apparent power consumption (kVAh).
TotalAparentPower	REAL	Total apparent power (kVA).
TotalPowerFactor	REAL	Total power factor.
AverageCurrent	REAL	Average current between the 3 phases (A).

PM_CFG_DDT Type

Name	Type	Description
WarningCode	WORD	Alarm code. Detected PM alarm code information. Takes the values from the <code>WarningCode</code> output pin. NOTE: <code>WarningCode</code> is not applicable for EMPM82xx
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred <code>FailCode[2]</code> .

PM82xx_CFG_DDT

Name	Type	Description
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred <code>FailCode[2]</code> .

PM1200_CFG_DDT Type

Name	Type	Description
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred <code>FailCode[2]</code> .

PM9C_CFG_DDT Type

Name	Type	Description
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred <code>FailCode[2]</code> .

PM_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
2	Module is running.
3	Inoperable device.
4	Alarm on the device or repetitive detected fault alarm requires resetting. NOTE: Not applicable for EMPM82xx
5	Communication interruption.
6	Requires resetting. <code>ResetFail</code> is required.
8	Refer to the <code>Resetting</code> output pin, page 129.
9	Refer to the <code>EnableDFB</code> input pin, page 127.

PM_ST.CFGW Word Structure

Bit	Description
0	Refer to the <code>ResetFail</code> input pin, page 127.
1	Owner.

PM_CFG.WarningCode Word Structure

Bit	Description for PM700
0	Phase 1 voltage out of range.
1	Phase 2 voltage out of range.
2	Phase 3 voltage out of range.
3	Phase 1 current out of range.
4	Phase 2 current out of range.
5	Phase 3 current out of range.
6	Frequency out of range or phase 1 voltage is insufficient to determine frequency.

Bit	Description for PM800
0	Summary bit (activated if any other bit is activated).
1	Incorrect configuration
2	Incorrect scaling.
3	Phase dropout.
4	Incorrect cabling.
5	Incremental energy could be incorrect due to the reset operation of the meter.
6	External demand synchronous waiting time.

Diagnostic Information Management

Overview

The diagnostic codes the device can return are read from the `FailCode` output variable.

Parameter Configuration Diagnostic Codes

This detected error indicates that a public variable parameter contains a value that is not allowed.

To reload new values, a rising edge is required on the `EnableDFB` input:

- `FailCode[0]: 16#0003`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

NOTE: The inoperable device reset is sent to the device.

Modbus Communications Diagnostic Codes

For Modbus communications, this code is used to indicate that communications have not been established. It can be reset.

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

After Modbus communications have been established, check Modbus client diagnostic codes for `FailCode [0]` and `FailCode [1]`. The components make a distinction between read request problems and write request problems:

- `FailCode [2]: 16#0001` - Read
- `FailCode [2]: 16#0002` - Write

Smart UPS Profile

What's in This Chapter

Description	140
DFB Representation	141
Inputs	141
Outputs	143
Inputs/Outputs	146
Diagnostic Information Management.....	148

Overview

This chapter describes the DFB of Smart UPS profile.

Description

General

The *MBSMARTUPS* DFB is the control block to manage SMART UPS uninterrupted power supplies based on a Modbus.

NOTE: The *MBSMARTUPS* DFB requires a Smart UPS device connected through an AP9622 communication card.

Function Description

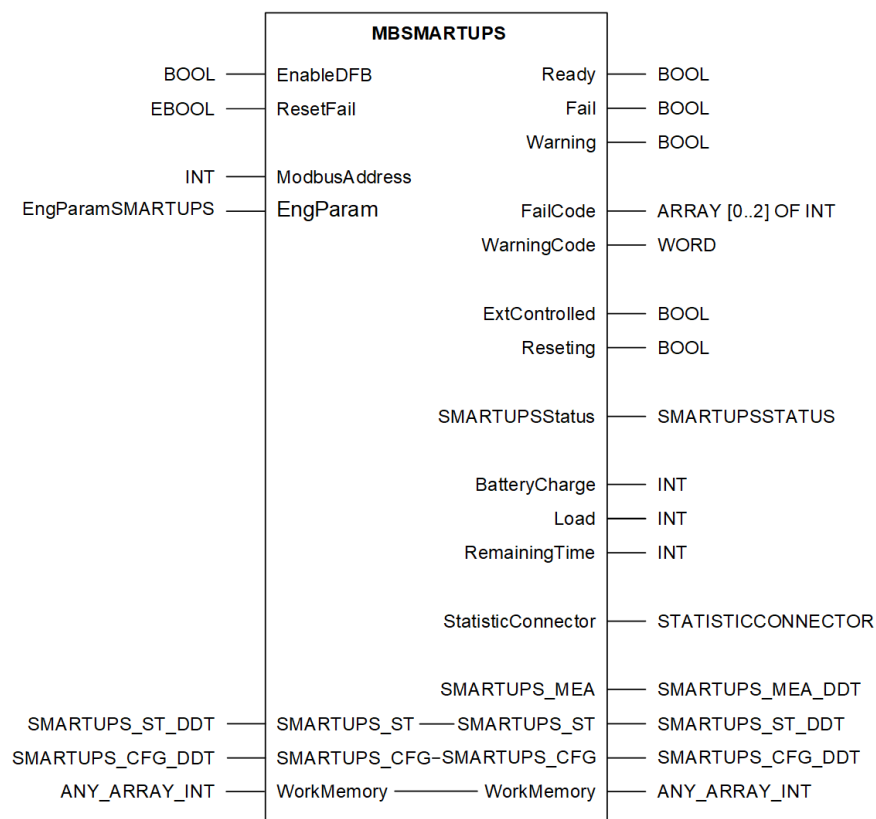
The main functions of the DFB are described in the following table:

Function	Description
Device detected failure	Monitors the inoperable devive.
Remote resetting	Enables you to reset communicationinterruptions.
Monitoring	Allows the device parameters to be monitored.

DFB Representation

Representation

The following figure represents MBSMARTUPS DFB in Control Expert:



Inputs

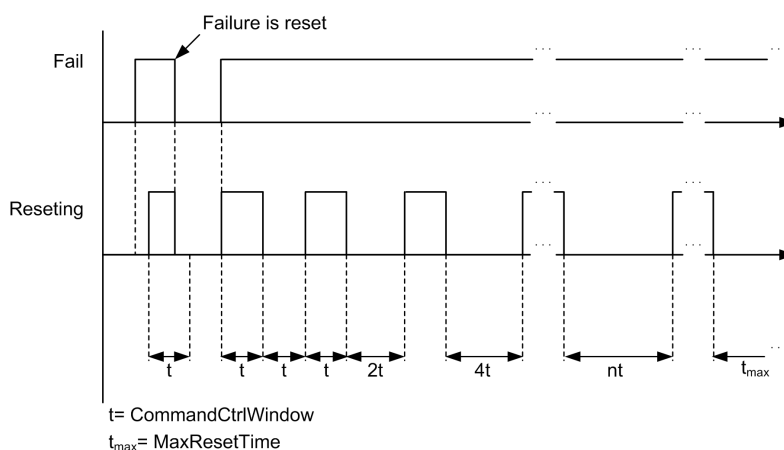
Input Parameter Description

Name	Data type	Description
<i>EnableDFB</i>	BOOL	This input enables the normal execution of the control block. <ul style="list-style-type: none"> 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. Public variable values are loaded during the first enabling cycle.
<i>ResetFail</i>	EBOOL	1 = Resets the <i>Fail</i> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <i>ControlCommand</i> is 1.
<i>ModbusAddress</i>	INT	Device address within the Modbus network.
<i>EngParam</i>	EngPar-amS-MART-UPS, page 142	Engineering parameters.

EngParamSMARTUPS

Name	Data type	Description
<i>Refresh</i>	TIME	Refresh time for device data on Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed.
<i>CommandCtrlWindow</i>	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. The PM only allows a communications <i>ResetFail</i> . If the failure detected continues after the <i>CommandCtrlWindow</i> time has elapsed, <i>Fail</i> remains TRUE and the <i>Reseting</i> resets (resets to FALSE).
<i>ScanTime</i>	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
<i>ResetMode</i>	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in <i>CommandCtrlWindow</i> is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in <i>CommandCtrlWindow</i> elapses. The second reset is carried out after <i>CommandCtrlWindow</i> * 2 elapses, so on. If the value of <i>CommandCtrlWindow</i> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable Value	Description
	FALSE	Communications are/the device is reset with the <i>ResetFail</i> variable.
	TRUE	Communications are/the device is reset automatically.
<i>MaxResetTime</i>	TIME	When in automatic <i>ResetMode</i> , this variable sets the maximum time between 2 consecutive resets. Refer to the timing diagram below.

Timing diagram:



Outputs

Output Parameter Description

Output	Type	Description	
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out any operation.	
Fail	BOOL	1 = A detected failure in the control block or in the device or communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode. NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.	
Warning	BOOL	1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset. The signal remains active until the cause for the alarm disappears.	
FailCode	ARRAY [0..2] OF INT	When Fail output is 1, it holds the code for the detected error. If Fail output is 0, it indicates the last detected error that occurred. The detected error source is specified by a 3 level structure. Refer to the Diagnostic Information Management, page 177 section.	
WarningCode	WORD	The WarningCode output holds a data structure with information on the alarm currently on the UPS.	
	Bit	Description	
	0	Replace battery.	
	1	Low battery.	
	2	Overload.	
ExtControlled	BOOL	1 = The device is being controlled from an external source (for example, from the monitoring system) to the system.	
Resetting	BOOL	1 = A reset is being carried out. The CommandCtrlWindow variable indicates the maximum time for resetting the detected failure. When a device or communication reset is carried out with ResetFail, the DFB tries to reset the detected failure within the time period defined in CommandCtrlWindow. If the detected failure is reset, the Fail and Resetting output variables are reset (set to FALSE). On the other hand, if the detected failure is not reset, the Resetting variable is set to FALSE and the Fail variable remains TRUE. The ResetFail is edge-based. Refer to the timing diagram below.	
SMARTUPSStatus	SMARTUPSSTATUS	The structure holds the data containing the information that the block extracts from the status word variable of the device. The following table describes the status information:	
	Parameter	Type	Description
	Status	SMART-UPS_DeviceStatus, page 144	General status of SMARTUPS.
	Warning	SMART-UPS_Warning, page 145	SMARTUPS notification condition.
	Fail	SMART-UPS_Fail, page 145	SMARTUPS failure detected condition.

Output	Type	Description	
<i>BatteryCharge</i>	INT	Indicates the remaining battery capacity as a percentage of the fully-charged condition (%) of the battery.	
<i>Load</i>	INT	Indicates the UPS output load as a percentage of full rated load in watts (%).	
<i>RemainingTime</i>	INT	Indicates the remaining runtime (min).	
<i>StatisticConnector</i>	STATISTICCONNECTOR	Information data used with Modbus communications to obtain statistics on the Modbus network (for example, requests carried out, time between requests, so on. This structure has been created for its use together with the Communications library Statistics module.	
	Parameter	Type	Description
	<i>Start</i>	BOOL	1 = The operation has started.
	<i>EndOk</i>	BOOL	1 = The operation has ended correctly.
	<i>EndNOK</i>	BOOL	1 = The operation has ended with a detected error.
	<i>PartialTime</i>	DINTt	Partial time.
<i>SMARTUPS_MEA</i>	SMARTUPS_MEA_DDT	Data structure with device measurement information.	
	Parameter	Type	Description
	<i>NominalOutput</i>	INT	Nominal output voltage (V).
	<i>ActualOutput</i>	INT	Actual output voltage (V).
	<i>InputVoltage</i>	INT	Input voltage (V).

SMARTUPS_DeviceStatus

The following table describes the *SMARTUPS_DeviceStatus* variables:

Parameter	Type	Description
<i>OnLine</i>	BOOL	1 = The device is on line.
<i>OnBattery</i>	BOOL	1 = The battery is on.
<i>Bypass</i>	BOOL	1 = The SMARTUPS is in bypass mode.
<i>Warning</i>	BOOL	1 = The SMARTUPS notification.
<i>Fail</i>	BOOL	1 = A SMARTUPS fault detected.
<i>AVRBoost</i>	BOOL	1 = AVRBoost.
<i>AVRTrim</i>	BOOL	1 = AVRTrim.
<i>TurningOn</i>	BOOL	1 = The device is turning on.
<i>BypassInternalFault</i>	BOOL	1 = The device is in bypass mode due to an internal fault detected, indicated through register 0002 or 0003.
<i>GoingBypass</i>	BOOL	1 = The device is switching to bypass mode due to a command.
<i>BypassCmd</i>	BOOL	1 = The device is in bypass mode due to a command.
<i>Returning</i>	BOOL	1 = The device is returning from bypass mode.
<i>BypassMan</i>	BOOL	1 = The device is in bypass mode as a result of manual bypass control.
<i>ReadyToPower</i>	BOOL	1 = The device is ready to provide the power to the load on the user command.
<i>ReadyToProvide</i>	BOOL	1 = The device is ready to provide the power to the load on the return of normal line voltage or on the user command.

SMARTUPS_Warning

The following table describes the *SMARTUPS_Warning* variables:

Parameter	Type	Description
<i>ReplaceBattery</i>	BOOL	1 = The battery needs to be replaced.
<i>LowBattery</i>	BOOL	1 = Low battery.
<i>Overload</i>	BOOL	1 = Overload.

SMARTUPS_Fail

The following table describes the *SMARTUPS_Fail* variables:

Parameter	Type	Description
<i>LowBattShutdown</i>	BOOL	1 = The UPS output is not receiving power due to low-battery shutdown.
<i>OverloadFault</i>	BOOL	1 = The UPS is unable to switch to on-battery operation due to overload.
<i>RelayMalfunction</i>	BOOL	1 = The main relay is not working.
<i>SleepMode</i>	BOOL	1 = The UPS is in sleep mode.
<i>ShutdownMode</i>	BOOL	1 = The UPS is in shutdown mode.
<i>BatteryCharger-Failure</i>	BOOL	1 = Battery charger failure detected.
<i>BypassRelay-Malfunction</i>	BOOL	1 = The bypass relay is not working.
<i>Temperature-Fault</i>	BOOL	1 = UPS fault detected, internal temperature exceeded nominal limits.
<i>FanFailure</i>	BOOL	1 = Electronic unit fan failure detected.
<i>IsolationFanFailure</i>	BOOL	1 = Isolation unit fan failure detected.
<i>BypassSupply-Failure</i>	BOOL	1 = Bypass supply failure detected.
<i>OutputFailure</i>	BOOL	1 = Output voltage selection failure detected – UPS is in bypass mode.
<i>UPSFault</i>	BOOL	1 = UPS fault detected – UPS is in bypass mode.
<i>NoBatteriesAttached</i>	BOOL	1 = The UPS commanded out of bypass mode with no batteries attached – UPS is in bypass mode.
<i>AVRFault</i>	BOOL	1 = An AVR Boost or trim relay fault detected.
<i>InverterFault</i>	BOOL	1 = Inverter fault detected.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
<i>SMARTUPS_ST</i>	SMARTUPS_ST_DDT	Device data structure that holds the minimum information required for performing control and monitoring. The information is used by the operator screen and can be read from or written to HMI/SCADA system.
<i>SMARTUPS_CFG</i>	SMARTUPS_CFG_DDT	Data structure with device information. Information used by the operator screen is readable from HMI/SCADA system.
<i>WorkMemory</i>	ANY_ARRAY_INT	The array is used in Modbus communications. This variable is to be used with a Modbus port that serializes Modbus requests in an optimum manner.

SMARTUPS_ST_DDT Type

Name	Type	Description
<i>STW</i>	WORD	Provides the device status. Access to the data held in this bit word is read-only.
<i>CFGW</i>	WORD	Enables controlling of the device from the monitoring subsystem or from the operator screen if <i>Owner</i> (1), or only from the monitoring subsystem if <i>Owner</i> (0). If <i>Owner</i> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
<i>BatteryCharge</i>	INT	Battery capacity. Remaining battery capacity as a percentage of the fully-charged condition (%) of the battery.
<i>Load</i>	INT	UPS output load as a percentage of full rated load in watts (%).
<i>RemainingTime</i>	INT	Remaining runtime (min).

SMARTUPS_CFG_DDT Type

Name	Type	Description
<i>DataStatus</i>	WORD	Provides the status information of the device for displaying in the background on the monitoring subsystem.
<i>WarningStatus</i>	WORD	Provides the notification information of the device for displaying in the background on the monitoring subsystem.
<i>FailStatus</i>	WORD	Provides detected failure information of the device for displaying, in the background on the monitoring subsystem.
<i>WarningCode</i>	WORD	Provides SMART UPS alarm code information. Takes values from the <i>WarningCode</i> output (W3).
<i>FailCode0</i>	INT	Indicates which detected error has occurred, <i>FailCode</i> [0].
<i>FailCode1</i>	INT	Indicates which detected error has occurred, <i>FailCode</i> [1].
<i>FailCode2</i>	INT	Indicates which detected error has occurred, <i>FailCode</i> [2].

SMARTUPS_ST.STW Word Structure

Bit	Description
0	Unknown technological module status. No variable refreshing.
1	Not ready.
2	Module is running.
3	Device failure detected.
4	Alarm on the device or repetitive detected fault alarm requires resetting.
5	Communication interruption.
6	Requires resetting. ResetFail is required.
8	Refer to the Resetting output pin, page 143.
9	Refer to the EnableDFB input pin, page 141.
10	On line.
11	Battery on.
12	Bypass mode.

SMARTUPS_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 141.
1	Owner.

SMARTUPS_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the OnLine status in the Status output pin, page 143.
1	Refer to the OnBattery status in the Status output pin, page 143.
2	Refer to the Bypass status in the Status output pin, page 143 .
3	Refer to the Warning status in the Status output pin, page 143.
4	Refer to the Fail status in the Status output pin, page 143.
5	Refer to the AVRBoost status in the Status output pin, page 143.
6	Refer to the AVRTrim status in the Status output pin, page 143.
7	Refer to the TurningOn status in the Status output pin, page 143.
8	Refer to the BypassInternalFault status in the Status output pin, page 143.
9	Refer to the GoingBypass status in the Status output pin, page 143.
10	Refer to the BypassCmd status in the Status output pin, page 143.
11	Refer to the Returning status in the Status output pin, page 143.
12	Refer to the BypassMan status in the Status output pin, page 143.
13	Refer to the ReadyToPower status in the Status output pin, page 143.
14	Refer to the ReadyToProvide status in the Status output pin, page 143.

SMARTUPS_CFG.WarningStatus Word Structure

Bit	Description
0	Refer to the <code>ReplaceBattery</code> output pin, page 143.
1	Refer to the <code>LowBattery</code> output pin, page 143.
2	Refer to the <code>Overload</code> output pin, page 143.

SMARTUPS_CFG.FailStatus Word Structure

Bit	Description
0	Refer to the <code>LowBattShutdown</code> status in the <i>FailStatus</i> output pin, page 143.
1	Refer to the <code>OverloadFault</code> status in the <i>FailStatus</i> output pin, page 143.
2	Refer to the <code>RelayMalfunction</code> status in the <i>FailStatus</i> output pin, page 143.
3	Refer to the <code>SleepMode</code> status in the <i>FailStatus</i> output pin, page 143.
4	Refer to the <code>ShutdownMode</code> status in the <i>FailStatus</i> output pin, page 143.
5	Refer to the <code>BatteryChargerFailure</code> status in the <i>FailStatus</i> output pin, page 143.
6	Refer to the <code>BypassRelayMalfunction</code> status in the <i>FailStatus</i> output pin, page 143.
7	Refer to the <code>TemperatureFault</code> status in the <i>FailStatus</i> output pin, page 143.
8	Refer to the <code>FanFailure</code> status in the <i>FailStatus</i> output pin, page 143.
9	Refer to the <code>IsolationFanFailure</code> status in the <i>FailStatus</i> output pin, page 143.
10	Refer to the <code>BypassSupplyFailure</code> status in the <i>FailStatus</i> output pin, page 143.
11	Refer to the <code>OutputFailure</code> status in the <i>FailStatus</i> output pin, page 143.
12	Refer to the <code>UPSFault</code> status in the <i>FailStatus</i> output pin, page 143.
13	Refer to the <code>NoBatteriesAttached</code> status in the <i>FailStatus</i> output pin, page 143.
14	Refer to the <code>AVRFault</code> status in the <i>FailStatus</i> output pin, page 143.
15	Refer to the <code>InverterFault</code> status in the <i>FailStatus</i> output pin, page 143.

SMARTUPS_CFG.WarningCode Word Structure

Bit	Description
5	Refer to the <code>Overload</code> in the <i>WarningCode</i> output pin, page 143.
6	Refer to the <code>Low battery</code> in the <i>WarningCode</i> output pin, page 143.
7	Refer to the <code>Replace battery</code> in the <i>WarningCode</i> output pin, page 143.

Diagnostic Information Management

Overview

The diagnostic codes the device can return are read from the `ErrorCode` output variable.

Parameter Configuration Diagnostic Codes

This detected error indicates that a public variable parameter contains a value that is not allowed.

To reload new values, a rising edge is required on the `EnableDFB` input:

- `FailCode[0]: 16#0003`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

Modbus Communications Diagnostic Codes

For Modbus communications, this code is used to indicate that communications have not been established. This code can be reset.

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

After Modbus communications have been established, check Modbus client diagnostic codes for `FailCode[0]` and `FailCode[1]`. The components make a distinction between read requests problems and write requests problems:

- `FailCode[2]: 16#0001 Read`
- `FailCode[2]: 16#0002 Write`

Diagnostic Code Example

For a detected error, the code is:

- `FailCode[1]: 16#0001`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

It can be reset.

ACCUSINE PCS Profile

What's in This Chapter

Description	150
DFB Representation	151
Inputs	151
Outputs	153
Inputs/Outputs	157
Public Variables	160
Diagnostic Information Management	160
Communication Needs And/Or Requirements	161

Overview

This chapter describes the DFB of Accusine PCS profile.

Description

General

The *EACCUSINE* DFB is the control block to manage family ACCUSINE PCS harmonic filters based on an Ethernet network.

NOTE: To use the *EACCUSINE* DFB, the Magelis HMI has to be running Active Filter Runtime Application version 5.1.12.1205 or higher.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Control	Run/Stop command.
Monitoring	Allows the device parameters to be monitored.
Remote resetting	Allows resetting of the communication interruption.
Owner	Manages the control system which is the owner (Operator or Program). Therefore, it is responsible for setting the control.

For DFBs communicating by I/O scanning, variables read from the device retain their last value when a communication interruption occurs. For details, refer to the description of the corresponding [output parameter](#), page 68.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

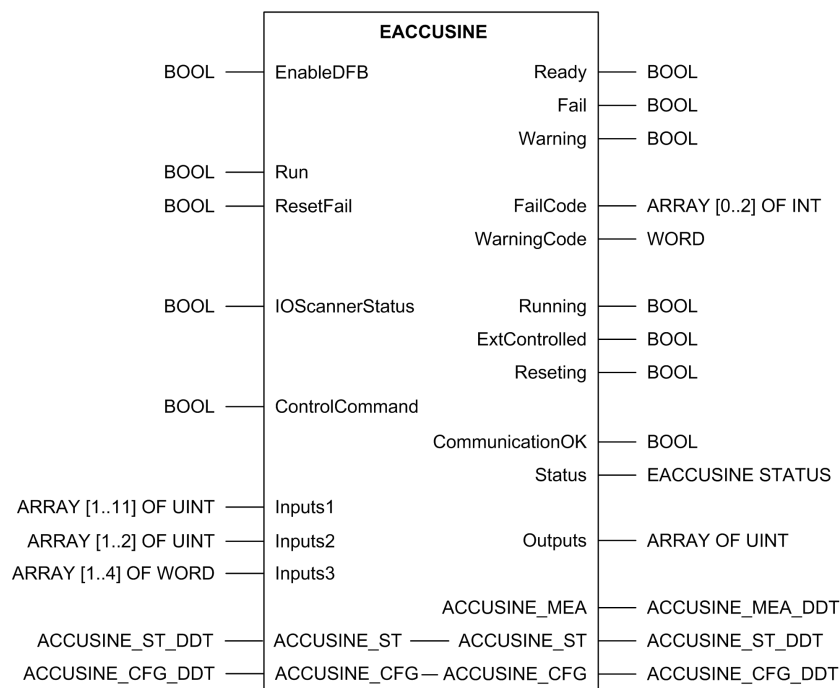
When you configure the *Last value* parameter of the I/O scanner line, take into consideration the behavior of the DFB when a communication interruption occurs.

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

DFB Representation

Representation

The following figure represents `EACCUSINE` DFB in Control Expert:



Inputs

Input Parameter Description

Parameter	Type	Description
<i>EnableDFB</i>	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = When the input is set to 0, the entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = When the input is set to 1, this input enables communications with the devices for their operation. <p>Public variable values are loaded during the first enabling cycle.</p>
<i>Run</i>	BOOL	1 = Starts the active filter.
<i>ResetFail</i>	BOOL	1 = Resets the <code>Fail</code> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <code>ControlCommand</code> is 1.
<i>IOScanner-Status</i>	BOOL	<p>1 = The node is present on the bus.</p> <p>You can find this variable in Ethernet communications.</p>
<i>ControlCommand</i>	BOOL	<p>Indicates to the DFB whether the motor is being controlled locally or from a source external to the DFB.</p> <ul style="list-style-type: none"> 0 = The DFB performs only read operations to monitor the status of the device and does not perform any control functions.

Parameter	Type	Description	
		<ul style="list-style-type: none"> 1 = The DFB performs read operations and performs control operations not conflicting with control commands coming from an external control source. 	
<i>Inputs1</i>	ARRAY [1...11] OF UINT	An array contains the read information obtained from the AccuSine unit through the IOScan. For the block to work properly, allocate the structure (%MWx) .	
	Parameter	Type	Description
	<i>Inputs1[1]</i>	UINT	Total rms load current, A phase.
	<i>Inputs1[2]</i>	UINT	Total rms load current, B phase.
	<i>Inputs1[3]</i>	UINT	Total rms load current, C phase.
	<i>Inputs1[4]</i>	UINT	AccuSine PCS output rms current, A phase.
	<i>Inputs1[5]</i>	UINT	AccuSine PCS output rms current, B phase.
	<i>Inputs1[6]</i>	UINT	AccuSine PCS output rms current, C phase.
	<i>Inputs1[7]</i>	UINT	Harmonic rms load current, A phase.
	<i>Inputs1[8]</i>	UINT	Harmonic rms load current, B phase.
	<i>Inputs1[9]</i>	UINT	Harmonic rms load current, C phase.
	<i>Inputs1[10]</i>	UINT	Reactive rms load current.
	<i>Inputs1[11]</i>	UINT	AccuSine PCS reactive rms output current.
<i>Inputs2</i>	ARRAY [1...2] OF UINT	An array contains the read information obtained from the AccuSine unit through the IOScan. For the block to work properly, allocate the structure (%MWx) .	
	Parameter	Type	Description
	<i>Inputs2[1]</i>	UINT	1: Running. 0: Stopped.
	<i>Inputs2[2]</i>	UINT	Enable remote status.
<i>Inputs3</i>	ARRAY [1...4] OF WORD	An array contains the read information obtained from the AccuSine unit through the IOScan. For the block to work properly, allocate the structure (%MWx) .	
	Parameter	Type	Description
	<i>Inputs3[1]</i>	WORD	Word containing information about the events with top priority.
	<i>Inputs3[2]</i>	WORD	Word 1 containing information about the events with lower priority.
	<i>Inputs3[3]</i>	WORD	Word 2 containing information about the events with lower priority.
	<i>Inputs3[4]</i>	WORD	Alarm status.

Outputs

Output Parameter Description

Parameter	Type	Description	
<i>Ready</i>	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out any operation.	
<i>Fail</i>	BOOL	<p>1 = A detected failure in the control block or in the device or communication interruption. To reset the <code>Fail</code> output pin, the <code>ResetFail</code> input has to be activated. The last detected error code is shown on FailCode.</p> <p>NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.</p>	
<i>Warning</i>	BOOL	<p>1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset.</p> <p>The signal remains active until the cause for the alarm disappears.</p>	
<i>FailCode</i>	ARRAY [0..2] OF INT	<p>When <code>Fail</code> output is 1, it holds the code for the detected error.</p> <p>If <code>Fail</code> output is 0, it indicates the last detected error that occurred.</p> <p>The detected error source is specified by a 3 level structure. Refer to the Diagnostic Information Management, page 160 section.</p>	
<i>WarningCode</i>	WORD	The <code>WarningCode</code> output holds a data structure with information on the alarm currently on the AccuSine unit.	
	Bit	Description	
	0	Maximum capacity alarm detected.	
	1	Thermal limit override.	
	2	Low-order harmonic alarm detected.	
<i>Running</i>	BOOL	1 = Indicates that the active filter is running.	
<i>ExtControlled</i>	BOOL	1 = The device is being controlled from an external source (for example, from the monitoring system) to the system.	
<i>Reseting</i>	BOOL	<p>1 = A reset is being carried out.</p> <p>The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected failure.</p> <p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code>.</p> <p>If the detected failure is reset, the <code>Fail</code> and <code>Reseting</code> output variables are reset (set to FALSE). If the detected failure is not reset, the <code>Reseting</code> variable sets to FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Refer to the timing diagram below.</p>	
<i>CommunicationOK</i>	BOOL	1 = Indicates that the node is present on the bus. You can find this variable in Ethernet communications.	
<i>Status</i>	EACCUSINEStatus	<p>The structure holds the data containing the information that the block extracts from the status word variable of the device.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	<i>Status</i>	EACCU-SINE_DeviceStatus	Provides the general status of the device. Access to the data held in this bit word is read-only.
	<i>Alarm</i>	EACCU-SINE_Alarm	Provides the information on the device status.
	<i>CriticalFail</i>	EACCU-SINE_CriticalFail	Provides the notification information of the device. Access to the data held in this bit word is read-only.

Parameter	Type	Description	
	<i>NonCriticalFail</i>	EACCU-SINE_NonCritical-Fail	Provides the notification information of the device. Access to the data held in this bit word is read-only.
	<i>State</i>	INT	Numerical code of device status.
	<i>Info</i>	INT	Code with information on statuses and required actions.
<i>Outputs</i>	ARRAY OF UINT	Holds an array structure with data sent to the device. The control exerted on the device can be influenced with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. For the control block to work properly, allocate the structure (%MWx) . Refer to the Communications Technologies, page 247.	
	Parameter	Type	Description
	<i>Outputs[1]</i>	UINT	Command: <ul style="list-style-type: none"> 0 to 1: Run 1 to 0: Stop
<i>ACCUSINE_MEA</i>	ACCUSINE_MEA_DDT	Data structure with device measurement information. The following table describes the status information:	
	Parameter	Type	Description
	<i>TOTAL_LOAD_A</i>	UINT	Total rms load current, A phase.
	<i>TOTAL_LOAD_B</i>	UINT	Total rms load current, B phase.
	<i>TOTAL_LOAD_C</i>	UINT	Total rms load current, C phase.
	<i>OUTPUT_A</i>	UINT	AccuSine PCS output rms current, A phase.
	<i>OUTPUT_B</i>	UINT	AccuSine PCS output rms current, B phase.
	<i>OUTPUT_C</i>	UINT	AccuSine PCS output rms current, C phase.
	<i>HARMONIC_A</i>	UINT	Harmonic rms load current, A phase.
	<i>HARMONIC_B</i>	UINT	Harmonic rms load current, B phase.
	<i>HARMONIC_C</i>	UINT	Harmonic rms load current, C phase.
	<i>TOTAL_LOAD_PF</i>	UINT	Reactive rms load current.
	<i>OUTPUT_PF</i>	UINT	AccuSine PCS reactive rms output current.

EACCUSINE_DeviceStatus

The following table describes the EACCUSINE_DeviceStatus variable:

Parameter	Type	Description
<i>Running</i>	BOOL	1 = The device is running.
<i>EnableRemote</i>	BOOL	1 = A remote status is enabled.
<i>Warning</i>	BOOL	1 = An ACCUSINE notification.
<i>Fail</i>	BOOL	1 = An ACCUSINE detected failure.

EACCUSINE_Alarm

The following table describes the EACCUSINE_Alarm variable:

Parameter	Type	Description
<i>MaxCapacity</i>	BOOL	1 = Maximum capacity alarm detected.
<i>ThermalLimit</i>	BOOL	1 = Thermal limit override.
<i>LowOrderHarmonic</i>	BOOL	1 = Low-order harmonic alarm detected.

EACCUSINE_CriticalFail

The following table describes the EACCUSINE_CriticalFail variable:

Parameter	Type	Description
<i>Fail_5V_OV</i>	BOOL	1 = A 5 V OV failure detected.
<i>Fail_5V_UV</i>	BOOL	1 = A 5 V UV failure detected.
<i>Fail_P15V_UV</i>	BOOL	1 = A +15 V UV failure detected.
<i>Fail_L15V_UV</i>	BOOL	1 = A -15 V UV failure detected.
<i>UserSetup</i>	BOOL	1 = User setup failure detected.
<i>TimeOut</i>	BOOL	1 = Timeout failure detected.
<i>AutoDetect</i>	BOOL	1 = Auto detect failure detected.
<i>HMIlost</i>	BOOL	1 = HMI lost failure detected.
<i>InvalidRating</i>	BOOL	1 = Invalid rating.
<i>PhaseRotation</i>	BOOL	1 = Phase rotation failure detected.

EACCUSINE_NonCriticalFail

The following table describes the EACCUSINE_NonCriticalFail variable:

Parameter	Type	Description
<i>ACLineLoss</i>	BOOL	1 = Detected failure in AC line.
<i>ACPhaseLoss</i>	BOOL	1 = Detected failure in AC phase.
<i>ACImbalance</i>	BOOL	1 = Detected failure in AC imbalance.
<i>ACOverVolt</i>	BOOL	1 = AC overvoltage.
<i>ACFreq</i>	BOOL	1 = Incorrect AC frequency.
<i>Sensor</i>	BOOL	1 = Sensor failure detected.
<i>Frequency</i>	BOOL	1 = Frequency failure detected.
<i>IGBT A</i>	BOOL	1 = IGBT A failure detected.
<i>IGBT B</i>	BOOL	1 = IGBT B failure detected.
<i>IGBT C</i>	BOOL	1 = IGBT C failure detected.
<i>IGBT1OverT</i>	BOOL	1 = IGBT1 excess temp failure detected.
<i>IGBT2OverT</i>	BOOL	1 = IGBT2 excess temp failure detected.
<i>IGBT3OverT</i>	BOOL	1 = IGBT3 excess temp failure detected.
<i>PhaseOverT</i>	BOOL	1 = 3 Ph excess temp failure detected.
<i>Filter</i>	BOOL	1 = Filter failure detected.
<i>EnclosureOverT</i>	BOOL	1 = Enclosure excess temp failure detected.
<i>DCBusOverV</i>	BOOL	1 = DC bus over voltage failure detected.
<i>DCBusLowV</i>	BOOL	1 = DC bus low voltage failure detected.

Parameter	Type	Description
<i>EnclosureUnderT</i>	BOOL	1 = Enclosure temp lower than minimum temp failure detected.
<i>F110COverT</i>	BOOL	1 = 110C excess temp failure detected.
<i>BoardOverT</i>	BOOL	1 = Circuit board excess temp failure detected.

State

The following table describes the *State* variable:

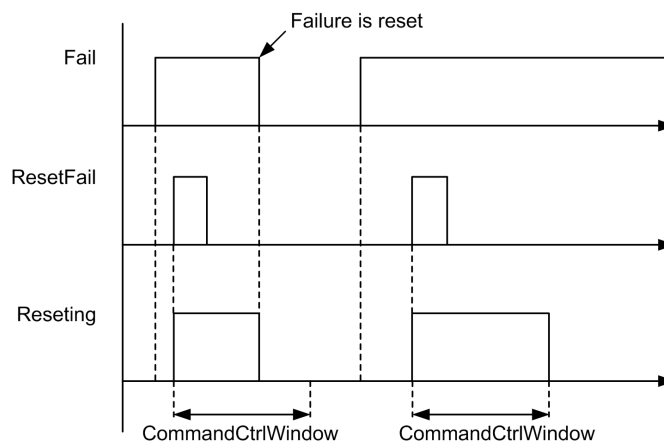
Variable value	Description
-2	Device has detected error.
-1	Not initialized. Waiting for data.
0	Disabled.
5	Ready.
8	Device failure detected.

Info

The following table describes the *Info* variable:

Variable value	Description
1	Incorrect configuration of DFB parameter.
8	Accusine running.
11	Missing <code>EnableDFB</code> . Initial state.
12	Missing <code>CommunicationOK</code> . Communication interruption.
24	Remove <code>ResetFail</code> . Reset again.
25	Accusine stopped.
81	Missing <code>ResetFail</code> . Device error detected.
82	A <code>Reset</code> is needed.

Timing diagram:



Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
<i>ACCUSINE_ST</i>	ACCUSINE_ST_DDT	Device data structure holds the minimum information required for performing control and monitoring. The information is used by the operator screen and can be read from or written to HMI/SCADA system.
<i>ACCUSINE_CFG</i>	ACCUSINE_CFG_DDT	Data structure with device information. Information used by the operator screen is readable from HMI/SCADA system.

ACCUSINE_ST_DDT Type

Name	Type	Description
<i>STW</i>	WORD	Provides the device status. Access to the data held in this bit word is read-only.
<i>CFGW</i>	WORD	Enables controlling the device from the monitoring subsystem or from the operator screen if <i>Owner</i> (1), or only from the monitoring subsystem if <i>Owner</i> (0). If <i>Owner</i> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.

ACCUSINE_CFG_DDT Type

Name	Type	Description
<i>DataStatus</i>	WORD	Provides the status information of the device for displaying in the background on the monitoring subsystem.
<i>FailStatus</i>	WORD	Provides information about the detected failure found in the device, for displaying in the background, on the monitoring subsystem.
<i>NonCriticalFailStatus</i>	WORD	Provides the status information of the device for displaying in the background on the monitoring subsystem.
<i>WarningCode</i>	WORD	Provides <i>EACCUSINE</i> alarm code information. Takes values from the <i>WarningCode</i> output (W3).
<i>FailCode0</i>	INT	Indicates which detected error has occurred, <i>FailCode</i> [0].
<i>FailCode1</i>	INT	Indicates which detected error has occurred, <i>FailCode</i> [1].
<i>FailCode2</i>	INT	Indicates which detected error has occurred, <i>FailCode</i> [2].

ACCUSINE_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
2	Module is running.
3	Inoperable device.
4	Alarm on the device or repetitive detected fault alarm requires resetting.

Bit	Description
5	Communication interruptions.
6	Requires resetting. ResetFail is required.
7	Refer to the ExtControlled output pin, page 153.
8	Refer to the Resetting output pin, page 153.
9	Refer to the EnabledDFB input pin, page 151.

ACCUSINE_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 151.
1	Owner.
6	Refer to the Run input pin, page 151.
7	Refer to the ControlCommand input pin, page 151.

ACCUSINE_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the Running status in the Status output pin, page 154.
1	Refer to the EnableRemote status in the Status output pin, page 154.
2	Refer to the Warning status in the Status output pin, page 154 .
3	Refer to the Fail status in the Status output pin, page 154.
4	Refer to the MaxCapacity status in the Status output pin, page 154.
5	Refer to the ThermalLimit status in the Status output pin, page 154.
6	Refer to the LowOrderHarmonic status in the Status output pin, page 154.

ACCUSINE_CFG.FailStatus Word Structure

Bits	Description
0	Refer to the Fail_5V_OV (CR) in the CriticalFail output pin, page 155.
1	Refer to the Fail_5V_UV (CR) in the CriticalFail output pin, page 155.
2	Refer to the Fail_P15V_UV in the CriticalFail output pin, page 155.
3	Refer to the Fail_L15V_UV in the CriticalFail output pin, page 155.
4	Refer to the UserSetup in the CriticalFail output pin, page 155.
5	Refer to the TimeOut in the CriticalFail output pin, page 155.
6	Refer to the AutoDetect in the CriticalFail output pin, page 155.
7	Refer to the HMI Lost in the CriticalFail output pin, page 155.
8	Refer to the InvalidRating in the CriticalFail output pin, page 155.
9	Refer to the PhaseRotation in the CriticalFail output pin, page 155.
10	–
11	Refer to the DCBusOverV in the CriticalFail output pin, page 155.

Bits	Description
12	Refer to the <code>DCBusLowV</code> in the <i>CriticalFail</i> output pin, page 155.
13	Refer to the <code>EnclosureUnderT</code> in the <i>CriticalFail</i> output pin, page 155.
14	Refer to the <code>F110CoverT</code> in the <i>CriticalFail</i> output pin, page 155.
15	Refer to the <code>BoardOverT</code> in the <i>CriticalFail</i> output pin, page 155.

ACCUSINE_CFG.NonCriticalFailStatus Word Structure

Bit	Description
0	Refer to the <code>ACLineLoss</code> in the <i>NonCriticalFail</i> output pin, page 155.
1	Refer to the <code>ACPhaseLoss</code> in the <i>NonCriticalFail</i> output pin, page 155.
2	Refer to the <code>ACImbalance</code> in the <i>NonCriticalFail</i> output pin, page 155.
3	Refer to the <code>ACOverVolt</code> in the <i>NonCriticalFail</i> output pin, page 155.
4	Refer to the <code>ACFreq</code> in the <i>NonCriticalFail</i> output pin, page 155.
5	Refer to the <code>Sensor</code> in the <i>NonCriticalFail</i> output pin, page 155.
6	Refer to the <code>Frequency</code> in the <i>NonCriticalFail</i> output pin, page 155.
7	Refer to the <code>IGBTA</code> in the <i>NonCriticalFail</i> output pin, page 155.
8	Refer to the <code>IGBTB</code> in the <i>NonCriticalFail</i> output pin, page 155.
9	Refer to the <code>IGBTC</code> in the <i>NonCriticalFail</i> output pin, page 155.
10	Refer to the <code>IGBT1OverT</code> in the <i>NonCriticalFail</i> output pin, page 155.
11	Refer to the <code>IGBT2OverT</code> in the <i>NonCriticalFail</i> output pin, page 155.
12	Refer to the <code>IGBT3OverT</code> in the <i>NonCriticalFail</i> output pin, page 155.
13	Refer to the <code>PhaseOverT</code> in the <i>NonCriticalFail</i> output pin, page 155.
14	Refer to the <code>Filter</code> in the <i>NonCriticalFail</i> output pin, page 155.
15	Refer to the <code>EnclosureOverT</code> in the <i>NonCriticalFail</i> output pin, page 155.

ACCUSINE_CFG.WarningCode Word Structure

Bit	Description
0	Refer to the <code>MaxCapacity</code> in the <i>WarningCode</i> output pin, page 154.
1	Refer to the <code>ThermalLimit</code> in the <i>WarningCode</i> output pin, page 154.
2	Refer to the <code>LowOrderHarmonic</code> in the <i>WarningCode</i> output pin, page 154.

Public Variables

Public Variable Description

Variable	Type	Description
<i>CommandCtrlWindow</i>	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. The PM only allows a communications <i>ResetFail</i> . If the detected failure, continues after the <i>CommandCtrlWindow</i> time has elapsed, <i>Fail</i> remains set to TRUE and the <i>Reseting</i> output resets (set to FALSE).
<i>ScanTime</i>	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
<i>ResetMode</i>	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in <i>CommandCtrlWindow</i> is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in <i>CommandCtrlWindow</i> elapses. The second reset is carried out after <i>CommandCtrlWindow</i> * 2 elapses, so on. If the value of <i>CommandCtrlWindow</i> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable Value	Description
	FALSE	Communications are/the device is reset with the <i>ResetFail</i> variable.
	TRUE	Communications are/the device is reset automatically.
<i>MaxResetTime</i>	TIME	When in automatic <i>ResetMode</i> , this variable sets the maximum time between 2 consecutive resets. Refer to the timing diagram below.

Diagnostic Information Management

Overview

The diagnostic codes, the device can return are read from the *ErrorCode* output variable.

Parameter Configuration Diagnostic Codes

This detected error indicates that a public variable parameter contains a value that is not allowed.

To reload new values, a rising edge is required on the *EnableDFB* input:

- *FailCode*[0]: 16#0003
- *FailCode*[1]: 16#0000
- *FailCode*[2]: 16#0004

Modbus Communications Diagnostic Codes

For Modbus communications, this code is used to indicate that communications have not been established. This code can be reset.

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

Diagnostic Code Example

For a detected error, the code will be:

- `FailCode[1]: 16#0001`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

Can be reset if `ControlCommand=TRUE`.

Communication Needs And/Or Requirements

IOSCAN Communications

The `EACCUSINE` component needs 3 IOScan lines:

- First IOScan line:
This IOScan line needs to be configured as follows:
 - `UnitID: 255`
 - `Slave Syntax: Index`
 - `RD Ref Slave: 1`
 - `RD Length: 11`
 - `Last value (Input): Set to 0`
 - `WR Ref Slave: 0`
 - `WR Length: 0`
- Second IOScan line:
This IOScan line needs to be configured as follows:
 - `UnitID: 255`
 - `Slave Syntax: Index`
 - `RD Ref Slave: 700`
 - `RD Length: 2`
 - `Last value (Input): Set to 0`
 - `WR Ref Slave: 0`
 - `WR Length: 0`

- Third IOScan line:

This IOScan line needs to be configured as follows:

- UnitID: 255
- Slave Syntax: Index
- RD Ref Slave: 600
- RD Length: 4
- Last value (Input): Set to 0
- WR Ref Slave: 1001
- WR Length: 1

The following figure shows the IO scanner line:

IP address	Device Name	Unit ID	Slave Syntax	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	RD Ref Slave	RD length	Last value (Input)	WR Master Object	WR Ref Slave	WR length	Description
192.168.2.19		255	Index	1500	60	%MW101	1	11	Hold last	%MW100	0	0	Accusine Device
192.168.2.19		255	Index	1500	60	%MW112	700	2	Hold last	%MW100	0	0	Accusine Device
192.168.2.19		255	Index	1500	60	%MW114	600	4	Hold last	%MW100	1001	1	Accusine Device

The **EACCUSINE** section needs the input/output values (RD/WR Master Object) and heart bit of the IOScan line.

- RD Master Object:

The **EACCUSINE** DFB needs the *Inputs1*, *Inputs2*, and *Inputs3*. In the diagram given above, *Inputs1* (11 words) is mapped to %MW101, *Inputs2* (2 words) is mapped to %MW112, and *Inputs3* (4 words) is mapped to %MW114 but the values needs to be filled from 3 different variable sections of **EACCUSINE** memory.

- WR Master Object:

The **EACCUSINE** DFB needs the outputs. In the diagram given above, outputs is mapped to %MW100. This value is only used in the third IOScan line.

Progressive Starters


What's in This Part

ATS Progressive Starter Profile 164

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Progressive Starters family.

These function blocks do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it 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.

ATS Progressive Starter Profile

What's in This Chapter

Description	164
DFB Representation	165
Inputs	166
Outputs	169
Inputs/Outputs	173
Diagnostic Information Management.....	177

Overview

This chapter describes the DFBs of ATS Progressive Starter profile.

Description

General

The ATS progressive starter profile is used to manage ATS soft starters on a Modbus network.

Function Description

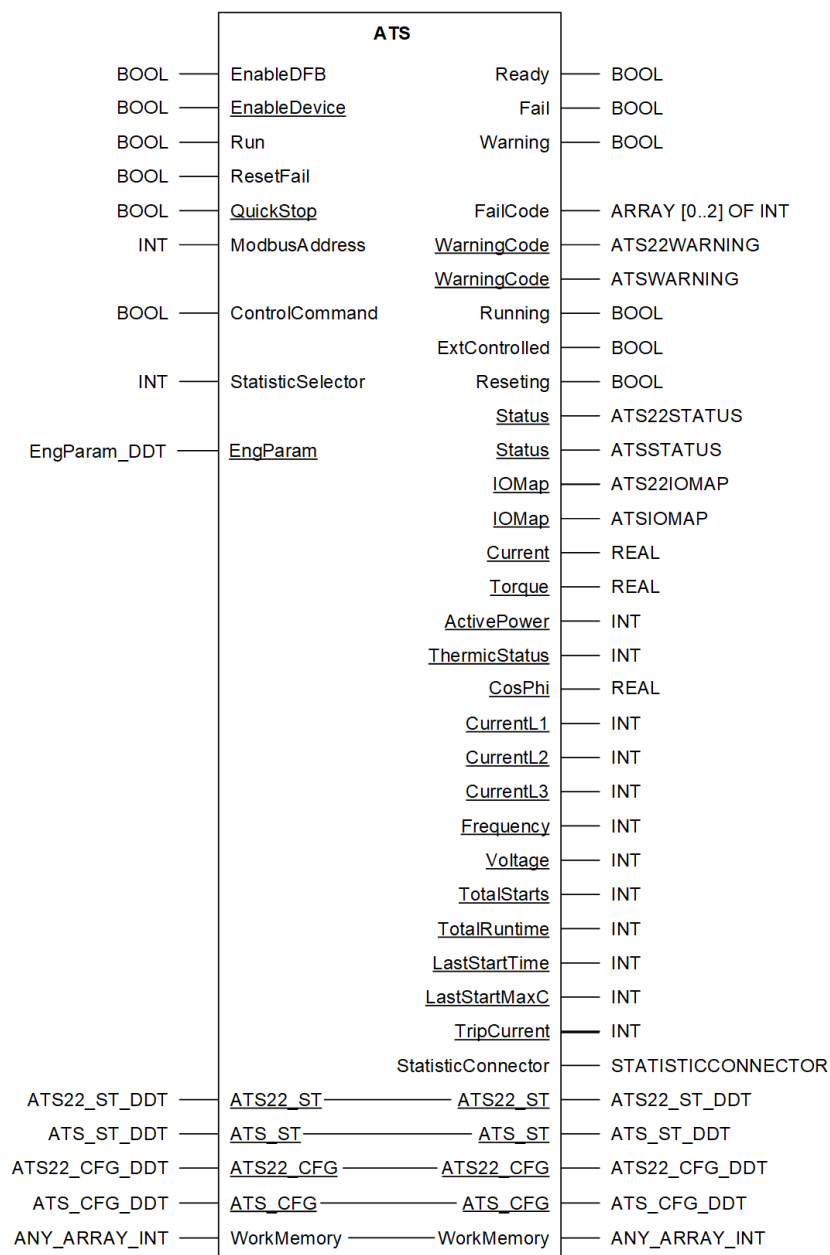
The main functions of the DFB are described in the following table:

Function	Description
Control	Start/Stop command.
Inoperable device	Monitors the inoperable device.
Remote resetting	Allows resetting of the device.
Control	Enables you to monitor the device. You can control from the controller or with the wired inputs/outputs of the speed driver.

DFB Representation

Representation

The following figure represents the functional module of ATS profile:



NOTE: The device can only work on serial Modbus mode.

NOTE: The underlined parameters are specific for some components.

The table shows the parameters available for specific components:

Parameters		Components	
		Modbus	
		MBATS22	MBATS48
Inputs	EnableDevice	—	X
	QuickStop	—	X
Outputs	WarningCode (ATS22WARNING)	X	—
	WarningCode (ATSWARNING)	—	X
	Status (ATS22STATUS)	X	—

Parameters		Components	
		Modbus	
		MBATS22	MBATS48
	Status (ATSSTATUS)	–	X
	IOMAP (ATS22IOMAP)	X	–
	IOMAP (ATSIOMAP)	–	X
	Current	–	X
	CurrentL1	X	–
	CurrentL2	X	–
	CurrentL3	X	–
	Torque	–	X
	ActivePower	–	X
	ThermicStatus	–	X
	CosPhi	–	X
	Frequency	X	–
	Voltage	X	–
	TotalStarts	X	–
	TotalRunTime	X	–
	LastStartTime	X	–
	LastStartMaxC	X	–
	TripCurrent	X	–
Inputs/Outputs	ATS22_ST	X	–
	ATS_ST	–	X
	ATS22_CFG	–	X
	ATS_CFG	–	X
X: Parameter is available.			
–: Parameter is not available.			

Inputs

Input Parameter Description

Name	Type	Description
EnableDFB	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. <p>Public variable values are loaded during the first enabling cycle.</p>
EnableDevice*	BOOL	<p>1 = Valid if the EnableDFB variable is active.</p> <p>The starter has to be enabled in order to be controlled.</p>
* These parameters are available only with specific components.		

If the device is reset and *Run* pin is active, then the device will auto start. If manual start of the device is required, then reset the *Run* pin followed by device.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Reset the `Run` variable before resuming operation.

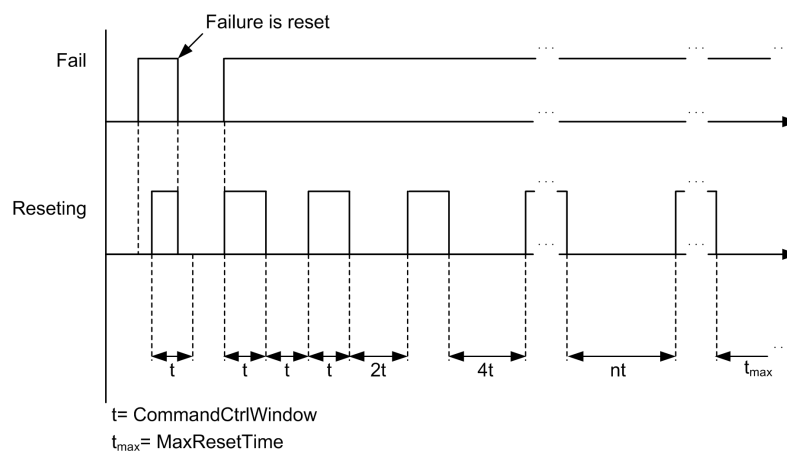
Failure to follow these instructions can result in equipment damage.

Name	Data type	Description
<code>Run</code>	BOOL	<p>1 = Starts the starter.</p> <p>If the device is being enabled or reset, reset the inputs so that unexpected starts do not occur. In these cases, first reset the input to resume the operation.</p>
<code>ResetFail</code>	BOOL	<p>1 = Resets the <code>Fail</code> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <code>ControlCommand</code> is 1.</p> <p>You can carry out communication resets when required. To carry out an automatic reset, use the <code>ResetMode</code> public variable.</p>
<code>QuickStop*</code>	BOOL	<p>1 = Stops the starter quickly with fast stop ramp. It is state-based.</p> <p>If there is a <code>QuickStop</code>, the <code>Run</code> bit has to be reset to resume operation.</p>
<code>ModbusAddress</code>	INT	Device address within the Modbus network.
<code>ControlCommand</code>	BOOL	<p>Indicates to the DFB whether the motor is being controlled locally or from a source external to the DFB.</p> <ul style="list-style-type: none"> 0 = Performs only read operations to monitor the status of the device and does not perform any control functions. 1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. <p>NOTE: This input does not configure the starter.</p>
<code>StatisticSelector</code>	INT	<p>Variable is used to obtain statistics for the Modbus network. This data provides information for using the <code>StatisticConnector</code> pin with the <code>StatisticCounter</code> DFB in General Purpose library for communication.</p> <p>The following table displays the <code>StatisticSelector</code> value:</p>
	Variable value	Description
	1	Read statistics, client.
	2	Write statistics, client.
<code>EngParam</code>	<p><code>EngParam_DDT</code>, page 168</p> <p><code>EngParamATS22/</code> <code>EngParamATS48</code></p>	Engineering parameters.
* These parameters are available only with specific components.		

EngParam_DDT

Name	Data type	Description
Refresh	TIME	Refresh time for device data on Modbus communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed.
CommandCtrlWindow	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is sent. The command that is controlled is Run. In the event of a ResetFail, this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the Resetting output.
ScanTime	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device. The time defined in CommandCtrlWindow is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in CommandCtrlWindow elapses. The second reset is carried out after CommandCtrlWindow * 2 elapses, so on. If the value of CommandCtrlWindow is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable Value	Description
	FALSE	Communications are/the device is reset with the ResetFail variable.
	TRUE	Communications are/the device is reset automatically.
MaxResetTime	TIME	When in automatic ResetMode, this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.

Timing diagram:



Outputs

Output Parameter Description

Output	Type	Description	
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out or carrying out any Run or Stop command.	
Fail	BOOL	<p>1 = A detected failure in the control block or in the device or a communication interruption. To reset the <code>Fail</code> output pin, the <code>ResetFail</code> input has to be activated. The last detected error code is shown on <code>FailCode</code>.</p> <p>NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.</p>	
Warning	BOOL	<p>1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset.</p> <p>This signal remains active until the cause of the alarm disappears.</p>	
FailCode	ARRAY [0..2] OF INT	<p>When <code>Fail</code> output is 1, it holds the code for the detected error.</p> <p>If <code>Fail</code> output is 0, it indicates the last detected error that occurred.</p> <p>The detected error source is specified by a 3-level structure. Refer to the <i>Diagnostic Information Management</i>, page 177 section.</p>	
WarningCode*	ATS22WARNING	The <code>WarningCode</code> output is a data structure with information on the alarm currently on the starter.	
	Name	Type	Description
	Order	BOOL	1 = An alarm is present on the device.
	ForcedLocalMode	BOOL	1 = A follow-up alarm. The device is not responding to the control command (<code>Run</code> , <code>QuickStop</code> , so on) within the time specified in <code>CommandCtrlWindow</code> .
	Device	BOOL	1 = The device is forced locally. It is controlled through the screw terminals.
WarningCode*	ATSWARNING	Holds a data structure with information on the alarm currently on the starter.	
	Name	Type	Description
	Order	BOOL	1 = A follow-up alarm. The device is not responding to the control command (<code>Run</code> , <code>QuickStop</code> , so on) within the time specified in <code>CommandCtrlWindow</code> .
	ForcedLocalMode	BOOL	1 = The device is forced locally. It is controlled through the screw terminal.
	Device	BOOL	1 = An alarm on the device (w 458.7).
	Current	BOOL	1 = A detected current limit alarm (w 459.11).
	Torque	BOOL	1 = A detected torque limit alarm (w 459.12).
	Thermal	BOOL	1 = A detected thermal limit alarm (w 459.7).
Running	BOOL	1 = The starter is running.	
ExtControlled	BOOL	<p>1 = The device is being controlled from an external source (for example, from the console, from a push-button panel, or from the monitoring system) to the system.</p> <p>Provides information for programming.</p> <p>NOTE: The <code>ControlCommand</code> signal, the <code>Owner</code> variable, and the <code>ForcedLocalMode</code> status are used to activate this signal. You cannot use this signal as a <code>ControlCommand</code> input.</p>	
Reseting	BOOL	<p>1 = A reset is being carried out.</p> <p>The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected failure.</p> <p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code>.</p>	

Output	Type	Description	
		<p>If the detected failure is reset, the <code>Fail</code> and <code>Resetting</code> output variables are reset (set to FALSE). On the other hand, if the detected failure is not reset, the <code>Resetting</code> variable is set to FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Refer to the Timing diagram below.</p>	
Status*	ATS22STATUS	<p>The structure holds data containing the information that the block extracts from the ETA variable (w256) of the starter.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	Ready	BOOL	1 = The device is ready to start or stop (w256.0).
	Running	BOOL	<ul style="list-style-type: none"> 0 = The starter is not running (w256.1). 1 = The starter is running (w256.1).
	Trip	BOOL	1 = A trip condition has occurred (w256.2).
	Alarm	BOOL	1 = An alarm is present on device (w256.3).
	ForcedLocalMode	BOOL	1 = The device is controlled by physical inputs (w256.14).
	Ramping	BOOL	1 = The starter is accelerating or slowing down (w256.15).
	Info	INT	<p>Numerical code with the information on statuses and required actions.</p> <p>Refer to the <code>Info</code> table below.</p>
Status*	ATSSTATUS	<p>The structure holds data containing the information that the block extracts from the status variable of the starter.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	ReadyToSwitchOn	BOOL	1 = The DSP402 device is ready to switch on (w458.0).
	SwitchedOn	BOOL	1 = The DSP402 device is switched on (w458.1).
	OperationEnabled	BOOL	1 = DSP402 device operation is enabled (w458.2).
	Malfunction	BOOL	1 = A detected failure on the device (w458.3).
	VoltageEnabled	BOOL	1 = Voltage at the device terminals (Not w458.4).
	QuickStop	BOOL	1 = Quick stop is activated (Not w458.5).
	SwitchOnDisabled	BOOL	1 = DSP402 device switch-on is disabled (w458.6).
	Alarm	BOOL	1 = An alarm is present on the device (w458.7).
	ForcedLocalMode	BOOL	1 = Device controlled by physical inputs (w458.9).
	LocalMode	BOOL	1 = Commands accepted through terminals for writing and reading. Drivecom disabled (w459.13 and w459.14).
	ResetAuthorization	BOOL	1 = Detected fault reset authorized (w459.0).
	Running	BOOL	1 = Motor running (w459.4).
	Accelerating	BOOL	<p>Start accelerating</p> <p>1 = Active (w459.9).</p>
	Decelerating	BOOL	<p>Start decelerating</p> <p>1 = Active (w459.10).</p>
	State	INT	<p>Numerical code corresponding to the state of the starter.</p> <p>Refer to the <code>State</code> table below.</p>
	Info	INT	Numerical code with the information on statuses and required actions.

Output	Type	Description	
			Refer to the Info table below.
IOMap*	ATS22IOMAP	Holds a data structure that contains information on the state of the starter inputs and outputs (w262 and w263). The following table describes the IOMap:	
	Parameter	Type	Description
	LI1	BOOL	1 = The state of the digital input LI1.
	LI2	BOOL	1 = The state of the digital input LI2.
	LI3	BOOL	1 = The state of the digital input LI3.
	R1	BOOL	1 = The state of the digital output R1.
	R2	BOOL	1 = The state of the digital output R2.
IOMap*	ATSIOMAP	Holds a data structure that contains information on the state of the starter inputs and outputs. The following table describes the IOMap:	
	Parameter	Type	Description
	LIRUN	BOOL	1 = The state of the digital input LIRUN.
	LISTOP	BOOL	1 = The state of the digital input LISTOP.
	LI3	BOOL	1 = The state of the digital input LI3.
	LI4	BOOL	1 = The state of the digital input LI4.
	LO1	BOOL	1 = The state of the digital output LO1.
	LO2	BOOL	1 = The state of the digital output LO2.
	R1	BOOL	1 = The state of the digital output R1.
	R2	BOOL	1 = The state of the digital output R2.
	R3	BOOL	1 = The state of the digital output R3.
	AO	INT	Range (0-10000). Shows the state of the analog output.
Current L1*	INT	Present current on starter line 1 in amperes (w 257).	
Current L2*	INT	Present current on starter line 2 in amperes (w 258).	
Current L3*	INT	Present current on starter line 3 in amperes (w 259).	
Current*	REAL	Motor current (A) (w 4062).	
Torque*	INT	Current motor torque in % (w 4063).	
ActivePower*	INT	Power currently consumed by the motor (w 4072).	
ThermicStatus*	INT	Motor thermic status in % (w 4064).	
CosPhi*	REAL	Current value of phi cosine (w4067).	
Frequency*	INT	Voltage frequency in Hertz (w 279).	
Voltage*	INT	Input voltage in volts (w 260).	
TotalStarts*	INT	Number of times that the ATS 22 has started (w 274).	
TotalRuntime*	INT	Total time (in hours) during which the ATS 22 has been operating (w 273).	
LastStartTime*	INT	Last starting time (w 275).	
LastStartMaxC*	INT	Maximum detected current during last start (w 276).	
TripCurrent*	INT	Current present when last tripping event occurred (w 280).	
StatisticConnector	STATISTICCONNECTOR	Information data is used with Modbus communications to obtain statistics on the Modbus network (requests carried out, time between requests, so on).	

Output	Type	Description	
		This structure has been created to be used within <code>StatisticCounter</code> DFB in General Purpose library for communication.	
	Parameter	Type	Description
	Start	BOOL	1 = The operation has started.
	EndOk	BOOL	1 = The operation has ended correctly.
	EndNOk	BOOL	1 = The operation has ended with a detected error.
	PartialTime	DINT	Partial time.
* These parameters are available only with specific components.			

Info

The following table describes the `Info` variable of *MBATS22*:

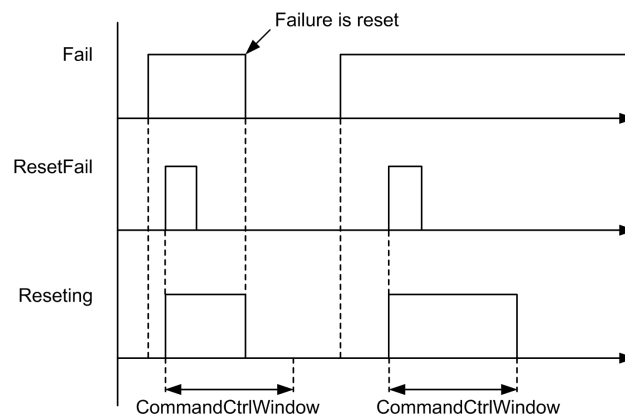
Variable value	Description
2	Waiting for Ready.
11	Missing <code>EnableDFB</code> .
12	Communication interruption.
20	Remove local forcing needs to be 0.
21	Waiting <code>Running</code> .
22	<code>Running</code> .
23	Waiting stopped.
24	Stopped.
25	Remove <code>ResetFail</code> needs to be 0.
30	In forced local mode.
31	Waiting <code>ForcedLocalMode</code> .
40	Remove <code>Run</code> needs to be 0.
81	Missing <code>ResetFail</code> . Inoperable starter.
82	Do starter reset.

The following table describes the `Info` variable of *MBATS48*:

Variable value	Description
2	Waiting <code>SwitchOnDisabled</code> .
3	Waiting <code>ReadyToSwitchOn</code> .
4	Waiting <code>SwitchOn</code> .
5	Waiting <code>OperationEnabled</code> .
6	Waiting <code>QuickStop</code> active.
11	Missing <code>EnableDFB</code> .
12	Communication Interruption.
13	ETA value is 0.
14	Remove local forcing needs to be 0.
21	Missing <code>EnableDevice</code> .
23	Remove <code>Run</code> needs to be 0.
24	Remove <code>ResetFail</code> . Reset again.

Variable value	Description
51	Starter stopped.
53	Run.
56	Waiting starter running
57	Waiting starter stop.
61	Remove QuickStop needs to be 0.
62	QuickStop is activated.
81	Missing ResetFail. Inoperable starter.
82	Do starter reset.
99	Unexpected state.

Timing diagram:



Inputs/Outputs

Input/Output Parameter Description

Name	Type	Description
ATS22_ST*	ATS22_ST_DDT, page 174	Device data structure holds the minimum information required for performing control and monitoring. The information is used by the operator screen and can be read from or written to HMI/SCADA system.
ATS_ST*	ATS_ST_DDT, page 174	Device data structure holds the minimum information required for performing control and monitoring functions. Information used by the operator screen is usable from the HMI or SCADA.
ATS22_CFG*	ATS22_CFG_DDT, page 175	Data structure with device information. Information used by the operator screen is readable from HMI/SCADA system.
ATS_CFG*	ATS_CFG_DDT, page 174	Device data structure with device information. Information used by the operator screen is readable from the HMI or SCADA system.
WorkMemory	ANY_ARRAY_INT	The array is used in Modbus communications. This variable is meant for use with a Modbus port that serializes Modbus requests in an optimum manner.
* These parameters are available only with specific components.		

ATS22_ST_DDT Type

Name	Type	Description
STW, page 175	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW, page 176	WORD	Device control. Enables to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
CurrentL1	REAL	Current consumed by the starter through line 1.
CurrentL2	REAL	Current consumed by the starter through line 2.
CurrentL3	REAL	Current consumed by the starter through line 3.

ATS_ST_DDT Type

Name	Type	Description
STW, page 175	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW, page 176	WORD	Device control. Provides the means to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0). If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
Current	REAL	Active current (A).

ATS_CFG_DDT Type

Name	Type	Description
DataStatus, page 176	WORD	Information on the device status. Provides the information on the <code>Status</code> output structure.
IOMap	WORD	Provides information on the state of the digital inputs/outputs of the device. Information on the <code>IOMap</code> output structure.
Info	INT	Starter information. Its value is <code>Info</code> status.
State	INT	Starter status code information. Its value is <code>State</code> status.
Torque	INT	Active torque in %.
ThermicStatus	INT	Motor thermal capacity in %.
ActivePower	INT	Power consumed by the motor in %.
CosPhi	REAL	Current value of phi cosine.
AO	INT	Value of the analog output. Its value is in <code>AO IOMAP</code> .
WarningCode	INT	<code>WarningCode</code> register.
FailCode0	INT	Code of last level 0 detected error. Indicates which detected error has occurred, <code>FailCode[0]</code> .
FailCode1	INT	Code of last level 1 detected error. Indicates which detected error has occurred, <code>FailCode[1]</code> .
FailCode2	INT	Code of last level 2 detected error. Indicates which detected error has occurred, <code>FailCode[2]</code> .

ATS22_CFG_DDT Type

Name	Type	Description
DataStatus, page 176	WORD	Provides the device status. Information on the Status output structure.
IOMap, page 177	WORD	Provides information on the state of the digital inputs/outputs of the device.
Info	INT	Provides device information.
WarningCode	INT	WarningCode register.
FailCode0	INT	Code of last level 0 detected error.
FailCode1	INT	Code of last level 1 detected error.
FailCode2	INT	Code of last level 2 detected error.
Frequency	INT	Voltage frequency in Hertz (w 279).
Voltage	INT	Input voltage in volts (w 260).
TotalStarts	INT	Number of times that the ATS22 has started (w 274).
TotalRuntime	INT	Total time (in hours) during which the ATS22 has been operating (w 273).
LastStartTime	INT	Total time that the last start lasted (w 275).
LastStartMaxC	INT	Maximum detected current during last start (w 276).
TripCurrent	INT	Current present when last tripping event occurred (w 280).

ATS22_ST.STW Word Structure

Bit	Description
0	Unknown technological module status. No variable refreshing.
1	Not ready.
2	Device is running.
3	Inoperable device.
4	Alarm on the device (follow-up requires resetting).
5	Communication interruption.
6	Requires resetting. ResetFail is required.
7	Refer to the ExtControlled output pin, page 169.
8	Refer to the Reseting output pin, page 169.
9	Refer to the EnableDFB input pin, page 166.

ATS_ST.STW Word Structure

Bit	Description
0	Unknown technological module status. No variable refreshing.
1	Not ready.
2	Device is running.
3	Inoperable device.
4	Alarm on the device or DFB (follow-up or screw terminal-based control).
5	Communication interruption.

Bit	Description
6	Requires resetting. ResetFail is required.
7	Refer to the ExtControlled output pin, page 169.
8	Refer to the Resetting output pin, page 169.
9	Refer to the EnabledDFB input pin, page 166.

ATS22_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 166.
1	Owner.
6	Refer to the Run input pin, page 166.
7	Refer to the ControlCommand input pin, page 166.

NOTE: The Owner bit enables to control the block from the ***_ST_DDT input/output structure ignoring the input signals of the block. It enables control from a monitoring system (HMI, SCADA, operator screen) in the Manual mode without using the programmed switching operation.

ATS_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 166.
1	Owner.
4	Refer to the QuickStop input pin, page 166
5	Refer to the EnableDevice input pin, page 166.
6	Refer to the Run input pin, page 166.
7	Refer to the ControlCommand input pin, page 166.

ATS22_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the Ready status in the Status output pin, page 169.
1	Refer to the Running status in the Status output pin, page 169.
2	Refer to the Trip status in the Status output pin, page 169.
3	Refer to the Alarm status in the Status output pin, page 169.
14	Refer to the ForcedLocalMode status in the Status output pin, page 169.
15	Refer to the Ramping status in the Status output pin, page 169.

ATS_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the ReadyToSwitchOn status in the Status output pin, page 169.
1	Refer to the SwitchedOn status in the Status output pin, page 169.
2	Refer to the OperationEnabled status in the Status output pin, page 169.
3	Refer to the Malfunction status in the Status output pin, page 169.
4	Refer to the VoltageEnabled status in the Status output pin, page 169.
5	Refer to the QuickStop status in the Status output pin, page 169.
6	Refer to the SwitchOnDisabled status in the Status output pin, page 169.
7	Refer to the Alarm status in the Status output pin, page 169.
8	Refer to the ForcedLocalMode status in the Status output pin, page 169.
9	Refer to the ResetAuthorization status in the Status output pin, page 169.
10	Refer to the Running status in the Status output pin, page 169.
11	Refer to the Accelerating status in the Status output pin, page 169.
12	Refer to the Decelaring status in the Status output pin, page 169.

ATS22_CFG.IOMap Word Structure

Bit	Description
0	Refer to the LI1 output pin, page 169.
1	Refer to the LI2 output pin, page 169.
2	Refer to the LI3 output pin, page 169.
3	Refer to the R1 output pin, page 169.
4	Refer to the R2 output pin, page 169.

Diagnostic Information Management

Overview

The diagnostic codes the device can return are read from the FailCode output variable.

Parameter Configuration Diagnostic Codes

This detected error indicates that a public variable parameter contains a value that is not allowed.

To reload new values, a rising edge is required on the EnableDFB input.

- FailCode[0]: 16#0003
- FailCode[1]: 16#0000
- FailCode[2]: 16#0004

The Failcode[0] variable can have the following codes:

Diagnostic Code	Symbol	Meaning
0	NOF	No detected failure.
1	UCF	Motor current below minimum limit.
2	OCF	Motor current above maximum limit.
3	PHbd	Phase imbalance.
4	Grdf	Ground fault.
5	OLF	Motor overload.
6	OtF	Excessively high motor temperature.
7	OHF	Excessively high motor temperature detected.
8	PIF	Inverted phase fault.
9	PHF	Phase or line fault, or inoperable motor.
10	USF	No voltage or voltage too low.
11	OSF	Voltage too high.
12	StF	Excessive starting time.
13	Snbf	Too many starts.
14	SSCr	Connection or short-circuit fault in thyristor.
15	EtF	External detected fault.
16	InF	Internal detected fault.
17	SLF	Modbus timeout.
18	trAP	Trap code.
19	SCF	Short-circuit.
20	bPF	Fault detected on bypass contactor.
21	CFE	Invalid configuration.

The inoperable device can be reset as long as `ControlCommand` is RUE. Otherwise, it can be reset on the starter through the screw terminal with the appropriate parameter configuration.

Modbus Communications Diagnostic Codes

For Modbus communications, this code is used to indicate that communications have not been established. This code can be reset.

- `FailCode[0]: 16#0002`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

After Modbus communications have been established, check Modbus client diagnostic codes for `FailCode [0]` and `FailCode [1]`. The components make a distinction between read requests problems and write requests problems:

- `FailCode[2]: 16#0001 Read`
- `FailCode[2]: 16#0002 Write`

Diagnostic Code Example

For a detected error, the code is:

- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0005`

Variable Speed Drives

What's in This Part

ATV Profile 180

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Variable Speed Drives family.

These function blocks do not reflect any specific installation.

<div>⚠ WARNING</div> <div>LOSS OF CONTROL<ul style="list-style-type: none">• Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.• Provide a fallback state for undesired control events or sequences.• Provide separate or redundant control paths wherever required.• Supply appropriate parameters, particularly for limits.• Review the implications of transmission delays and take actions to mitigate.• Review the implications of communication link interruptions and take actions to mitigate.• Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.• Apply local accident prevention and safety regulations and guidelines. ¹• Test each implementation of this library for proper operation before placing it into service.<p>Failure to follow these instructions can result in death, serious injury, or equipment damage.</p></div>
--

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.

ATV Profile

What's in This Chapter

Description	180
DFB Representation	182
Inputs	186
Outputs	194
Inputs/Outputs	205
Public Variables	211
Diagnostics Information Management	212

Overview

This chapter describes the DFBs of ATV profile.

Description

General

The ATV profile is used to manage the Altivar variable speed drives on different communication networks.

NOTE: *ASATV31* and *MBATV31* are deprecated control functions.

Function Description

The main functions of the DFB are described in the following table:

Function	Description
Control	Forward/reverse direction of rotation.
Speed	Allows the speed set-point of the device to be sent.
Device status indication	Displays the status of the device.
Remote resetting	Allows resetting of the device.
Control or Monitoring	Enables you to monitor the device. You can control the command or set-point from the controller or through the wired inputs/outputs of the speed drive. Also allows the command and set-point to be controlled separately.
Owner	Manages the control system which is the owner (Operator or Program). Therefore, it is responsible for setting the control.
Torque	Allows the drive to run based on Torque set point.

I/O Scanning Information

The below table explains the configuration for I/O scanning messaging.

Function Block	Modbus ID	Read Address	Read Length	Write Address	Write Length
ATV7161	255	0	9	0	7
ATV6xx	255	0	12	0	5
ATV9xx	255	0	12	0	6

Function Block	Modbus ID	Read Address	Read Length	Write Address	Write Length
EATV32	255	0	7	0	5
ATV6xxx	255	0	14	0	6

DTM Profile Selection

To select a DTM profile:

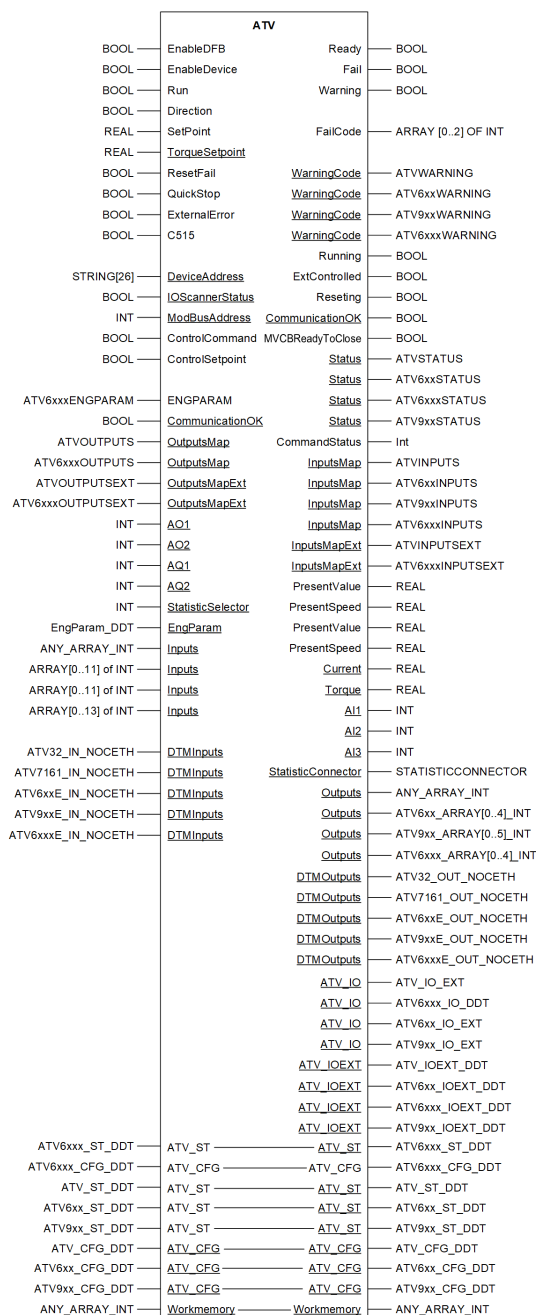
- For ATV6xx, you have to select **ATV6xx DFB** as its DTM profile.
- For ATV9xx, you have to select **ATV9xx DFB** as its DTM profile.
- For ATV6xxx, you have to select **ATV6xxx DFB** as its DTM profile.

NOTE: ATV blocks are designed for `DRIVECOM` profile. In the `Command and reference` parameter, `CHCF` (Channel configuration) is set to `Combined` channel mode as current value.

DFB Representation

Representation

The following figure represents the function module of ATV profile:



NOTE:

- The underlined parameters are specific for some components.
- DTM Inputs and DTM Outputs pins are not supported in EcoStruxure Process Expert.

For DFBs communicating by I/O scanning, variables read from the device retain their last value when a communication interruption occurs. For details, refer to the description of the corresponding output parameter, page 68.

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

When you configure the *Last value* parameter of the I/O scanner line, take into consideration the behavior of the DFB when a communication interruption occurs.

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

The table shows the parameters available for specific components:

Parameters		Com- po- nent- s								
			Mod- bus	Advan- tys STB	Ethernet					Profi- bus
					All	All	I/O scanning			
ATV7161	EAT- V32	ATV9xx	ATV6xx	ATV6xxx						
In- put- s	DeviceAddress	–	–	X	–	–	–	-	X	–
	IOScannerStatus	–	–	X	X	X	X	X	–	X
	ModBusAddress	X	–	–	–	–	–	-	–	–
	CommunicationOK	–	X	–	–	–	–	-	–	–
	OutputsMap (ATVOUTPUTS)	–	–	X	X	–	–	-	–	–
	OutputsMapExt (ATVOUTPUTSEXT)	–	–	X	X	–	–	-	–	–
	OutputsMap (ATV6xxOUTPUTS)	–	–	–	–	–	X	-	–	–
	OutputsMapExt (ATV6xxOUTPUTSEXT)	–	–	–	–	–	X	–	–	–
	OutputsMap (ATV6xxxOUTPUTS)	–	–	–	–	–	–	X	–	–
	OutputsMapExt (ATV6xxxOUTPUTSEXT)	–	–	–	–	–	–	X	–	–
	OutputsMap (ATV9xxOUTPUTS)	–	–	–	–	X	–	–	–	–
	OutputsMapExt (ATV9xxOUTPUTSEXT)	–	–	–	–	X	–	–	–	–
	TorqueSetpoint	–	–	–	–	X	–	X	–	–
	AO1	–	–	X	X	–	X	-	–	–
	AO2	–	–	X	–	–	X	-	–	–
	AQ1	–	–	–	–	X	–	X	–	–
	AQ2	–	–	–	–	X	–	X	–	–
	StatisticSelector	X	–	–	–	–	–	-	X	–
	EngParam	X	–	X	–	–	–	-	X	–
	ExternalError	–	–	–	–	X	X	X	–	–
	C515	–	–	–	–	X	X	X	–	–
	Inputs (ANY_ARRAY_INT)	–	X	X	X	–	–	-	–	X
	Inputs (ATV6xx_IN_DDT)	–	–	–	–	–	X	–	–	–
	Inputs (ATV6xxx_IN_DDT)	–	–	–	–	–	–	X	–	–

Parameters		Com- po- nent- s								
		Mod- bus	Advan- tys STB	Ethernet					Profi- bus	
		All	All	I/O scanning					Mes- saging	All
				ATV7161	EAT- V32	ATV9xx	ATV6xx	ATV6xxx		
In- put- s	<i>Inputs (ATV9xx_IN_DDT)</i>	–	–	–	–	X	–	–	–	–
	<i>DTMInputs (ATV32_IN_NOCETH)</i>	–	–	–	X	–	–	–	–	–
	<i>DTMInputs (ATV7161_IN_NOCETH)</i>	–	–	X	–	–	–	–	–	–
	<i>DTMInputs (ATV6xxE_IN_NOCETH)</i>	–	–	–	–	–	X	–	–	–
	<i>DTMInputs (ATV6xxxE_IN_NOCETH)</i>	–	–	–	–	–	–	X	–	–
	<i>DTMInputs (ATV9xxE_IN_NOCETH)</i>	–	–	–	–	X	–	–	–	–
Ou- tpu- ts	<i>WarningCode (ATVWARNING)</i>	X	X	X	X	–	–	–	X	X
	<i>WarningCode (ATV6xxWARNING)</i>	–	–	–	–	–	X	–	–	–
	<i>WarningCode (ATV6xxxWARNING)</i>	–	–	–	–	–	–	X	–	–
	<i>WarningCode (ATV9xxWARNING)</i>	–	–	–	–	X	–	–	–	–
	<i>CommunicationOK</i>	–	–	X	–	X	X	X	–	X
	<i>Status (ATVSTATUS)</i>	X	X	X	X	–	–	–	X	X
	<i>Status (ATV6xxSTATUS)</i>	–	–	–	–	–	X	–	–	–
	<i>Status (ATV6xxxSTATUS)</i>	–	–	–	–	–	–	X	–	–
	<i>Status (ATV9xxSTATUS)</i>	–	–	–	–	X	–	–	–	–
	<i>CommandStatus</i>	–	–	–	–	–	–	X	–	–
	<i>InputsMap (ATVINPUTS)</i>	–	–	X	X	–	–	–	–	–
	<i>InputsMapExt (ATVINPUTSEXT)</i>	–	–	X	X	–	–	–	–	–
	<i>InputsMap (ATV6xxINPUTS)</i>	–	–	–	–	–	X	–	–	–
	<i>InputsMap (ATV6xxxINPUTS)</i>	–	–	–	–	–	–	X	–	–
	<i>InputsMapExt (ATV6xxINPUTSEXT)</i>	–	–	–	–	–	X	–	–	–
	<i>InputsMapExt (ATV6xxxINPUTSEXT)</i>	–	–	–	–	–	–	X	–	–
	<i>InputsMap (ATV9xxINPUTS)</i>	–	–	–	–	X	–	–	–	–
	<i>InputsMapExt (ATV9xxINPUTSEXT)</i>	–	–	–	–	X	–	–	–	–
	<i>Current</i>	–	–	X	X	X	X	X	–	–
	<i>Torque</i>	–	–	X	X	X	X	X	–	–
Ou- tpu- ts	<i>Power</i>	–	–	–	–	X	X	X	–	–
	<i>AI1</i>	–	–	X	–	X	X	X	–	–
	<i>AI2</i>	–	–	X	–	X	X	X	–	–
	<i>AI3</i>	–	–	–	–	X	X	X	–	–
	<i>StatisticConnector</i>	X	–	–	–	–	–	–	–	–

Parameters		Com- po- nent- s								
			Mod- bus	Advan- tys STB	Ethernet					Profi- bus
					All	All	I/O scanning			
ATV7161	EAT- V32	ATV9xx	ATV6xx	ATV6xxx						
	Outputs (ANY_ARRAY_INT)	–	X	X	X	–	–	–	–	X
	Outputs (ATV6xx_OUT_DDT)	–	–	–	–	–	X	–	–	–
	Outputs (ATV6xxx_OUT_DDT)	–	–	–	–	–	–	X	–	–
	Outputs (ATV9xx_OUT_DDT)	–	–	–	–	X	–	–	–	–
	DTMOutputs (ATV32_OUT_NOCETH)	–	–	–	X	–	–	–	–	–
	DTMOutputs (ATV7161_OUT_NOCETH)	–	–	X	–	–	–	–	–	–
	DTMOutputs (ATV6xxE_OUT_NOCETH)	–	–	–	–	–	X	–	–	–
	DTMOutputs (ATV6xxxE_OUT_NOCETH)	–	–	–	–	–	–	X	–	–
	DTMOutputs (ATV9xxE_OUT_NOCETH)	–	–	–	–	X	–	–	–	–
	ATV_IO (ATV_IO_EXT)	–	–	X	X	–	–	–	–	–
	ATV_IOEXT (ATV_IOEXT_DDT)	–	–	X	X	–	–	–	–	–
	ATV_IO (ATV6xx_IO_EXT)	–	–	–	–	–	X	–	–	–
	ATV_IO (ATV6xxx_IO_EXT)	–	–	–	–	–	–	X	–	–
	ATV_IOEXT (ATV6xx_IOEXT_DDT)	–	–	–	–	–	X	–	–	–
	ATV_IOEXT (ATV6xx_IOEXT_DDT)	–	–	–	–	–	–	X	–	–
	ATV_IO (ATV9xx_IO_DDT)	–	–	–	–	X	–	–	–	–
	ATV_IOEXT (ATV9xx_IOEXT_DDT)	–	–	–	–	X	–	–	–	–
In- put- s/ out- put- s	ATV_ST (ATV_ST_DDT)	X	X	X	X	–	–	–	X	X
	ATV_ST (ATV6xx_ST_DDT)	–	–	–	–	–	X	–	–	–
	ATV_ST (ATV6xxx_ST_DDT)	–	–	–	–	–	–	X	–	–
	ATV_ST (ATV9xx_ST_DDT)	–	–	–	–	X	–	–	–	–
	ATV_CFG (ATV_CFG_DDT)	X	X	X	X	–	–	–	X	X
	ATV_CFG (ATV6xx_CFG_DDT)	–	–	–	–	–	X	–	–	–
	ATV_CFG (ATV6xxx_CFG_DDT)	–	–	–	–	–	–	X	–	–
	ATV_CFG (ATV9xx_CFG_DDT)	–	–	–	–	X	–	–	–	–
	Workmemory	X	–	–	–	–	–	–	X	–

X: Parameter is available.
–: Parameter is not available.

Inputs

Input Parameter Description

Name	Data type	Description
EnableDFB	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = The entire DFB is restarted (statuses, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation. <p>Public variable values are loaded during the first enabling cycle.</p>
EnableDevice	BOOL	<p>1 = Enables the devices if the EnableDFB variable is active.</p> <p>The speed drive has to be enabled in order to be controlled.</p>

If the device is reset and *Run* pin is active, then the device will auto start. If manual start of the device is required, then reset the *Run* pin followed by device.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Reset the *Run* variable before resuming operation.

Failure to follow these instructions can result in equipment damage.

NOTICE

UNINTENDED EQUIPMENT OPERATION

Configure the Torque setpoint through the drive.

Failure to follow these instructions can result in equipment damage.

Parameter	Type	Description
Run	BOOL	<p>1 = Starts the motor run in the direction selected with the <i>Direction</i> input variable.</p>
Direction	BOOL	<p>Direction of rotation of the motor.</p> <ul style="list-style-type: none"> 0 = Activates the reverse direction drive. 1 = Activates the forward direction drive. <p>You cannot change the direction of rotation by changing the sign in <i>SetPoint</i> input variable. <i>SetPoint</i> only accepts positive values.</p>
SetPoint	REAL	<p>The speed set-point is requested from the speed drive and only accepts positive values.</p> <p>It is measured in engineering units and you can configure these units with the following public variables:</p> <ul style="list-style-type: none"> HighRangeRpm LowRangeRpm HighRangeEngUnit LowRangeEngUnit <p>NOTE: Verify that the unit is within the correct range.</p> <p>The DFB makes the conversion between engineering units and speed drive rpms.</p>

Parameter	Type	Description
		<p>(engineering units) HighRangeEngUnit</p> <p>LowRangeEngUnit</p> <p>LowRangeRpm HighRangeRpm (rpm)</p>
TorqueSetPoint ⁵ and ⁶	REAL	The torque set-point is requested from the speed drive.
ResetFail	BOOL	1 = Resets the Fail output parameter to 0 or in case of inoperable device, sends a reset command to the device if ControlCommand is 1.
QuickStop	BOOL	1 = Stops the speed drive quickly with fast stop ramp. If there is a QuickStop, the Run bit has to be reset to resume operation.
ExternalError ⁶	BOOL	Indicates the drive that a detected process failure has occurred.
C515 ⁶	BOOL	User configurable digital input. For additional information refer to the respective ATV device user manual.
CommunicationOK*	BOOL	1 = The node is present on the bus. You can find this variable in Ethernet communications. For more information, refer to the CommunicationOk, page 248 variable.
IOScannerStatus*	BOOL	1 = The node is present on the bus. You can find this variable in Ethernet communications.
ModbusAddress*	INT	Device address within the Modbus network. You can find this variable in Modbus communications.
DeviceAddress ⁷	STRING[26]	Device address within the Ethernet network. Depending on the platform, the following definitions apply:
	Platform	IP Addressing DeviceAddress (variable)
	M340	'{IP} ID'
	M580	'{IP} ID'
	Quantum	'{IP} ID'
	NOTE: ID is 0.	
ControlCommand*	BOOL	<p>Indicates to the DFB whether the motor is being controlled locally or from a source external to the DFB.</p> <ul style="list-style-type: none"> 0 = Performs only read operations to monitor the status of the device and does not perform any control functions. 1 = Performs read operations and performs control operations not conflicting with control commands coming from an external control source. <p>If the speed driver is an ATV 61/32/12/212/6xx/6xxx/9xx unit and you need to use this input to switch between sending the set-point through the network and sending it through the screw terminal, you have to configure the speed driver so that it works accordingly. Refer to the Preparing the Device, page 250 for more information on how to set this configuration.</p>
ControlSetpoint	BOOL	<p>Indicates to the DFB whether the speed of the device is being controlled by the DFB or from a source external to the DFB.</p> <ul style="list-style-type: none"> 0 = Speed of the device is being controlled from an external source. 1 = Speed of the device is being controlled by the DFB.

Parameter	Type	Description	
		If the speed driver is an ATV 61/32/12/212/6xx/6xxx/9xx unit and you need to use this input to switch between sending the set-point through the network and sending it through the screw terminal, you have to configure the speed driver so that it works accordingly. Refer to the <i>Preparing the Device</i> , page 250 for more information on how to set this configuration.	
ENGPARAM ⁶	ATV6xx- xENGPARAM/ EngPara- mATV/ EngPara- mATV212/ EngPara- mATV7161/ EngParamE- MATV7161	Engineering parameters.	
	Name	Data type	Description
	Refresh	TIME	Time to refresh the cyclic data.
	ScanTime	TIME	Minimum time to maintain detected warning signals.
	MaxReset- Time	TIME	Maximum time between two resets.
	Scaling- FactorCur- rent	REAL	Define scaling factor for current depending on drive rate.
	HighRan- geEngUnit	REAL	Maximum inverter speed measured in user units.
	LowRan- geEngUnit	REAL	Minimum inverter speed measured in user units.
	HighRan- geRpm	INT	Maximum inverter speed measured in rpm.
	LowRan- geRpm	INT	Minimum inverter speed measured in rpm.
	ResetMode	BOOL	False = Manual reset True = Automatic reset
OutputsMap [*]	ATVOUT- PUTSMAP	Holds a data structure that is used to control the speed driver outputs. The information is available on Ethernet networks. You can only control the outputs of the speed driver with this input variable. You cannot control the outputs of the speed driver from the HMI/SCADA system. The following table describes the OutputsMap:	
	Parameter	Type	Description
	R1	BOOL	1 = Controls the state of the R1 relay output
	R2	BOOL	1 = Controls the state of the R2 relay output.
OutputsMap [*]	ATV6xxOUT- PUTS/ ATV9xxOUT- PUTS/ ATV6xx- xOUTPUTS	Holds a data structure that is used to control the speed driver outputs. You can only control the outputs of the speed driver with this input variable. You cannot control the outputs of the speed driver from the HMI/SCADA system. The following table describes the Outputs Map:	
	Parameter	Type	Description
	R1	BOOL	1 = Controls the state of the R1 relay output.
	R2	BOOL	1 = Controls the state of the R2 relay output.
	R3	BOOL	1 = Controls the state of the R3 relay output.

Parameter	Type	Description
OutputsMapExt*	ATVOUT-PUTSMAPEXT	<p>Holds a data structure that is used to control the speed driver outputs for ATV61 and ATV71 on Ethernet networks and for ATV32 extended cards on Ethernet networks.</p> <p>You can only control the outputs of the speed driver with this input variable. You cannot control the outputs of the speed driver from the HMI/SCADA system.</p> <p>The following table describes the OutputsMapExt:</p>
	Output	Type
	R3	BOOL
	R4	BOOL
	LO1	BOOL
	LO2	BOOL
	LO3	BOOL
OutputsMapExt*	LO4	BOOL
	LO4	BOOL
	LO4	BOOL
	LO4	BOOL
	LO4	BOOL
	LO4	BOOL
	LO4	BOOL
OutputsMapExt*	ATV6xxOUT-PUTSEXT/ ATV9xxOUT-PUTSEXT/ ATV6xx- xOUTPUT- SEXT	<p>Holds a data structure that is used to control the speed driver outputs.</p> <p>You can only control the outputs of the speed driver with this input variable. You cannot control the outputs of the speed driver from the HMI/SCADA system.</p> <p>The following table describes the OutputsMapExt:</p>
	Parameter	Type
	R4	BOOL
	R5	BOOL
	R6	BOOL
	DQ1 ⁶	BOOL
	DQ11	BOOL
	DQ12	BOOL
A01*	INT	Controls the value of the A01 analog output. You can find this value on devices on Ethernet networks. You can only control the analog output of the speed driver with this input variable. You cannot control the analog output of the speed from the HMI/SCADA system.
A02*	INT	Controls the value of the A02 analog output. This value can be found on ATV61 and ATV71 on Ethernet networks. You can only control the analog output of the speed driver with this input variable. You cannot control the analog output of the speed from the HMI/SCADA system.
AQ1 ^{5,6,8}	INT	Controls the value of the AQ1 analog output. The speed driver analog output can only be controlled with this input variable; it cannot be controlled from the HMI/SCADA system..
AQ2 ^{5,6,8}	INT	Controls the value of the AQ2 analog output. The speed driver analog output can only be controlled with this input variable; it cannot be controlled from the HMI/SCADA system..
StatisticSelector*	INT	<p>Variable is used to obtain statistics for the Modbus network (requests carried out, time between requests, so on). This data provides information for using StatisticConnector pin within the StatisticCounter DFB in General Purpose library for communication.</p> <p>The following table displays the StatisticSelector value:</p>
	Variable value	Description

Parameter	Type	Description
	1	Read statistics, client.
	2	Write statistics, client.
Inputs ¹	ANY_ARRAY_ INT	<p>Holds an array structure with the data obtained from the device. You can control the speed driver with this input variable. This input is reserved for the DFB, and you cannot use this input directly. To make the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247.</p> <p>The Inputs pin should not be connected when DTMIInputs pin is connected, otherwise the function detects an incorrect configuration.</p> <p>The information available for the ATV61 and ATV71 on Ethernet is shown in the following table:</p>
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
	Inputs [2]	INT
	Inputs [3]	INT
	Inputs [4]	INT
	Inputs [5]	INT
	Inputs [6]	INT
	Inputs [7]	INT
Inputs ²	ANY_ARRAY_ INT	<p>Holds an array structure with the data obtained from the device. You can control the speed driver with this input variable. This input is reserved for the DFB, and you cannot use this input directly. To make the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247.</p> <p>The Inputs pin should not be connected when DTMIInputs pin is connected, otherwise the function detects an incorrect configuration.</p> <p>The information available for the ATV32 on Ethernet is shown in the following table:</p>
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
	Inputs [2]	INT
	Inputs [3]	INT
	Inputs [4]	INT
	Inputs [5]	INT
Inputs ³	ANY_ARRAY_ INT	<p>Holds an array structure with the data obtained from the device. You can control the speed driver with this input variable. This input variable is reserved for the DFB, and you cannot use this variable directly. To make the control block to work properly, allocate the structure (%MWx). Refer to the Communications Technologies, page 247.</p> <p>The information available for speed drivers on an Advantys STB is shown in the following table:</p>
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
Inputs ⁴	ARRAY [0..27] OF BYTE	Holds a structure with the data obtained from the inverter.

Parameter	Type	Description
Inputs ^{5,8}	ARRAY [0..11] OF INT	<p>Holds an array structure with the data obtained from the device. You can control the speed driver with this input variable. This input is reserved for the DFB, and you cannot use this input directly. To make the control block to work properly, allocate the structure (%MWx).</p> <p>The Inputs pin should not be connected when DTMInputs pin is connected, otherwise the function detects an incorrect configuration.</p> <p>The information available for the ATV6xx/ATV9xx on Ethernet is shown in the following table:</p>
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
	Inputs [2]	INT
	Inputs [3]	INT
	Inputs [4]	INT
	Inputs [5]	INT
	Inputs [6]	INT
	Inputs [7]	INT
	Inputs [8]	INT
	Inputs [9]	INT
	Inputs [10]	INT
	Inputs [11]	INT
Inputs ⁶	ARRAY [0..13] OF INT	<p>Holds an array structure with the data obtained from the device. You can control the speed driver with this input variable. This input is reserved for the DFB, and you cannot use this input directly. To make the control block to work properly, allocate the structure (%MWx).</p> <p>The Inputs pin should not be connected when DTMInputs pin is connected, otherwise the function detects an incorrect configuration.</p> <p>The information available for the ATV6xxx on Ethernet is shown in the following table:</p>
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
	Inputs [2]	INT
	Inputs [3]	INT
	Inputs [4]	INT
	Inputs [5]	INT
	Inputs [6]	INT
	Inputs [7]	INT
	Inputs [8]	INT
	Inputs [9]	INT
	Inputs [10]	INT
	Inputs [11]	INT
	Inputs [12]	INT

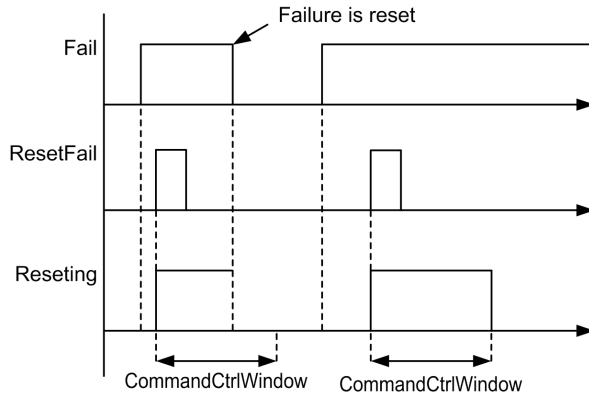
Parameter	Type	Description	
	Inputs [13]	INT	Drive command status (CMDS)
DTMInputs ¹	ATV7161_IN_NOCETH	<p>Holds an input structure for the data to be obtained from device DTM. You can read device monitoring information with this input variables. This input pin has to be used with the device DTM.</p> <p>The <i>DTMInputs</i> pin should not be connected when Inputs pin is connected, otherwise the function detects an incorrect configuration</p> <p>The information available for the ATV7161_IN_NOCETH on Ethernet is shown in the following table:</p>	
	Parameter	Type	Description
	RESERVED	UINT	Reserved.
	ETA	UINT	Status word.
	RFRD	INT	Output velocity.
	ERRD	UINT	CIA402 diagnostic code.
	LCR	INT	Motor current.
	OTR	INT	Motor torque.
	IL1R	INT	Logic input states.
	AI1C	INT	Analog input 1.
	AI2C	INT	Analog input 2.
DTMInputs ²	ATV32_IN_NOCETH	<p>Holds an input structure for the data to be obtained from device DTM. You can read device monitoring information with this input variables. This input pin has to be used with the device DTM.</p> <p>The <i>DTMInputs</i> pin should not be connected when Inputs pin is connected, otherwise the function detects an incorrect configuration</p> <p>The information available for the ATV32_IN_NOCETH on Ethernet is shown in the following table:</p>	
	Parameter	Type	Description
	ETA	UINT	Status word.
	RFRD	INT	Output velocity.
	ERRD	UINT	CIA402 diagnostic code.
	LCR	UINT	Motor current.
	OTR	INT	Motor torque.
	IL1R	UINT	Logic input states.
DTMInputs ^{5,8}	ATV6XXE_IN_NOCETH/ ATV9XXE_IN_NOCETH	<p>Holds an input structure for the data to be obtained from device DTM. You can read device monitoring information with this input variables. This input pin has to be used with the device DTM.</p> <p>The <i>DTMInputs</i> pin should not be connected when Inputs pin is connected, otherwise the function detects an incorrect configuration</p> <p>The information available for the ATV6XXE_IN_NOCETH/ATV9XXE_IN_NOCETH on Ethernet is shown in the following table:</p>	
	Parameter	Type	Description
	ETA	WORD	Status word.
	RFRD	INT	Output Velocity.
	ERRD	WORD	CIA402 diagnostic code.
	LCR	UINT	Motor current.
	OTR	INT	Motor torque.

Parameter	Type	Description	
	IL1R	WORD	Logic input states.
	AI1C	INT	Analog input 1.
	AI2C	INT	Analog input 2.
	AI3C	INT	Analog input 3.
	CCC	WORD	Active command channel.
	CRC	WORD	Active reference channel.
	EPRW	INT	Active electrical output power estimation.
DTMInputs ⁶	ATV6xxxE_IN_NOCETH	<p>Holds an input structure for the data to be obtained from device DTM. You can read device monitoring information with this input variables. This input pin has to be used with the device DTM.</p> <p>The <i>DTMInputs</i> pin should not be connected when Inputs pin is connected, otherwise the function detects an incorrect configuration</p> <p>The information available for the ATV6xxxE_IN_NOCETH on Ethernet is shown in the following table:</p>	
	Parameter	Type	Description
	ETA	WORD	Status word.
	RFRD	INT	Output Velocity.
	ERRD	WORD	CIA402 diagnostic code.
	LCR	UINT	Motor current.
	OTR	INT	Motor torque.
	IL1R	WORD	Logic input states.
	AI1C	INT	Analog input 1.
	AI2C	INT	Analog input 2.
	AI3C	INT	Analog input 3.
	CCC	WORD	Active command channel.
	CRC	WORD	Active reference channel.
	EPRW	INT	Active electrical output power estimation.
	BWS0	WORD	Mains and Inrush circuit breaker word status.
	CMDS	INT	Drive command status
<p>*: Parameters are available only for specific components.</p> <p>1: Parameter is available only for <i>ATV7161</i>.</p> <p>2: Parameter is available only for <i>EATV32</i>.</p> <p>3: Parameter is available only for <i>ASATV31</i> and <i>ASATV7161</i>.</p> <p>4: Parameter is available only for <i>PBATV7161</i>.</p> <p>5: Parameter is available only for <i>ATV9xx</i>.</p> <p>6: Parameter is available only for <i>ATV6xx</i>, <i>ATV9xx</i> and <i>ATV6xxx</i>.</p> <p>7: Parameter is applicable only for Ethernet messaging.</p> <p>8: Parameter is available only for <i>ATV6xx</i> and <i>All modbus and ethernet messaging ATV</i>.</p>			

Outputs

Output Parameter Description

Parameter	Type	Description
Ready	BOOL	<p>1 = The device is enabled and free of detected errors. The device is ready to carry out or carrying out any Run or Stop command.</p> <p>If QuickStop is enabled, this output resets to 0. This variable is TRUE as long as there are no communication interruption and the device is in DSP402 state 5.</p>
Fail	BOOL	<p>1 = A detected failure in the control block or in the device or a communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode.</p> <p>NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.</p>
Warning	BOOL	<p>1 = An alarm has been activated for the device. It cannot be reset because the signal remains active until the cause of the alarm is removed or until the maximum set ScanTime is reached.</p>
FailCode	ARRAY [0..2] OF INT	<p>When the Fail output is 1, it holds the code for the detected error.</p> <p>If the Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3 level structure. Refer to the Diagnostic Information Management, page 212 for more details.</p>
WarningCode*	ATVWARNING	<p>Holds a data structure with the alarm information currently on the speed driver.</p> <p>The following table describes the WarningCode:</p>
	Parameter	Type
	Device	1 = An alarm present on the device.
	Order	1 = An alarm. The device is not responding to the control command (Run , QuickStop) within the time specified in CommandCtrlWindow .
	ForcedLocalMode	1 = The device is forced locally. It is controlled through the screw terminals.
WarningCode ^{5,6}	ATV6xxWARNING/ ATV6xxxWARNING/ ATV9xxWARNING	<p>Holds a data structure with the alarm information currently on the speed driver.</p> <p>The following table describes the WarningCode:</p>
	Parameter	Type
	Device	1 = An alarm is present on the device.
	Order	1 = Follow-up alarm. The device is not responding to the control command (Run , QuickStop , so on) within the time specified in CommandCtrlWindow .
	ForcedLocalMode	1 = The device is forced locally. It is controlled through the screw terminals.
	ReservedReference-Channel	1 = The drive reference channel is reserved by DTM (Control Expert, SoMove) or WEB server.
	ReservedCommandChannel	1 = The drive command channel is reserved by DTM (Control Expert, SoMove) or WEB server.
Running	BOOL	<p>1 = The speed driver is running and has an output frequency.</p> <p>NOTE: If SetPoint is 0 and Run is active, the Running signal is activated.</p>
ExtControlled	BOOL	<p>1 = The device is being controlled from an external source (for example, from the console, from a push-button panel, or from the monitoring system) to the system.</p> <p>Provides information for programming.</p> <p>NOTE: The ControlCommand signal, the Owner variable, and the ForcedLocalMode status are used to activate this signal. You cannot use this signal as a ControlCommand input.</p>
Reseting	BOOL	<p>1 = A reset is being carried out.</p> <p>The CommandCtrlWindow variable indicates the maximum time for resetting the detected failure.</p>

Parameter	Type	Description	
		<p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code>.</p> <p>If the detected failure is reset, the <code>Fail</code> and <code>Resetting</code> output variables are reset (set to FALSE). If the detected failure is not reset, the <code>Resetting</code> variable is set to FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Refer to the timing diagram below.</p> 	
<code>CommunicationOk*</code>	BOOL	1 = Communication ok for the device present in the I/O scanner.	
<code>MVCBReadyToClose⁶</code>	BOOL	Medium voltage circuit breaker ready to close.	
<code>Status*</code>	ATVSTATUS	<p>The structure holds data containing the information that the block extracts from the status variable of the speed driver.</p> <p>The following table describes the status information:</p>	
	Parameter	Type	Description
	<code>ReadyToSwitchOn</code>	BOOL	1 = The device is ready to switch on (<code>DRIVECOM [ETA.0]</code>).
	<code>SwitchedOn</code>	BOOL	1 = The device is switched on (<code>ETA.1</code>).
	<code>OperationEnabled</code>	BOOL	1 = Operation is enabled (<code>ETA.2</code>).
	<code>Malfunction</code>	BOOL	1 = A detected failure on the device (<code>ETA.3</code>).
	<code>VoltageEnabled</code>	BOOL	1 = Voltage on the device terminals (<code>ETA.4</code>).
	<code>QuickStop</code>	BOOL	1 = Quick stop is activated (<code>ETA.5</code>).
	<code>SwitchOnDisabled</code>	BOOL	1 = Switch-on is disabled (<code>ETA.6</code>).
	<code>Alarm</code>	BOOL	1 = An alarm is present on the device (<code>ETA.7</code>).
	<code>ForcedLocalMode</code>	BOOL	1 = Forced local mode (<code>ETA.9</code>).
	<code>ReferenceReached</code>	BOOL	1 = The speed set-point has been reached (<code>ETA.10</code>).
	<code>ReferenceExceeded</code>	BOOL	1 = The speed set-point has exceeded limits (<code>ETA.11</code>).
	<code>StopImposed</code>	BOOL	1 = Stop is forced by remote control STOP key (<code>ETA.14</code>).
	<code>ForwardReverseRotation</code>	BOOL	<ul style="list-style-type: none">0 = Reverse running direction (<code>ETA.15</code>).1 = Forward running direction (<code>ETA.15</code>).
	<code>State</code>	INT	<p>Numerical code corresponding to the state of the speed driver.</p> <p>Refer to the <i>State</i>, page 201 table below.</p>
	<code>Info</code>	INT	<p>Numerical code with the information on statuses and required actions.</p> <p>Refer to the <i>Info</i>, page 201 table below.</p>
	<code>Status⁷</code>	<code>MBATV212STATUS</code>	<p>The structure holds data containing the information that the function block extracts from the status variable of the speed driver. The following table describes the status information:</p>
Parameter		Type	Description
<code>OutputFailure⁷</code>		BOOL	<ul style="list-style-type: none">1 = The device is active (<code>ETA.0</code>).

Parameter	Type	Description	
			<ul style="list-style-type: none">0 = The device is not active (ETA . 0).
	Tripped ⁷	BOOL	1 = The device is tripped (ETA . 1).
	Alarm ⁷	BOOL	1 = An alarm is present on the device (ETA . 2), page 214.
	MOFF ⁷	BOOL	1 = Main circuit undervoltage alarm is issued (ETA . 3).
	MotorSelectionTHR ⁷	BOOL	<ul style="list-style-type: none">1 = Motor 2 (THR2) is selected (ETA . 4).0 = Motor 1 (THR1) is selected (ETA . 4).
	PIControl ⁷	BOOL	1 = PI control is prohibited (ETA . 5)
	AccelerationPattern ⁷	BOOL	<ul style="list-style-type: none">1 = Acceleration/deceleration pattern 2 (AD2) is selected (ETA . 6).0 = Acceleration/deceleration pattern 1 (AD1) is selected (ETA . 6).
	DCBraking ⁷	BOOL	1 = Forced DC braking (ETA . 7).
	ForwardReverseRotation ⁷	BOOL	<ul style="list-style-type: none">1 = Reverse running direction (ETA . 9).0 = Forward running direction (ETA . 9).
	Run ⁷	BOOL	1 = Device is running (ETA . 10).
	CoastStop ⁷	BOOL	1 = ST is off (ETA . 11).
	EmergencyStop ⁷	BOOL	1 = Emergency stop status (ETA . 12).
	StandbyST ⁷	BOOL	1 = The device is standby ST mode (ETA . 13).
	Standby ⁷	BOOL	1 = The device is standby mode (ETA . 14).
	LocalRemote ⁷	BOOL	<ul style="list-style-type: none">1 = The device is in local mode (ETA . 15).0 = The device is in remote mode (ETA . 15).
Status ^{5,6}	ATV6xxSTATUS/ ATV9xxSTATUS/ ATV6xxxSTATUS	The structure holds data containing the information that the function block extracts from the status variable of the speed driver. The following table describes the status information:	
	Parameter	Type	Description
	ReadyToSwitchOn	BOOL	1 = The device is ready to switch on (DRIVECOM [ETA . 0]).
	SwitchedOn	BOOL	1 = The device is switched on (ETA . 1).
	OperationEnabled	BOOL	1 = Operation is enabled (ETA . 2).
	Malfunction	BOOL	1 = A detected failure on the device (ETA . 3).
	VoltageEnabled	BOOL	1 = Voltage on the device terminals (ETA . 4).
	QuickStop	BOOL	1 = Quick stop is activated (ETA . 5).
	SwitchOnDisabled	BOOL	1 = Switch-on is disabled (ETA . 6).
	Alarm	BOOL	1 = An alarm is present on the device (ETA . 7).
	ForcedLocalMode	BOOL	1 = Forced local mode (ETA . 9).
	ReferenceReached	BOOL	1 = The speed set-point has been reached (ETA . 10).
	ReferenceExceeded	BOOL	1 = The speed set-point has exceeded limits (ETA . 11).
	StopImposed	BOOL	1 = Stop is forced by remote control STOP key (ETA . 14).
	ForwardReverseRotation	BOOL	<ul style="list-style-type: none">1 = Forward running direction (ETA . 15).0 = Reverse running direction (ETA . 15). NOTE: <ul style="list-style-type: none">Drive is in ready state, default forward status will be active.Drive is in run state, status gets updated as per direction selection.
	ReservedReferenceChannel	BOOL	1 = The drive reference channel is reserved by DTM (Control Expert, SoMove) or WEB server (w8441 . 15) CRC.
	ReservedCommandChannel	BOOL	1 = The drive command channel is reserved by DTM (Control Expert, SoMove) or WEB server (w8442 . 15) CCC.
	State	INT	Numerical code corresponding to the state of the speed driver.

Parameter	Type	Description	
			Refer to the <i>State</i> , page 201 table.
	Info	INT	Numerical code with the information on statuses and required actions. Refer to the <i>Info</i> , page 201 table.
CommandStatus ⁶	INT	Indicate to controller program, from which source speed driver command is controlled. <ul style="list-style-type: none"> • 0 - No command selected • 1 - Remote command selected • 2 - Local command selected • 3 - Panel command selected 	
InputsMap*	ATVINPUTS	Holds a data structure with the information on the state of the inputs of the speed driver. The information is available on Ethernet networks. The following table describes the InputsMap:	
	Input	Type	Description
	L1	BOOL	1 = The state of the digital input L1.
	L2	BOOL	1 = The state of the digital input L2.
	L3	BOOL	1 = The state of the digital input L3.
	L4	BOOL	1 = The state of the digital input L4.
	L5	BOOL	1 = The state of the digital input L5.
	L6	BOOL	1 = The state of the digital input L6.
InputsMap*	ATV6xxINPUTS/ ATV9xxINPUTS/ ATV6xxxINPUTS	Holds a data structure with the information on the state of the inputs of the speed driver. The following table describes the InputsMap:	
	Parameter	Type	Description
	DI1	BOOL	1 = State of the digital input DI1.
	DI2	BOOL	1 = State of the digital input DI2.
	DI3	BOOL	1 = State of the digital input DI3.
	DI4	BOOL	1 = State of the digital input DI4.
	DI5	BOOL	1 = State of the digital input DI5.
	DI6	BOOL	1 = State of the digital input DI6.
	DI7 ⁶	BOOL	1 = State of the digital input DI7.
	DI8 ⁶	BOOL	1 = State of the digital input DI8.
InputsMapExt*	ATVINPUTSEXT	Holds a data structure with the information on the state of the inputs of the speed driver for ATV61 and ATV71 extended cards on Ethernet networks. The following table describes the InputsMapExt:	
	Input	Type	Description
	L7	BOOL	1 = The state of the digital input L7.
	L8	BOOL	1 = The state of the digital input L8.
	L9	BOOL	1 = The state of the digital input L9.
	L10	BOOL	1 = The state of the digital input L10.
	L11	BOOL	1 = The state of the digital input L11.
	L12	BOOL	1 = The state of the digital input L12.
	L13	BOOL	1 = The state of the digital input L13.
	L14	BOOL	1 = The state of the digital input L14.
InputsMapExt ^{5,6}	ATV6xxINPUTSEXT/ ATV9xxINPUTSEXT/ ATV6xxxINPUTSEXT	Holds a data structure with the information on the state of the inputs of the speed driver.	

Parameter	Type	Description	
		The following table describes the InputsMapExt:	
	Parameter	Type	Description
	DI7 ⁵	BOOL	1 = State of the digital input DI7.
	DI8 ⁵	BOOL	1 = State of the digital input DI8.
	DI11	BOOL	1 = State of the digital input DI11.
	DI12	BOOL	1 = State of the digital input DI12.
	DI13	BOOL	1 = State of the digital input DI13.
	DI14	BOOL	1 = State of the digital input DI14.
	DI15	BOOL	1 = State of the digital input DI15.
	DI16	BOOL	1 = State of the digital input DI16.
PresentValue	REAL	Current speed of rotation in engineering units (EU).	
PresentSpeed	REAL	Current motor speed in rpm (rotation per minute).	
Current*	REAL	Present motor current in percentage (%). You can find this value on speed drives on ATV61 and ATV71 units on Ethernet networks and on speed drivers on ATV32 unit on Ethernet networks..	
Torque*	REAL	Current motor torque in 0.1 A increment. You can find this value on ATV61 and ATV71 on Ethernet networks and on ATV32 unit on Ethernet networks..	
Power*	REAL	Actual drive electrical power in kW. (Resolution is 0.00).	
AI1*	INT	The value of the AI1 analog device input. You can find this value on devices ATV61 and ATV71 units on Ethernet networks.	
AI2*	INT	The value of the AI2 analog device input. You can find this value on ATV61 and ATV71 on Ethernet networks.	
AI3*	INT	The value of the AI3 analog device input.	
ATV_IO*	ATV_IO_DDT, page 202	Device data structure holds the information for performing monitoring functions. The information used by the operator screen is readable from HMI or SCADA system.	
ATV_IOEXT*	ATV_IOEXT_DDT, page 202	Device data structure holds the information for performing monitoring functions. The information used by the operator screen is readable from HMI or SCADA system.	
ATV_IO ^{5,6}	ATV6xx_IO_DDT, page 203/ATV9xx_IO_DDT, page 203/ATV6xxx_IO_DDT, page 203	Device data structure holds the information for performing monitoring functions. The information used by the operator screen is readable from HMI or SCADA system.	
ATV_IOEXT ^{5,6}	ATV6xx_IOEXT_DDT, page 203/ATV9xx_IOEXT_DDT, page 203/ATV6xxx_IOEXT_DDT, page 203	Device data structure holds the information for performing monitoring functions. The information used by the operator screen is readable from HMI or SCADA system.	
Outputs ¹	ANY_ARRAY_INT	Holds an array structure with data sent to the device. You can control the speed driver with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. To make the control block work properly, allocate the structure (% MWx). Refer to the Communications Technologies, page 247.	
		The following table describes the ANY_ARRAY_INT information available for EATV61 and EATV71 units on Ethernet networks:	
	Parameter	Type	Description
	Outputs [0]	INT	Control word (CMD).
	Outputs [1]	INT	Speed set-point (LFRD).
	Outputs [2]	INT	Logic output states (OL1R).
	Outputs [3]	INT	Analog output 1 physical value (AO1C).
	Outputs [4]	INT	Analog output 2 physical value (AO2C).
	Outputs [5]	INT	Analog output 3 physical value (AO3C).
Outputs ²	ANY_ARRAY_INT	Holds an array structure with data sent to the device. You can control the speed driver with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. To make the control block work properly, allocate the structure (% MWx). Refer to the Communications Technologies, page 247.	

Parameter	Type	Description	
		The following table describes the ANY_ARRAY_INT information available for ATV32E on Ethernet networks:	
	Parameter	Type	Description
	Outputs [0]	INT	Control word (CMD).
	Outputs [1]	INT	Speed set-point (LFRD).
	Outputs [2]	INT	Logic output states (OL1R).
	Outputs [3]	INT	Analog output 1 physical value (AO1C).
Outputs ³	ANY_ARRAY_INT	<p>Holds an array structure with data sent to the device. You can control the speed driver with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. To make the control block work properly, allocate the structure (%MWx). Refer to the Communication Technologies, page 247.</p> <p>The information available for speed drivers on an Advantys STB is shown in the following table:</p>	
	Parameter	Type	Description
	Outputs [0]	INT	Control word (CWD).
	Outputs [1]	INT	Speed set-point.
Outputs ⁴	ARRAY [0..27] OF BYTE	Holds an array structure with data sent to the inverter. (Reserved for DFB usage).	
Outputs ⁵	ARRAY [0..4] OF INT	<p>Holds an array structure with data sent to the device. You can control the speed driver with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. To make the control block work properly, allocate the structure (%MWx).</p> <p>The following table describes the ATV6xx_OUT_DDT information available for ATV6xx on Ethernet networks:</p>	
	Parameter	Type	Description
	Outputs [0]	INT	Control word (CMD).
	Outputs [1]	INT	Speed set-point (LFRD).
	Outputs [2]	INT	Logic output states (OL1R).
	Outputs [3]	INT	Analog output 1 physical value (AO1C).
	Outputs [4]	INT	Analog output 2 physical value (AO2C).
Outputs ⁶	ARRAY [0..5] OF INT	<p>Holds an array structure with data sent to the device. You can control the speed driver with this output variable. This variable is reserved for the DFB, and you cannot use this variable directly. To make the control block work properly, allocate the structure (%MWx).</p> <p>The following table describes the ATV9xx_OUT_DDT/ATV6xxx_OUT_DDT information available for ATV9xx/ATV6xxx on Ethernet networks:</p>	
	Parameter	Type	Description
	Outputs [0]	INT	Control word (CMD).
	Outputs [1]	INT	Speed set-point (LFRD).
	Outputs [2]	INT	Torque set-point (LFR).
	Outputs [3]	INT	Logic output states (OL1R).
	Outputs [4]	INT	Analog output 1 physical value (AO1C).
	Outputs [5]	INT	Analog output 2 physical value (AO2C).
DTMOutputs ¹	ATV7161E_OUT_NOETH	<p>Holds an output structure for the data to be written from controller to the device via Device DTM. This output pin has to be used when the device DTM is used.</p> <p>The following table describes the ATV7161E_OUT_NOETH information available for ATV7161 on Ethernet networks:</p>	
	Parameter	Type	Description
	Reserved.	UINT	Reserved.
	CMD	UINT	Control word.
	LFRD	INT	Speed set-point.

Parameter	Type	Description	
	OL1R	UINT	Logic output states.
	AO1C	INT	Analog output 1 physical value.
	AO2C	INT	Analog output 2 physical value.
	AO3C	INT	Analog output 3 physical value.
DTMOutputs ²	ATV32_OUT_NOCETH	Holds an output structure for the data to be written from controller to the device via Device DTM. This output pin has to be used when the device DTM is used The following table describes the ATV32_OUT_NOCETH information available for ATV32E on Ethernet networks:	
	Parameter	Type	Description
	CMD	UINT	Control word.
	LFRD	INT	Speed set-point.
	OL1R	UINT	Logic output states.
	AO1C	INT	Analog output 1 physical value.
DTMOutputs ⁵	ATV6xxE_OUT_NOCETH	Holds an output structure for the data to be written from controller to the device via Device DTM. This output pin has to be used when the device DTM is used The following table describes the ATV6xxE_OUT_NOCETH information available for ATV6xx on Ethernet networks:	
	Parameter	Type	Description
	CMD	INT	Control word.
	LFRD	INT	Speed set-point.
	OL1R	INT	Logic output states.
	AO1C	INT	Analog output 1 physical value.
	AO2C	INT	Analog output 2 physical value.
DTMOutputs ⁶	ATV9xxE_OUT_NOCETH/ ATV6xxxE_OUT_NOCETH	Holds an output structure for the data to be written from controller to the device via Device DTM. This output pin has to be used when the device DTM is used The following table describes the ATV9xx_OUT_NOCETH/ATV6xxxE_OUT_NOCETH information available for ATV9xx/ATV6xxx on Ethernet networks:	
	Parameter	Type	Description
	CMD	INT	Control word.
	LFRD	INT	Speed set-point.
	LTR	INT	Torque set-point.
	OL1R	INT	Logic output states.
	AO1C	INT	Analog output 1 physical value.
	AO2C	INT	Analog output 2 physical value.

Parameter	Type	Description	
Statistic-Connector*	STATISTICCONNECTOR	Information data is used with Modbus communication to obtain statistics on the Modbus network (requests carried out, time between requests). This structure has been created for its use together with the <code>StatisticCounter</code> DFB in General Purpose library for communication. The following table describes the <code>StatisticConnector</code> :	
	Parameter	Type	Description
	Start	BOOL	1 = The operation has started.
	EndOk	BOOL	1 = The operation has ended correctly.
	EndNOk	BOOL	1 = The operation has ended with a detected error.
	PartialTime	DINT	Partial time.
<p>*: Parameters are available for specific components.</p> <p>1: Parameter is available only for <i>ATV7161</i>.</p> <p>2: Parameter is available only for <i>EATV32</i>.</p> <p>3: Parameter is available only for <i>ASATV31</i> and <i>ASATV7161</i>.</p> <p>4: Parameter is available only for <i>PBATV7161</i>.</p> <p>5: Parameter is available only for <i>ATV6xx</i>.</p> <p>6: Parameter is available only for <i>ATV9xx</i> and <i>ATV6xxx</i>.</p> <p>7: Parameter is available only for <i>MBATV212</i>.</p>			

State

The following table describes the `State` variable:

Variable value	Description
-2	Detected error in DFB.
-1	Not initialized. Waiting for data.
0	Disabled.
2	Switch on disabled (Nst status).
3	Ready to switch on (Nst status).
4	Switched on (Nst status).
5	Operation enabled (Rdy status).
6	Quick stop (Fst status).
8	Inoperable device (Flt status).

Info

The code with the information is shown on the Control Expert operator screen. This variable is for informational purposes only. Do not use it to program switching operations. The following table describes the `Info` variable:

Variable value	Description
1	Incorrect configuration of DFB parameter.
2	Waiting <code>SwitchOnDisabled</code> .
3	Waiting <code>ReadyToSwitchOn</code> .
4	Waiting <code>SwitchOn</code> .
5	Waiting <code>OperationEnabled</code> .

Variable value	Description
6	Waiting <code>QuickStop</code> active.
10	Waiting for device information.
11	Missing <code>EnableDFB</code> .
12	Communication interruption.
13	ETA value is 0.
14	Remove local forcing has to be 0.
21	Missing <code>EnableDevice</code> .
23	Remove <code>Run</code> has to be 0.
24	Remove <code>ResetFail</code> . Reset again.
51	Speed driver stopped.
52	Speed driver running with speed 0.
53	Speed driver running.
54	Speed driver at rated running
55	Waiting speed-driver operation (Run).
56	Wait speed driver speed.
57	Waiting speed driver stop.
58	Speed set-point is 0.
59	Speed outside limits.
61	Remove <code>QuickStop</code> has to be 0.
62	<code>QuickStop</code> is activated.
81	Missing <code>ResetFail</code> . Inoperable speed driver.
82	Do speed driver reset.
99	Unexpected state.

ATV_IO_DDT Type

Name	Type	Description
<code>OutputsMap</code> , page 203	WORD	Data with the outputs of the speed driver.
<code>AO1</code>	INT	Device control. Value of the <code>AO1</code> analog output (value of the <code>AO1</code> input variable).
<code>Current</code>	REAL	Present current value. Current motor current in 0.1 A increment (value of the <code>Current</code> output variable).

ATV_IOEXT_DDT Type

Name	Type	Description
<code>InputsMap</code> , page 204	WORD	State of the digital inputs of the speed driver.
<code>AI1*</code>	INT	<code>AI1</code> analog input. Value of the <code>AI1</code> analog input (value of the <code>AI1</code> output variable)
<code>AI2*</code>	INT	<code>AI2</code> analog input. Value of the <code>AI2</code> analog output (value of the <code>AI2</code> output variable).
<code>AO2*</code>	INT	<code>AO2</code> analog output. Value of the <code>AO2</code> analog output (Value of the <code>AO2</code> input variable).

Name	Type	Description
Torque	DINT	Current torque value. Current motor torque in 0.1 A increment (value of the <code>Torque</code> output variable).
*: Parameters are available for specific components.		

ATV6xx_IO_DDT/ATV9xx_IO_DDT/ATV6xxx_IO_DDT Type

Name	Type	Description
OutputsMap, page 203	WORD	Data with the outputs of the speed driver.
AQ1	INT	Device control. Value of the AQ1 analog output (value of the AQ1 input variable).
Current	REAL	Motor current value. Actual motor current in Amps.

ATV6xx_IOEXT_DDT/ATV9xx_IOEXT_DDT/ATV6xxx_IOEXT_DDT Type

Name	Type	Description
InputsMap, page 205	WORD	State of the digital inputs of the speed driver.
AI1	INT	AI1 analog input. Value of the AI1 analog input (value of the AI1 output variable)
AI2	INT	AI2 analog input. Value of the AI2 analog output (value of the AI2 output variable).
AI3	INT	AI3 analog input. Value of the AI3 analog output (value of the AI3 output variable).
AQ2	INT	AQ2 analog output. Value of the AQ2 analog output (Value of the AQ2 input variable).
Torque	DINT	Motor torque value. Actual drive torque in %.
Power	REAL	Electrical power value. Actual driver power in kW.

OutputsMap Word Structure (ATV6xx, ATV9xx, ATV6xx)

Provides control over the device outputs. The following table describes the OutputsMap word structure:

Bit	Description
0	R1 output state (<i>OutputsMap.R1</i>)
1	R2 output state (<i>OutputsMap.R2</i>).
2	R3 output state (<i>OutputsMap.R3</i>).
3	R4 output state (<i>OutputsMapExt.R4</i>).
4	R5 output state (<i>OutputsMapExt.R5</i>).
5	R6 output state (<i>OutputsMapExt.R6</i>).
8	DQ1 output state (<i>OutputsMapExt.DQ1</i>).
12	DQ11 output state (<i>OutputsMapExt.DQ11</i>).
13	DQ12 output state (<i>OutputsMapExt.DQ12</i>).

OutputsMap Word Structure (ATV7161)

Provides control over the device outputs. The following table describes the OutputsMap word structure:

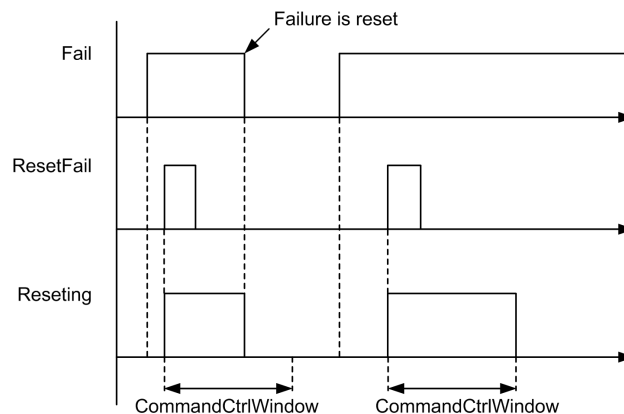
Bit	Description
0	R1 output state (OutputsMap.R1)
1	R2 output state (OutputsMap.R2).
2	R3 output state (OutputsMap.R3) for ATV 71 and ATV 61.
3	R4 output state (OutputsMap.R4) for ATV 71 and ATV 61.
4	LO1 output state (OutputsMapExt.LO1) for ATV 71 and ATV 61.
5	LO2 output state (OutputsMapExt.LO2) for ATV 71 and ATV 61.
6	LO3 output state (OutputsMapExt.LO3) for ATV 71 and ATV 61.
7	LO4 output state (OutputsMapExt.LO4) for ATV 71 and ATV 61.

InputsMap Word Structure (ATV7161)

Provides the value of the device inputs. The following table describes the InputsMap word structure:

Bit	Description
0	Value of the digital input DI1 (<i>InputsMap.DI1</i>).
1	Value of the digital input DI2 (<i>InputsMap.DI2</i>).
2	Value of the digital input DI3 (<i>InputsMap.DI3</i>).
3	Value of the digital input DI4 (<i>InputsMap.DI4</i>).
4	Value of the digital input DI5 (<i>InputsMap.DI5</i>).
5	Value of the digital input DI6 (<i>InputsMap.DI6</i>).
6	Value of the digital input DI7 (<i>InputsMap.DI7</i>).
7	Value of the digital input DI8 (<i>InputsMap.DI8</i>).
10	Value of the digital input DI11 (<i>InputsMap.DI11</i>).
11	Value of the digital input DI12 (<i>InputsMap.DI12</i>).
12	Value of the digital input DI13 (<i>InputsMap.DI13</i>).
13	Value of the digital input DI14 (<i>InputsMap.DI14</i>).
14	Value of the digital input DI15 (<i>InputsMap.DI15</i>).
15	Value of the digital input DI16 (<i>InputsMap.DI16</i>).

Timing diagram:



InputsMap Word Structure (ATV6xx, ATV9xx, ATV6xxx)

Provides the value of the device inputs. The following table describes the InputsMap word structure:

Bit	Description
0	Value of the digital input L1 (InputsMap.L1).
1	Value of the digital input L2 (InputsMap.L2).
2	Value of the digital input L3 (InputsMap.L3).
3	Value of the digital input L4 (InputsMap.L4).
4	Value of the digital input L5 (InputsMap.L5).
5	Value of the digital input L6 (InputsMap.L6).
6	Value of the digital input L7 (InputsMap.L7).
7	Value of the digital input L8 (InputsMap.L8).
8	Value of the digital input L9 (InputsMap.L9).
9	Value of the digital input L10 (InputsMap.L10).
10	Value of the digital input L11 (InputsMap.L11).
11	Value of the digital input L12 (InputsMap.L12).
12	Value of the digital input L13 (InputsMap.L13).
13	Value of the digital input L14 (InputsMap.L14).
14	Value of the digital input L15 (InputsMap.L15).

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
ATV_ST ¹	ATV_ST_DDT, page 206	Device data structure that holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is readable/writable from the HMI or SCADA system.
ATV_CFG ¹	ATV_CFG_DDT, page 206	Data structure with the device information. The information is used by the operator screen and is readable from the HMI or SCADA system.

Parameter	Type	Description
ATV_ST ²	ATV6xx_ST DDT, page 207/ ATV9xx_ST DDT, page 207/ ATV6xxx_ST DDT, page 207	Device data structure holds the minimum information required for performing control and monitoring functions. The information used by the operator screen, is readable/writable from the HMI/SCADA system.
ATV_CFG ²	ATV6xx_CFG DDT, page 208/ ATV9xx_CFG DDT, page 208/ ATV6xxx_CFG DDT, page 207	Data structure with the device information. The information is used by the operator screen, is readable from the HMI/SCADA system.
WorkMemory ¹	ANY_ARRAY_INT	Array is used for Modbus communications. This variable is to be used with a Modbus port that serializes Modbus requests in an optimum manner.
¹ : Parameters are available only for specific components. ² : Parameters are available only for ATV6xx, ATV9xx and AVT6xxx.		

ATV_ST_DDT Type

Name	Type	Description
STW, page 209	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW, page 207	WORD	Device control. Provides the means necessary to control the device from the monitoring subsystem or from the operator screen if Owner (1), or only from the monitoring subsystem if Owner (0). If Owner is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
PresentValue	REAL	Current speed value. Current speed of rotation.
SetPoint	REAL	Set-point variable. The speed set-point is requested from the speed driver.

ATV_CFG_DDT Type

Name	Type	Description
DataStatus, page 209	WORD	Information on the device status. Information on the Status output structure.
Info	INT	Device information. Speed driver information. Its value is Info status.
Warning-Code, page 210	WORD	Alarm code information. Takes the values from the WarningCode output.
State	INT	Speed driver status code. Speed driver status code information. Its value is State status.
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred FailCode[0].
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred FailCode[1].
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred FailCode[2].

ATV6xx_ST_DDT/ATV9xx_ST_DDT/ATV6xxx_ST_DDT Type

Name	Type	Description
STW	WORD	Provides information on the device status. Access to the data held in this bit word is read-only.
CFGW, page 207	WORD	Device control. Provides the means necessary to control the device from the monitoring subsystem or from the operator screen if <i>Owner</i> (1), or only from the monitoring subsystem if <i>Owner</i> (0). If <i>Owner</i> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.
PresentValue	REAL	Actual speed value. Present speed of rotation
SetPoint	REAL	Speed set-point variable. The speed set-point is requested from the speed driver.
TorqueSet-Point*	REAL	Torque set-point variable is requested from the speed driver.
*: Parameters are available for 9xx and 6xxx.		

ATV6xx_CFG_DDT/ATV9xx_CFG_DDT/ATV6xxx_CFG_DDT Type

Name	Type	Description
DataStatus	WORD	Provides information on the device status. Information on the <i>Status</i> output structure.
Channelstatus, page 210	WORD	Provides information on the channel status. Information on the <i>Status</i> output structure.
ExtendedStatus, page 210*	WORD	Extended device status.
Info	INT	Device information. Speed driver information. Its value is <i>Info</i> status.
State	INT	Speed driver status code. Speed driver status code information. Its value is <i>State</i> status.
WarningCode	WORD	Notification code register.
FailCode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred <i>FailCode</i> [0].
FailCode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred <i>FailCode</i> [1].
FailCode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred <i>FailCode</i> [2].
*: Parameter is applicable only for ATV6xxx.		

ATV_ST.STW/ATV6xx_ST.STW/ATV9xx_ST.STW/ATV6xxx_ST.STW Word Structure

Bit	Description
0	Unknown technological module status. No variable refreshing.
1	Not ready.
2	Technological module is running.
3	Inoperable device.
4	Alarm on the device or DFB (follow-up or screw terminal-based control).

Bit	Description
5	Communication interruption.
6	Requires resetting. ResetFail is required.
7	Refer to the ExtControlled output pin, page 194.
8	Refer to the Resetting output pin, page 194.
9	Refer to the EnabledDFB input pin, page 186.
15	Direction of rotation.

ATV_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 186.
1	Owner.
3	Refer to the Direction input pin, page 186.
4	Refer to the QuickStop input pin, page 186.
5	Refer to the EnableDevice input pin, page 186.
6	Refer to the Run input pin, page 186.
7	Refer to the ControlCommand input pin, page 186.
8	Refer to the ControlSetPoint input pin, page 186.

ATV6xx_ST.CFGW/ATV9xx_ST.CFGW/ATV6xxx_ST.CFGW Word Structure

Bit	Description
0	Refer to the ResetFail input pin, page 186.
1	Owner.
3	Refer to the Direction input pin, page 186.
4	Refer to the QuickStop input pin, page 186.
5	Refer to the EnableDevice input pin, page 186.
6	Refer to the Run input pin, page 186.
7	Refer to the ControlCommand input pin, page 186.
8	Refer to the ControlSetPoint input pin, page 186.
9	Refer to the ExternalError input pin, page 186.

NOTE: The `Owner` bit enables to control the block from the `***_ST_DDT` input/output structure ignoring the input signals of the block. It enables control from a monitoring system (HMI, SCADA, Operator screen) in the Manual mode without using the programmed switching operation.

ATV_CFG.DataStatus/ATV6xx_CFG.DataStatus/ATV9xx_CFG.DataStatus/ATV6xxx_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the ReadyToSwitchOn status in the Status output pin, page 194.
1	Refer to the SwitchedOn status in the Status output pin, page 194.
2	Refer to the OperationEnabled status in the Status output pin, page 194.
3	Refer to the Malfunction status in the Status output pin, page 194.
4	Refer to the VoltageEnabled status in the Status output pin, page 194.
5	Refer to the QuickStop status in the Status output pin, page 194.
6	Refer to the SwitchOnDisabled status in the Status output pin, page 194.
7	Refer to the Alarm status in the Status output pin, page 194.
8	Refer to the ForcedLocalMode status in the Status output pin, page 194.
9	Refer to the ReferenceReached status in the Status output pin, page 194.
10	Refer to the ReferenceExceeded status in the Status output pin, page 194.
11	Refer to the StopImposed status in the Status output pin, page 194.
12	Refer to the ForwardReverseRotation status in the Status output pin, page 194.

ATV_CFG.DataStatus Word Structure for MBATV212 Variable Speed Drive

Bit	Description
0	Refer to the OutputFailure status in the Status output pin, page 194.
1	Refer to the Tripped status in the Status output pin, page 194.
2	Refer to the Alarm status in the Status output pin, page 194.
3	Refer to the MOFF status in the Status output pin, page 194.
4	Refer to the MotorSelectionTHR status in the Status output pin, page 194.
5	Refer to the PIControl status in the Status output pin, page 194.
6	Refer to the AccelerationPattern status in the Status output pin, page 194.
7	Refer to the DCBraking status in the Status output pin, page 194.
9	Refer to the ForwardReverseRotation status in the Status output pin, page 194.
10	Refer to the Run status in the Status output pin, page 194.
11	Refer to the CoastStop status in the Status output pin, page 194.
12	Refer to the EmergencyStop status in the Status output pin, page 194.
13	Refer to the StandbyST status in the Status output pin, page 194.
14	Refer to the Standby status in the Status output pin, page 194.
15	Refer to the LocalRemote status in the Status output pin, page 194.

ATV6xx_CFG.ChannelStatus/ATV9xx_CFG.ChannelStatus Word Structure

Bit	Description
0	Refer to the ReservedReferenceChannel status in the Status output pin, page 194.
1	Refer to the ReservedCommandChannel status in the Status output pin, page 194.

ATV6xxx_CFG.ExtendedStatus Word Structure

Bit	Description
0	Refer to the ReservedReferenceChannel status in the Status output pin, page 194.
1	Refer to the ReservedCommandChannel status in the Status output pin, page 194.
2	Refer to the MVCBReadyToClose status in the Status output pin, page 194.
3	No command selected. Refer to CommandStatus output pin, page 194.
4	Remote command selected. Refer to CommandStatus output pin, page 194
5	Local command selected. Refer to CommandStatus output pin, page 194
6	Panel command selected. Refer to CommandStatus output pin, page 194

ATV_CFG.WarningCode Word Structure

Bit	Description
1	Refer to the Device in the WarningCode output pin, page 194.
2	Refer to the Order in the WarningCode output pin, page 194.
3	Refer to the ForcedLocalMode in the WarningCode output pin, page 194.

ATV6xx_CFG.WarningCode/ATV9xx_CFG.WarningCode/ ATV6xxx_CFG.WarningCode Word Structure

Bit	Description
1	Refer to the Device in the WarningCode output pin, page 194.
2	Refer to the Order in the WarningCode output pin, page 194.
3	Refer to the ForcedLocalMode in the WarningCode output pin, page 194.
4	Refer to ReservedReferenceChannel in the WarningCode output pin, page 194.
5	Refer to ReferenceCommandChannel in the WarningCode output pin, page 194.

Public Variables

Public Variable Description

Name	Data type	Description
CommandCtrlWindow	TIME	Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, an alarm is issued. The commands that are controlled are <i>EnableDevice</i> , <i>Run</i> , and <i>QuickStop</i> . In the event of a <i>ResetFail</i> , this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the <i>Reseting</i> output.
ScanTime	TIME	Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.
HighRangeRpm	INT	High range of the input signal in rpm. NOTE: The parameter value has to be higher than <i>LowRangeRpm</i> .
LowRangeRpm	INT	Low range of the input signal in rpm. NOTE: The parameter value has to be higher than 0.
HighRangeEngUnit	REAL	High range for the measurement in engineering units (Refer to the <i>SetPoint</i> input pin, page 186).

For operating the motor at different rpm, *LowRangeEngUnit* has to be less than the *HighRangeEngUnit*. If both the values are same, motor operates at *HighRangeEngUnit*.

NOTICE

UNEXPECTED EQUIPMENT BEHAVIOR

Do not configure *LowRangeEngUnit* to be same as *HighRangeEngUnit*.

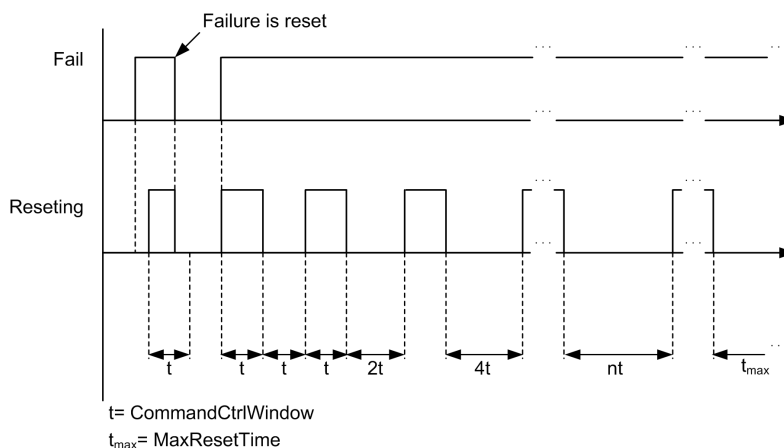
Failure to follow these instructions can result in equipment damage.

LowRangeEngUnit	REAL	Low range for the measurement in engineering units (Refer to the <i>SetPoint</i> input pin, page 186). NOTE: The parameter value has to be higher than <i>LowRangeRpm</i> .
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and device interruptions. The time defined in <i>CommandCtrlWindow</i> is used to define the interval after which a reset has to be carried out. The first reset is carried out after the time defined in <i>CommandCtrlWindow</i> elapses. The second reset is carried out after <i>CommandCtrlWindow</i> * 2 elapses, so on. If the value of <i>CommandCtrlWindow</i> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications with the device is reset with the <i>ResetFail</i> variable.
	TRUE	Communications with the device is reset automatically.

MaxResetTime	TIME	When in automatic ResetMode, this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.
ScalingFactorCurrent	REAL	Define scaling factor for current depending on drive rate.

NOTE: Public variable parameters are not applicable for all Modbus and Ethernet messaging ATV.

Timing diagram:



Diagnostics Information Management

Overview

The diagnostic codes that the device can return are read on the `FailCode` output variable.

Parameter Configuration Diagnostic Codes

This diagnostic code indicates that the function block has incorrect configuration.

- `FailCode[0]: 16#0003`
- `FailCode[1]: 16#0000`
- `FailCode[2]: 16#0004`

This diagnostic code can occur for any of the below conditions:

- Wrong array size at `Inputs` or `Outputs` pins of the function block.
- Variables connected to both `DTMInputs` and `Inputs` pins of the function Block.

During the above detected `FailCode` the function block does not process any inputs and the function blocks output displays the last processed state.

This detected `FailCode` can be reset by a rising edge to the `EnableDFB` input pin after correcting the configuration of the function block.

Ethernet Communication Diagnostic Codes

This code indicates that communications have not been established and can be reset:

- `FailCode[0]: 16#0002`

- FailCode[1]: 16#0000
- FailCode[2]: 16#0004

After the communications have been established, check Ethernet client diagnostic codes for FailCode [0] and FailCode [1]. The components make a distinction between detected read request and write request interruptions:

- FailCode[2]: 16#0001 Read
- FailCode[2]: 16#0002 Write

Diagnostic Code Example

For a detected error, the code is:

- FailCode[1]: 16#0000
- FailCode[2]: 16#0005

The FailCode[0] can have one of the following codes:

Diagnostic code	Symbol	Meaning
16#0000	nOF	No detected failure.
16#1000	CrF OLF SOF	Detected failure in capacitor pre-charge. Motor overload or Overspeed.
16#2310	OCF	Overcurrent.
16#2320	SCF	Short-circuit impedance. Power stage detected failure.
16#2330	SCF	Motor ground fault.
16#2340	SCF	Short-circuit between motor phases.
16#3110	OSF	Supply over voltage detected failure.
16#3120	USF	Low bus voltage.
16#3130	PHF	Phase dropout detected failure.
16#3310	ObF OPF	Over voltage on DC bus. Motor phase dropout.
16#4210	OHF	Drive overheating.
16#5520	EEF	Inoperable EEPROM memory.
16#6100	InF	Internal detected failure.
16#6300	CFF CFI	Incorrect (parameter) configuration or Invalid parameters.
16#7300	LFF	Inoperable 4...20 mA AI3 input.
16#7510	SLF	Modbus is not operational.
16#9000	EPF	External detected failure.
16#FF00	tnF	Inoperable auto-tuning.
16#FF01	bLF	Brake control unit is not operational.

The inoperable device can be reset if `ControlCommand` is TRUE. If this condition is not met, the detected failure can be reset on the speed driver or through the screw terminal with the correct parameter configuration.

NOTE: The diagnostic codes of the device are available for speed drives, except for those that communicate through the Advantys STB Island.

Diagnostic Code Example for MBATV212 Variable Speed Drive

The `FailCode[0]` can have one of the following codes:

Diagnostic bit	Meaning
0	Overcurrent.
1	Drive overload.
2	Motor overload.
3	Overheat.
4	Overvoltage.
5	Main circuit undervoltage.
6	Reserved.
7	Undercurrent.
8	Over-torque.
9	Reserved.
10	Cumulative operation hours reached.
11	Reserved.
12	Reserved.
13	Main circuit undervoltage alarm same as MS-relay status.
14	At the time of the instant blackout, Forced deceleration/stop.
15	An automatic stop during the lower limit frequency continuance.

Weighing Module

What's in This Part

Weighing Module Profile	216
-------------------------------	-----

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Weighing Module.

These function blocks do not reflect any specific installation.

Weighing Module Profile

What's in This Chapter

Description	216
DFB Representation	217
Inputs	217
Outputs	219
Inputs/Outputs	220
Public Variables	224
Diagnostic Information Management	225
Configuration of PMESWT Module	226
PMESWT DFB Work Flow	231

Overview

This chapter describes the DFB of Weighing module profile.

Description

General

EIPMPMESWT is a high performance weighing module controller which is used for vessels and silos weighing, dosing process and dynamic weighing application on Ethernet IP using Explicit unconnected messaging.

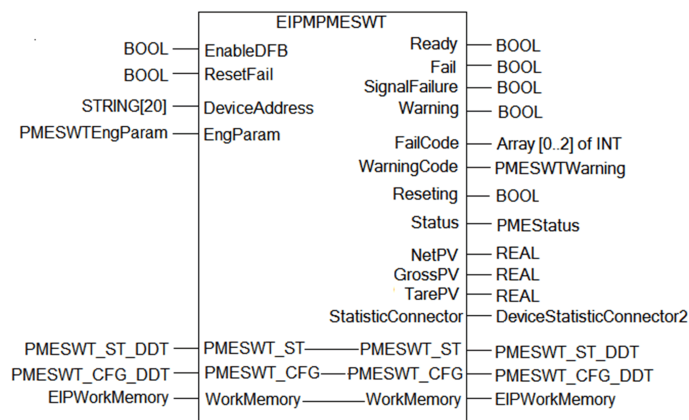
Function Description

Function	Description
Control and Monitoring	<p>When owner is program, functional commands (Tare, Cancel Tare and Zero Adjustment) are executed through user sequence DFB.</p> <p>Also it enables you to monitor the device.</p> <p>You can control the functional commands from the user sequence DFB.</p>
Device status indication	Displays the status of the device.
Remote resetting	Allows resetting of the device.
Owner	Manages the control system which is the owner (Operator or Program).
Failure management	<p>Fail output pin will be high based on the following conditions. It requires ResetFail/Reseting when failure occurred.</p> <ul style="list-style-type: none"> • Inoperable Device • Communication interruption • Configuration interruption <p>SignalFailure output pin will be high based on the following conditions. It requires ResetFail/Reseting when failure occurred.</p> <ul style="list-style-type: none"> • Fail • OOS mode

DFB Representation

Representation

The following figure represents the functional module of EIPMPMESWT.



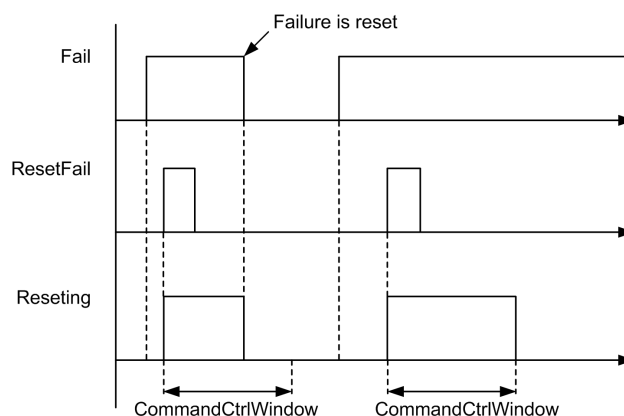
Inputs

Input Parameter Description

Parameter	Type	Description
EnableDFB	BOOL	This input enables the normal execution of the control block. 0 = The entire DFB is restarted (statuses and output values are lost) and output values are set to 0 and OOS mode is set to true. 1 = Enables communication with the devices for their operation.
ResetFail	BOOL	1 = Resets the <code>Fail</code> and <code>SignalFailure</code> output pin to 0. NOTE: <ul style="list-style-type: none"> When there is a detected configuration mismatch, output pins <code>Fail</code> and <code>SignalFailure</code> cannot be reset to 0. When Owner is in Program and no detected failure, output pins <code>Fail</code> and <code>SignalFailure</code> can be reset to 0.
DeviceAddress	STRING[20]	Device address on Ethernet IP network
EngParam	PMESWTEngParam	Engineering parameters.
	PMESWTParam	The structure holds data containing the engineering parameter that the block requires. (Constants enabled). The following table describes parameter information.
	Parameter	Type Description
	CommandCtrlWindow	TIME Control time for operations. This is the time that the block waits for the operations to be carried out by the device. If a command has been sent and the command is not executed within the time indicated by this variable, an alarm is issued. The commands that are controlled are <code>Tare</code> , <code>Cancel Tare</code> , and <code>Zero Adjustment</code> . In the event of a <code>ResetFail</code> , this is not interpreted as an alarm. Instead, the detected failure continues, and you have to reset the <code>Resetting</code> output.
	Refresh	TIME Refresh time for device data on Ethernet IP communications. NOTE: This refresh operation is carried out on read variables. Write requests are carried out when needed.
	Scantime	TIME Allows you to configure the time for which the alarm signals are kept active. Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.

Parameter	Type	Description		
	MaxResetTime	TIME	When in automatic <code>ResetMode</code> , this variable is used to define the maximum time that can elapse between 2 consecutive resets.	
	EngUnit	UINT	User defined Engineering unit. Possible values are: <ul style="list-style-type: none">0 = g1 = kg2 = t3 = oz4 = lb5 = N NOTE: <ul style="list-style-type: none">DTM calibration window allows you to configure µdef and N.m unit. This measurement unit is mainly used in strain gauge and torque based applications respectively. Since this control function will be used for weighing application, µdef and N.m unit are not taken into consideration.If user enters other than predefined engineering unit, then control function will consider the last configured engineering unit.	
EngParam	ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and device interruption. The time defined in <code>CommandCtrlWindow</code> is used to define the interval after which a reset has to be carried out. The first reset is carried out after the time defined in <code>CommandCtrlWindow</code> elapses. The second reset is carried out after <code>CommandCtrlWindow * 2</code> elapses, so on. If the value of <code>CommandCtrlWindow</code> is 0 s, its value is not used and is instead replaced with a value of 1 s. The following table describes the type of the reset:	
			Value	Description
			False	Communications with the device is reset with the <code>ResetFail</code> variable.
			True	Communications with the device is reset automatically.
	CommParam	The structure holds data containing the engineering parameter that the client block requires. The following table describes parameter information.		
	Parameter	Type	Description	
	CommFailRetries	INT	Number of retries in case of a communication interruption.	
Timeout	TIME	Time for which client will wait for a response, once the request is initiated, before moving fail state.		
InactivityTime	TIME	The period for which the client will not initiate a request after the retries are expired.		

Timing Diagram



Outputs

Output Parameter Description

Parameter	Type	Description
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out weighing functional commands (Tare, Cancel Tare and Zero Adjustment).
Fail	BOOL	1 = A detected failure in the control block or in the device or a communication interruption. To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode. NOTE: If a communication interruption occurs, the variables being read from the device cease to be refreshed as a refresh operation can no longer be carried out. The variables keep their last value.
SignalFailure	BOOL	1 = A detected failure in the control block or in the device or a communication interruption or OOS mode is enabled. NOTE: To reset the Fail output pin, the ResetFail input has to be activated. The last detected error code is shown on FailCode.
Warning	BOOL	1 = An alarm has been activated for the device. It cannot be reset because the signal remains active until the cause of the alarm is removed or until the maximum set ScanTime is reached.
FailCode	Array [0..2] of INT	When the Fail output is 1, it holds the code for the detected error. If Fail bit is 0, it indicates the last detected error that occurred. The detected error source is specified by using a 3 level structure. Refer to Diagnostic Information Management, page 225 for additional information.
WarningCode	PMESWTWarning	Holds a data structure with the alarm information currently on the device. The following table describes the WarningCode
	Parameter	Type Description
	Device	BOOL Detected warning when executing weighing functional commands (Tare, Cancel Tare and Zero Adjustment).
	Order	BOOL 1 = An alarm. The device is not responding to the command within the time specified in CommandCtrlWindow.
Reseting	BOOL	1 = A reset is being carried out. The CommandCtrlWindow variable indicates the maximum time for resetting the detected failure. When a communication reset is carried out with ResetFail, the DFB tries to reset the detected failure within the time period defined in CommandCtrlWindow. If the detected failure is reset, the Fail and Reseting output variables are reset (set to FALSE). If the detected failure is not reset, the Reseting variable is set to FALSE and the Fail variable remains TRUE. The ResetFail is edge-based.
StatisticConnector	DeviceStatisticConnector2	StatisticConnector has two client data, Read client and Write client.
	Parameter	Type Description
	ReadClient	StatisticConnector1 StatisticConnector of Read client.
	WriteClient	StatisticConnector1 StatisticConnector of Write client.
	StatisticConnector1	Information data is used with EIP communication to obtain statistics on the bus network (requests carried out and time between requests). This structure has been created for its use together with the StatisticCounter DFB in General Purpose library for communication.
	Parameter	Type Description
	Start	BOOL 1 = The operation has started.
	EndOk	BOOL 1 = The operation has ended successfully.
	EndNok	BOOL 1 = The operation has ended with a detected error.
	TotalTime	INT Total time taken for current request.
Status	PMESStatus	The structure holds data containing the information that the function block extracts from the status variable of the weighing module.

Parameter	Type	Description	
	Parameter	Type	Description
	DeviceStatus	WORD	Information on device calibration of the weighing module.
	MeasurementStatus	WORD	Measurement and device status of the weighing module.
	ResponseStatus	WORD	Response status provides the state of the command currently being processed by the device.
NetPV	REAL	Net measurement value in Engineering units.	
GrossPV	REAL	Gross measurement value in Engineering units.	
TarePV	REAL	Tare measurement value in Engineering units.	

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
PMESWT_ST	PMESWT_ST_DDT	Device data structure that holds the minimum information required for performing control and monitoring functions. The information used by the operator screen is readable/writable from the HMI or SCADA system.
PMESWT_CFG	PMESWT_CFG_DDT	Data structure with the device information. The information is used by the operator screen and is readable from the HMI or SCADA system.
Workmemory	ANY_ARRAY_INT	Memory area which contains the client reference is used for Ethernet/IP communications.

PMESWT_ST_DDT

Parameter	Type	Description
NetPV	REAL	Net measurement value in Engineering units.
GrossPV	REAL	Gross measurement value in Engineering units.
TarePV	REAL	Tare measurement value in Engineering units.
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	Device control. Provides the necessary means to control the device from the supervision in operator mode (Owner = 1).
EngUnit	UINT	Unit for Processing the measurement values in Engineering unit. Possible values are: <ul style="list-style-type: none"> • 0 = g • 1 = kg • 2 = t • 3 = oz • 4 = lb • 5 = N

PMESWT_CFG_DDT

Parameter	Type	Description
MaxCapacity	UDINT	Maximum capacity of the weighing module.
CalibratedMaxCapacity	UDINT	Last calibrated maximum capacity of the weighing module.
CalibrationLoad	UDINT	Calibration load for calibration the device.
MeasurementStatus	WORD	Measurement and device status of the weighing module.
DeviceStatus	WORD	Informations on Device calibration of the weighing module.
ResponseStatus	WORD	Response status provides the state of the command currently being processed by the device.
Warningcode	WORD	Alarm code information. Captures the values from the WarningCode output.
Failcode0	INT	Code of last level 0 detected error. Indicates that a detected error has occurred FailCode[0].
Failcode1	INT	Code of last level 1 detected error. Indicates that a detected error has occurred FailCode[1].
Failcode2	INT	Code of last level 2 detected error. Indicates that a detected error has occurred FailCode[2].
ScaleInterval	UINT	Minimal difference between two consecutive indicated values (either gross or net). Read & write access. Possible value are: 1, 2, 5, 10, 20, 50
CalibratedScaleInterval	UINT	Indicates last calibrated Scale interval parameter.
Stepcalibration	UINT	Number of the step calibration in progress. <ul style="list-style-type: none"> 1 = Calibration parameter mismatch 2 = Physical calibration done 3 = Zero acquisition done. 4 = Load - 1 acquisition done. 5 = Calibration done and parameters saved. 6 = Calibration canceled. 7 = Device not responding. <p>NOTE: In Step Calibration = 6, the last calibrated values are returned back to the device register.</p>
CalibratedCalibrationUnit	UINT	Indicates the lasts calibrated unit.
CalibrationUnit	UINT	Engineering unit which is used for calibrating the device. Read and write access. <p>Possible values are:</p> <ul style="list-style-type: none"> 0 = g 1 = kg 2 = t 3 = oz 4 = lb 5 = N

PMESWT_CFG.MeasurementStatus Structure

Bit	Description
0	Measurement is OK.
1	Gross weight is less than negative Maximum capacity.

Bit	Description
2	Gross weight is greater than maximum capacity.
3	Converter saturation or Analog signal out of A/D input range.
4	0 = Measurement is unstable. 1 = Measurement is stable.
5	Measurement out of ¼ division flag.
6	0 = Memory Ok. 1 = Detected memory failure.
7	0 = none. 1 = At least one preset tare is processed.
8	0 = Logical Input I0 state is disabled. 1 = Logical Input I0 state is enabled.
9	0 = Logical Input I1 state is disabled. 1 = Logical Input I1 state is enabled
10	0 = Logical Output S0 state is disabled. 1 = Logical Output S0 state is enabled.
11	0 = Logical Output S1 state is disabled. 1 = Logical Output S1 state is enabled.
12	0 = Logical Output S2 state is disabled. 1 = Logical Output S2 state is enabled.
13	0 = Logical Output S3 state is disabled. 1 = Logical Output S3 state is enabled
14	Tare process flag.
15	Zero/Tare process flag.

PMESWT_CFG.DeviceStatus Structure

Bit	Description
0	User dosing cycle in progress.
1	At least one user calibration is processed.
2	Physical calibration is set.
3	Theoretical calibration is set.
4	Load cell wiring mode: <ul style="list-style-type: none"> 0 = 4 wire 1 = 6 wire
5	Termination resistor is set.
6	Theoretical zero done.
7	Theoretical calibration done.
8	Zero adjustment done.
9	Span adjustment done.
10	Start physical calibration done.
11	Zero acquisition done.
12	Load 1 acquisition done
13	Load 2 acquisition done.

Bit	Description
14	Load 3 acquisition done.
15	Save calibration done.

PMESWT_CFG.ResponseStatus Structure

Bit	Description
0	Ready to accept a new command.
1	Command execution in progress.
2	Command execution complete.
3	Detected error during command execution.

PMESWT_ST.STW Structure

Bit	Description
0	Unknown state.
1	Not ready.
2	-
3	Inoperable device.
4	Detected warning.
5	Communication interruption.
6	Requires <code>Reseting</code> .
7	-
8	<code>Reseting</code> .
9	Enable DFB.
10	<code>SignalFailure</code> .
11	Stability.
12	Cancel calibration disable.
13	<code>REARM</code> enable.
14	<code>ExtCalibration</code> - If OOS not enabled & calibration parameters are modified. Then this bit is triggered to enable Accept Device Values .

PMESWT_ST.CFGW Structure

Bit	Description
0	<code>Reset Fail</code> .
1	Owner. <ul style="list-style-type: none">0 - Program1 - Operator
2	Tare command.
3	Cancel tare command.
4	Zero adjustment command.
5	Physical calibration command.

Bit	Description
6	Validation step calibration.
7	Cancel calibration command.
10	Accept new values.
15	OOS mode. Enables to determine whether the device is out of service (1) or in use (0).

Warning Bit Structure

Bit	Description
0	Refer to Device in the <code>WarningCode</code> , page 219 output pin .
1	Refer to Order in the <code>WarningCode</code> , page 219 output pin.

Public Variables

Public Variable Description

Parameter	Type	Description
SC	PMESWT_SC_DDT	Provides the frequently needed data for monitoring and controlling the DFB from sequence control.

PMESWT_SC_DDT

Parameter	Type	Description
NetPV	REAL	Net measurement value in Engineering unit.
GrossPV	REAL	Gross measurement value in Engineering unit.

Parameter	Type	Description
TarePV	REAL	Tare measurement value in Engineering unit
Properties	WORD	<p>Bits are expressed as ranked bit.</p> <ul style="list-style-type: none"> Bit 0 (Owner): Provides the data needed to configure the DFB normally from the monitoring subsystem. <ul style="list-style-type: none"> Read access 0 = Program 1 = Operator Bit 1 (OOS): 1 = Out-Of-Service mode is selected. Read access. Bit 2 (Fail): A detected failure in the control block or in the device or a communication interruption. Bit 3 (CONFIGFail): Configuration mismatch. Bit 4 (Stable): 1 = Measurement is stable, 0 = Measurement is not stable. Read only. Bit 5 (TareDone): 1 = At least one tare is processed. Read only. Bit 6 (ZeroDone): 1 = At least one zero is processed. Read only. Bit 7 (CMDReady): Ready to accept a new command. Ready only. Bit 8 (CMDInProgress): Command execution in progress. Ready only. Bit 9 (CMDComplete): Command execution complete. Ready only. Bit 10 (CMDError): Detected error during command execution. Ready only. Bit 11: (TareCMD): Execution of tare command. Read/ Write access. Bit 12 (CancelTareCMD): Execution of cancel tare command. Read/ Write access. Bit 13 (ZeroADJCMD): Execution of Zero adjustment command. Read/ Write access.

Diagnostic Information Management

Overview

The diagnostic codes the device can return are read from the `FailCode` output variable.

Parameter Configuration Diagnostic Codes

Following are the Parameter configuration diagnostic codes:

- `FAILCODE [0] : 16#0003`
- `FAILCODE [1] : 16#0000`
- `FAILCODE [2] : 16#0004`

The above mentioned parameter configuration diagnostic codes occurs due to the given below conditions:

- If user enters other than predefined engineering units through other source (DTM).
- If there is a difference in calibrated calibration parameters from device register parameters and when Out Of Service mode is not enabled.
- When Out Of Service mode is enabled to start the calibration operation.

EthernetIP Communication Diagnostic Codes

This code indicates that communications have not been established and can be reset:

- FailCode[0] : 16#0002
- FailCode[1] : 16#0000
- FailCode[2] : 16#0004

After the communications have been established, check EthernetIP client diagnostic codes for FailCode [0] and FailCode [1]. The components make a distinction between detected read request and write request interruptions:

- FailCode[2] : 16#0001 Read
- FailCode[2] : 16#0002 Write

Device Diagnostic Codes

Following are the Device diagnostic codes:

- FailCode[0] : 16#0001
- FailCode[1] : 16#0000
- FailCode[2] : 16#0005

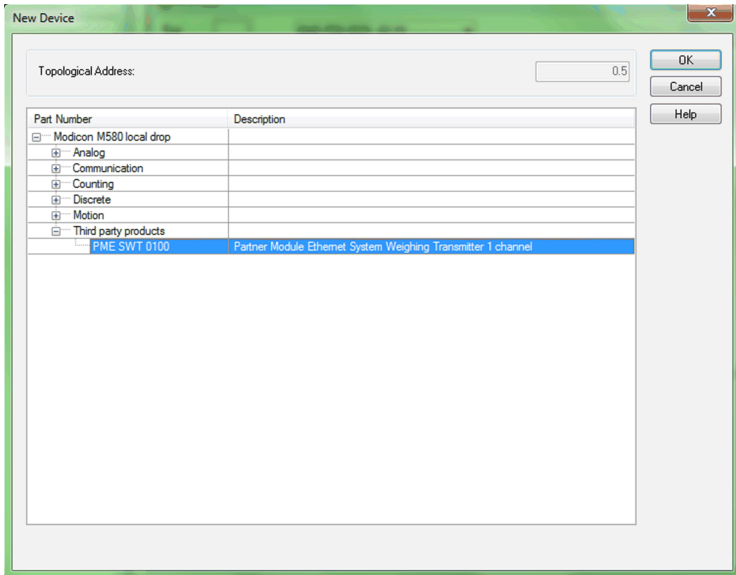
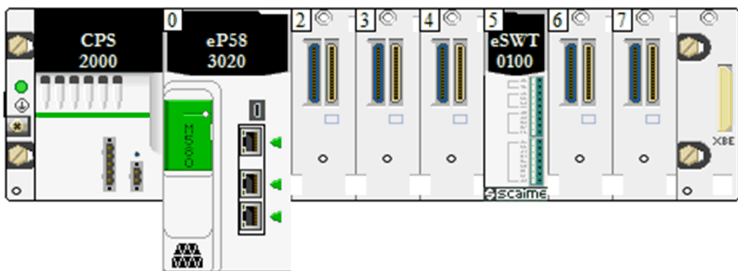
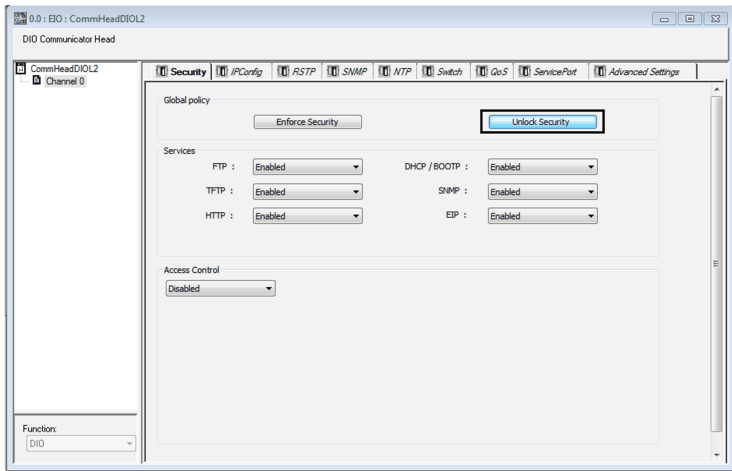
The above mentioned device diagnostic codes occurs due to the given below conditions:

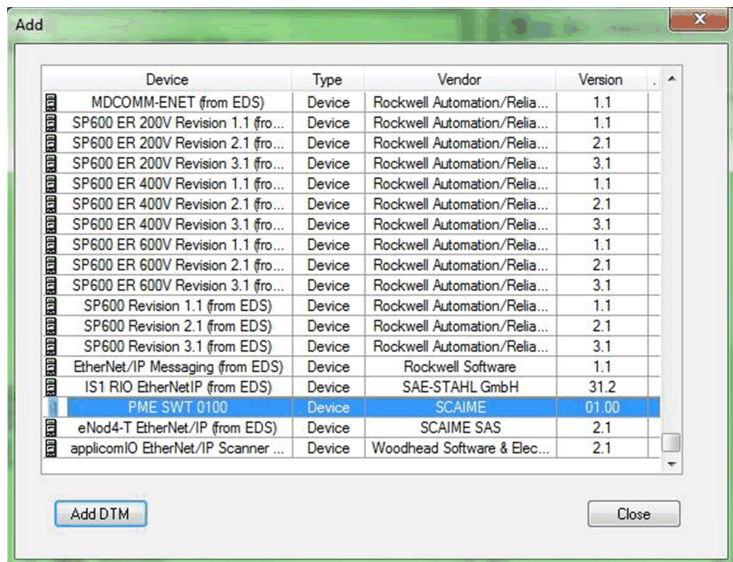
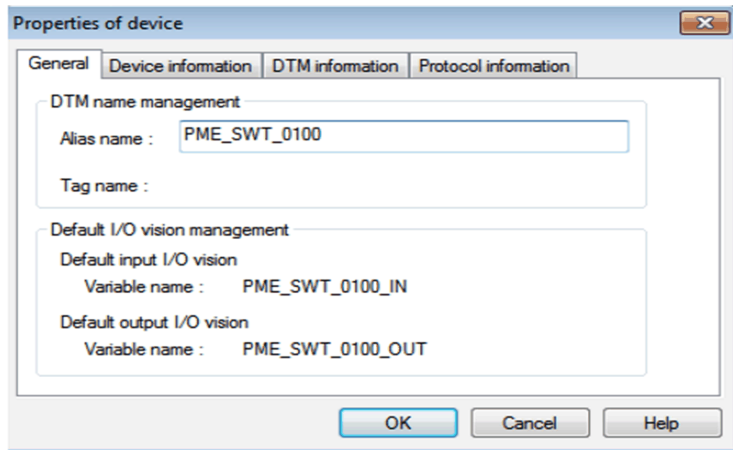
- Detected Memory failure.
- Gross weight is less than negative maximum capacity.
- Gross weight is greater than maximum capacity.
- Converter saturation or analog signal out of A/D input range.

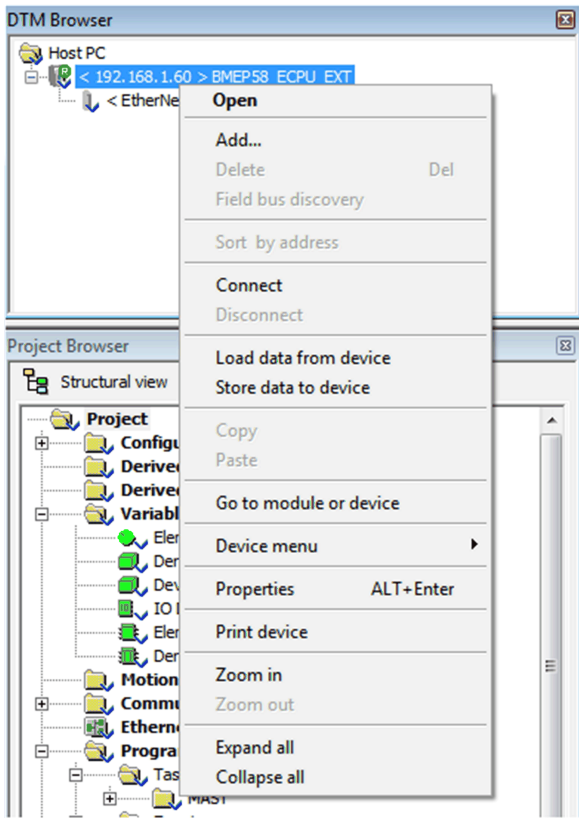
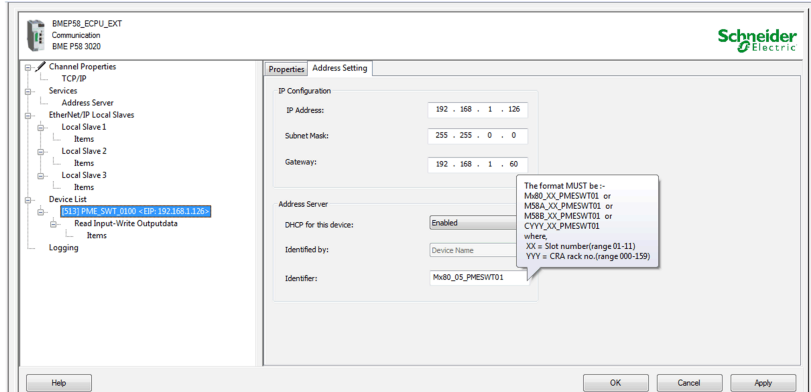
Configuration of PMESWT Module

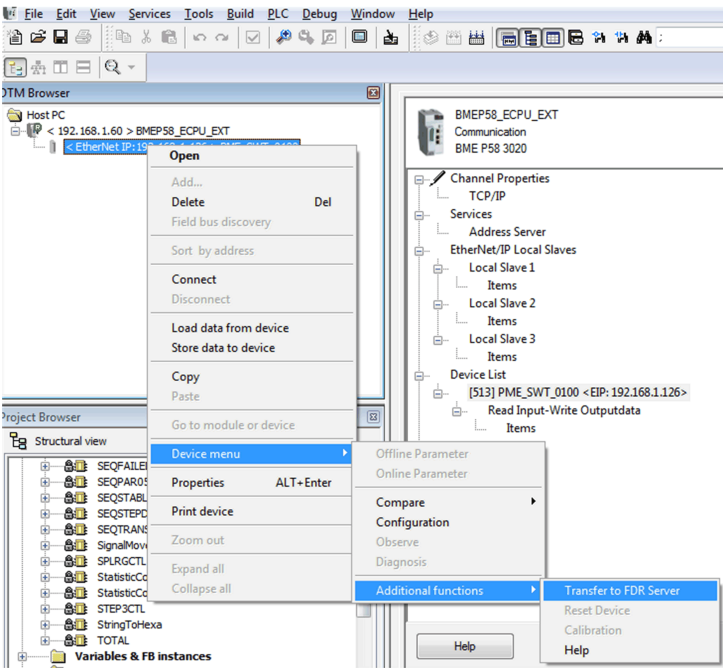

Procedure to Configure PMESWT Module on M580 Controller

Step	Action
1	Create new Control Expert project by selecting M580 CPU.
2	Add PME SWT module by right clicking on empty slot of M580 Ethernet backplane and select PME SWT module from new device window.

Step	Action
	<div data-bbox="268 197 1007 770">  </div> <div data-bbox="277 797 1018 1066">  </div> <p data-bbox="316 1081 1442 1128">NOTE: Verify that you have not selected slot number 2, 8, 10 and 11 to add PME SWT device as Ethernet Port is absent in those slots.</p>
3	<p data-bbox="268 1149 1458 1218">Navigate to PLC bus (Project Browser → Configuration → PLC bus). Click on the Ethernet Port available on the CPU which opens the port configuration. On the Security tab, click the Unlock Security button to enable FTP, TFTP and HTTP communication.</p> <div data-bbox="268 1236 1003 1704">  </div>
4	<p data-bbox="268 1727 1426 1771">Open DTM browser screen from Menu Tools → DTM Browser. Right click the master DTM in the DTM browser and select PME SWT 0100 device DTM and click Add DTM.</p>

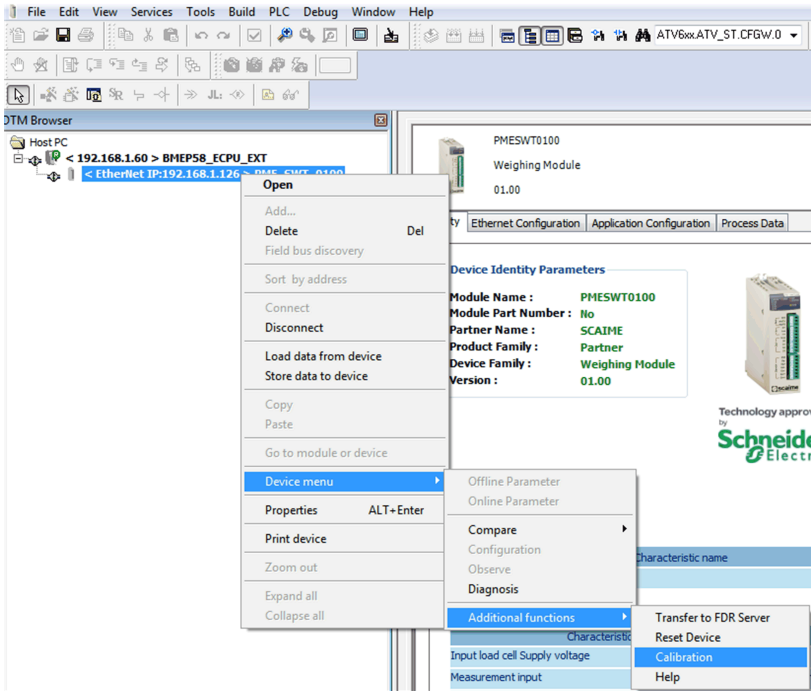
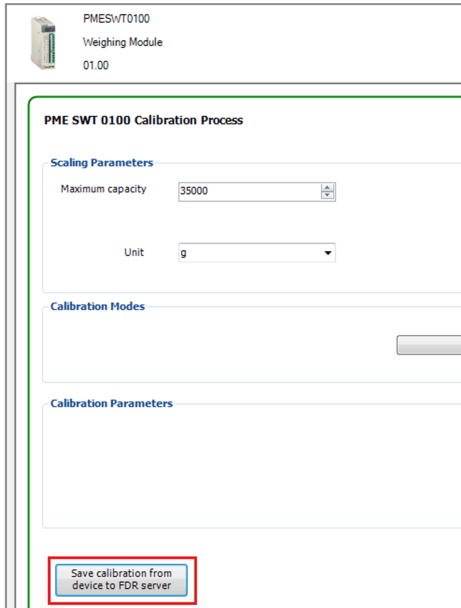
Step	Action
	<div></div>
5	<p>Provide the Alias name for the DTM of your choice and click OK on Properties of device window.</p> <div></div>
6	<p>In the DTM browser, right click on the CPU master DTM and click Open. The Master DTM properties window opens.</p>

Step	Action
	
7	<p>In the Master DTM properties window, navigate to Device List → PMESWT, select Address Setting tab to configure the device IP address and address server configuration and then click Apply.</p> 
8	Build and transfer the Control Expert configuration file to the PLC.
9	In the DTM browser, right click on the M580 CPU DTM and then select Connect .
10	In the DTM browser, right click on the PMESWT device DTM and then navigate to Device menu → Additional functions → Transfer to FDR Server .

Step	Action
	
11	<p>After successful transfer of FDR, device will become healthy and the following message window displays.</p> 
12	<p>To verify that the device is connected or not, In the DTM browser, right click on the PMESWT device DTM and select Connect. Now, the IP address and name of PMESWT module appears in bold lettering, indicating both devices are connected.</p>

Procedure to save Calibration from Device to FDR Server

It is recommended to perform the below steps at the end of each calibration from Supervision.

Step	Action
1	<p>In the DTM browser, right click on the PMESWT device DTM and then navigate to Device menu → Additional functions → Calibration.</p> 
2	<p>In the DTM calibration window, click Save calibration from device to FDR server.</p> 

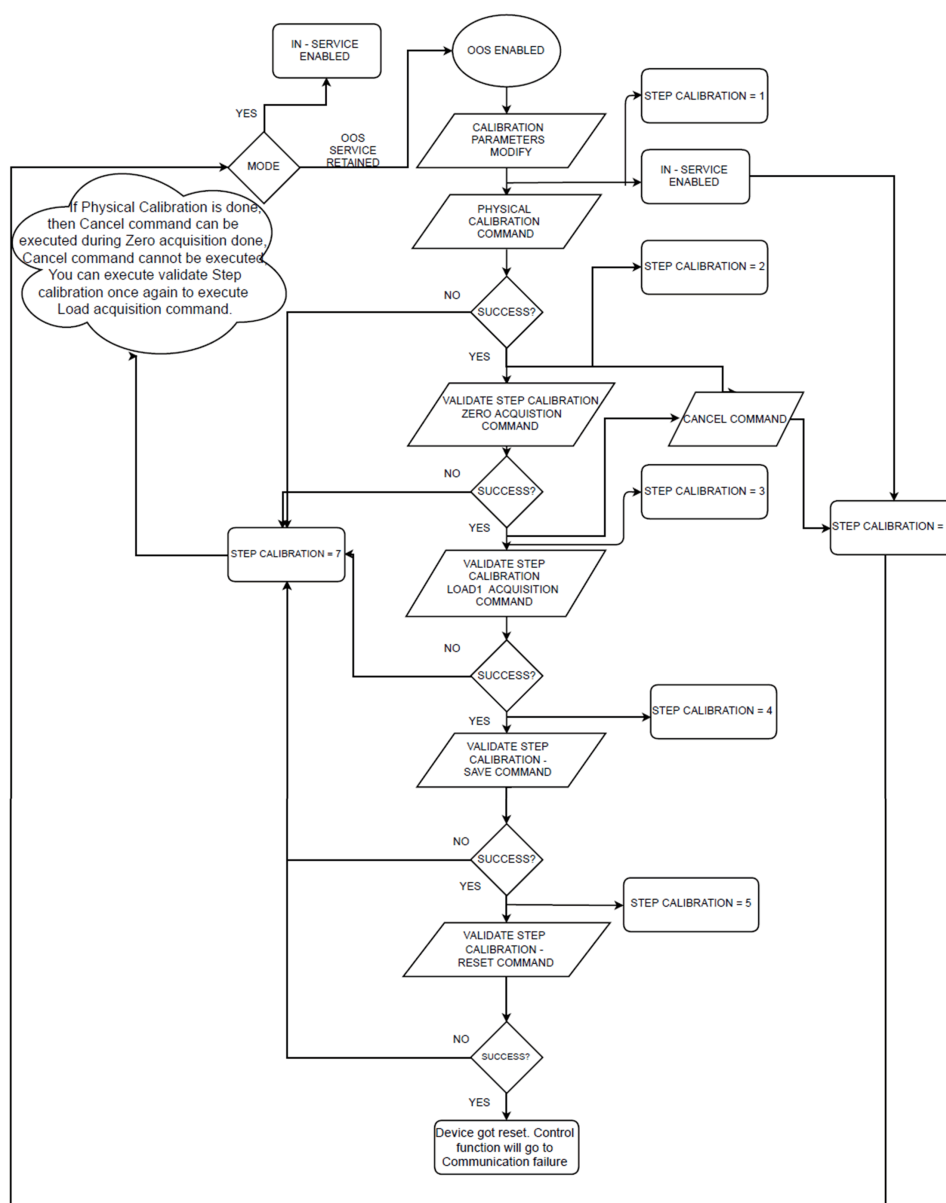
PMESWT DFB Work Flow

Using the Control Function Block

Step	Action
1	During rising edge of <code>EnableDFB</code> or first download, control function is set to Owner (Program) and Mode (In-Service).
2	Once the device is ready for operation, control function reads the measurement values and calibration parameters from the device and process the EU conversion accordingly.

Step	Action
3	If you want to perform calibration, then you need to set Owner (Operator) and Mode (Out Of Service). For execution of calibration commands refer Weighing Module Calibration Commands Flow Chart, page 232.
4	For execution of functional commands refer Weighing Module Functional Commands Flow Chart, page 233.
5	<p>During operation, if you perform calibration from external source (DTM), then control function will generate detected configuration failure because the last calibrated value and new value read from device register are different (calibration parameter mismatch). In such scenario, if you have performed calibration through external source (DTM), then Accept Device Values button in the calibration faceplate gives flexibility to accept the calibrated values. Accept Device Values button is enabled only when there is calibrated parameter mismatch (detected configuration failure).</p> <p>You can clear this detected failure by clicking Accept Device Values button or start calibration procedure from the faceplate, by taking the object in Out Of Service mode.</p>

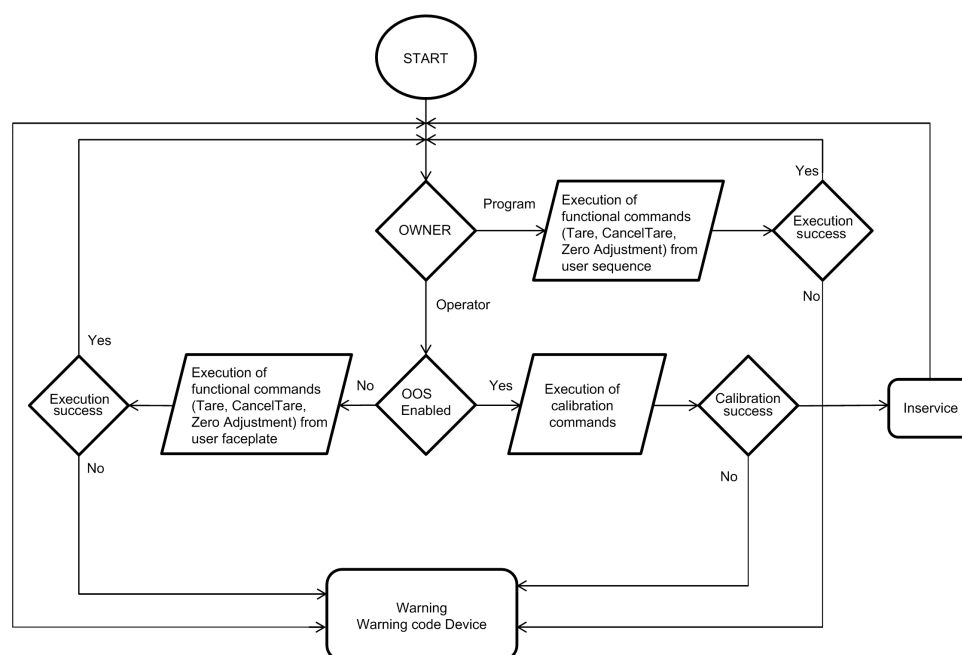
Weighing Module Calibration Commands Flow Chart



NOTE:

- Initially you have to configure the weighing module through **DTM**.
- It is suggested to perform the calibration from the faceplate only.
- Before the start of calibration you need to verify that calibration parameters are entered properly.
- In Load acquisition state (after zero acquisition done) during calibration you need to verify that a physical load is placed on the load cell.
- At the end of each calibration, once the device reset is performed and before giving `ResetFail` you need to verify that the physical device is healthy.
- In Step Calibration = 6, the last calibrated values are returned back to the device register.

Weighing Module Functional Commands Flow Chart



Safety Module


What's in This Part

MBXPSMC - XPSMC Safety Module (Modbus) 235

Overview

This part provides the detailed description, pin layout, pin description, operator screen of the device control blocks of the Safety Module family.

These function blocks do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it 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.

MBXPSMC - XPSMC Safety Module (Modbus)

What's in This Chapter

DFB Representation	235
Inputs	236
Outputs	237
Inputs/Outputs	242

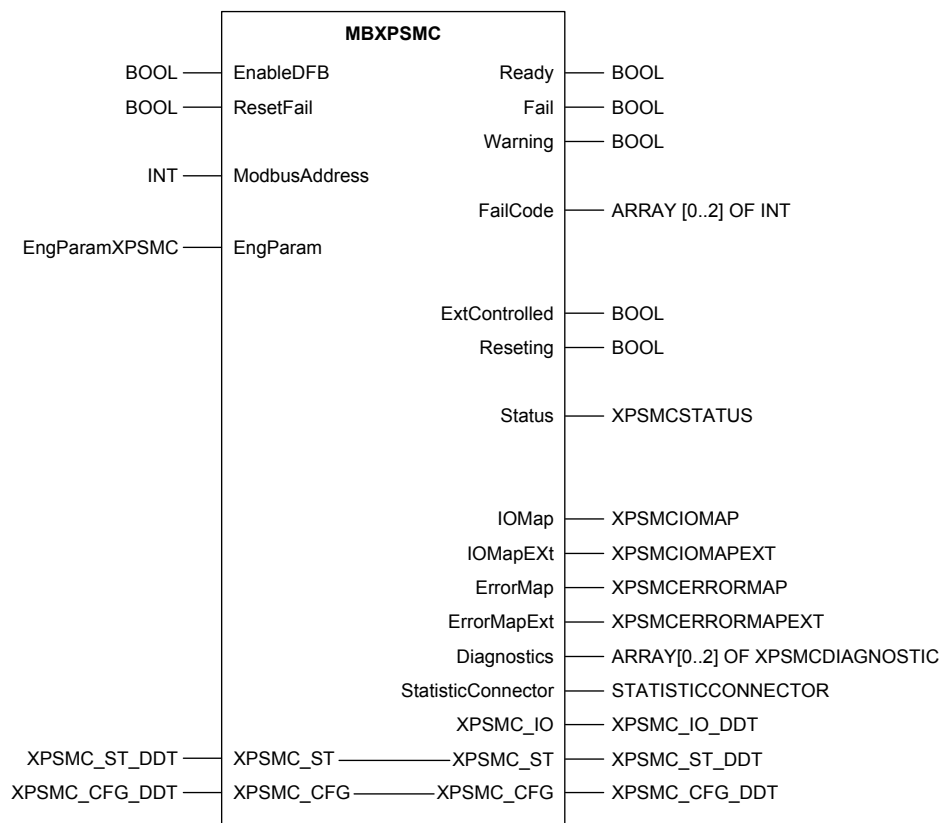
Overview

This chapter describes the MBXPSMC DFB.

DFB Representation

Representation

The following figure represents MBXPSMC DFB in Control:



Operator Screen Representation

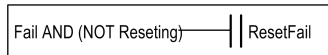
The component includes an operator screen that enables interaction with the communication interface blocks.

The operator screen data is used if you are handling a monitoring system by using the following:

- XPSMC_ST_DDT and XPSMC_CFG_DDT inputs/outputs structures.
- XPSMC_IO output variable depends on the specific component.

Inputs

Input Parameter Description

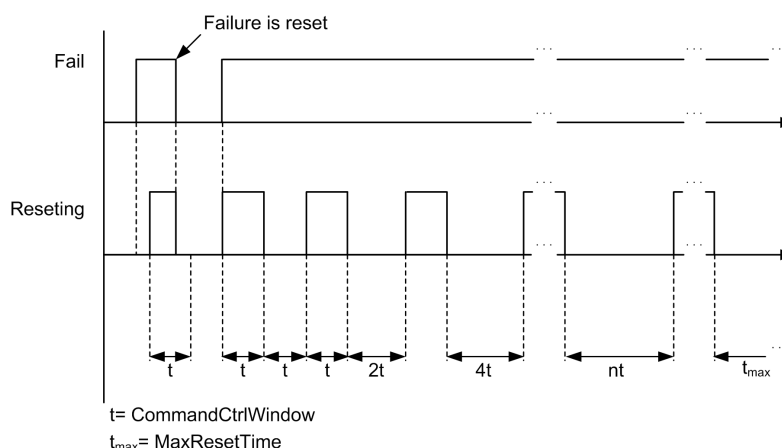
Name	Data type	Description
EnabledDFB	BOOL	<p>This input enables the normal execution of the control block.</p> <ul style="list-style-type: none"> 0 = The entire DFB is restarted (states, output values, counters are lost) and output values are set to 0. 1 = Enables communications with the devices for their operation.
ResetFail	BOOL	<p>1 = Resets the <code>Fail</code> output parameter to 0 or in case of inoperable device, sends a reset command to the device if <code>ControlCommand</code> is 1.</p> <p>The following figure shows the <code>ResetFail</code> input:</p>  <pre> graph LR A[Fail AND (NOT Resetting)] --> B[ResetFail] </pre>
Inputs	ARRAY	<p>Holds an array structure with data obtained from the device. You can control the starter/controller with this input variable. This input is reserved for the DFB, and you cannot use this input directly. For the control block to work properly, allocate the structure (%MWx). Refer to the <i>Communications Technologies</i>, page 247.</p> <p>Based on the device and the relevant communications, the following structure is used.</p> <p>The information available for controllers on an XPSMC reader is shown in the following table:</p>
	Parameter	Type
	Inputs [0]	INT
	Inputs [1]	INT
	Inputs [2]	INT
	Inputs [3]	INT
ModbusAddress	INT	Device address within the Modbus network.
EngParam	EngParamXPSMC, page 236	Engineering Parameters

EngParamXPSMC

Name	Data type	Description
Refresh	TIME	Time to refresh the device data.
CommandCtrlWindow	TIME	<p>Control time for operations. The block waits for the operations to be carried out by the device during this time.</p> <p>If a command has been sent and the command is not executed within the time indicated by this variable, a follow-up alarm is issued.</p> <p>The command that is controlled is <code>Run</code>. In case of <code>ResetFail</code>, this is not interpreted as an alarm. Instead, the detected failure continues and you have to reset the <code>Resetting</code> output.</p>
ScanTime	TIME	<p>Allows to configure the time during which the alarm signals are kept active.</p> <p>Helps the monitoring subsystem to acquire the data for the alarms that are automatically reset.</p>

Name	Data type	Description
ResetMode	BOOL	Enables to configure the type of reset. This type of reset is used for communication interruption and inoperable device.
		The time defined in <code>CommandCtrlWindow</code> is used to define the interval after which a reset needs to be carried out. The first reset is carried out after the time defined in <code>CommandCtrlWindow</code> elapses. The second reset is carried out after <code>CommandCtrlWindow * 2</code> elapses, and so on. If the value of <code>CommandCtrlWindow</code> is 0 s, its value is not used and is instead replaced with a value of 1 s.
		The following table describes the type of the reset:
	Variable value	Description
	FALSE	Communications are/the device is reset with the <code>ResetFail</code> variable.
	TRUE	Communications are/the device is reset automatically.
MaxResetTime	TIME	When in automatic <code>ResetMode</code> , this variable is used to define the maximum time that can elapse between 2 consecutive resets. Refer to the Timing diagram below.

Timing diagram:



Outputs

Output Parameter Description

Output	Type	Description
Ready	BOOL	1 = The device is enabled and free of detected errors. The device is ready to carry out or carrying out any Run or Stop command.
Fail	BOOL	1 = A detected failure in the control block or in the device or communication interruption. To reset the <code>Fail</code> output pin, the <code>ResetFail</code> input has to be activated. The last detected error code is shown on <code>FailCode</code> .
Warning	BOOL	1 = An alarm has been activated for the device. It does not affect the block operation and does not need to be reset. The signal remains active until the cause for the alarm disappears.
FailCode	ARRAY [0..2] OF INT	When <code>Fail</code> output is 1, this variable holds the code for the detected error. If <code>Fail</code> output is 0, it indicates the last detected detected error that occurred. The detected error source is specified by a three-level structure.

Output	Type	Description
ExtControlled	BOOL	<p>1 = The device is being controlled from an external source (for example, from the console, from a push-button panel, or from the monitoring system) to the system.</p> <p>Provides information for programming.</p> <p>NOTE: The <code>ControlCommand</code> signal, the <code>Owner</code> variable, and the <code>ForcedLocalMode</code> status are used to activate this signal. You cannot use this signal as a <code>ControlCommand</code> input.</p>
Reseting	BOOL	<p>1 = A reset is being carried out.</p> <p>The <code>CommandCtrlWindow</code> variable indicates the maximum time for resetting the detected failure.</p> <p>When a device or communication reset is carried out with <code>ResetFail</code>, the DFB tries to reset the detected failure within the time period defined in <code>CommandCtrlWindow</code>.</p> <p>If the detected failure is reset, the <code>Fail</code> and <code>Reseting</code> output variables are reset (set to FALSE). Alternatively, if the detected failure is not reset, the <code>Reseting</code> variable is set to FALSE and the <code>Fail</code> variable remains TRUE. The <code>ResetFail</code> is edge-based.</p> <p>Refer to the Timing diagram.</p>
Status	XPSMCSTATUS	<p>Holds data structure containing the information that the module extracts from the status variable of XPSMC control blocks.</p> <p>The following table describes the status information:</p>
	Parameter	Type
	Run	BOOL
	LoadingConfiguration	BOOL
	InternalError	BOOL
	ExternalError	BOOL
	Stop	BOOL
	FromRunToStop	BOOL
	RebootSwitch	BOOL
	DeviceActive	BOOL
	XPSMC	<p>BOOL</p> <p>1 = The device is an XPSMC16. 0 = The device is an XPSMC32.</p>
	Checking	BOOL
	ConfigurationOK	BOOL
	OrderStop	BOOL
IOMap	XPSMCIOMAP	<p>Holds a data structure with information on the state of the inputs and outputs of the controller.</p> <p>The following table describes the IOMap:</p>
	Parameter	Type
	I1	BOOL
	I2	BOOL
	I3	BOOL

Output	Type	Description	
	I4	BOOL	1 = The state of the digital input I4.
	I5	BOOL	1 = The state of the digital input I5.
	I6	BOOL	1 = The state of the digital input I6.
	I7	BOOL	1 = The state of the digital input I7.
	I8	BOOL	1 = The state of the digital input I8.
	I9	BOOL	1 = The state of the digital input I9.
	I10	BOOL	1 = The state of the digital input I10.
	I11	BOOL	1 = The state of the digital input I11.
	I12	BOOL	1 = The state of the digital input I12.
	I13	BOOL	1 = The state of the digital input I13.
	I14	BOOL	1 = The state of the digital input I14.
	I15	BOOL	1 = The state of the digital input I15.
	I16	BOOL	1 = The state of the digital input I16.
	O1	BOOL	1 = The state of the digital output O1.
	O2	BOOL	1 = The state of the digital output O2.
	O3	BOOL	1 = The state of the digital output O3.
	O4	BOOL	1 = The state of the digital output O4.
	O5	BOOL	1 = The state of the digital output O5.
	O6	BOOL	1 = The state of the digital output O6.
	R1	BOOL	1 = The state of the digital output R1.
	R2	BOOL	1 = The state of the digital output R2.
IOMapExt	XPSMCIOMAPEXT	Standard inputs/outputs information of the controllers. The following table describes the IOMapExt:	
	Parameter	Type	Description
	I17	BOOL	1 = The state of the digital input I17.
	I18	BOOL	1 = The state of the digital input I18.
	I19	BOOL	1 = The state of the digital input I19.
	I20	BOOL	1 = The state of the digital input I20.
	I21	BOOL	1 = The state of the digital input I21.
	I22	BOOL	1 = The state of the digital input I22.
	I23	BOOL	1 = The state of the digital input I23.
	I24	BOOL	1 = The state of the digital input I24.
	I25	BOOL	1 = The state of the digital input I25.
	I26	BOOL	1 = The state of the digital input I26.
	I27	BOOL	1 = The state of the digital input I27.
	I28	BOOL	1 = The state of the digital input I28.
	I29	BOOL	1 = The state of the digital input I29.
	I30	BOOL	1 = The state of the digital input I30.
	I31	BOOL	1 = The state of the digital input I31.
	I32	BOOL	1 = The state of the digital input I32.
ErrorMap	XPSMCERROR-MAP	Holds a data structure with information on the detected errors of the inputs and outputs of the controller.	

Output	Type	Description	
		The following table describes the <code>ErrorMap</code> :	
	Parameter	Type	Description
	I1	BOOL	1 = Signals a detected error of the digital input I1.
	I2	BOOL	1 = Signals a detected error of the digital input I2.
	I3	BOOL	1 = Signals a detected error of the digital input I3.
	I4	BOOL	1 = Signals a detected error of the digital input I4.
	I5	BOOL	1 = Signals a detected error of the digital input I5.
	I6	BOOL	1 = Signals a detected error of the digital input I6.
	I7	BOOL	1 = Signals a detected error of the digital input I7.
	I8	BOOL	1 = Signals a detected error of the digital input I8.
	I9	BOOL	1 = Signals a detected error of the digital input I9.
	I10	BOOL	1 = Signals a detected error of the digital input I10.
	I11	BOOL	1 = Signals a detected error of the digital input I11.
	I12	BOOL	1 = Signals a detected error of the digital input I12.
	I13	BOOL	1 = Signals a detected error of the digital input I13.
	I14	BOOL	1 = Signals a detected error of the digital input I14.
	I15	BOOL	1 = Signals a detected error of the digital input I15.
	I16	BOOL	1 = Signals a detected error of the digital input I16.
	C1	BOOL	1 = Signals a detected error of the digital output C1.
	C2	BOOL	1 = Signals a detected error of the digital output C2.
	C3	BOOL	1 = Signals a detected error of the digital output C3.
	C4	BOOL	1 = Signals a detected error of the digital output C4.
	C5	BOOL	1 = Signals a detected error of the digital output C5.
	C6	BOOL	1 = Signals a detected error of the digital output C6.
	C7	BOOL	1 = Signals a detected error of the digital output C7.
	C8	BOOL	1 = Signals a detected error of the digital output C8.
ErrorMapExt	XPSMCERRORMA-PEXT	Holds a data structure with information on the extended detected errors of the inputs and outputs of the controller. The following table describes the <code>ErrorMapExt</code> :	
	Parameter	Type	Description
	I17	BOOL	1 = Signals a detected error of the digital input I17.
	I18	BOOL	1 = Signals a detected error of the digital input I18.
	I19	BOOL	1 = Signals a detected error of the digital input I19.
	I20	BOOL	1 = Signals a detected error of the digital input I20.
	I21	BOOL	1 = Signals a detected error of the digital input I21.
	I22	BOOL	1 = Signals a detected error of the digital input I22.
	I23	BOOL	1 = Signals a detected error of the digital input I23.
	I24	BOOL	1 = Signals a detected error of the digital input I24.
	I25	BOOL	1 = Signals a detected error of the digital input I25.
	I26	BOOL	1 = Signals a detected error of the digital input I26.
	I27	BOOL	1 = Signals a detected error of the digital input I27.
	I28	BOOL	1 = Signals a detected error of the digital input I28.

Output	Type	Description	
	I29	BOOL	1 = Signals a detected error of the digital input I29.
	I30	BOOL	1 = Signals a detected error of the digital input I30.
	I31	BOOL	1 = Signals a detected error of the digital input I31.
	I32	BOOL	1 = Signals a detected error of the digital input I32.
Diagnostics	ARRAY[0..2] OF XPSMCDIAGNOSTIC	Advanced diagnostics of the device.	
XPSMC_IO	XPSMC_IO_DDT	Standard inputs/outputs information of the controllers for the controller base (used by HMI/SCADA). The following table describes the XPSMC_IO:	
	Parameter	Type	Description
	InputsMapExt	WORD	Extended input status on the controller base. Refer to the InputsMapExt table below.
	InputsErrorMapExt	WORD	Status of the detected errors in the extended input on the controller base. Refer to the InputsErrorMapExt table below.
StatisticConnector	STATISTICCONNECTOR	Information data is used with Modbus communications to obtain statistics on the Modbus network (requests carried out and time between requests). This structure has been created for its use together with the StatisticCounter DFB in General Purpose Library for communication. The following table describes STATISTICCONNECTOR:	
	Parameter	Type	Description
	Start	BOOL	1 = The operation has started.
	EndOK	BOOL	1 = The operation has ended correctly.
	EndNOK	BOOL	1 = The operation has ended with a detected error.
	PartialTime	DINT	Partial time.

InputMapExt

The following table describes the InputMapExt variable:

Bit	Description
0	Status of the extended input I17 on a controller base.
1	Status of the extended input I18 on a controller base.
2	Status of the extended input I19 on a controller base.
3	Status of the extended input I20 on a controller base.
4	Status of the extended input I21 on a controller base.
5	Status of the extended input I22 on a controller base.
6	Status of the extended input I23 on a controller base.
7	Status of the extended input I24 on a controller base.
8	Status of the extended input I25 on a controller base.
9	Status of the extended input I26 on a controller base.
10	Status of the extended input I27 on a controller base.
11	Status of the extended input I28 on a controller base.
12	Status of the extended input I29 on a controller base.
13	Status of the extended input I30 on a controller base.

Bit	Description
14	Status of the extended input I31 on a controller base.
15	Status of the extended input I32 on a controller base.

InputErrorMapExt

The following table describes the `InputErrorMapExt` variable:

Bit	Description
0	Status of the detected error in the extended input I17 on a controller base.
1	Status of the detected error in the extended input I18 on a controller base.
2	Status of the detected error in the extended input I19 on a controller base.
3	Status of the detected error in the extended input I20 on a controller base.
4	Status of the detected error in the extended input I21 on a controller base.
5	Status of the detected error in the extended input I22 on a controller base.
6	Status of the detected error in the extended input I23 on a controller base.
7	Status of the detected error in the extended input I24 on a controller base.
8	Status of the detected error in the extended input I25 on a controller base.
9	Status of the detected error in the extended input I26 on a controller base.
10	Status of the detected error in the extended input I27 on a controller base.
11	Status of the detected error in the extended input I28 on a controller base.
12	Status of the detected error in the extended input I29 on a controller base.
13	Status of the detected error in the extended input I30 on a controller base.
14	Status of the detected error in the extended input I31 on a controller base.
15	Status of the detected error in the extended input I32 on a controller base.

Inputs/Outputs

Input/Output Parameter Description

Parameter	Type	Description
XPSMC_ST	XPSMC_ST_DDT	Device data structure holds the minimum information required for performing control and monitor functions. The information used by the operator screen is readable/writable from the HMI/SCADA system.
XPSMC_CFG	XPSMC_CFG_DDT	Data structure with device information. The information used by the operator screen is readable from the HMI/SCADA system.

XPSMC_ST_DDT Type

Name	Type	Description
STW	WORD	Provides the device status. Access to the data held in this bit word is read-only.
CFGW	WORD	<p>Device control.</p> <p>Provides the data to control the device from the monitoring subsystem or from the operator screen if <code>Owner</code> (1), or only from the monitoring subsystem if <code>Owner</code> (0).</p> <p>If <code>Owner</code> is 0, it takes the input variables of the DFB as a value for reading from the HMI/SCADA system.</p>

XPSMC_CFG_DDT Type

Name	Type	Description
DataStatus	WORD	Information on the device status.
InputsMap	WORD	Status of the standard inputs.
OutputMaps	WORD	Status of the outputs.
InputErrorMaps	WORD	Status of the detected errors in the standard inputs.
OutputErrorMaps	WORD	Status of the found detected errors in the outputs.
DiagnosticIndex1	INT	Diagnostic device number 1.
DiagnosticIndex2	INT	Diagnostic device number 2.
DiagnosticIndex3	INT	Diagnostic device number 3.
DiagnosticMessage1	INT	Diagnostic message 1.
DiagnosticMessage2	INT	Diagnostic message 2.
DiagnosticMessage3	INT	Diagnostic message 3.
FailCode0	INT	Code of last level 0 detected error. Indicates which detected error has occurred.
FailCode1	INT	Code of last level 1 detected error. Indicates which detected error has occurred.
FailCode2	INT	Code of last level 2 detected error. Indicates which detected error has occurred.

XPSMC_ST.STW Word Structure

Bit	Description
0	Unknown device status or communication interruption. No variable refreshing.
1	Not ready.
4	Alarm on the device or repetitive detected fault alarm requires resetting.
5	Communication interruption.
6	Requires resetting. <code>ResetFail</code> is required.
9	Refer to the <code>EnableDFB</code> input pin, page 236.

XPSMC_ST.CFGW Word Structure

Bit	Description
0	Refer to the <code>ResetFail</code> input pin, page 236.
1	Owner.

NOTE: The `Owner` bit enables to control the block from the `XPSMC_ST_DDT` input/output structure ignoring the input signals of the block. It enables control from a monitoring system (HMI, SCADA, operator screen) in the Manual mode without using the programmed switching operation.

XPSMC_CFG.DataStatus Word Structure

Bit	Description
0	Refer to the <code>Run</code> in the <code>Status</code> output pin, page 237.
1	Refer to the <code>LoadingConfiguration</code> in the <code>Status</code> output pin, page 237.
2	Refer to the <code>InternalError</code> in the <code>Status</code> output pin, page 237.
3	Refer to the <code>ExternalError</code> in the <code>Status</code> output pin, page 237.
4	Refer to the <code>Stop</code> in the <code>Status</code> output pin, page 237.
5	Refer to the <code>FromRunToStop</code> in the <code>Status</code> output pin, page 237.
6	Refer to the <code>RebootSwitch</code> in the <code>Status</code> output pin, page 237.
7	Refer to the <code>DeviceActive</code> in the <code>Status</code> output pin, page 237.
8	Refer to the <code>XPSMC</code> in the <code>Status</code> output pin, page 237.
9	Refer to the <code>Checking</code> in the <code>Status</code> output pin, page 237.
10	Refer to the <code>ConfigurationOK</code> in the <code>Status</code> output pin, page 237.
11	Refer to the <code>OrderStop</code> in the <code>Status</code> output pin, page 237.

XPSMC_CFG.InputsMap Word Structure

Bit	Description
0	Status of the standard input <code>I1</code> on a controller base.
1	Status of the standard input <code>I2</code> on a controller base.
2	Status of the standard input <code>I3</code> on a controller base.
3	Status of the standard input <code>I4</code> on a controller base.
4	Status of the standard input <code>I5</code> on a controller base.
5	Status of the standard input <code>I6</code> on a controller base.
6	Status of the standard input <code>I7</code> on a controller base.
7	Status of the standard input <code>I8</code> on a controller base.
8	Status of the standard input <code>I9</code> on a controller base.
9	Status of the standard input <code>I10</code> on a controller base.
10	Status of the standard input <code>I11</code> on a controller base.
11	Status of the standard input <code>I12</code> on a controller base.
12	Status of the standard input <code>I13</code> on a controller base.
13	Status of the standard input <code>I14</code> on a controller base.

Bit	Description
14	Status of the standard input I15 on a controller base.
15	Status of the standard input I16 on a controller base.

XPSMC_CFG.OutputsMap Word Structure

Bit	Description
0	Status of the output O1 on a controller base.
1	Status of the output O2 on a controller base.
2	Status of the output O3 on a controller base.
3	Status of the output O4 on a controller base.
4	Status of the output O5 on a controller base.
5	Status of the output O6 on a controller base.
6	Status of the output R1 on a controller base.
7	Status of the output R2 on a controller base.

XPSMC_CFG.InputsErrorMap Word Structure

Bit	Description
0	Status of the detected errors in the standard input I1 on a controller base.
1	Status of the detected errors in the standard input I2 on a controller base.
2	Status of the detected errors in the standard input I3 on a controller base.
3	Status of the detected errors in the standard input I4 on a controller base.
4	Status of the detected errors in the standard input I5 on a controller base.
5	Status of the detected errors in the standard input I6 on a controller base.
6	Status of the detected errors in the standard input I7 on a controller base.
7	Status of the detected errors in the standard input I8 on a controller base.
8	Status of the detected errors in the standard input I9 on a controller base.
9	Status of the detected errors in the standard input I10 on a controller base.
10	Status of the detected errors in the standard input I11 on a controller base.
11	Status of the detected errors in the standard input I12 on a controller base.
12	Status of the detected errors in the standard input I13 on a controller base.
13	Status of the detected errors in the standard input I14 on a controller base.
14	Status of the detected errors in the standard input I15 on a controller base.
15	Status of the detected errors in the standard input I16 on a controller base.

XPSMC_CFG.OutputsErrorMap Word Structure

Bit	Description
0	Status of the detected errors in the output C1 on a controller base.
1	Status of the detected errors in the output C2 on a controller base.

Bit	Description
2	Status of the detected errors in the output C3 on a controller base.
3	Status of the detected errors in the output C4 on a controller base.
4	Status of the detected errors in the output C5 on a controller base.
5	Status of the detected errors in the output C6 on a controller base.
6	Status of the detected errors in the output C7 on a controller base.
7	Status of the detected errors in the output C8 on a controller base.


Communication Technologies

What's in This Part

Ethernet Technology.....	248
Modbus over ULP Technology.....	282
Modbus Technology	287
Profibus Technology	308
CANOpen Technology	312

Overview

This part describes the communication technologies used in the Devices library.
These function blocks do not reflect any specific installation.

 **WARNING**

LOSS OF CONTROL

- Perform a Failure Mode and Effects Analysis (FMEA) of your application, and apply preventive and detective controls before implementation.
- Provide a fallback state for undesired control events or sequences.
- Provide separate or redundant control paths wherever required.
- Supply appropriate parameters, particularly for limits.
- Review the implications of transmission delays and take actions to mitigate.
- Review the implications of communication link interruptions and take actions to mitigate.
- Provide independent paths for control functions (for example, emergency stop, over-limit conditions, and fault conditions) according to the safety analysis and applicable codes, and regulations.
- Apply local accident prevention and safety regulations and guidelines. ¹
- Test each implementation of this library for proper operation before placing it 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.

Ethernet Technology

What's in This Chapter

<i>CommunicationOK</i> Variable	248
ATV71/61 Control Block on Ethernet with IO Scanning	249
Tesys T Control Block on Ethernet	256
Modbus Devices Used Through Gateway	259
Modbus TCPIP - Implicit with NOE	264
Modbus TCPIP - Implicit with NOC	268
Modbus TCPIP - Explicit	273
Modbus TCPIP - Explicit with Gateway	278

Overview

This chapter describes configuration details of Ethernet technology on various devices.

CommunicationOK Variable

Overview

This section provides the information about the *CommunicationOK* variable.

Description

IODDT Variable Description

To check the presence of a device on an Ethernet network, you can check the IODDT variable for Ethernet communications.

NOTE: The variable is pre-defined by the manufacturer, which includes input and output language objects that are part of the channel corresponding to a module used for a specific function.

This table describes Ethernet communication IODDTs that depend on the type of hardware installed:

SI.NO	IODDT Name	Module Name	Platform
1	T_COM_ETH_BMX	—	M340 Platform

System Word Description

The System Word is used instead of IODDT variable in Quantum platform device on an Ethernet network. For example, %SW160 and %SW167.

This figure shows the IO Scanning rows on Ethernet communication:

Name	Type	Address	Value	Comment
Ethernet	T_COM_ETHCOPRO	%ch0.0.3		
CH_ERROR	BOOL	%i0.0.3.ERR		Channel error
REFRESH_IO_1	BOOL	%i0.0.3.1.0		Remote IO scanning refresh indicator : device 1
REFRESH_IO_2	BOOL	%i0.0.3.1.1		Remote IO scanning refresh indicator : device 2
REFRESH_IO_3	BOOL	%i0.0.3.1.2		Remote IO scanning refresh indicator : device 3
REFRESH_IO_4	BOOL	%i0.0.3.1.3		Remote IO scanning refresh indicator : device 4
REFRESH_IO_5	BOOL	%i0.0.3.1.4		Remote IO scanning refresh indicator : device 5

Declaring a variable as an IODDT enables to obtain an indicator of the communication status for each IOScanning row.

NOTE: Templates generated with an Ethernet communications network by the SGStudio program have the Ethernet variable correctly declared and allocated. If a user template is used, declare this variable.

In the case of the Quantum Copro platform, the IODDT variable is not used. System words are used instead (%SW160 to %SW167). For example, IOScanning line 1 corresponds to %SW160.0.

In the case of the Quantum NOE platform, the IODDT variable is not used. System words are used instead (%IW1 to %IW8). For example, IOScanning line 1 corresponds to %IW1.0.

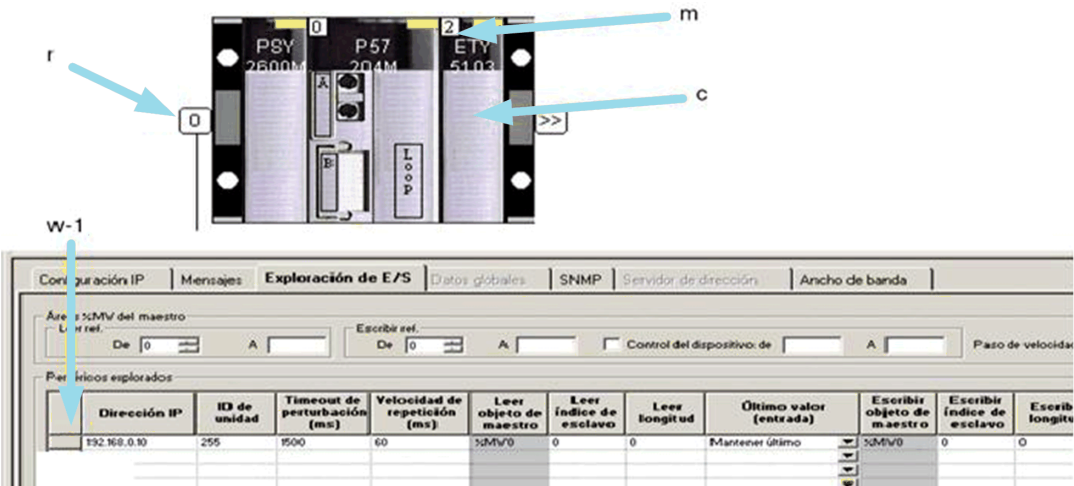
For example, if you use the SG Workbench template, the `CommunicationOK` input is shown as `Ethernet.REFRESH_IO_X`, which has to be replaced with `Ethernet.REFRESH_IO_1.0` for line 1.

The IODDT variable has to be addressed with %cha.b.c, where:

a	= Rack number
b	= Number of slots inside the PLC
c	= Number of channels in the Ethernet module

In the following example, the address for the IODDT variable is %CH0.2.0.

As for the device defined on the first IOScanning line: Ethernet.REFRESH_IO_1.



ATV71/61 Control Block on Ethernet with IO Scanning

Overview

This section provides the configuration and setup details involving the Altivar 71 and Altivar 61 variable speed drives connected through Modbus TCP (Ethernet) in the Modicon M340, M580, and Quantum automation platforms.

Introducción

The Device library currently incorporates many elements with Ethernet technology. One example is the ATV 71 and ATV 61 speed drivers and the Tesys T motor protection devices. You can use any device in serial Modbus mode by using a serial Modbus–Modbus TCP gateway and the `EGtwMB` block from the Communication library.

You can operate both ATV 61 and ATV 71 devices and Tesys T devices with messaging or IO scanning. This documentation refers explicitly to device configurations designed to work with IO Scanning because working with messaging requires the exact same steps until the step of assigning an IP address to the device. After assigning an IP address to the device is completed, the address has to be fed to the `DeviceAddress` input on each one of the devices.

ATV71/61 Control Block on Ethernet

⚠ WARNING

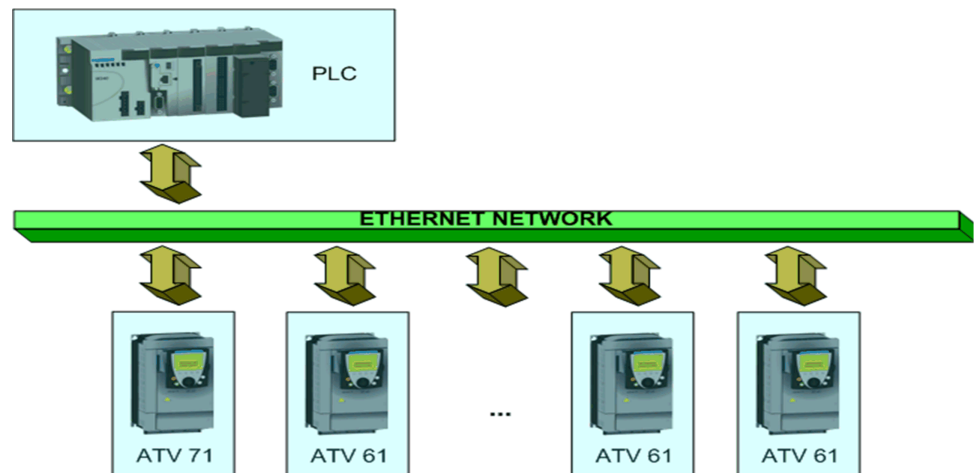
UNINTENDED EQUIPMENT OPERATION

Adapt the below examples to configure device or communication network parameters before you implement them.

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

Refer to the configuration manual for communication networks and Control devices.

This figure represents the hardware configuration:



Preparing the Device

Introduction

Configure each Altivar 71/61 device in such a way that it is on the same network as the PLC that controls it.

Graphic Terminal Settings Description

The following settings need to be defined from the graphic terminal.

1. DRIVE MENU

1.9 COMMUNICATIONS

ETHERNET

IP card (the device IP address)

IP mask (Mask)

IP Gate IP (Gate IP, if any)

IP Process (IP of the controller that controls the device)

You can configure the Process IP section if you want only one single specific controller to control the speed driver. In this case, only the device with the specified IP can access the speed driver. If access control is not required, do not specify an IP, that is, enter 0.0.0.0.

After you have configured this data, you can access the device through the Ethernet network.

ATV 71/61 IO Scanner Configuration

Overview

To implement I/O exchange operations between the speed driver and the PLC, both the ATV 61/ATV 71 IO Scanner and I/O scanning of the controllers have to be configured.

IO Scanner Configuration Procedure


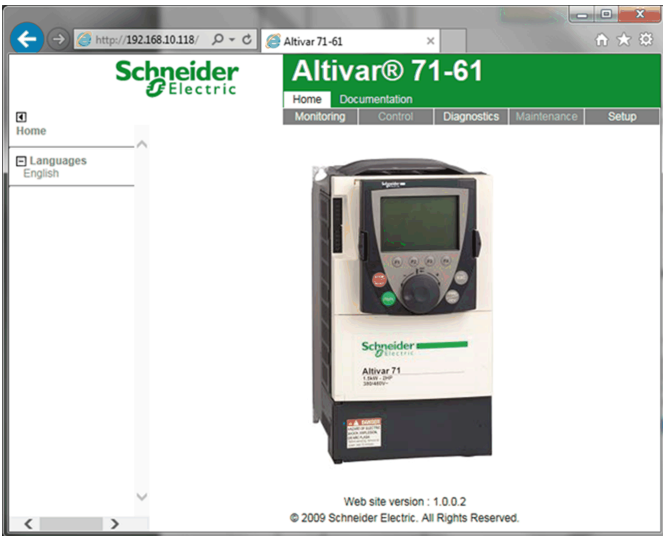
To configure the IO Scanner, you have to access the ATV61/ATV71 webpage. To do this, open Windows Explorer and enter the IP address that has been specified for the device. For example, the address is: <http://197.129.17.118/>.


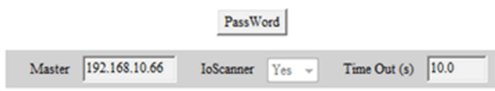
NOTE: To access the device from the computer, the computer has to be on the same network as the speed driver.

For example, if IP address of the speed driver is 197.129.17.118 and the subnet mask is 255.255.255.0, the valid IP address for the PC will be 192.129.17.xxx, where:

xxx	= Can be any number except for 118 (which is the number corresponding to the speed driver).
-----	---

This table shows the step-by-step configuration of the ATV71/61 I/O scanner:

STEP	ACTION	RESULT
1	Enter the Speed drivers IP address into widows explorer	<p>The speed drivers login screen appears.</p> <p>This figure represents the speed drivers login screen:</p>  <p>The default values are:</p> <p>user name: USER</p> <p>Password: USER</p> <p>NOTE: If you change these values, store the values because you cannot access the webpage if the values are not known.</p>
2	Enter the user name and password.	<p>The Home screen for the speed drivers appears:</p> <p>This figure represents the Speed drivers Home screen:</p> 


STEP	ACTION	RESULT																																				
3	Click Setup tab to access the Configuration menu.	<p>Setup menu appears.</p> <p>This figure represents the home screen with Setup menu:</p> 																																				
4	Select the IO Scanner option (on the left menu) to display the IO SCANNER screen.	<p>IO SCANNER screen appears.</p> <p>This figure represents the default configuration values of the latest configuration:</p> <table><tr><th colspan="3">Output Parameters</th><th colspan="3">Input Parameters</th></tr><tr><th></th><th>Parameter</th><th>Device variable</th><th></th><th>Parameter</th><th>Device variable</th></tr><tr><td>1</td><td>CMD</td><td>8501</td><td>1</td><td>ETA</td><td>3201</td></tr><tr><td>2</td><td>LFRD</td><td>8602</td><td>2</td><td>RFRD</td><td>8604</td></tr><tr><td></td><td></td><td></td><td>3</td><td>ERRD</td><td>8606</td></tr><tr><td></td><td></td><td></td><td>4</td><td>LCR</td><td>3204</td></tr></table>  <p>Change the configuration to match the inputs and outputs expected by the device defined in Control.</p> <p>The configuration required corresponds to the inputs (ATV7161_IN_DDT) and outputs (ATV7161_OUT_DDT).</p>	Output Parameters			Input Parameters				Parameter	Device variable		Parameter	Device variable	1	CMD	8501	1	ETA	3201	2	LFRD	8602	2	RFRD	8604				3	ERRD	8606				4	LCR	3204
Output Parameters			Input Parameters																																			
	Parameter	Device variable		Parameter	Device variable																																	
1	CMD	8501	1	ETA	3201																																	
2	LFRD	8602	2	RFRD	8604																																	
			3	ERRD	8606																																	
			4	LCR	3204																																	

Modifying the ATV7161 IOScanner Configuration

Procedure

After the IOScanner is configured, follow the procedure to modify the IOScanner configuration:

STEP	ACTION	RESULT																																																
1	<p>Click PassWord button on the IO SCANNER home screen.</p> <p>Enter the default password (USER).</p>	<p>IO SCANNER home screen refreshes displaying Enter your password field instead of PassWord button to enter the appropriate password.</p> <p>This figure represents the IO SCANNER home screen:</p> <div> <p>Output Parameters</p> <table> <tr> <th></th><th>Parameter</th><th>Device variable</th></tr> <tr> <td>1</td><td>CMD</td><td>8501</td></tr> <tr> <td>2</td><td>LFRD</td><td>8602</td></tr> </table> </div> <div> <p>Input Parameters</p> <table> <tr> <th></th><th>Parameter</th><th>Device variable</th></tr> <tr> <td>1</td><td>ETA</td><td>3201</td></tr> <tr> <td>2</td><td>RFRD</td><td>8604</td></tr> <tr> <td>3</td><td>ERRD</td><td>8606</td></tr> <tr> <td>4</td><td>LCR</td><td>3204</td></tr> </table> </div> <p>NOTE: You can change this password but store it in the memory to modify this configuration in future.</p>		Parameter	Device variable	1	CMD	8501	2	LFRD	8602		Parameter	Device variable	1	ETA	3201	2	RFRD	8604	3	ERRD	8606	4	LCR	3204																								
	Parameter	Device variable																																																
1	CMD	8501																																																
2	LFRD	8602																																																
	Parameter	Device variable																																																
1	ETA	3201																																																
2	RFRD	8604																																																
3	ERRD	8606																																																
4	LCR	3204																																																
2	<p>After entering the password, disable IO SCANNER by selecting No from IO Scanner list.</p>	<p>IO SCANNER home screen with disable option appears:</p> <div> <p>Output Parameters</p> <table> <tr> <th></th><th>Parameter</th><th>Device variable</th></tr> <tr> <td>1</td><td>CMD</td><td>8501</td></tr> <tr> <td>2</td><td>LFRD</td><td>8602</td></tr> </table> </div> <div> <p>Input Parameters</p> <table> <tr> <th></th><th>Parameter</th><th>Device variable</th></tr> <tr> <td>1</td><td>ETA</td><td>3201</td></tr> <tr> <td>2</td><td>RFRD</td><td>8604</td></tr> </table> </div>		Parameter	Device variable	1	CMD	8501	2	LFRD	8602		Parameter	Device variable	1	ETA	3201	2	RFRD	8604																														
	Parameter	Device variable																																																
1	CMD	8501																																																
2	LFRD	8602																																																
	Parameter	Device variable																																																
1	ETA	3201																																																
2	RFRD	8604																																																
3	<p>Enter the modified configuration.</p>	<p>The modified final table appears.</p> <p>This figure represents modified configuration screen:</p> <div> <p>Output Parameters</p> <table> <tr> <th></th><th>Parameter</th><th>Device variable</th></tr> <tr> <td>1</td><td>CMD</td><td>8501</td></tr> <tr> <td>2</td><td>LFRD</td><td>8602</td></tr> <tr> <td>3</td><td>OL1R</td><td>5212</td></tr> <tr> <td>4</td><td>AO1C</td><td>5271</td></tr> <tr> <td>5</td><td>AO2C</td><td>5272</td></tr> <tr> <td>6</td><td>AO3C</td><td>5273</td></tr> </table> </div> <div> <p>Input Parameters</p> <table> <tr> <th></th><th>Parameter</th><th>Device variable</th></tr> <tr> <td>1</td><td>ETA</td><td>3201</td></tr> <tr> <td>2</td><td>RFRD</td><td>8604</td></tr> <tr> <td>3</td><td>ERRD</td><td>8606</td></tr> <tr> <td>4</td><td>LCR</td><td>3204</td></tr> <tr> <td>5</td><td>OTR</td><td>3205</td></tr> <tr> <td>6</td><td>IL1R</td><td>5202</td></tr> <tr> <td>7</td><td>AI1C</td><td>5242</td></tr> <tr> <td>8</td><td>AI2C</td><td>5243</td></tr> </table> </div>		Parameter	Device variable	1	CMD	8501	2	LFRD	8602	3	OL1R	5212	4	AO1C	5271	5	AO2C	5272	6	AO3C	5273		Parameter	Device variable	1	ETA	3201	2	RFRD	8604	3	ERRD	8606	4	LCR	3204	5	OTR	3205	6	IL1R	5202	7	AI1C	5242	8	AI2C	5243
	Parameter	Device variable																																																
1	CMD	8501																																																
2	LFRD	8602																																																
3	OL1R	5212																																																
4	AO1C	5271																																																
5	AO2C	5272																																																
6	AO3C	5273																																																
	Parameter	Device variable																																																
1	ETA	3201																																																
2	RFRD	8604																																																
3	ERRD	8606																																																
4	LCR	3204																																																
5	OTR	3205																																																
6	IL1R	5202																																																
7	AI1C	5242																																																
8	AI2C	5243																																																

STEP	ACTION	RESULT
4	After you have finished setting the configuration, click Save button.	This action causes the speed driver to store the new configuration in its memory.
5	Activate the IO SCANNER again to enable data exchange with the PLC by selecting Yes .	<p>The reactivated IO SCANNER screen appears.</p> <p>This figure represents reactivated IOScanner configuration screen:</p> 

NOTE: If you have followed the above steps, the **IO SCANNER** home screen with the data configured in the speed driver appears when it is activated.

Configuring the IO Scanning of the Controller

Description

To control the speed driver, configure the I/O scanning of the controller.

This configuration is different for each module and needs to match corresponding IO Scanner configuration of the devices.

In the Modicon M340 automation platform, IO scanning is only available if the hardware configuration features a NOE communications module.

This parameter configuration is used to define the relationship between the function I/Os and the device I/Os, which are configured with the IO Scanner.

The **ATV7161** function requires 8 input words in the *****_Inputs** variable (**ANY_ARRAY_INT**) and 6 output words for the *****_Outputs** variable (**ANY_ARRAY_INT**).

The first I/O scanning word is reserved, this means that an I/O Scanning row with 9 read words and 7 write words has to be configured.

The following table describes data variables:

DFB Variable	Parameter	Device variable	Type	Direction
Input0		Reserved	Read	%MWx
Input1	ETA	Status Word (ETA)	Read	%MWx+1
Input2	RFRD	Output Velocity (rpm)	Read	%MWx+2
Input3	ERRD	Code of last detected error	Read	%MWx+3
Input4	LCR	Present Current	Read	%MWx+4
Input5	OTR	Current Torque	Read	%MWx+5

DFB Variable	Parameter	Device variable	Type	Direction
Input6	IL1R	IL1I (Digital inputs)	Read	%MWx+6
Input7	AI1C	AI1 (Analog input 1)	Read	%MWx+7
Input8	AI2C	AI2 (Analog input 2)	Read	%MWx+8
Output0		Reserved	Write	%MWy
Output1	CMD	CWD (Control Word)	Write	%MWy+1
Output2	LFRD	Speed Set-Point	Write	%MWy+2
Output3	OL1R	OL1R (ATV digital outputs)	Write	%MWy+3
Output4	AO1C	AO1 (Analog output 1)	Write	%MWy+4
Output5	AO2C	AO2 (Analog output 2)	Write	%MWy+5
Output6	AO3C	AO3 (Analog output 3)	Write	%MWy+6

Scanning Row Appearance

The following figure shows an example of how a scanning row would look in Control Expert.

In this example, 9 words are read starting with %MW500 and 7 words are written starting with %MW550. The slave variables need to have a value of 0.

The following figure shows the controller IO scanning rows.

	IP address	Device Name	Unit ID	Slave Syntax	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	RD Ref Slave	RD length	Last value (Input)	VR Master Object	VR Ref Slave	VR length	Gate Bri Des
1	192.168.1.118		255	Index	1500	64	%dMV500	0	9	Hold last	%dMV550	0	7	

When this configuration is used, the data used by the controller is as follows

The following figure shows configuration data used by the controller:

Nombre	Tipo	Dirección	Valor	Comentario
EATV71_OutputData	ATV7161_OUT_ETH	%Mw550		
Reserved	INT	%Mw550		Reserved
Data	ATV7161_OUT_DDT	%Mw551		Output Data to ATV71 or ATV61 (MemoryBlock %Mwxc6 linked to DFB)
Data[0]	INT	%Mw551		
Data[1]	INT	%Mw552		
Data[2]	INT	%Mw553		
Data[3]	INT	%Mw554		
Data[4]	INT	%Mw555		
Data[5]	INT	%Mw556		
EATV71_InputData	ATV7161_IN_ETH	%Mw500		
Reserved	INT	%Mw500		Reserved
Data	ATV7161_IN_DDT	%Mw501		Input Data from ATV71 or ATV61 (MemoryBlock %Mwxc8 linked to DFB)
Data[0]	INT	%Mw501		
Data[1]	INT	%Mw502		
Data[2]	INT	%Mw503		
Data[3]	INT	%Mw504		
Data[4]	INT	%Mw505		
Data[5]	INT	%Mw506		
Data[6]	INT	%Mw507		
Data[7]	INT	%Mw508		

Tesys T Control Block on Ethernet

Overview

This section provides the complete connection and setup details of TeSys T devices connected through Modbus TCP (Ethernet) on Modicon M340, M580, or Quantum automation platforms.

Tesys T Hardware Configuration

⚠ WARNING

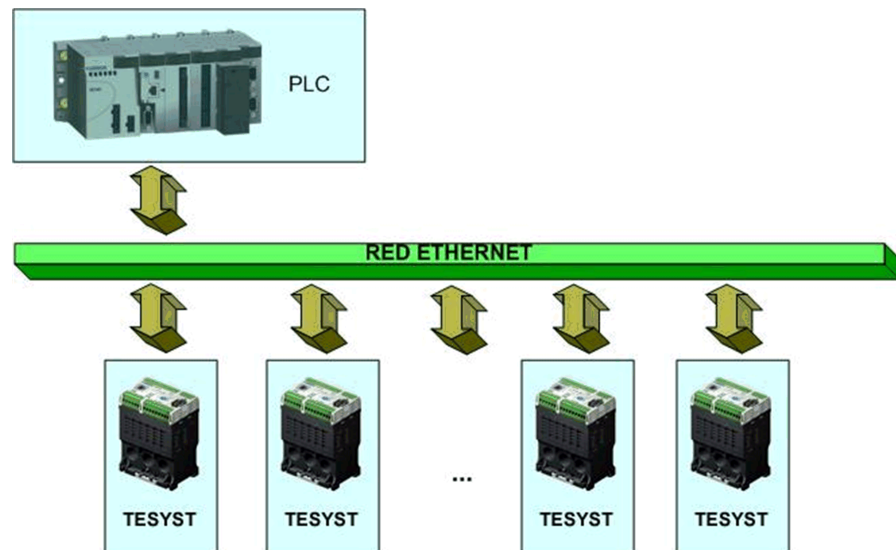
UNINTENDED EQUIPMENT OPERATION

Adapt the below examples to configure device or communication network parameters before you implement them.

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

Refer to the configuration manual for communication networks and Control devices.

The following figure represents the hardware configuration of the Tesys T devices:



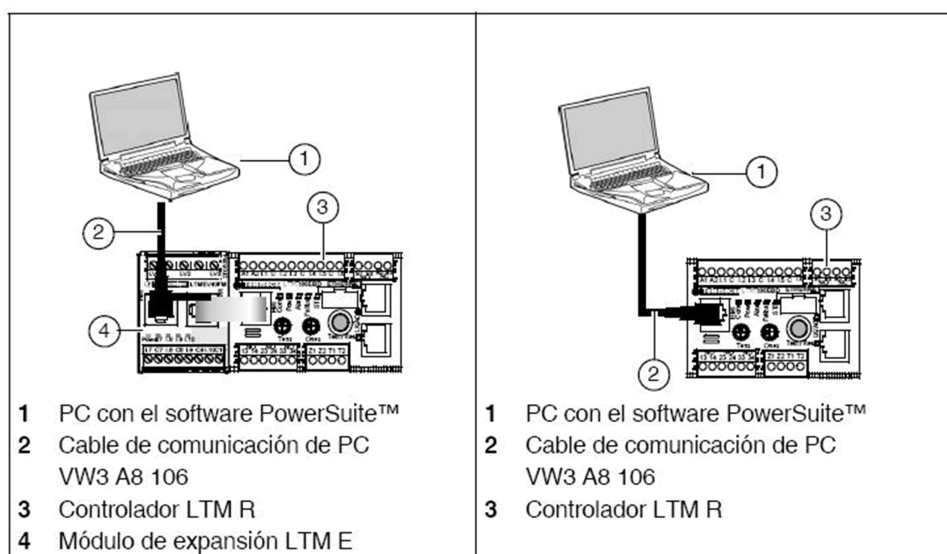
Preparing the Device

Introduction

Configure each Tesys T devices on the same network as the PLC that controls it.

Configuration with and without Expansion Module

To configure the Tesys T, Use the Power Suite software tool by connecting the system exactly as shown in the following figure.



After you are connected, you have to define the options that appear in the following figure.

Network

IP Parameters configuration format

IP address: 197 . 129 . 17 . 150 ☒ Big Endian ☐ Little Endian

Subnet mask: 255 . 255 . 255 . 0 Frame Type: Ethernet II

Default gateway: 0 . 0 . 0 . 0

Communication Loss

Master IP address: 0 . 0 . 0 . 0

☐ Warning

☐ Fault Fault time: 60,00 Seconds FallBack: L01_L02_OFF

Configuration Mode

☐ Configuration via Network port ☒ Local Configuration

☐ Faulty Device Replacement Enabled ☐ Auto Backup

Auto Backup Period: 120 Seconds

Configuring the IO Scanning of the Controller

Description

To control the Tesys T, configure the I/O scanning of the controller.

In the Modicon M340 automation platform, IO Scanning is only available if the hardware configuration features a NOE communications module.

This parameter configuration is used to define the relationship between the function I/Os and the device I/Os.

2 different Tesys T functions use I/O scanning. A function (ETESYST) uses limited amount of words that are refreshed very quickly and the other function (EIOSTESYST) reads a large portion of the words available but at lower speeds.

The ETESYST function requires 4 input words in the Inputs variable (ANY_ARRAY_INT structure) and 3 output words in the Outputs variable (ANY_ARRAY_INT structure). The input words are read starting on word 2502, and the write words are written starting on word 2506.

The EIOSTESYST function requires 64 input words in the Inputs variable (ANY_ARRAY_INT structure) and 5 output words in the Outputs variable (ANY_ARRAY_INT structure). The input words are read starting on word 451, and the output words are read starting on word 700. The refresh rate has to be more than or equal to 200 ms.

The data used in the controller by the ETESYST DFB is as follows::

DFB Variable	Device Variable	Type	Direction
Input0	Status register 1	Read	%MWx
Input1	Status register 2	Read	%MWx+1
Input2	Logic input status	Read	%MWx+2
Input3	Logic output status	Read	%MWx+3
Output0	Output control	Write	%MWy
Output1	CWD (Control Word)	Write	%MWy+1
Output2	Reserved	Write	%MWy+2

Scanning Row Appearance

The following figure shows an example of a scanning line for an ETESYST DFB in Control Expert.

In this example, 4 words are read starting with %MW500 and 3 words are written starting with %MW550. The slave ID needs to be 0.

The following figure shows the controller IO scanning rows.

IP address	Unit ID	Health Timeout (ms)	Repetitive rate (ms)	RD Master Object	RD Slave Index	RD length	Last value (Input)	VR Master Object	VR Slave Index	VR length	Description
1 192.192.17.118	255	1500	60	%MW500	0	9	Hold last	%MW550	0	0	ATV780

Modbus Devices Used Through Gateway

Overview

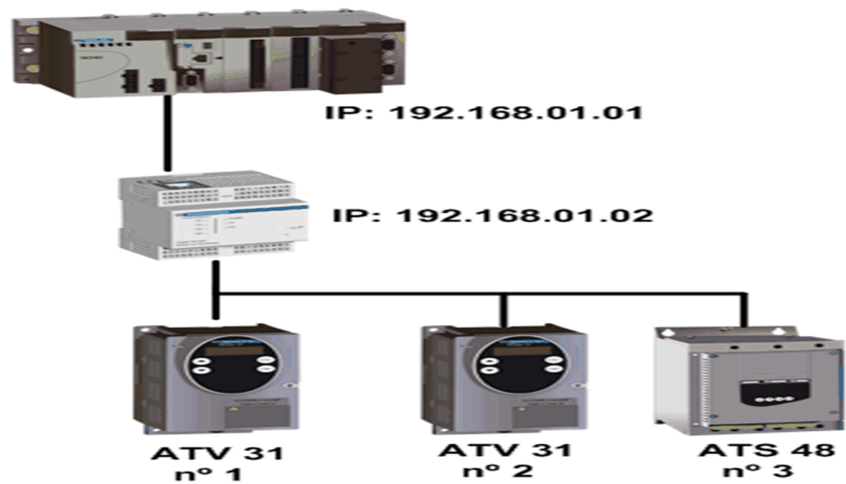
This section provides the complete connection details of several devices on an Ethernet network using an M340 controller and a TSXETG100 gateway.

Modbus Devices Used Through Gateway

M340 Controller

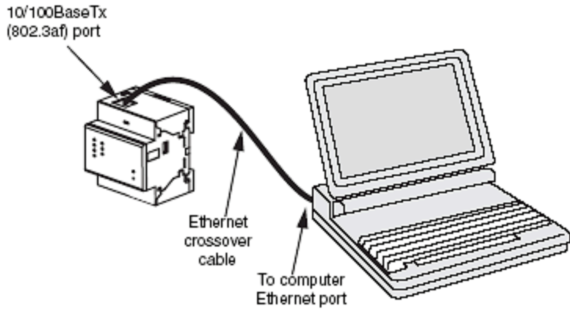
This section includes an example that shows how to connect several devices on an Ethernet network using M340 controller and a TSXETG100 gateway. Refer to the hardware manual and the Communication library manual for more details regarding the elements used in this example.

This figure represents the hardware configuration of M340 controller:



Configuration Procedure

This table describes the procedure to configure the IP address within the Ethernet network:

STEP	ACTION	DESCRIPTION
1	Access the home screen of the gateway to configure the IP address within the Ethernet network by using your computer.	<p>Check the manual for the selected gateway chosen to find the factory IP address and the login/password required to access the gateway for the first time from the computer.</p> <p>This figure shows the hardware configuration:</p> 
2	Configure the gateway address.	<p>Configure the new IP address (which needs to be within the working range), the media type (type of cable), and the baud rate for communicating with the devices on the Ethernet network.</p> <p>This figure represents the IP address configuration:</p> <p style="text-align: center;">Ethernet & TCP/IP</p> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">Ethernet</p> <p>MAC Address - 00:80:67:80:4B:B0</p> <p>Frame Format: Ethernet II</p> <p>Media Type: 10T/100Tx Auto</p> </div> <div style="border: 1px solid black; padding: 5px;"> <p style="text-align: center;">IP Parameters</p> <p>IP Address: 192 . 168 . 1 . 10</p> <p>Subnet Mask: 255 . 255 . 255 . 0</p> <p>Default Gateway: 192 . 168 . 1 . 1</p> <p style="text-align: right;">Apply</p> </div>
3	Configure the Modbus network.	<p>This figure represents the Modbus network configuration:</p> <p style="text-align: center;">Serial Port</p> <div style="border: 1px solid black; padding: 5px;"> <p>Mode: Master</p> <p>Physical Interface: RS485 2-wire</p> <p>Transmission Mode: Automatic</p> <p>Baud Rate: 19200</p> <p>Parity: None</p> <p>Response Timeout: 0.4 (Seconds)</p> <p style="text-align: right;">Apply</p> </div> <p>Refer to the TSXETG100 Configuration Description, page 261.</p>

TSXETG100 Gateway Configuration Description

General

The TSXETG100 gateway can operate either in Master mode or in Slave mode. Normally, it needs to operate in the Master mode for SGU solutions.

Master Mode

The gateway function as the Master for the Modbus network under it, forcing a communication speed and parity for the entire network that it controls and a time out in case of a communications interruption.

Timeout

The configurable `TimeOut` time is the Hardware time out.

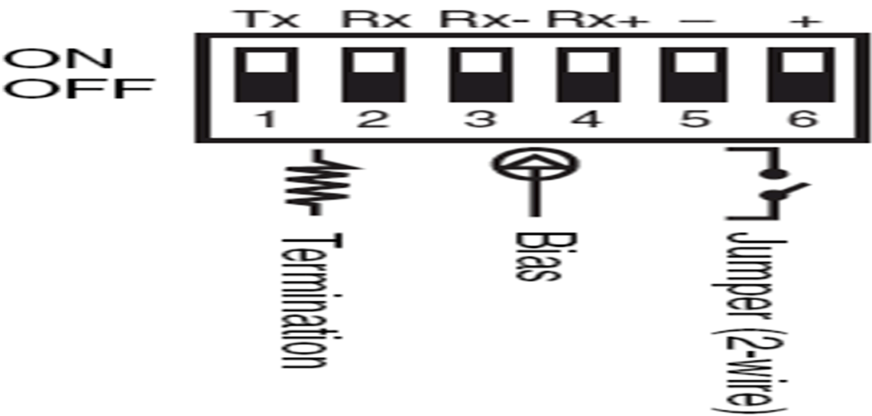
Slave Mode

This gateway model can also work in Slave mode, in which case it serves as an additional slave in the Modbus network. This is useful in networks in which the network Master is a PLC on a Modbus network . To this you can connect an Ethernet slave, that is, the exact opposite route when compared to the one described in this manual.

Polarity

Regarding the polarization of the Modbus network and the addition of terminating resistors, the EGT100 gateway has micro switches available for their configuration.

This figure shows the gateway with Polarization circuit:



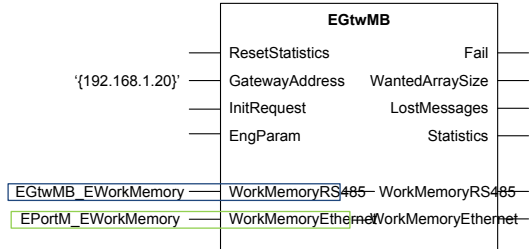
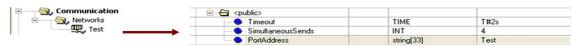
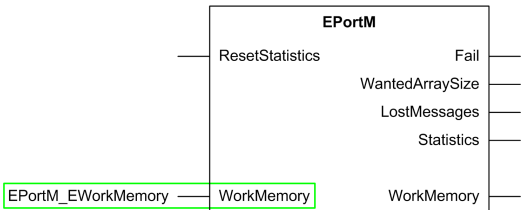
Control Program Example

Program Procedure

The control program for the 2 variable speed drives and for the ATS 48 starter using the CPU's own Ethernet port is as follows.

This table describes the declaration and configuration of the parameters for the devices on the Modbus network.

STEP	ACTION	RESULT
1	Declare and configure the first speed driver with address 1.	<p>The first speed driver is configured with address 1.</p> <p>This figure represents the configured first speed driver</p>
2	Declare and configure the second speed driver with address 2.	<p>The second speed driver is configured with address 2.</p> <p>This figure represents the configured second speed driver.</p>
3	Declare and configure the starter with address 3.	<p>The starter is configured with address 3.</p> <p>This figure represents the configured starter:</p>

STEP	ACTION	RESULT
4	Insert EGtwMB DFB.	<p>The IP address needs to be entered in the GatewayAddress input variable.</p> <p>This figure represents the EGtwMB DFB:</p> 
5	The DFB has a public variable in which the name of the configured channel of the PLC (IP + services) needs to be entered.	<p>This is the channel through which the PLC sends its requests to the slave.</p> <p>This figure represents the slave configuration:</p>  <p>This figure represents Ethernet port configuration:</p> 

NOTE: The name of the EGtwMB_MBWorkMemory variable needs to match in the ATV, ATS, and EGtwMB.

NOTE: The name of the EPortM_EWorkMemory variable needs to match in the EGtwMB and EPortM.

Modbus TCPIP - Implicit with NOE

Overview

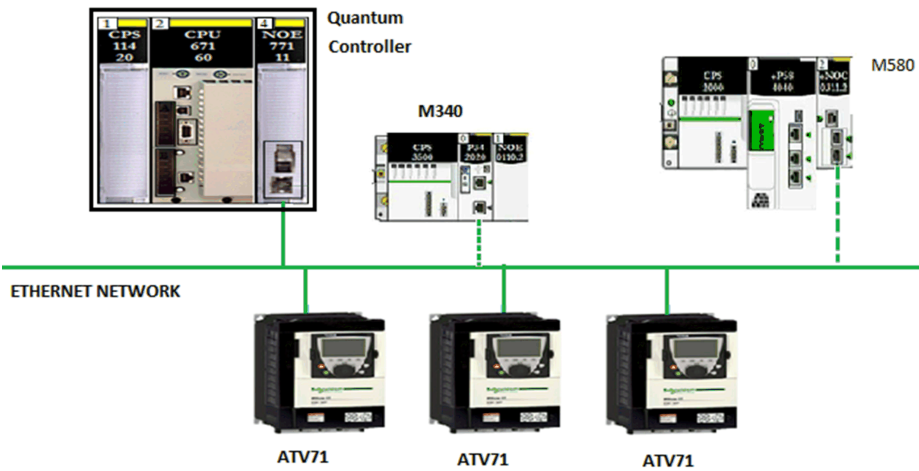
This section describes the procedure to create an application using Modbus TCPIP communication technology (implicit). In this example ATV71 and Quantum controller with NOE is used as a reference. However, same procedure can be followed for creating an application with any other ModBus TCPIP devices or Controller.

Creating an Application using Modbus TCPIP Communication Technology (Implicit with NOE)

Overview

In this example ATV71 and **Quantum** controller with **NOE** is used as a reference. However, same procedure can be followed for creating an application with any other Modbus devices or controller.

Architecture



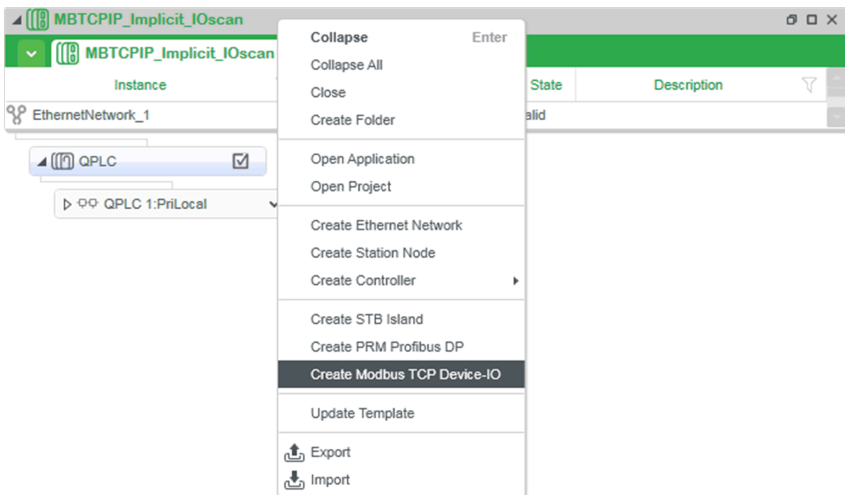
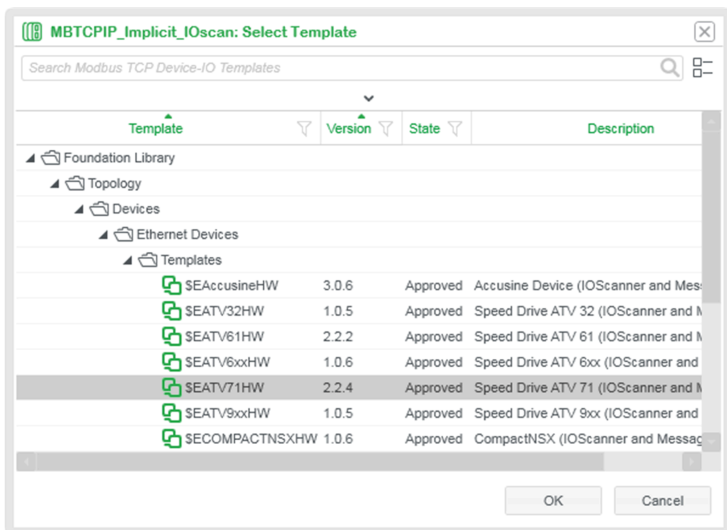
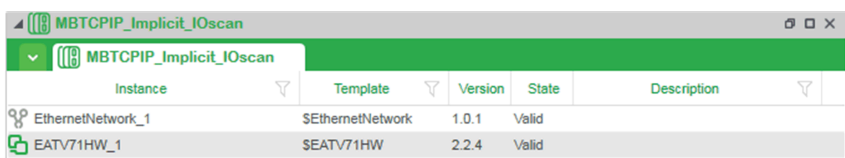
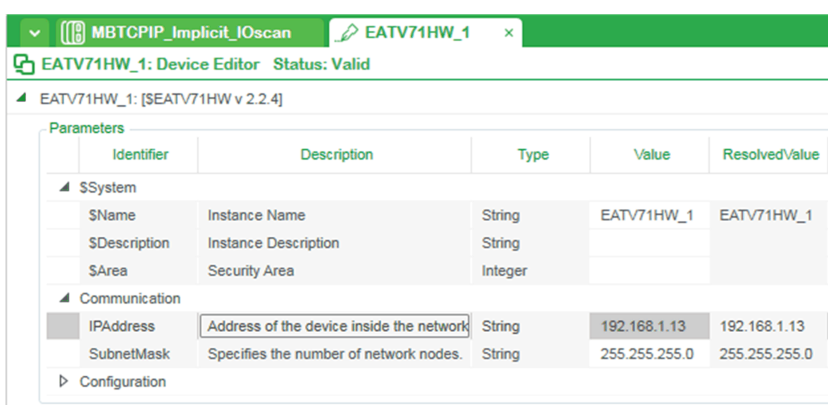
Application Explorer

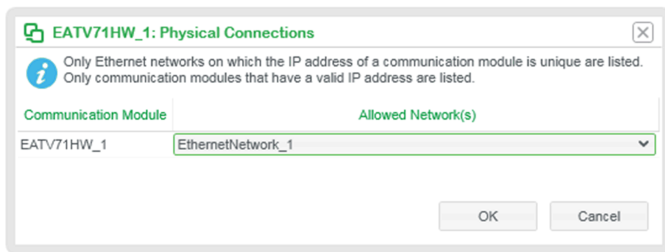
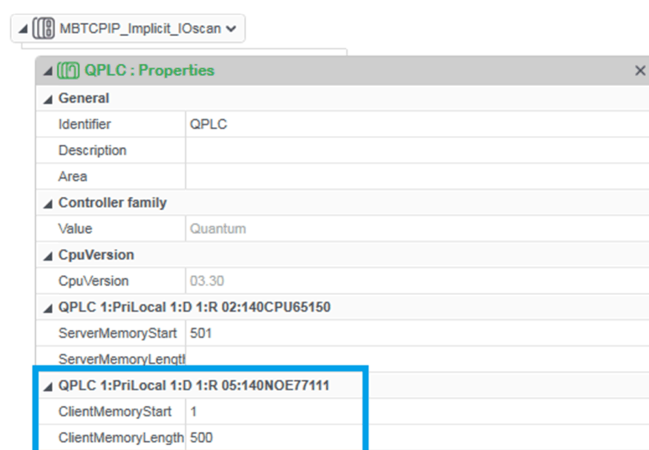
Step	Action
1	Instantiate <code>ATV71E</code> and <code>Motor2</code> templates in Application Explorer .
2	Right click on instance of <code>ATV71E</code> and select Edit Links . Link the objects as shown below. <div><p>The screenshot shows two templates in the Application Explorer. On the left is <code>ATV71E_2</code> (S ATV71E [5.1.8]) with a 'Control' folder containing 'Logic' and 'EnableDevice'. On the right is <code>Motor2_2</code> (SMotor2 [2.7.0]) with a 'Control' folder containing 'ILCK', 'OP1', 'OP2', 'DEV1S2D', and 'DEV2S1D'. An orange line connects the 'EnableDevice' property of <code>ATV71E_2</code> to the 'DEV1S2D' property of <code>Motor2_2</code>. The <code>Motor2_2</code> template also shows a 'Motor2' folder with 'Logic' and 'RSPSEL', and a 'Supervision' folder with 'ForwardRunningSignal', 'ReverseRunningSignal', 'ForwardFailSignal', 'ReverseFailSignal', 'ExternalControl', 'OP1DOSignal', 'OP2DOSignal', 'Local Panel', and 'Supervision'.</p></div>

Topology Explorer

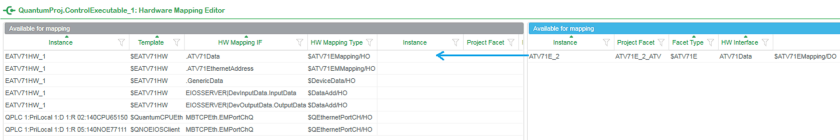
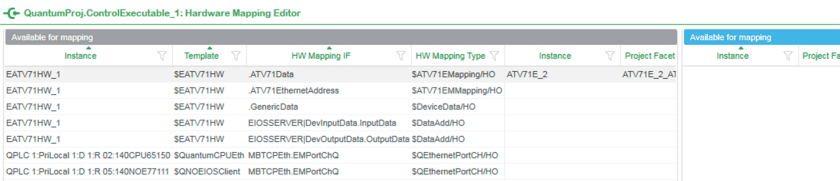
Step	Action
1	In Topology, create PLC hardware configuration with Quantum controller.
2	Link the controller to Ethernet network.

Create Hardware Template

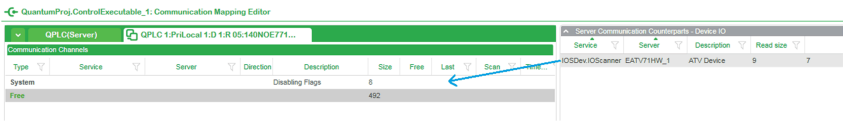
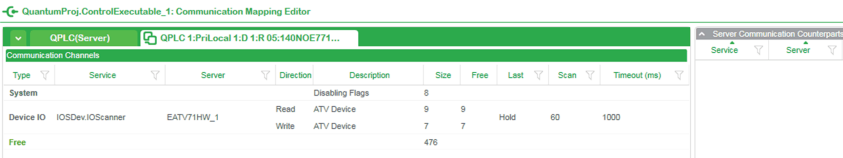
Step	Action																																													
1	<p>Right click on Topology and select Create Modbus TCP Device - IO.</p> 																																													
2	<p>A dialog box appears on the screen. Select \$EATV71HW template and click OK as shown.</p>  <table><thead><tr><th>Template</th><th>Version</th><th>State</th><th>Description</th></tr></thead><tbody><tr><td>\$EAccusineHW</td><td>3.0.6</td><td>Approved</td><td>Accusine Device (IOScanner and Mes</td></tr><tr><td>\$EATV32HW</td><td>1.0.5</td><td>Approved</td><td>Speed Drive ATV 32 (IOScanner and M</td></tr><tr><td>\$EATV61HW</td><td>2.2.2</td><td>Approved</td><td>Speed Drive ATV 61 (IOScanner and M</td></tr><tr><td>\$EATV6xxHW</td><td>1.0.6</td><td>Approved</td><td>Speed Drive ATV 6xx (IOScanner and</td></tr><tr><td>\$EATV71HW</td><td>2.2.4</td><td>Approved</td><td>Speed Drive ATV 71 (IOScanner and M</td></tr><tr><td>\$EATV9xxHW</td><td>1.0.5</td><td>Approved</td><td>Speed Drive ATV 9xx (IOScanner and</td></tr><tr><td>\$ECompactNSXHW</td><td>1.0.6</td><td>Approved</td><td>CompactNSX (IOScanner and Messag</td></tr></tbody></table>	Template	Version	State	Description	\$EAccusineHW	3.0.6	Approved	Accusine Device (IOScanner and Mes	\$EATV32HW	1.0.5	Approved	Speed Drive ATV 32 (IOScanner and M	\$EATV61HW	2.2.2	Approved	Speed Drive ATV 61 (IOScanner and M	\$EATV6xxHW	1.0.6	Approved	Speed Drive ATV 6xx (IOScanner and	\$EATV71HW	2.2.4	Approved	Speed Drive ATV 71 (IOScanner and M	\$EATV9xxHW	1.0.5	Approved	Speed Drive ATV 9xx (IOScanner and	\$ECompactNSXHW	1.0.6	Approved	CompactNSX (IOScanner and Messag													
Template	Version	State	Description																																											
\$EAccusineHW	3.0.6	Approved	Accusine Device (IOScanner and Mes																																											
\$EATV32HW	1.0.5	Approved	Speed Drive ATV 32 (IOScanner and M																																											
\$EATV61HW	2.2.2	Approved	Speed Drive ATV 61 (IOScanner and M																																											
\$EATV6xxHW	1.0.6	Approved	Speed Drive ATV 6xx (IOScanner and																																											
\$EATV71HW	2.2.4	Approved	Speed Drive ATV 71 (IOScanner and M																																											
\$EATV9xxHW	1.0.5	Approved	Speed Drive ATV 9xx (IOScanner and																																											
\$ECompactNSXHW	1.0.6	Approved	CompactNSX (IOScanner and Messag																																											
3	<p>EATV71HW hardware template gets added in Topology instance.</p>  <table><thead><tr><th>Instance</th><th>Template</th><th>Version</th><th>State</th><th>Description</th></tr></thead><tbody><tr><td>EthernetNetwork_1</td><td>\$EthernetNetwork</td><td>1.0.1</td><td>Valid</td><td></td></tr><tr><td>EATV71HW_1</td><td>\$EATV71HW</td><td>2.2.4</td><td>Valid</td><td></td></tr></tbody></table> <p>Right click on EATV71HW_1 instance and select Properties.</p>	Instance	Template	Version	State	Description	EthernetNetwork_1	\$EthernetNetwork	1.0.1	Valid		EATV71HW_1	\$EATV71HW	2.2.4	Valid																															
Instance	Template	Version	State	Description																																										
EthernetNetwork_1	\$EthernetNetwork	1.0.1	Valid																																											
EATV71HW_1	\$EATV71HW	2.2.4	Valid																																											
4	<p>Set the corresponding IP address and subnet mask of MBTCPIP device as shown below.</p>  <table><thead><tr><th>Identifier</th><th>Description</th><th>Type</th><th>Value</th><th>ResolvedValue</th></tr></thead><tbody><tr><td colspan="5">\$System</td></tr><tr><td>\$Name</td><td>Instance Name</td><td>String</td><td>EATV71HW_1</td><td>EATV71HW_1</td></tr><tr><td>\$Description</td><td>Instance Description</td><td>String</td><td></td><td></td></tr><tr><td>\$Area</td><td>Security Area</td><td>Integer</td><td></td><td></td></tr><tr><td colspan="5">Communication</td></tr><tr><td>IPAddress</td><td>Address of the device inside the network</td><td>String</td><td>192.168.1.13</td><td>192.168.1.13</td></tr><tr><td>SubnetMask</td><td>Specifies the number of network nodes.</td><td>String</td><td>255.255.255.0</td><td>255.255.255.0</td></tr><tr><td colspan="5">Configuration</td></tr></tbody></table>	Identifier	Description	Type	Value	ResolvedValue	\$System					\$Name	Instance Name	String	EATV71HW_1	EATV71HW_1	\$Description	Instance Description	String			\$Area	Security Area	Integer			Communication					IPAddress	Address of the device inside the network	String	192.168.1.13	192.168.1.13	SubnetMask	Specifies the number of network nodes.	String	255.255.255.0	255.255.255.0	Configuration				
Identifier	Description	Type	Value	ResolvedValue																																										
\$System																																														
\$Name	Instance Name	String	EATV71HW_1	EATV71HW_1																																										
\$Description	Instance Description	String																																												
\$Area	Security Area	Integer																																												
Communication																																														
IPAddress	Address of the device inside the network	String	192.168.1.13	192.168.1.13																																										
SubnetMask	Specifies the number of network nodes.	String	255.255.255.0	255.255.255.0																																										
Configuration																																														
5	<p>Right-click on EATV71HW_1 instance and select Physical Connections.</p>																																													

Step	Action
	 <p>Select the Ethernet network address in Allowed Network(s) as shown above.</p>
6	In Topology, right click on Controller and open Properties .
7	<p>Define ClientMemoryStart address and ClientMemoryLength to access under the NOE.</p> 

Project Explorer

Step	Action
1	Create a project for Quantum controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .
4	Navigate to Executables section, right-click on Executables and select Map Hardware .
5	<p>Map the Hardware interfaces by drag and drop of application interfaces, as shown below.</p> <p>Map - ATV71 Data to .ATV71Data of \$EATV71HW.</p>  <p>In this example, NOE port for communication is used.</p> <p>After Mapping</p> 

Map Communication

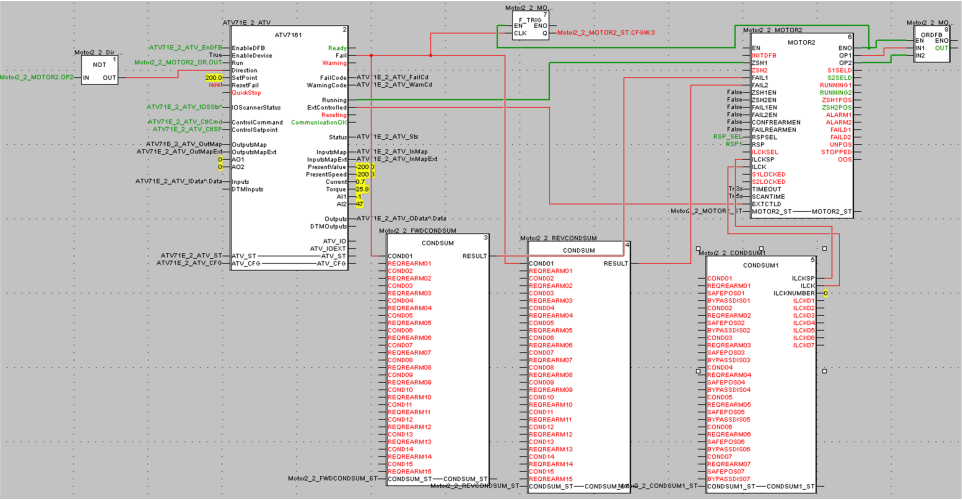
Step	Action
1	Navigate to Executables section, right-click on Executables and select Map Communication .
2	Map the communication counterparts - IO device to Communication Channels .  <p>This figure shows the performing of communication mapping.</p> 

Action Steps

Step	Action
1	Now, generate and build your project and open the project online.

Read/Write Devices

Now, you should be able to read/write devices.



Modbus TCPIP - Implicit with NOC

Overview

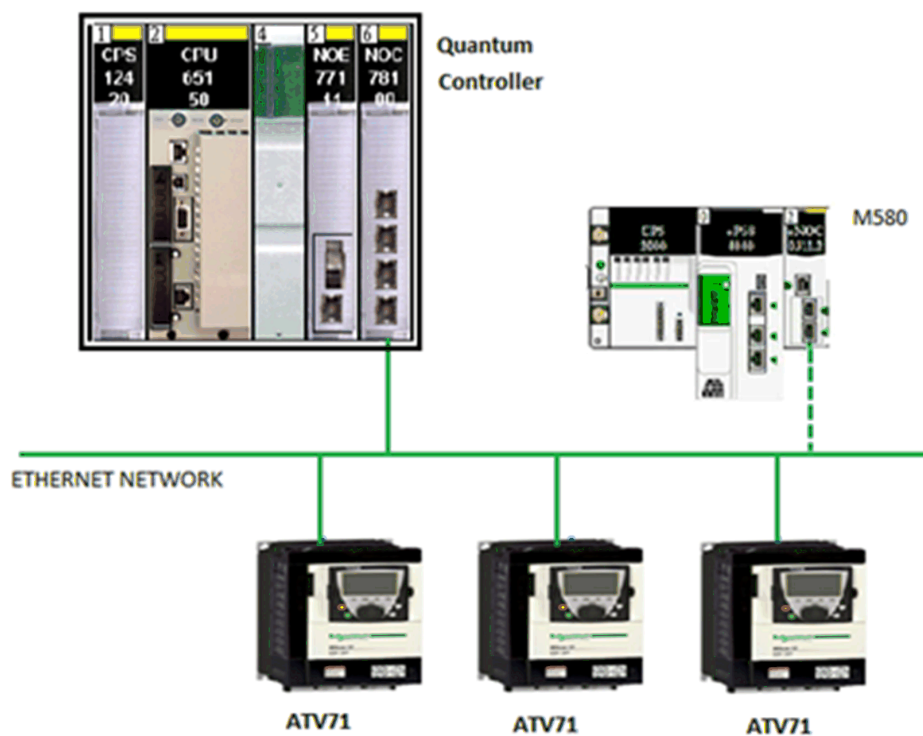
This section describes the procedure to create an application using Modbus TCPIP communication technology (implicit). In this example ATV71 and **Quantum** controller with **NOC** is used as a reference. However, same procedure can be followed for creating an application with any other ModBus TCPIP devices or Controller.

Modbus TCPIP - Implicit with NOC

Overview

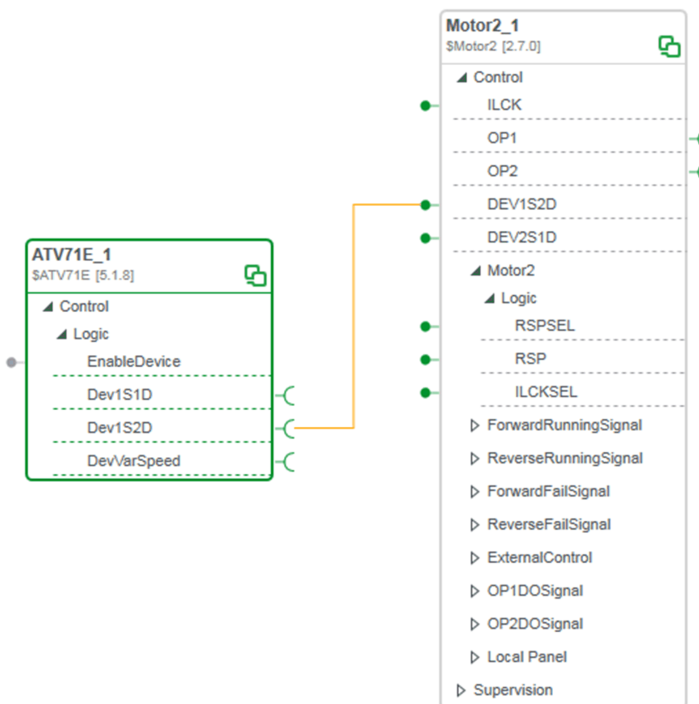
This document explains the process of creating an application using Modbus TCPIP implicit Communication Technology through Quantum controller - NOC and ATV71 is used as a reference. However, same procedure can be used for creating an application with other devices.

Architecture



Application Explorer

Step	Action
1	Instantiate ATV71E and Motor2 templates in Application Explorer .
2	Right click on instance of ATV71E and select Edit Links . Link the objects as shown below.

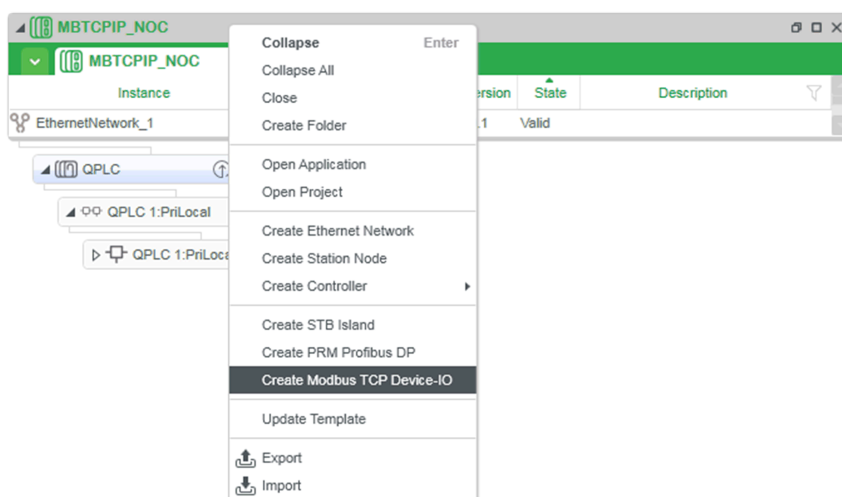


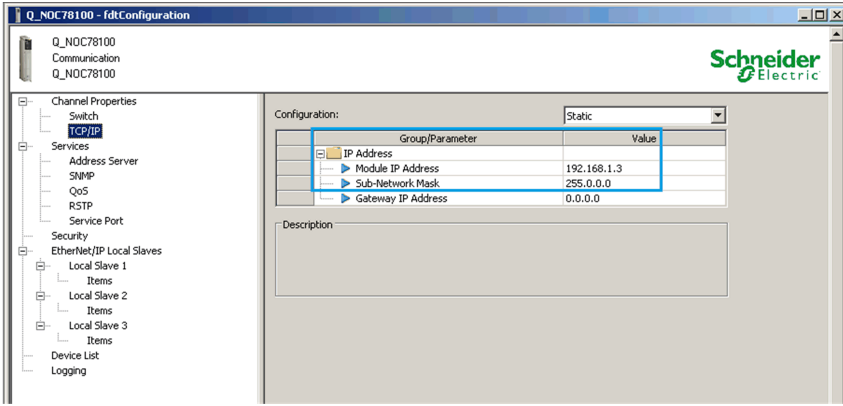
Topology Explorer

Step	Action
1	In Topology, create PLC hardware configuration with Quantum controller.
2	Link the controller to Ethernet network.

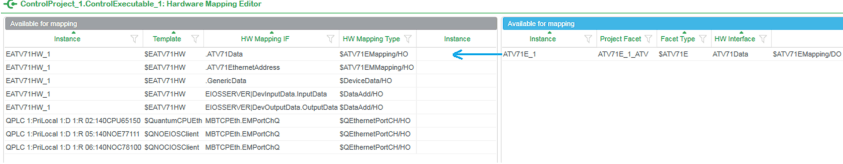
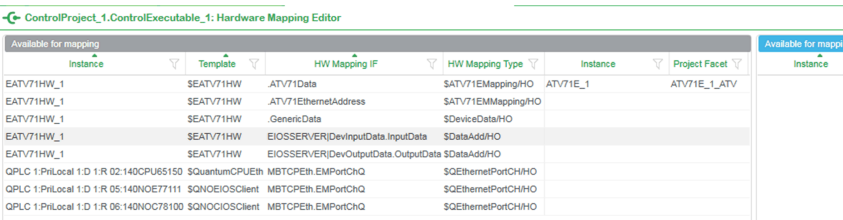
Create Hardware Template

Step	Action
1	Right-click on Topology and select Create Modbus TCP Device - IO .
2	A dialog box appears, select \$EATV71HW template and click OK as shown in the following figure.

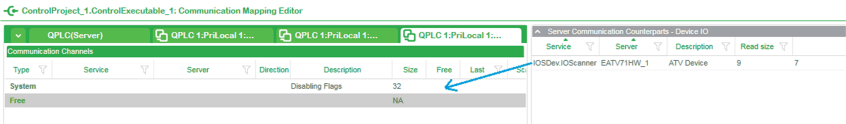
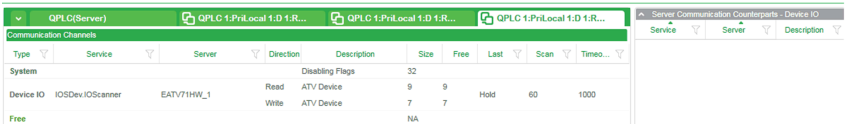


Step	Action
7	<p>Open DTM device configuration, under TCP/IP section provide the IP address of NOC device.</p> 
8	Click Apply and OK to confirm the settings.

Project Explorer

Step	Action
1	Create a project for Quantum controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .
4	Navigate to Executables section, right-click on Executables and select Map Hardware .
5	<p>Map the Hardware interfaces by drag and drop of application interfaces, as shown in the following figure.</p> <p>Map - ATV71 Data to ATV71Data of \$EATV71HW.</p> <p>This mapping is required to pass ATV71 data to DFB.</p>  <p>After Mapping</p> 

Map Communication

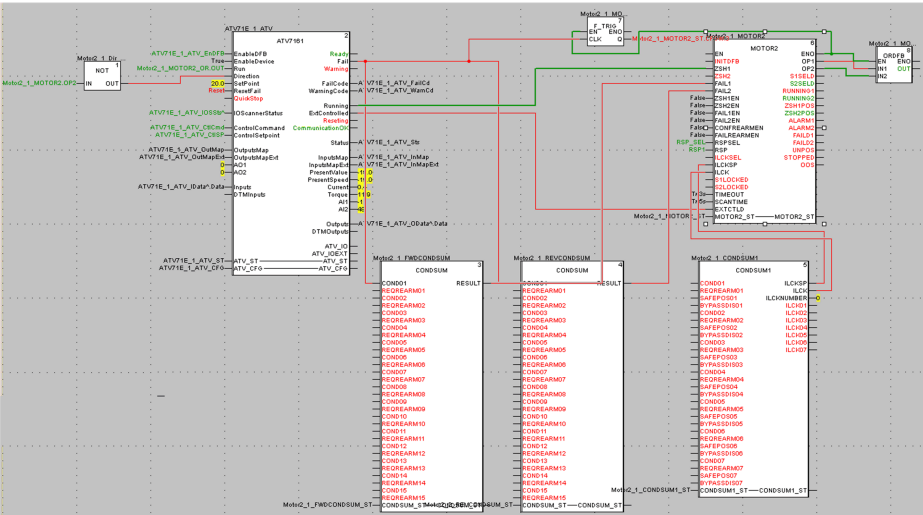
Step	Action
1	Navigate to Executables section, right-click on Executables and select Map Communication .
2	Map the communication counterparts - IO device to Communication Channels .  After Mapping 

Action Steps

Step	Action
1	Now, generate and build your project and open the project online.

Read/Write Devices

Now, you should be able to read/write devices.



Modbus TCPIP - Explicit

Overview

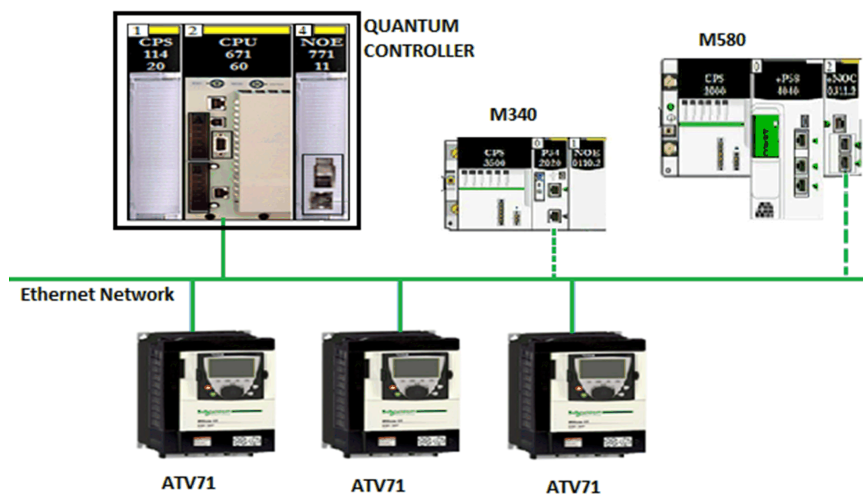
This section describes the procedure to create an application using Modbus TCPIP communication technology (explicit). In this example **ATV71** and **Quantum** controller is used as a reference. However, same procedure can be followed for creating an application with any other ModBus devices or controller.

Creating an Application using Modbus TCPIP Communication Technology (Explicit)

Overview

In this example, ATV71 and Quantum controller is used as a reference. However, same procedure can be followed for creating an application with any other Modbus devices or controller.

Architecture



Application Explorer

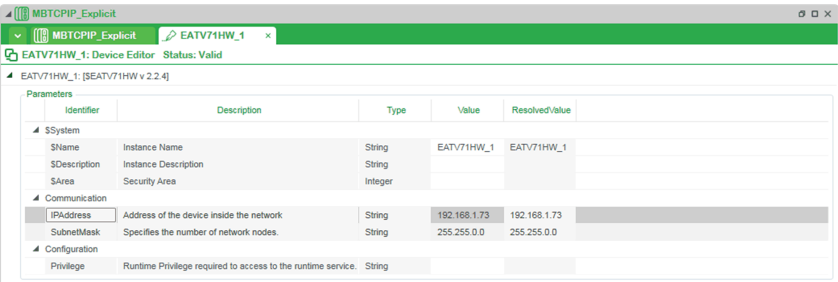
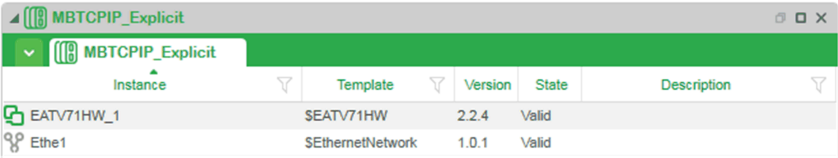
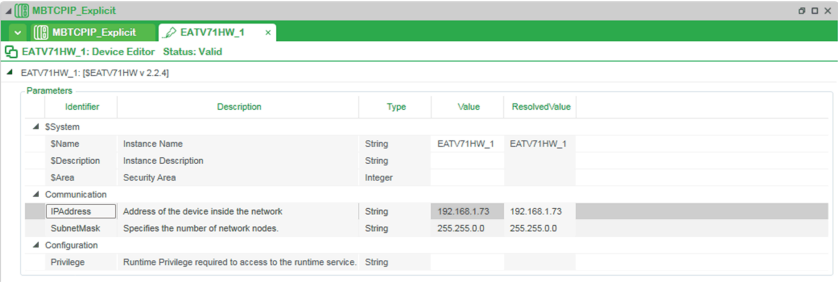
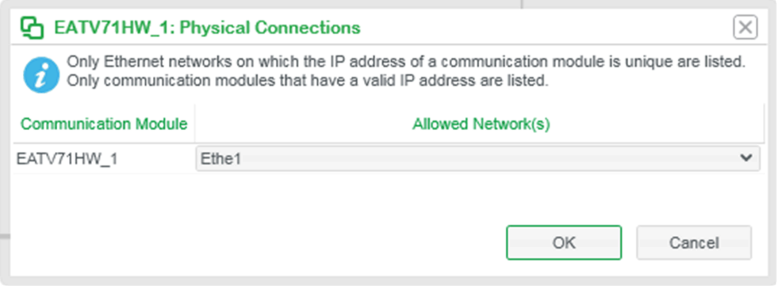
Step	Action
1	Instantiate ATV71EM , Motor2 and EMPortQ templates in Application Explorer .
2	Right click on EMPortQ_2 instance and select Properties . Provide the memory address as shown in the following figure. <div data-bbox="608 356 1461 629" data-label="Image"> <p>The screenshot shows the 'EMPortQ_2 Instance Editor' with the 'Properties' tab selected. The 'Holding Register starting address' is set to '%MW1'. Other parameters like 'Number of Active Sends' (4), 'Time to wait a response' (00:00:02), and 'Max. number of retry' (3) are also visible.</p> </div>
3	Right-click on instance of ATV71EM and select Edit Links . Link the objects as shown in the following figure. <div data-bbox="608 719 1382 1200" data-label="Diagram"> <p>The diagram illustrates the linking of three instances: EMPortQ_2, ATV71EM_2, and Motor2_2. EMPortQ_2 is linked to ATV71EM_2 via the 'Ethernet Port' and 'EnableDevice' properties. ATV71EM_2 is linked to Motor2_2 via the 'Dev1S1D', 'Dev1S2D', and 'DevVarSpeed' properties. The Motor2_2 instance also shows various other properties like 'ILCK', 'OP1', 'OP2', 'DEV1S2D', 'DEV2S1D', 'RSPSEL', 'RSP', 'ILCKSEL', 'ForwardRunningSignal', 'ReverseRunningSignal', 'ForwardFailSignal', 'ReverseFailSignal', 'ExternalControl', 'OP1DOSignal', 'OP2DOSignal', 'Local Panel', and 'Supervision'.</p> </div>

Topology Explorer

Step	Action
1	In Topology, create PLC hardware configuration with Quantum controller.
2	Link the controller to Ethernet network.

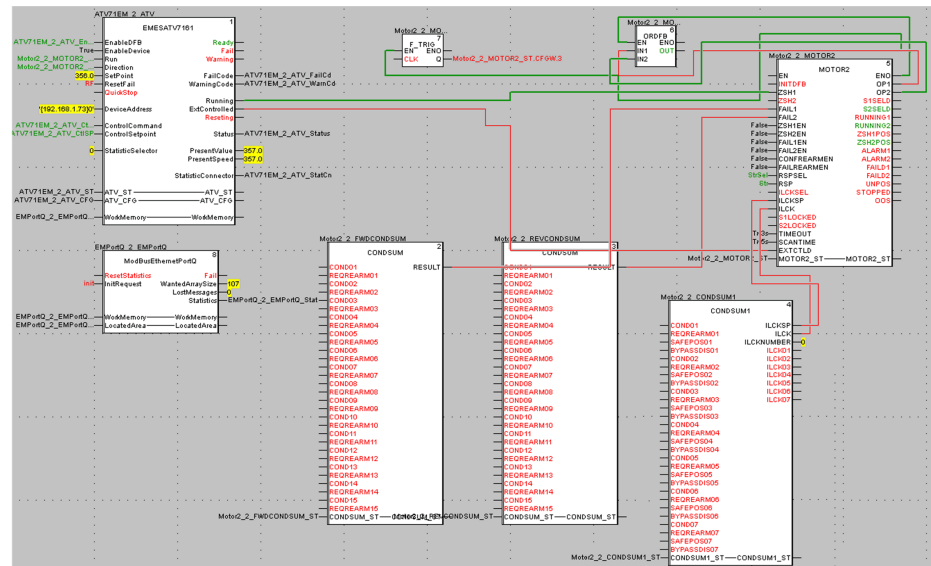
Project Explorer

Step	Action
1	Create a project for Quantum controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .
4	Navigate to Executables section, right click on Executables and select Map Hardware .
5	Map the application interfaces by drag and drop in front of respective field, as shown in the following figure. Map - EMPortChQ to MBTCPeth.EMPortChQ of \$QuantumCPUeth. This mapping is required to pass Quantum PLC Port address to DFB.

Step	Action
	
3	<p>EATV71HW hardware template gets added in Topology instance</p>  <p>Right click on EATV71HW_1 instance and go to Properties.</p>
4	<p>Set the corresponding IP address and subnet mask of MBTCP_IP device as shown in the following figure.</p> 
5	<p>Right-click on EATV71HW_1 instance and go to Physical Connections.</p> <p>Select the Ethernet network address in Allowed Network(s) as shown in the following figure.</p> 

Read/Write Devices

Now, you should be able to read/write devices.



Modbus TCPIP - Explicit with Gateway

Overview

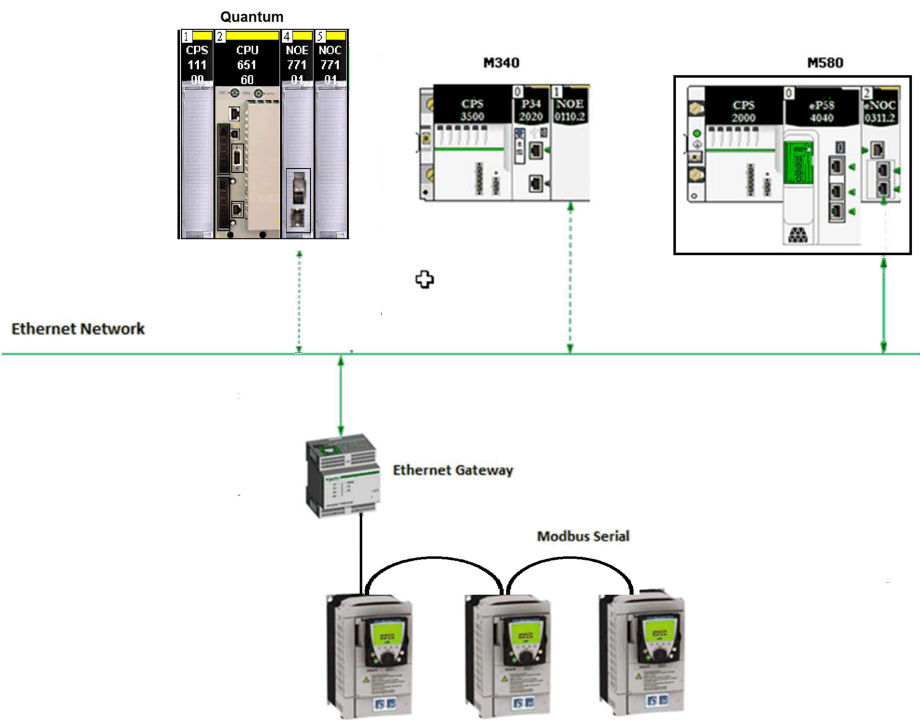
This section describes the procedure to create an application using Modbus TCPIP communication technology (explicit). In this example ATV71 and M580 is used as a reference. However, same procedure can be followed for creating an application with any other Modbus TCPIP devices or controller.

Creating an Application using Modbus Communication Technology (Explicit) with Gateway

Overview

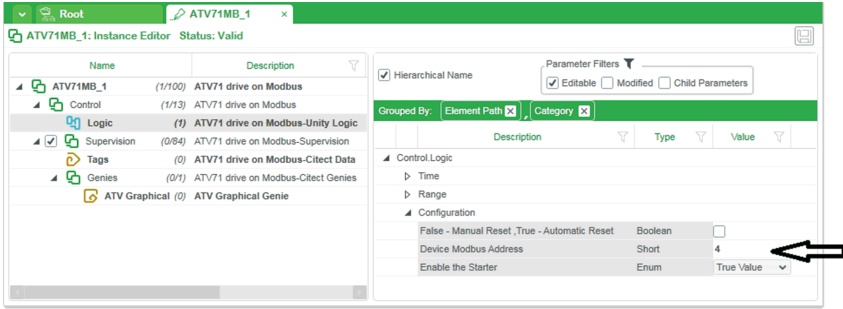
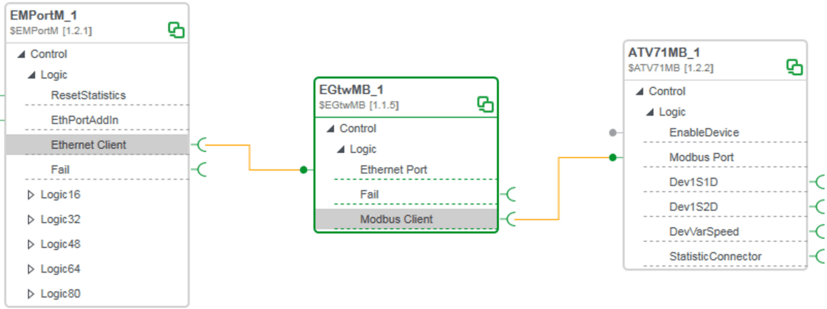
In this example, ATV71 and M580 controller is used as a reference. However, same procedure can be followed for creating an application with any other Modbus devices or controller.

Architecture



Application Explorer


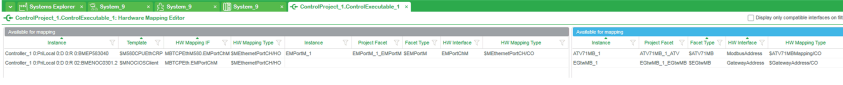
Step	Action
1	Instantiate ATV71MB, <i>Motor</i> , EGtwMB and EMPortM templates in Application Explorer .
2	In Application Explorer right click on instance of EGtwMB and select properties to set the IP address of the gateway. <div></div>

Step	Action
3	<p>In Application Explorer, right click on instance of ATV71MB and select properties to set the Modbus address of the ATV71.</p> 
4	<p>Right-click on instance of EGtwMB_1 and select Edit Links. Link the objects as shown in the following figure.</p> 

Topology Explorer

Step	Action
1	In Topology, create PLC hardware configuration with M580 controller.
2	Link the controller to Ethernet network.

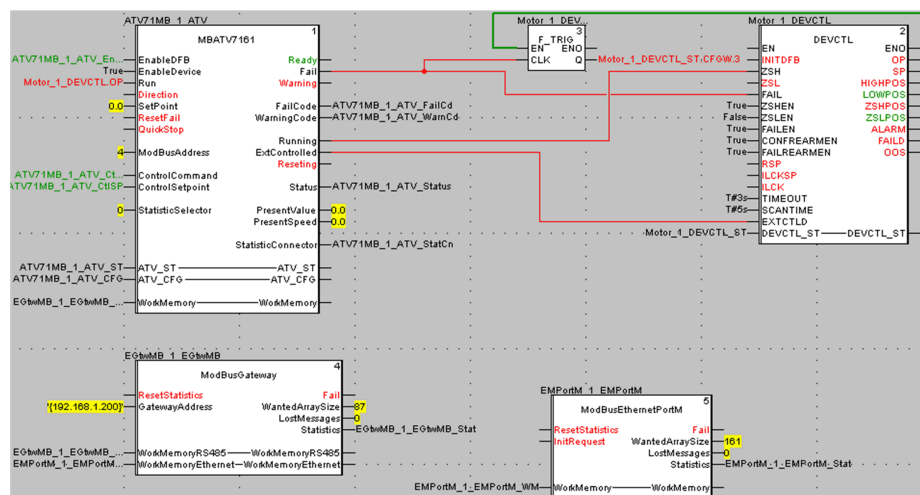
Project Explorer

Step	Action
1	Create a project for M580 controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .
4	Navigate to Executables section, right-click on Executables and select Map Hardware .
5	<p>Map the application interfaces by drag and drop in front of respective field, as shown in the following figure:</p> <p>Map - EMPortM_1 to MBTCPeth.EMPortCHM of \$M580CPUETHCRP.</p> <p>This mapping is required to pass <i>M580</i> PLC Port address to DFB. In this example we are using <i>M580</i> CPU port for communication.</p>  <p>After Mapping,</p> 
6	Build the control project and open the built project.

Step	Action																																																																							
7	<p>In Function block section, check public parameters for <code>EMPortM</code> block.</p> <table><tr><th>Name</th><th>no.</th><th>Type</th><th>Value</th><th>Comment</th></tr><tr><td>ATV71MB_1_ATV</td><td></td><td>MBATV71...</td><td></td><td></td></tr><tr><td>EGtwMB_1_EGtwMB</td><td></td><td>ModBusG...</td><td></td><td></td></tr><tr><td>EMPortM_1_EMPortM</td><td></td><td>ModBusEt...</td><td></td><td></td></tr><tr><td><inputs></td><td></td><td></td><td></td><td></td></tr><tr><td><outputs></td><td></td><td></td><td></td><td></td></tr><tr><td><inputs/outputs></td><td></td><td></td><td></td><td></td></tr><tr><td><public></td><td></td><td></td><td></td><td></td></tr><tr><td>Timeout</td><td>TIME</td><td>T#2s</td><td></td><td>Time to wait a response. Must be greater than the hardware timeout</td></tr><tr><td>SimultaneousSends</td><td>INT</td><td>4</td><td></td><td>Number of active sendings</td></tr><tr><td>PortAddress</td><td>string[33]</td><td>0.0.3</td><td></td><td>Name of the network to send the data</td></tr><tr><td>MaxRetryAfterSwitch</td><td>INT</td><td>3</td><td></td><td>Max. number of retry for first request to be successful after the controller switchover</td></tr></table> <p>Verify the port addressing as per below table:</p> <table><tr><th colspan="2">Controller</th><th>Port Addressing</th></tr><tr><td rowspan="2">Quantum</td><td>CPU</td><td>'254'</td></tr><tr><td>NOE/NOC</td><td>'Slot'</td></tr><tr><td>M340/M580</td><td>CPU/NOE/NOC</td><td>'Rack.Slot.channel'</td></tr></table>	Name	no.	Type	Value	Comment	ATV71MB_1_ATV		MBATV71...			EGtwMB_1_EGtwMB		ModBusG...			EMPortM_1_EMPortM		ModBusEt...			<inputs>					<outputs>					<inputs/outputs>					<public>					Timeout	TIME	T#2s		Time to wait a response. Must be greater than the hardware timeout	SimultaneousSends	INT	4		Number of active sendings	PortAddress	string[33]	0.0.3		Name of the network to send the data	MaxRetryAfterSwitch	INT	3		Max. number of retry for first request to be successful after the controller switchover	Controller		Port Addressing	Quantum	CPU	'254'	NOE/NOC	'Slot'	M340/M580	CPU/NOE/NOC	'Rack.Slot.channel'
Name	no.	Type	Value	Comment																																																																				
ATV71MB_1_ATV		MBATV71...																																																																						
EGtwMB_1_EGtwMB		ModBusG...																																																																						
EMPortM_1_EMPortM		ModBusEt...																																																																						
<inputs>																																																																								
<outputs>																																																																								
<inputs/outputs>																																																																								
<public>																																																																								
Timeout	TIME	T#2s		Time to wait a response. Must be greater than the hardware timeout																																																																				
SimultaneousSends	INT	4		Number of active sendings																																																																				
PortAddress	string[33]	0.0.3		Name of the network to send the data																																																																				
MaxRetryAfterSwitch	INT	3		Max. number of retry for first request to be successful after the controller switchover																																																																				
Controller		Port Addressing																																																																						
Quantum	CPU	'254'																																																																						
	NOE/NOC	'Slot'																																																																						
M340/M580	CPU/NOE/NOC	'Rack.Slot.channel'																																																																						

Read/Write Devices

Now you should be able to read/write devices.

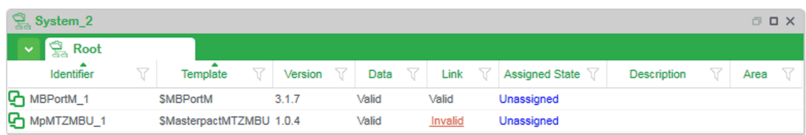
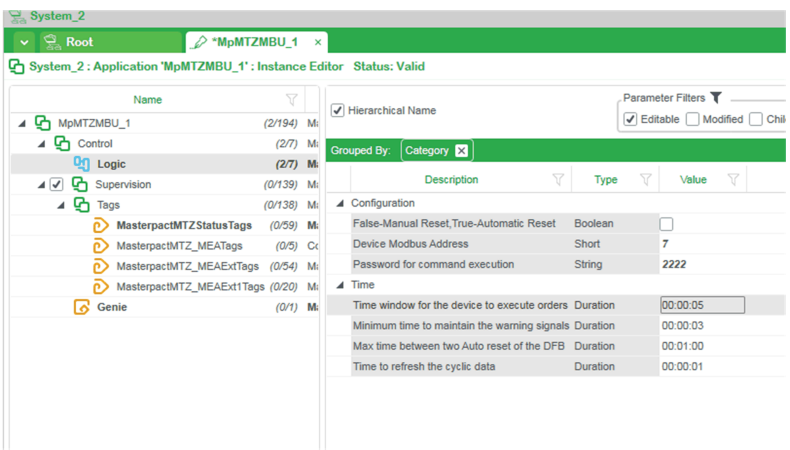
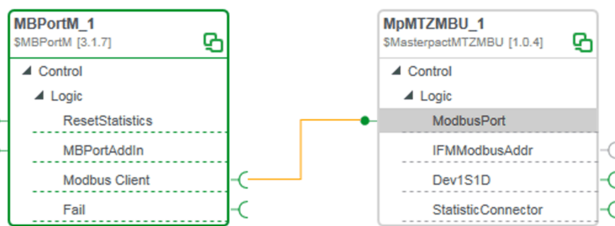


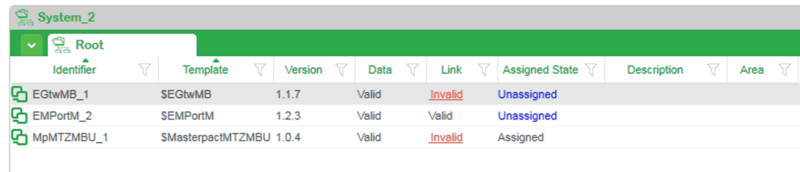
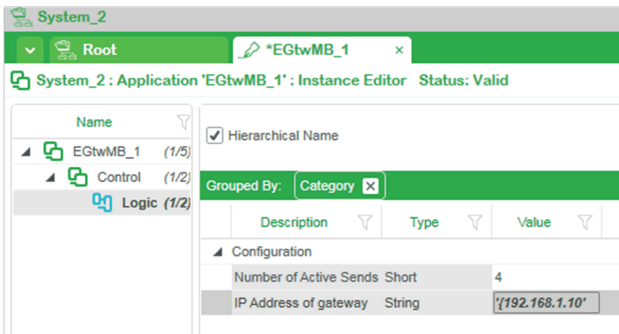
Modbus over ULP Technology

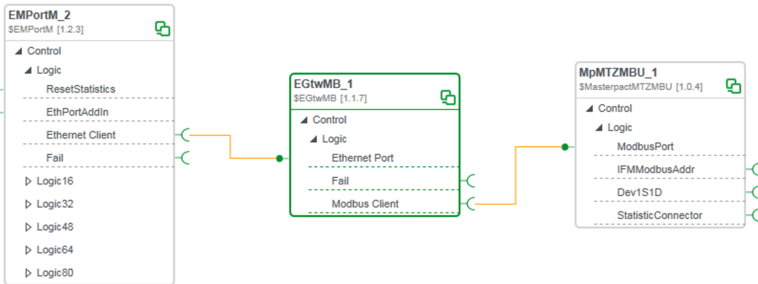
What's in This Chapter

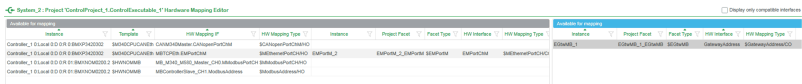
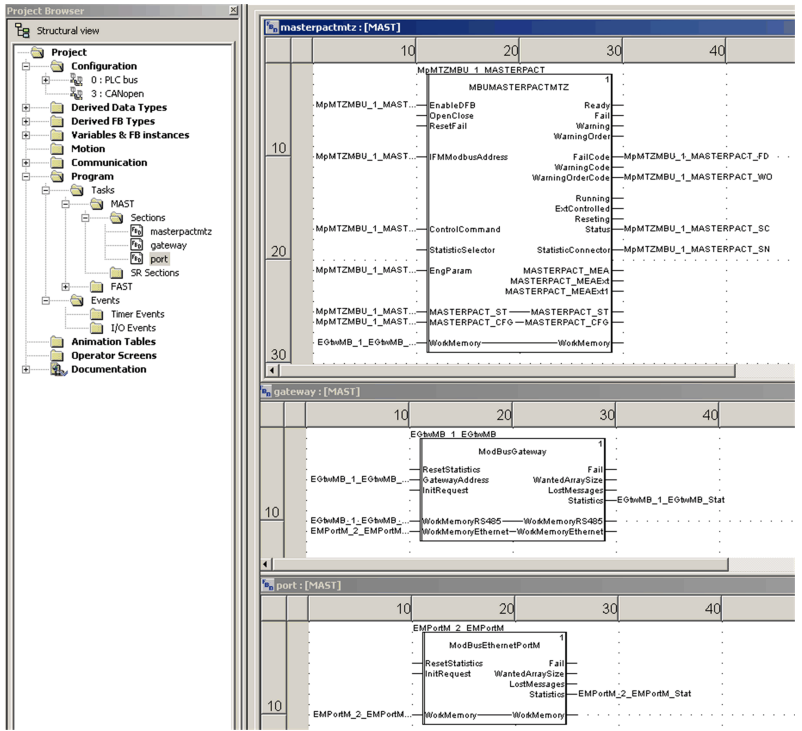
Creating a Project with \$MASTERPACTMTZMBU for IFM connected to Masterpact MTZ and Modbus Serial RJ45 Ports	282
Creating a Project with \$MasterpactMTZMBU for IFE or IFE stacked with IFM connected to Masterpact MTZ and Modbus Ethernet RJ45 Ports	284

Creating a Project with \$MASTERPACTMTZMBU for IFM connected to Masterpact MTZ and Modbus Serial RJ45 Ports

Step	Action
1	<p>Instantiate \$MasterpactMTZMBU and \$MBPortM as shown.</p> 
2	<p>Configure the parameters as shown.</p> 
3	<p>Perform edit links as shown.</p> 
4	<p>Create MAST sections and assign the facets accordingly and perform Generate.</p>

Step	Action																																
1	<p>Instantiate \$MasterpactMTZMBU, \$EGtwMB, \$MBPortM as shown.</p>  <table><thead><tr><th>Identifier</th><th>Template</th><th>Version</th><th>Data</th><th>Link</th><th>Assigned State</th><th>Description</th><th>Area</th></tr></thead><tbody><tr><td>EGtwMB_1</td><td>SEGtwMB</td><td>1.1.7</td><td>Valid</td><td>Invalid</td><td>Unassigned</td><td></td><td></td></tr><tr><td>EMPortM_2</td><td>\$EMPortM</td><td>1.2.3</td><td>Valid</td><td>Valid</td><td>Unassigned</td><td></td><td></td></tr><tr><td>MpMTZMBU_1</td><td>\$MasterpactMTZMBU</td><td>1.0.4</td><td>Valid</td><td>Invalid</td><td>Assigned</td><td></td><td></td></tr></tbody></table>	Identifier	Template	Version	Data	Link	Assigned State	Description	Area	EGtwMB_1	SEGtwMB	1.1.7	Valid	Invalid	Unassigned			EMPortM_2	\$EMPortM	1.2.3	Valid	Valid	Unassigned			MpMTZMBU_1	\$MasterpactMTZMBU	1.0.4	Valid	Invalid	Assigned		
Identifier	Template	Version	Data	Link	Assigned State	Description	Area																										
EGtwMB_1	SEGtwMB	1.1.7	Valid	Invalid	Unassigned																												
EMPortM_2	\$EMPortM	1.2.3	Valid	Valid	Unassigned																												
MpMTZMBU_1	\$MasterpactMTZMBU	1.0.4	Valid	Invalid	Assigned																												
2	<p>Configure the parameters as shown.</p>  <p>The screenshot shows the 'Instance Editor' for 'EGtwMB_1'. The left sidebar shows a tree structure: Name > EGtwMB_1 (1/5) > Control (1/2) > Logic (1/2). The main area shows the 'Configuration' tab with the following parameters:</p> <table><thead><tr><th>Grouped By</th><th>Description</th><th>Type</th><th>Value</th></tr></thead><tbody><tr><td>Category</td><td colspan="3"></td></tr><tr><td colspan="4">Configuration</td></tr><tr><td></td><td>Number of Active Sends</td><td>Short</td><td>4</td></tr><tr><td></td><td>IP Address of gateway</td><td>String</td><td>'192.168.1.10'</td></tr></tbody></table>	Grouped By	Description	Type	Value	Category				Configuration					Number of Active Sends	Short	4		IP Address of gateway	String	'192.168.1.10'												
Grouped By	Description	Type	Value																														
Category																																	
Configuration																																	
	Number of Active Sends	Short	4																														
	IP Address of gateway	String	'192.168.1.10'																														
3	<p>Perform edit links as shown.</p>																																

Step	Action																																						
	<div></div>																																						
4	<div><p>Create MAST sections and assign the facets accordingly and perform Generate.</p><div><div><p>System_2 : Project 'ControlProject_1' Assignment Editor</p><p>Instances</p><table><tr><th>Instance</th><th>Template</th><th>Version</th><th>Assigned State</th><th>Description</th></tr><tr><td>EGtwMB_1</td><td>\$EGtwMB</td><td>1.1.7</td><td>Assigned</td><td>Application Explorer</td></tr><tr><td>EMPortM_2</td><td>\$EMPortM</td><td>1.2.3</td><td>Assigned</td><td></td></tr><tr><td>MpMTZMBU_1</td><td>\$MasterpactMTZMBU</td><td>1.0.4</td><td>Assigned</td><td></td></tr></table></div><div><div><p>ControlProject_1 - Containers</p><table><tr><th>Path</th><th>Container</th></tr><tr><td>.</td><td>ControlProject_1</td></tr><tr><td>.\ControlProject_1\Tasks\I Mast</td><td>gateway</td></tr><tr><td>.\ControlProject_1\Tasks\I Mast</td><td>masterpactmtz</td></tr><tr><td>.\ControlProject_1\Tasks\I Mast</td><td>port</td></tr></table></div><div><p>gateway - Assignments</p><table><tr><th>Container</th><th>Instance</th><th>Instance Template</th><th>State</th></tr><tr><td>gateway</td><td>EGtwMB_1</td><td>SEGtwMB</td><td>Valid</td></tr></table></div></div><p>NOTE: The port DFB should have higher execution order than MBUMASTERPACTMTZ DFB.</p></div></div>	Instance	Template	Version	Assigned State	Description	EGtwMB_1	\$EGtwMB	1.1.7	Assigned	Application Explorer	EMPortM_2	\$EMPortM	1.2.3	Assigned		MpMTZMBU_1	\$MasterpactMTZMBU	1.0.4	Assigned		Path	Container	.	ControlProject_1	.\ControlProject_1\Tasks\I Mast	gateway	.\ControlProject_1\Tasks\I Mast	masterpactmtz	.\ControlProject_1\Tasks\I Mast	port	Container	Instance	Instance Template	State	gateway	EGtwMB_1	SEGtwMB	Valid
Instance	Template	Version	Assigned State	Description																																			
EGtwMB_1	\$EGtwMB	1.1.7	Assigned	Application Explorer																																			
EMPortM_2	\$EMPortM	1.2.3	Assigned																																				
MpMTZMBU_1	\$MasterpactMTZMBU	1.0.4	Assigned																																				
Path	Container																																						
.	ControlProject_1																																						
.\ControlProject_1\Tasks\I Mast	gateway																																						
.\ControlProject_1\Tasks\I Mast	masterpactmtz																																						
.\ControlProject_1\Tasks\I Mast	port																																						
Container	Instance	Instance Template	State																																				
gateway	EGtwMB_1	SEGtwMB	Valid																																				
5	Create topology project.																																						

Step	Action
6	<p>Perform map hardware as shown.</p>  <p>The screenshot shows the 'Hardware Mapping Editor' with a table of hardware components and their mappings. The table has columns for 'Instance', 'Type', 'HW Mapping ID', 'Instance', 'Project Name', 'Point Type', 'HW Instance', and 'HW Mapping Type'. The data includes various components like 'Control_1', 'Control_2', 'Control_3', and 'Control_4' mapped to specific hardware addresses and types.</p>
7	<p>Perform build to obtain project as shown.</p>  <p>The screenshot shows the 'Project Browser' on the left and the 'masterpactmtz : [MAST]' project structure on the right. The Project Browser shows a tree view of the project structure, including 'Configuration', 'Derived Data Types', 'Derived FB Types', 'Variables & FB Instances', 'Motion', 'Communication', 'Program', 'Tasks', 'Sections', 'Events', 'Timer Events', 'I/O Events', 'Animation Tables', 'Operator Screens', and 'Documentation'. The 'masterpactmtz : [MAST]' project structure shows a table of components and their mappings, including 'MpmTZMBU_1_MASTERPACT', 'gateway : [MAST]', and 'port : [MAST]'. The table has columns for 'Instance', 'Type', 'HW Mapping ID', 'Instance', 'Project Name', 'Point Type', 'HW Instance', and 'HW Mapping Type'.</p>

Modbus Technology

What's in This Chapter

General	287
Configuring the Devices	290
Modbus Serial - Explicit	298

Overview

This chapter describes the configuration and setup details of the devices connected through Modbus on the Modicon M340 automation platform.

General

Overview

This section describes the complete configuration and setup details of the devices connected through Modbus on the Modicon M340 automation platform.

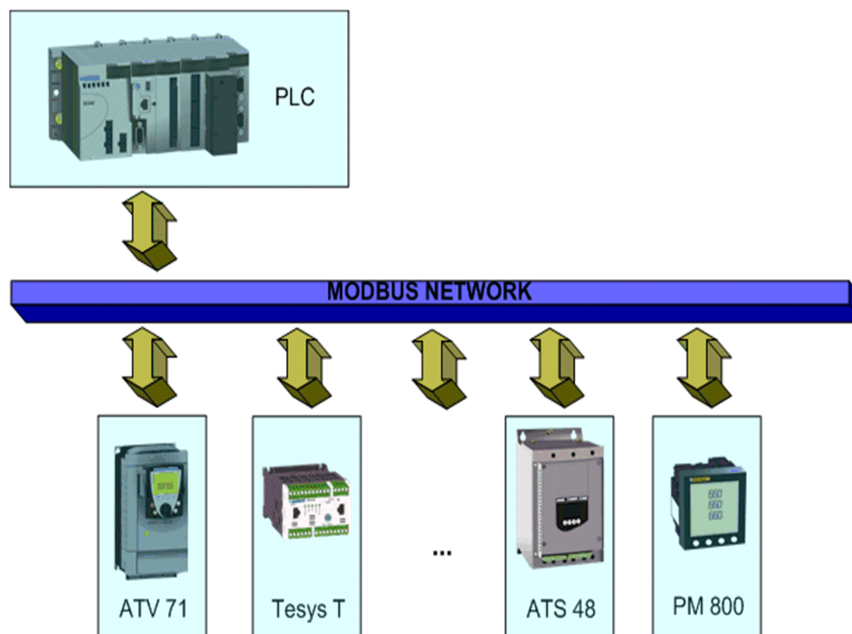
Modbus Technology

Modbus Network Configuration

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION
Adapt the below examples to configure device or communication network parameters before you implement them.
Failure to follow these instructions can result in death, serious injury, or equipment damage.

Refer to the configuration manual for communication networks and Control devices.

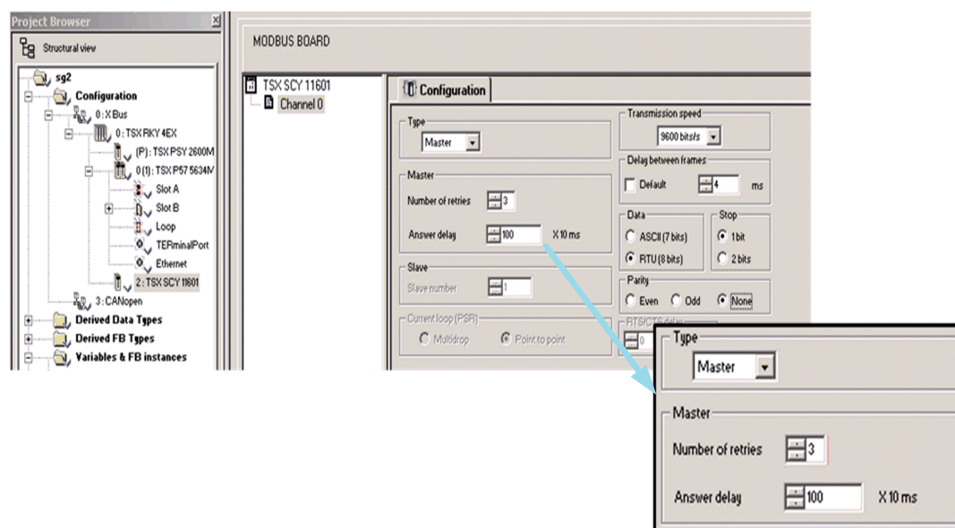
This figure shows the complete configuration and setup details for the devices connected through Modbus on the Modicon M340 automation platforms.



Determining Timeout Times

Introduction

This section provides guidelines for correctly selecting `Timeout` and `Refresh` times based on the known parameters by using an example with a Modbus connection close to 10 ATV 31 devices:



Hardware Timeout

The timeout setting is selected on the MB physical port (PCMCIA or module). During this time, the PLC HW uses as the maximum permissible time that can elapse before indicating a detected failure if it does not receive a response to any sent request.

With a Modbus TSX SCY21601 module, for example, the module Timeout is 3 seconds (1 second x 3 retries). When connecting 10 ATV31 devices on Modbus, a good setting would be to set the time to 200 ms and 1 retry (400 ms).

Software Timeout


The timeout setting is selected on the serializer (ModbusPort - component that manages the Modbus port). During this time, the SW uses as the maximum permissible time that can elapse before indicating a detected failure if it does not receive a response from any of the devices.

The following is the minimum time that needs to be selected:

With Hardware Timeout = tHW

Software Timeout > tHW x No. of simultaneous send operations (Modbus port `SimultaneousSends` public parameter).

The default number of simultaneous requests managed by the ModbusPort is 4. (The SGS on the Modbus port function generates a value of 4 for simultaneous transmissions.)

Software Timeout > 400 ms x 4 sim.sends  Software Timeout > 1600 ms

NOTE: Check the default value for the public variable of the ModbusPort function if you want to modify the software Timeout.

Device Timeout

This timeout is selected on the speed driver. During this time, the device uses as the maximum permissible time that can elapse before indicating a detected failure if it does not receive any requests. It tends to be a large number, but as a guideline:

Device Timeout > tHW x No. of devices connected to the Modbus port

For instance, if the network has 10 ATV 31 devices, check the timeout value that is configured in the parameters of each communications menu of the ATV 31:

Device Timeout > 400msec x 10 devices > Device Timeout > 4 s

The guidelines are as follows:

- The value of this timeout depends on the conditions that you establish for the ATV 31 when operating it. That is, a timeout of 4 seconds for each ATV 31 means that if the ATV 31 ceases to communicate, it stops or indicate a detected failure after 4 seconds.
- If there is a communications interruption with an ATV 31, the master attempts to establish communications with the device, and with that device only, during the time specified in the Hardware Timeout.

In this example, the master attempts to establish communications with the ATV 31 that is having a communication interruption of 400 ms.

As a result, if the device Timeout < Hardware Timeout (that is, lower than 400 ms in this example), the remaining network devices begins to drop out (indicate a detected error) consequently due to a timeout while the master is attempting to establish communications with the first ATV 31.

NOTE: The device Timeout need to be higher than the `Refresh` time (Refresh).

Refresh

Refresh is the data refresh time, that is, the time between the cyclic requests sent to each one of the devices.

This time must be sufficient for the data to be refreshed quickly but not so short as to saturate the bus with cyclic requests.

An estimated value between 200 and 1500 is necessary to refresh their data depending on the type of devices and the number of requests.

Configuring the Devices

Overview

This section explains the configuration of the devices.

ATV 31, ATV 312, and ATV 12

Description

The speed drives are connected to Modbus networks through an RJ45 connector.

Configure each Altivar 31, Altivar 312, and Altivar 12 devices with the same baud rate and with a different station number so that each device is uniquely identified on the Modbus bus.

From the actual ATV without a programming console, in the **Con** menu for ATV 31 units and the **Conf > FULL > Con** menu for ATV 312 and ATV 12 units:

- PrO : Protocol (Modbus RTU)
- Adrc or Add: Modbus address
- Bdr or tpr: Baud rate (19.2 = 19200 bps)
- Tfo: Communications mode (normally 8E1)
- tto : Communications Timeout (10 s by default)

NOTE: Communication parameters are recognized after the device is energized, that is, when the parameters are edited, the device has to be power cycled so that the parameters are applied.

ATV 71 and ATV 61

Description

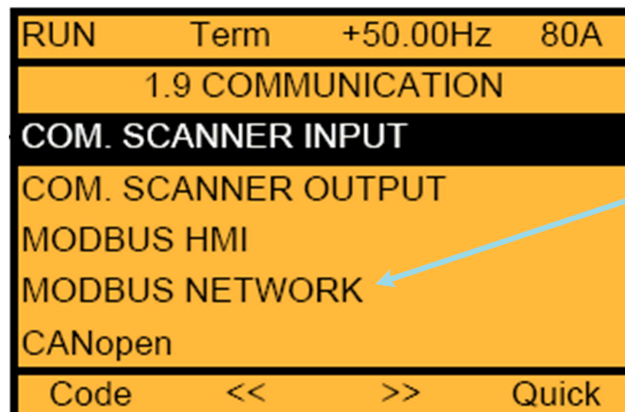
The ATV 71/61 speed driver is also connected to the Modbus network through a RJ45 connector.

That is, you can use an infrastructure identical to that used for the ATV 31.

Configure each Altivar 71 device with the same baud rate and with a different station number so that each device is uniquely identified on the Modbus bus.

From the actual ATV without a programming console, in the menu; or from the **1.9 COMMUNICATION** menu if there is a parameter configuration console:

- Add : Modbus address
- Tbr : Baud rate (19.2 = 19200bps)
- Tfo : Format (8-N-1 by default)
- tto : Communications Timeout (10 s by default)



NOTE: Communication parameters are recognized after the device is energized.

Tesys U

Description

The LULC033 is the communications module for Tesys U starters and controllers that is used for communicating with the M340 platform directly on the Modbus.

The Tesys U communications module is connected to the Modbus network through an RJ45 connector.

The available protection and control information depends on the control unit with which the Modbus communications module is associated.

The Tesys U communications module has 3 types of control unit:

- Standard (reference: LUCA)
- Advanced (references: LUCB/C/D, LUCBT/DT)
- Multifunction (references: LUCM/MT)

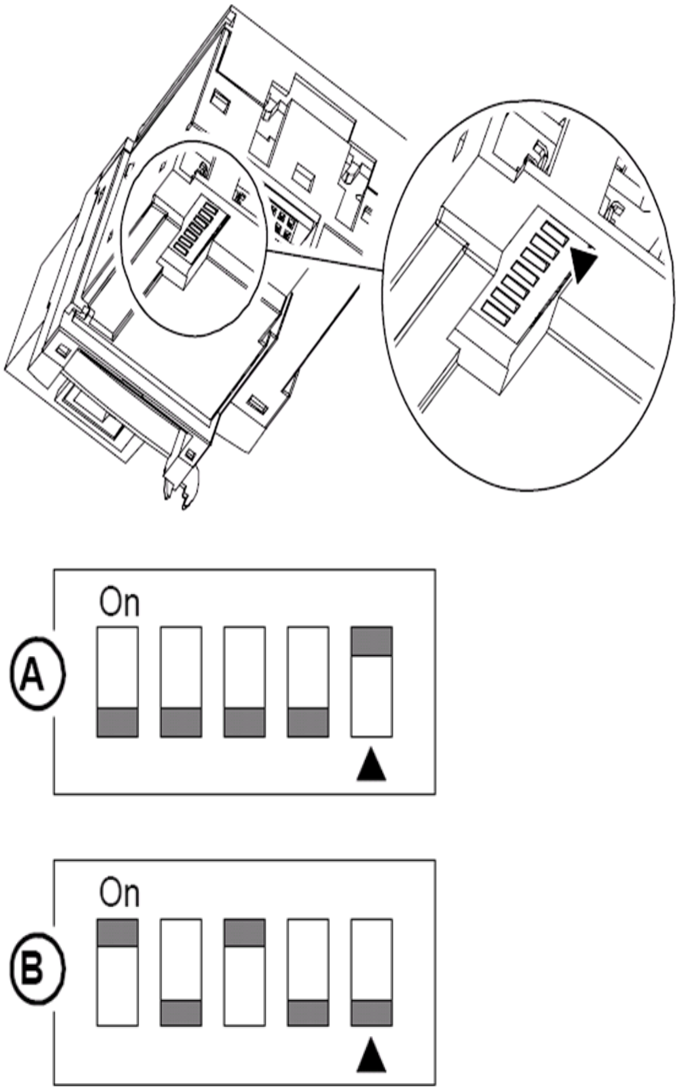
The following table shows the information and controls that can be accessed with each type of control unit:

Information/ Controls	Control unit		
	Standard (LUCA)	Advanced (LUCB/C/ D, LUCBT/DT)	Multifunction (LUCM/ MT)
Start and stop controls *	X	X	X
Statuses (ready, running, detected failure) *	X	X	X
Alarm *	—	X	X
Automatic resetting and remote resetting through the bus *	—	X	X
Indication of motor load *	—	X	X
Detected failure differentiation *	—	X	X
Parameter configuration and remote query for all functions	—	—	X
Log function	—	—	X

Information/ Controls	Control unit		
	Standard (LUCA)	Advanced (LUCB/C/ D, LUCBT/DT)	Multifunction (LUCM/ MT)
Monitoring function	–	–	X
Meaning of symbols used:			
X	Applicable		
–	Not Applicable		
* The control block implemented in the GPL enables to work with the information marked with an asterisk. Refer to <i>Tesys U profile</i> for more information regarding control based on the device and communications used.\			

Modbus Module Address

The address of the Modbus communications module is defined by switches that are located on the bottom of the module. To access these switches, module has to be turned around:



A) The module is delivered with address 1 (factory settings).

B) Configuration example for address 20.

The range of accepted values goes from 1 to 31, and the address code is binary. The least significant bit is on the right and is indicated with ▲.

NOTE: Addressing is recognized after the communications module is energized.

The control block implemented in the SGU is programmed to work with the factory default configuration.

The parameter configuration of the communications module enables you to define the following:

- The operating mode
- The reset mode for a reset due to a thermal overload event
- The correspondence between the communication module outputs and the LUTM controller inputs

The parameter configuration registers (Modbus protocol) can be accessed in read/write mode or with the Power Suite software.

The following table shows factory default values:

Item	Register	Factory settings	Meaning
Reset mode for a reset due to a detected thermal overload fault	602.0	1	Manual mode
Return mode for the control outputs in case of a communication interruption	682	2	Forced Stop Power base: OA1 and OA3 to 0 Controller base: 13 and 23 to 0
In power base, assignment for:	685 LSB	2	LO1 is the image of control bit 700.0
• LO1 output	685 MSB	12	OA1 is the image of control bit 704.0
• OA1 output	686 MSB	13	OA3 is the image of control bit 704.1
• OA3 output	687 LSB	12	13 is the image of control bit 704.0
In control base, assignment for:	687 LSB	13	23 is the image of control bit 704.1
• Output 13			
• Output 23			
Recovery mode after stop	688	0	The outputs revert to the state before the power supply interrupted.

Controller

If a controller is used, 7 inputs are used in a configuration by using a communications module.

Input	Function	Output	Comment
I.1	In Local mode: Each input controls the output relay.	13	The input image
I.2		23	The input image
I.1	In Remote through bus mode: These inputs are available for sending external information through the communication bus (writing to a register).	13	The register image
I.2		23	The register image
NOTE: If the voltage is too low, the output opens. Detected external and control unit problems do not affect the state of this relay.			
I.3 or I.4	These inputs are dedicated to the status return of the contactor controlled by the output relay.	–	–
I.5 RST (Reset)	This dedicated input allows you to reset the	–	Through Reset button on the front side of the box or panel.

Input	Function	Output	Comment
	controller manually after a inoperable unit.		
I.6 SF (System detected fault or inoperable system)	This dedicated input enables sending trip information when a protection mechanism that complements the controller is tripped.	–	Example: Reception of information provided by the Trip NC contact: => From a circuit breaker, from a relay with a probe, from a voltage relay => from a chain of these contacts.
I.10	In Local or Remote through bus mode. This input sends external information through the communication bus (writing to a register).		
	In Local mode.	Output	= The input image
	In Remote through bus mode.	Output	= The register image
	In combined mode: This input sends local or remote through the bus information and allows the controller to manage the control priorities.		
	If I.10 = 1: Local mode.	Output	= The input image
	If I.10 = 0: Remote through bus mode:	Output	= The register image

This table describes the dedicated input:

Input	Function	Comment
I.7 SR (System ready)	This dedicated input allows system availability to be communicated through the bus. If the correlation Ready bit is not used, I.7 is available for sending the rest of the information.	Example: Reception of information provided by the circuit breaker Ready NO contact.
I.8	This input sends external information through the communication bus (writing to a register). It has no effect on the controller operation.	Example: Status of the Emergency Stop button.
I.9		Example: In case of a cabinet in test position panel.

Tesys T

Description

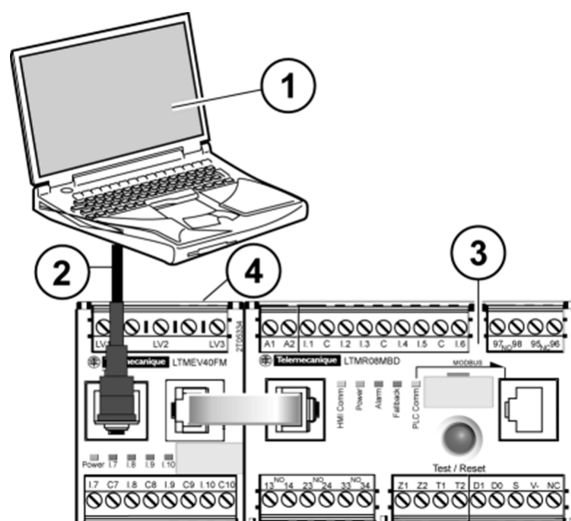
To connect a Tesys T device on a Modbus network, the following communication parameters need to be configured.

Parameter	Value
Network	Modbus (Device Settings menu)
Address	From 1 to 247 (Network Port menu)
Baud rate	From 1200 bps to 19200 bps (Network Port menu)
Parity	Even or No Parity (Network Port menu)

Use PowerSuite program to configure the Tesys T.

Configuration with Expansion Module

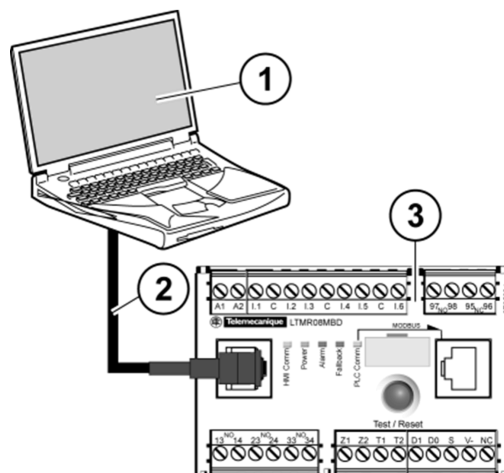
This figure shows the connection of a Tesys T module with an expansion module.



- 1 PC with PowerSuite software
- 2 VW3 A8 106 cable
- 3 LMT R controller
- 4 Expansion module

Configuration without Expansion Module

This figure shows the connection of a Tesys T module without an expansion module.



- 1 PC with PowerSuite software
- 2 VW3 A8 106 cable
- 3 LMT R controller

ATS 22 and ATS 48

Description

NOTE: A reference to the ATS 48 unit is also referred to the ATS 22 unit unless otherwise specified.

The ATS 48 starter is connected to the Modbus network through an RJ45 connector.

Configure each ATS 48 device with the same baud rate and with a different station number so that each device is uniquely identified on the Modbus bus.

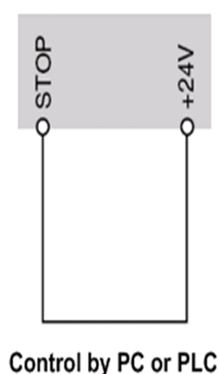
Access the **COP** menu and configure the following:

Parameter	Description
Add	ATS 48 Modbus address within the Modbus network. Address from 0 to 31 for ATS 48 units and from 0 to 247 for ATS 22 units.
tbr	Network baud rate at 4800, 9600, or 19200 bps.
FOr	Configure the amount of communication bits (8 bits), parity (even, odd, or no parity), and 1 or 2 Stop bits.
TLP/TTO	(TTO for ATS 22 units) Time out. You can configure with a value of 0.1 to 60.0 in 0.1 s increments.
Ctrl	Only for ATS 22 units. Enables you to configure whether control is Modbus-based or screw terminal-based. Configure dbS for Modbus-based control.

The timeout configured on the ATS 48 device has to be longer than the time configured in the `Refresh` variable of the generated component.

NOTE: Communication parameters are recognized after the device is energized.

To use PLC-based control, the following connection needs to be made.



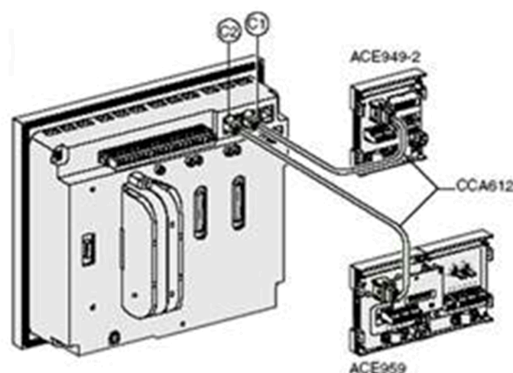
SEPAM

Description

Modbus communication allows connecting Sepam units to a monitor that is equipped with a master Modbus communications path.

To connect this unit, an ACE949-2 (2-wire RS485 network connection interface) or ACE949 (4-wire RS485 network connection interface) interface module with a 12 VCC or 24 VCC power supply is required.

This figure shows the Sepam series 80 configuration:



To connect a Sepam device to a Modbus network, configure the following communication parameters with the configuration software:

Parameter	Value
Bus	Modbus
Address	From 1 to 247
Baud Rate	From 1200 bps to 19200 bps
Parity	Even, odd, or no parity

PM

Description

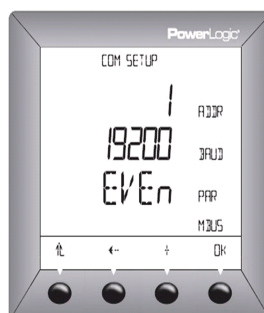
Configure each PM device with the same baud rate and with a different station number so that each device is uniquely identified on the Modbus bus.

After accessing the internal menu of the signal analyzer, configure the following:

Set Up Communications

1. In SETUP mode, press \rightarrow until COM is visible.
2. Press COM.
3. Enter the ADDR (meter address): 1 to 247.
4. Press OK.
5. Select the BAUD (baud rate): 2400, 4800, 9600, or 19200.
6. Press OK.
7. Select the parity: EVEN, ODD, or NONE.
8. Press OK to return to the SETUP MODE screen.

NOTE: Default values are displayed.



Compact and Masterpact

Description

Both Compact and Masterpact units are configured the same way in the Modbus section. This configuration is implemented with the TRV00210 device.

This device automatically detects the baud rate of the bus as long as it is working with 8 bits, even parity, and 1 stop bit.



The rotary selectors on the front of the device are used to configure the Modbus node address (address from 1 to 99). To control the device remotely, the last selector has to be pointing at the open lock symbol.

Modbus Serial - Explicit

Overview

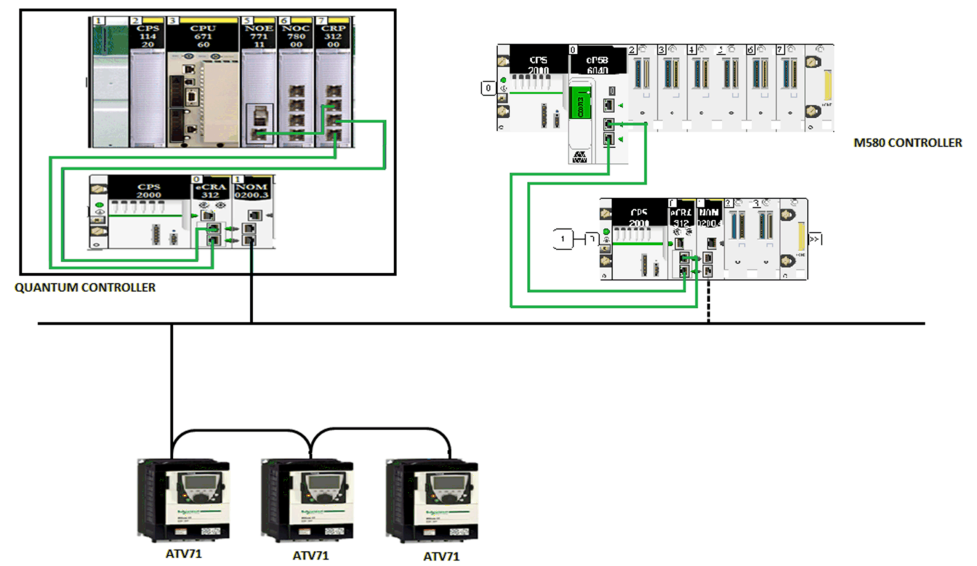
This section describes the procedure to create an application using Modbus communication technology (Explicit).

Creating an Application using Modbus Communication Technology (Explicit)

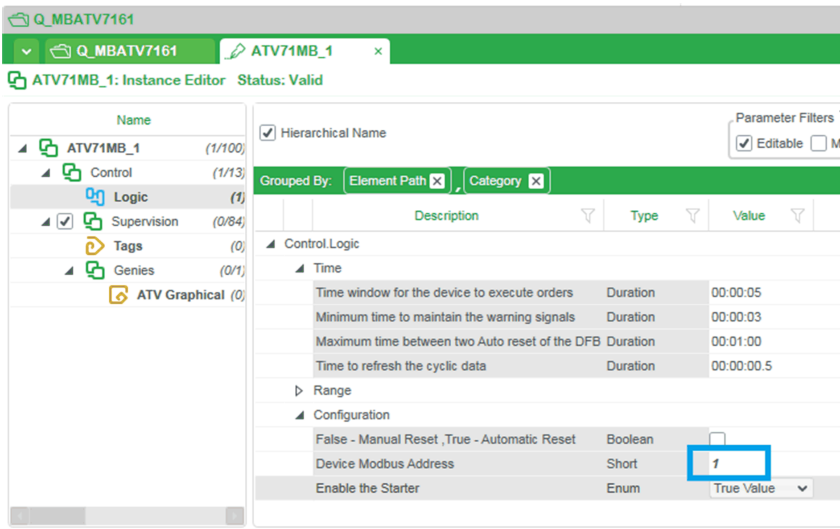
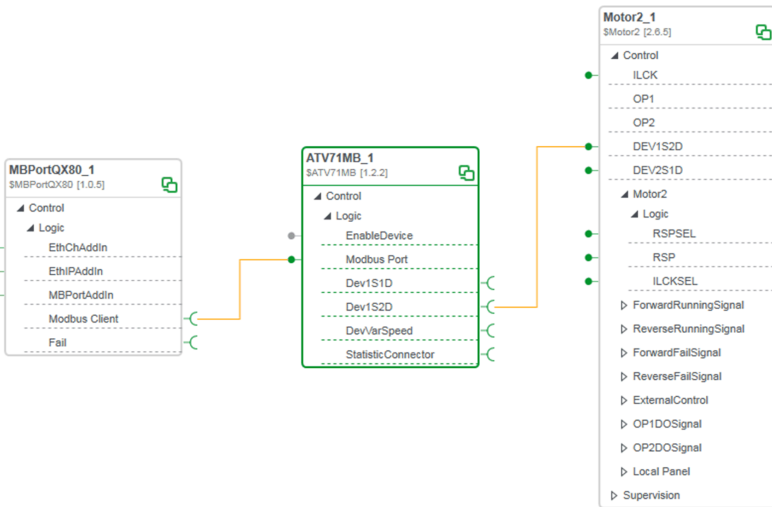
Overview

In this example, *ATV71* and Quantum controller is used as a reference. However, same procedure can be followed for creating an application with any other Modbus devices or controller.

Architecture



Application Explorer

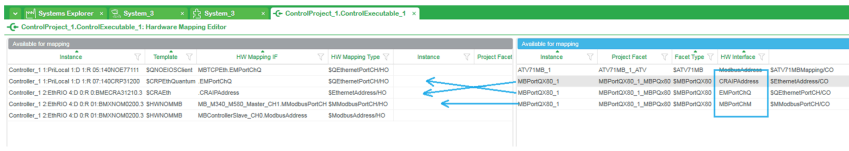
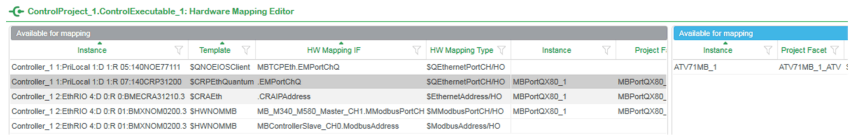
Step	Action
1	Instantiate ATV71MB , Motor2 and MBPortQX80 templates.
2	<p>Right-click on ATV71MB instance and navigate to Properties. Enter the address in Device Modbus Address field under Logic Section as shown below.</p> 
3	<p>Right click on instance of ATV71MB and select Edit Links. Link the objects as shown in below.</p> 

Topology Explorer

Step	Action
1	In Topology, create PLC hardware configuration with Quantum controller.
2	Link the controller to Ethernet network.

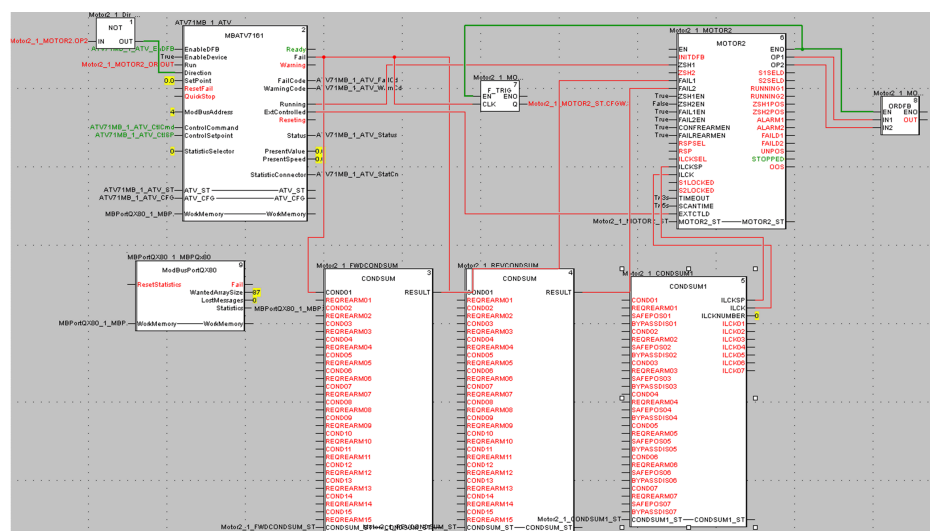
Project Explorer

Step	Action
1	Create a project for Quantum controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .

Step	Action
4	Navigate to Executables section, right-click on Executables and select Map Hardware .
5	<p>Map the application interfaces by drag and drop in front of respective field, as shown below.</p> <p>Map:</p> <ul style="list-style-type: none"> CRAIPAddress to CRAIPAddress EMPortChQ to EMPortChQ MBPortChM to MB_M340_M580_Master_CH1.MModbusPortCH <p>This mapping is required to pass the IP address of CRA, Port address of CRP and Port address of NOM module to DFB.</p>  <p>After Mapping</p> 
6	Build the control project and open the built project.
7	In Function Block section, check public parameters for MBPortQX80 block.
8	Verify the IP address of CRA , Port address of CRP and Port address of NOM module is correct.

Read/Write Devices

Now, you should be able to read/write devices.

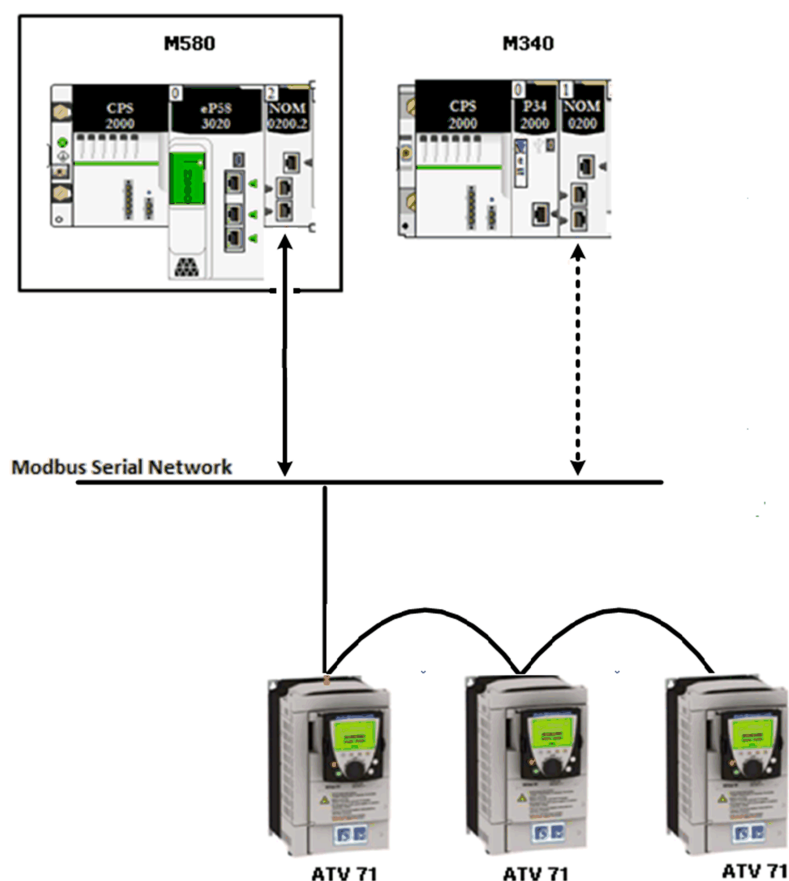


Creating an Application using Modbus Communication Technology (Explicit) with M580/M340 Controller

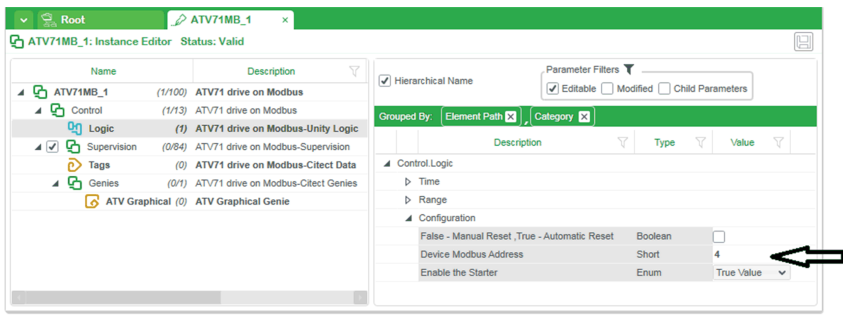
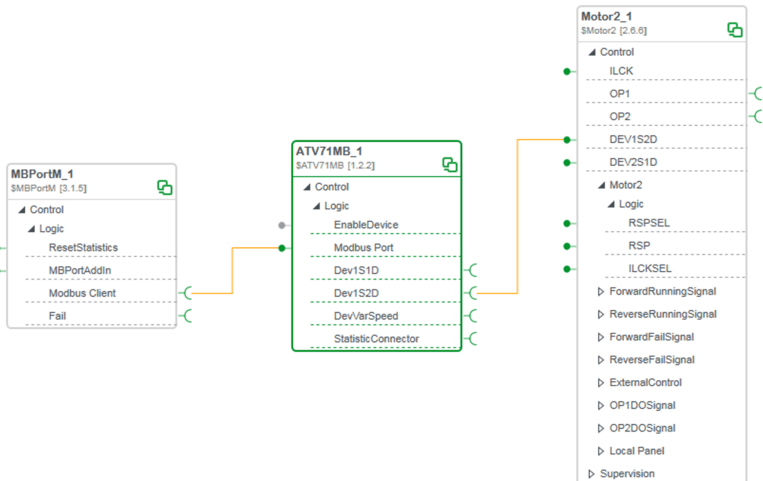
Overview

In this example, ATV71 and M580 controller is used as a reference. However, same procedure can be followed for creating an application with any other Modbus devices or controller.

Architecture



Application Explorer

Step	Action
1	Instantiate ATV71MB , Motor and MBPortM templates in Application Explorer .
2	<p>In Application Explorer, right-click on instance of ATV71MB and select properties to set the Modbus address of the ATV71.</p> 
3	<p>Right-click on instance of ATV71MB and select Edit Links. Link the objects as shown in below.</p> 

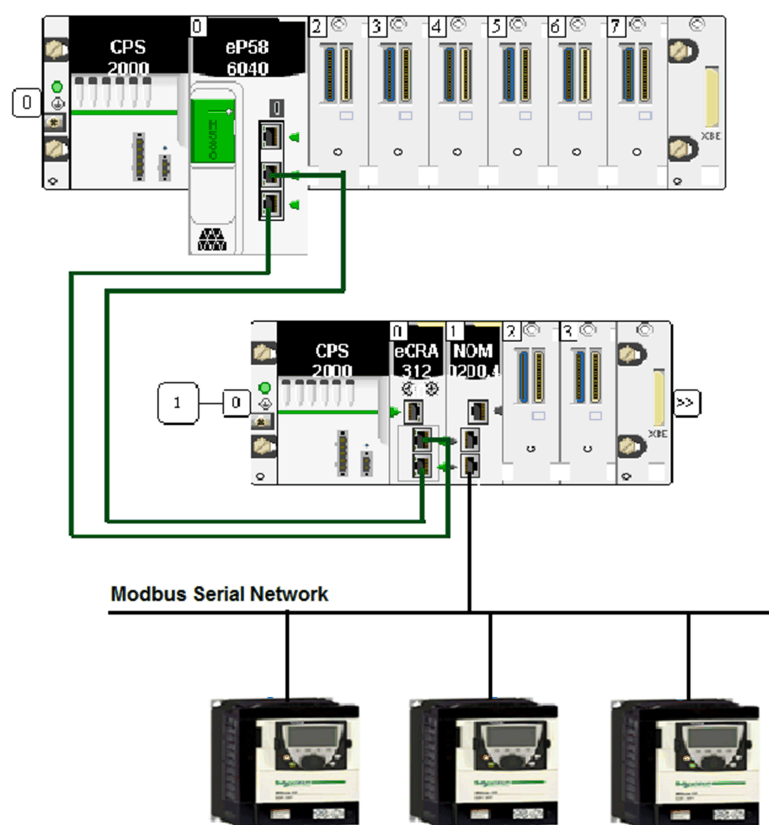
Topology Explorer

Step	Action
1	In Topology, create PLC hardware configuration with M580 controller.
2	Link the controller to Ethernet network.

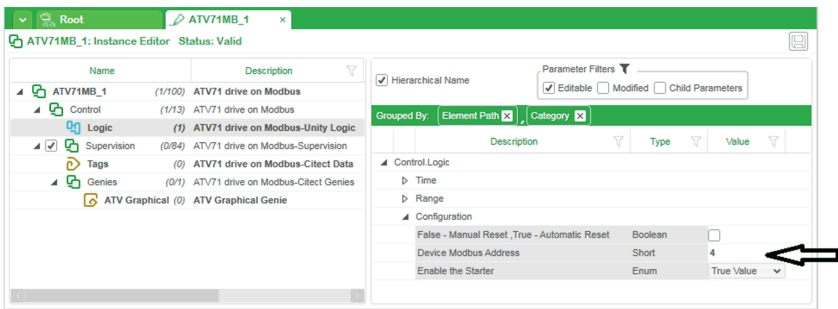
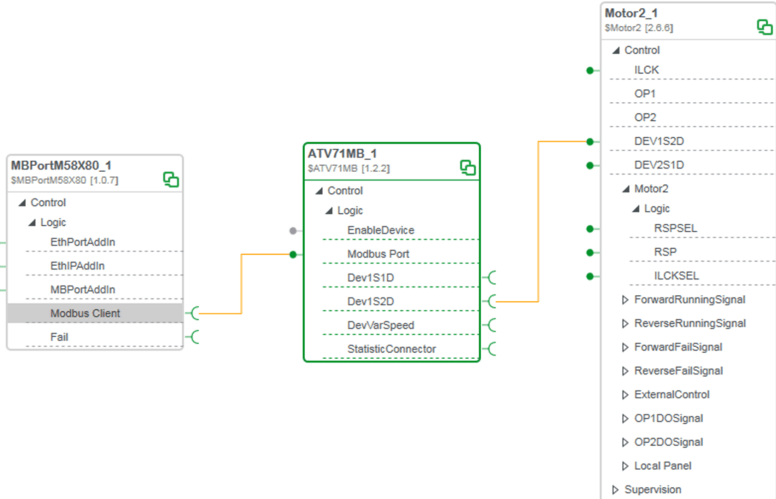
Project Explorer

Step	Action
1	Create a project for M580 controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .
4	Navigate to Executables section, right click on Executables and select Map Hardware .
5	<p>Map the application interfaces by drag and drop in front of the respective field, as shown below.</p> <p>Map - MBPortM_1 to MB_M340_M580_Master_CH1.MModbusPortCH of \$HWNOMMB.</p> <p>This mapping is required to pass Port address to DFB. In this example we are using NOM port (Channel 0) for communication.</p>

Architecture



Application Explorer

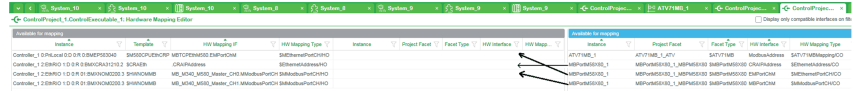
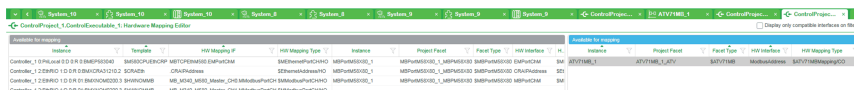
Step	Action
1	Instantiate ATV71MB , Motor2 , MBPortM58X80 templates in Application Explorer .
2	In Application Explorer , right-click on instance of ATV71MB and select properties to set the Modbus address of the ATV71 . 
3	In Application Explorer , right click on instance of ATV71MB and select properties to set the Modbus address of the ATV71 . 

Topology Explorer

Step	Action
1	In Topology, create PLC hardware configuration with M580 controller.
2	Link the controller to Ethernet network.

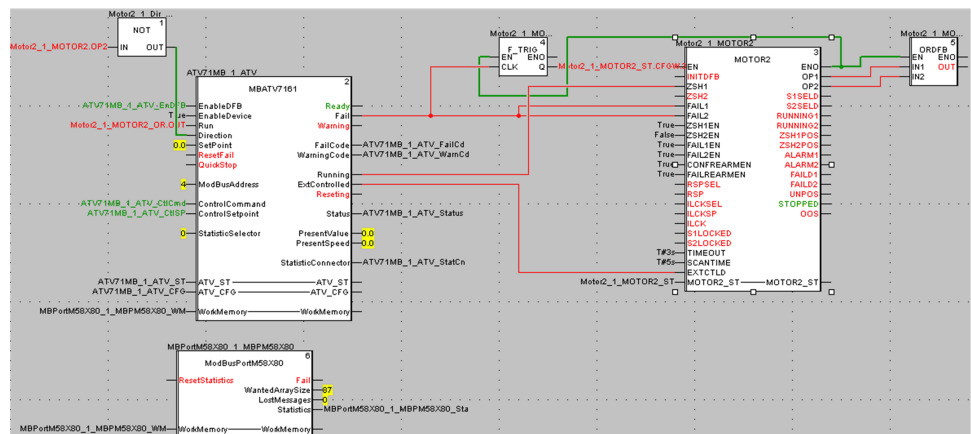
Project Explorer

Step	Action
1	Create a project for M580 controller.
2	Assign the facets and generate the control project.
3	Create Executables and do Map Service .
4	Navigate to Executables section, right click on Executables and select Map Hardware .
5	Map the application interfaces by drag and drop in front of respective field, as shown below. MAP: <ul style="list-style-type: none"> CRAIPAddress to .CRAIPAddress EMPortChQ to .EMPortCh

Step	Action																																																																		
	<ul style="list-style-type: none"> MBPortChM to .MB_M340_M580_Master_CH1.MModbusPortCH0 <p>This mapping is required to pass M580 PLC Port address, CRA IP address and NOM port address to DFB. In this example, M580 CPU port for communication with CRA is used and NOM (Channel 0) is used to communicate with ATV71.</p>  <p>After Mapping,</p> 																																																																		
6	Build the control project and open the built project.																																																																		
7	<p>In Function block section, check public parameters for MBPORTM block.</p> <table border="1"> <thead> <tr> <th>Name</th><th>no.</th><th>Type</th><th>Value</th><th>Comment</th></tr> </thead> <tbody> <tr> <td>ATV71MB_1 ATV</td><td></td><td>no.</td><td></td><td></td></tr> <tr> <td>MBPortM580_1_MBP580_1</td><td></td><td>ModBusP...</td><td></td><td></td></tr> <tr> <td><inputs></td><td></td><td></td><td></td><td></td></tr> <tr> <td><outputs></td><td></td><td></td><td></td><td></td></tr> <tr> <td><inputs/outputs></td><td></td><td></td><td></td><td></td></tr> <tr> <td><public></td><td></td><td></td><td></td><td></td></tr> <tr> <td>Timeout</td><td></td><td>TIME</td><td>T#2s</td><td>Time to wait a response. Must be greater than the hardware timeout</td></tr> <tr> <td>EthPortAddress</td><td></td><td>STRING</td><td>0.0.3</td><td>Ethernet Port Address - Rack.Slot.Channel</td></tr> <tr> <td>CRAIPAddress</td><td></td><td>STRING</td><td>(192.168.1.239)</td><td>CRA IP Address - (ip.ip.ip)</td></tr> <tr> <td>MBPortAddress</td><td></td><td>STRING</td><td>0.01.0</td><td>Modbus PortAddress - Rack.Slot.Channel</td></tr> <tr> <td>SimultaneousSends</td><td></td><td>INT</td><td>4</td><td>Number of active sendings</td></tr> </tbody> </table> <p>Verify the port addressing as per below table:</p> <table border="1"> <thead> <tr> <th colspan="2">Controller</th><th>Port Addressing</th></tr> </thead> <tbody> <tr> <td>M580</td><td>CPU/NOE/NOC</td><td>'Rack.Slot.channel'</td></tr> </tbody> </table>	Name	no.	Type	Value	Comment	ATV71MB_1 ATV		no.			MBPortM580_1_MBP580_1		ModBusP...			<inputs>					<outputs>					<inputs/outputs>					<public>					Timeout		TIME	T#2s	Time to wait a response. Must be greater than the hardware timeout	EthPortAddress		STRING	0.0.3	Ethernet Port Address - Rack.Slot.Channel	CRAIPAddress		STRING	(192.168.1.239)	CRA IP Address - (ip.ip.ip)	MBPortAddress		STRING	0.01.0	Modbus PortAddress - Rack.Slot.Channel	SimultaneousSends		INT	4	Number of active sendings	Controller		Port Addressing	M580	CPU/NOE/NOC	'Rack.Slot.channel'
Name	no.	Type	Value	Comment																																																															
ATV71MB_1 ATV		no.																																																																	
MBPortM580_1_MBP580_1		ModBusP...																																																																	
<inputs>																																																																			
<outputs>																																																																			
<inputs/outputs>																																																																			
<public>																																																																			
Timeout		TIME	T#2s	Time to wait a response. Must be greater than the hardware timeout																																																															
EthPortAddress		STRING	0.0.3	Ethernet Port Address - Rack.Slot.Channel																																																															
CRAIPAddress		STRING	(192.168.1.239)	CRA IP Address - (ip.ip.ip)																																																															
MBPortAddress		STRING	0.01.0	Modbus PortAddress - Rack.Slot.Channel																																																															
SimultaneousSends		INT	4	Number of active sendings																																																															
Controller		Port Addressing																																																																	
M580	CPU/NOE/NOC	'Rack.Slot.channel'																																																																	

Read/Write Devices

Now, you should be able to read/write devices.



Profibus Technology

What’s in This Chapter

Steps to Create a Working Project with Profibus Templates 308

Overview

This chapter describes the configuration and setup details of the devices connected through Profibus.

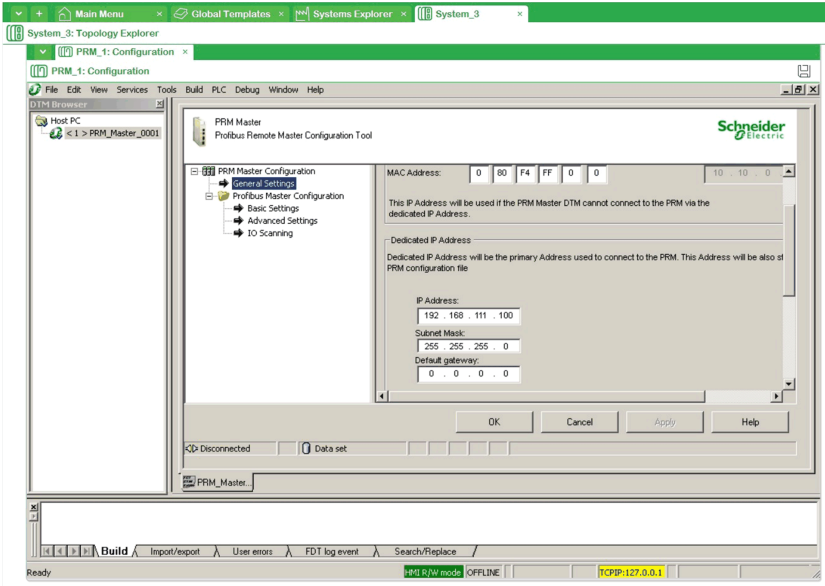
Steps to Create a Working Project with Profibus Templates

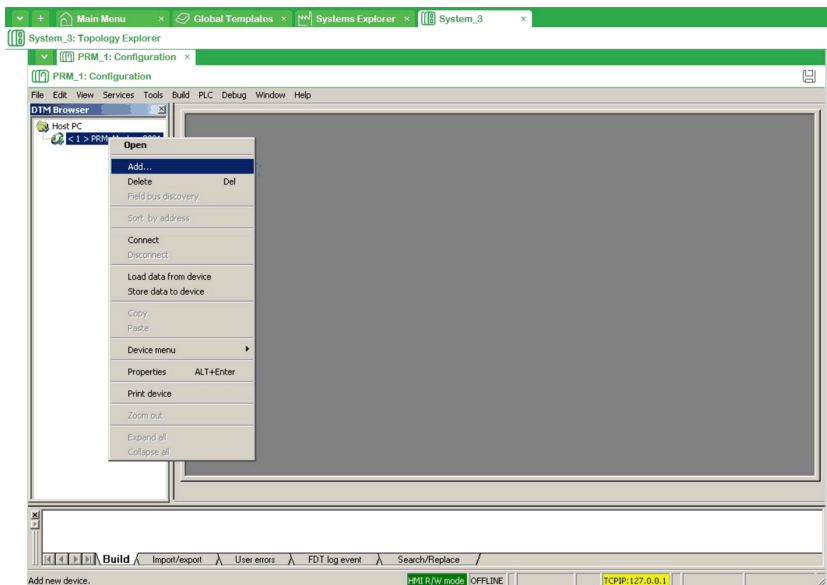
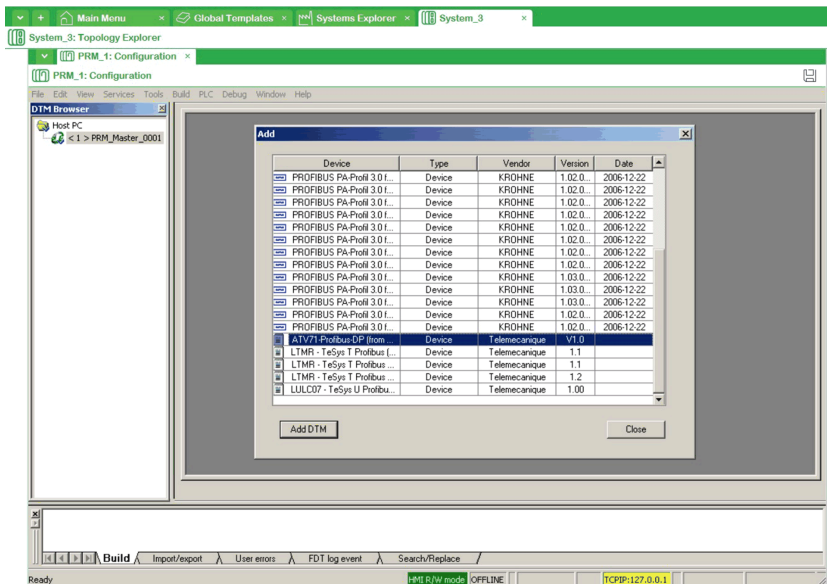
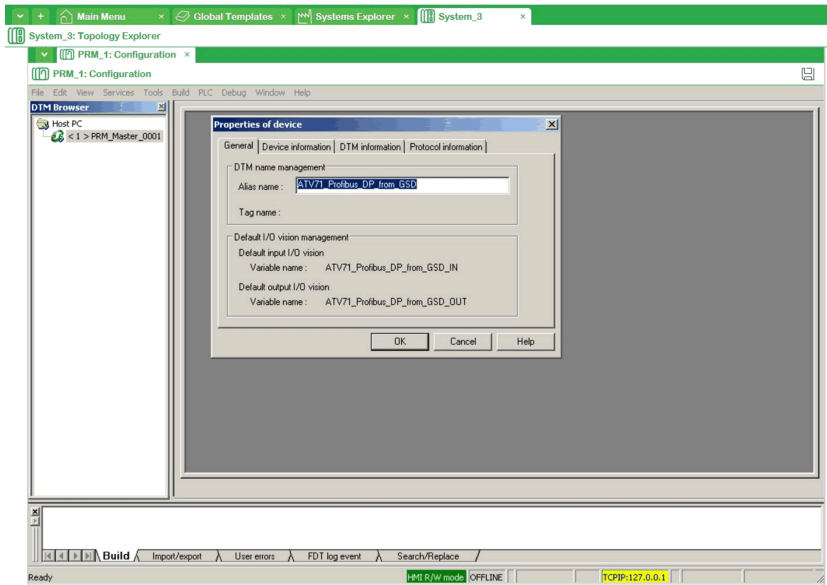
Overview

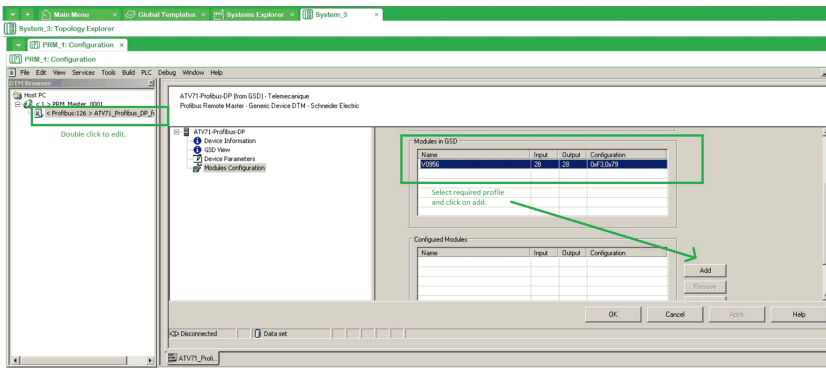
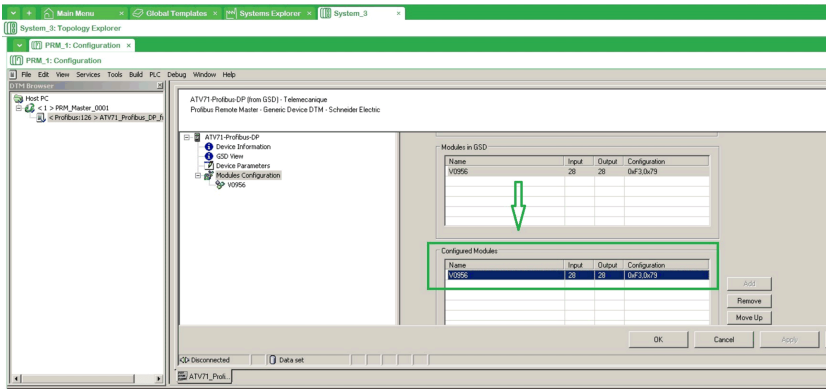
This section provides the PRM configuration and project creation details involving the Profibus templates.

Configuring PRM Device

The following table describes the steps for configuring a PRM device:

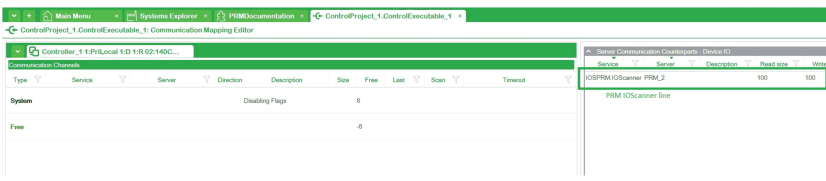
Step	Action
1	Create a PRM device in Topology Explorer > System > Create PRM Profibus DP .
2	Configure the created PRM device PRM (right click) > Configure . It opens the Configuration editor for the topological entity.
3	Configure the PRM hardware settings as required. 
4	In the DTM Configuration window PRM (right click) > Add .

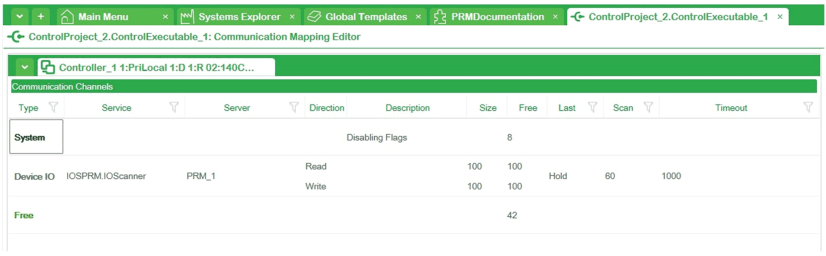
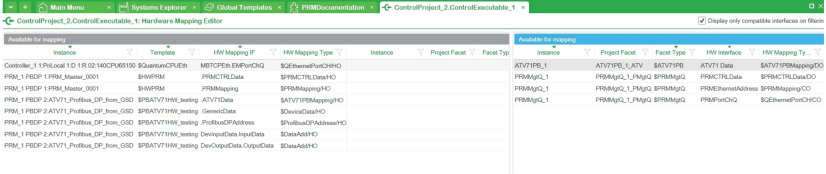
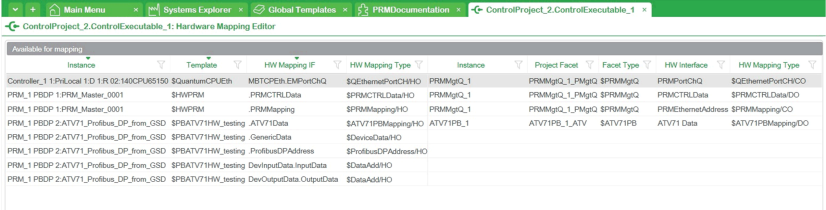
Step	Action																																																																																										
	 <p>The screenshot shows the SIMATIC Manager interface with the 'System_3: Topology Explorer' window. The 'DTM Browser' is open, and a right-click context menu is displayed. The menu options include: Add..., Delete, Field bus discovery, Sort: by address, Connect, Disconnect, Load data from device, Store data to device, Copy, Paste, Device menu, Properties (ALT+Enter), Print device, Zoom out, Expand all, and Collapse all. The 'Add...' option is highlighted.</p>																																																																																										
5	<p>Select the required GSD file to be added based on the device. Here ATV71 is shown in the following figure.)</p>  <p>The screenshot shows the SIMATIC Manager interface with the 'System_3: Topology Explorer' window. The 'DTM Browser' is open, and the 'Add' dialog box is displayed. The dialog box contains a table with the following columns: Device, Type, Vendor, Version, and Date. The table lists several devices, including 'PROFIBUS PA-Profit 3.0.1...' and 'ATV71 Profibus DP from...'. The 'ATV71 Profibus DP from...' device is selected. The 'Add DTM' button is visible at the bottom of the dialog box.</p> <table border="1"><thead><tr><th>Device</th><th>Type</th><th>Vendor</th><th>Version</th><th>Date</th></tr></thead><tbody><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>PROFIBUS PA-Profit 3.0.1...</td><td>Device</td><td>KROHNE</td><td>1.02.0...</td><td>2006-12-22</td></tr><tr><td>ATV71 Profibus DP from...</td><td>Device</td><td>Telemecanique</td><td>V1.0</td><td></td></tr><tr><td>LTMR - TeSys T Profibus...</td><td>Device</td><td>Telemecanique</td><td>1.1</td><td></td></tr><tr><td>LTMR - TeSys T Profibus...</td><td>Device</td><td>Telemecanique</td><td>1.1</td><td></td></tr><tr><td>LTMR - TeSys T Profibus...</td><td>Device</td><td>Telemecanique</td><td>1.2</td><td></td></tr><tr><td>LULC07 - TeSys U Profibus...</td><td>Device</td><td>Telemecanique</td><td>1.00</td><td></td></tr></tbody></table>	Device	Type	Vendor	Version	Date	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22	ATV71 Profibus DP from...	Device	Telemecanique	V1.0		LTMR - TeSys T Profibus...	Device	Telemecanique	1.1		LTMR - TeSys T Profibus...	Device	Telemecanique	1.1		LTMR - TeSys T Profibus...	Device	Telemecanique	1.2		LULC07 - TeSys U Profibus...	Device	Telemecanique	1.00	
Device	Type	Vendor	Version	Date																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
PROFIBUS PA-Profit 3.0.1...	Device	KROHNE	1.02.0...	2006-12-22																																																																																							
ATV71 Profibus DP from...	Device	Telemecanique	V1.0																																																																																								
LTMR - TeSys T Profibus...	Device	Telemecanique	1.1																																																																																								
LTMR - TeSys T Profibus...	Device	Telemecanique	1.1																																																																																								
LTMR - TeSys T Profibus...	Device	Telemecanique	1.2																																																																																								
LULC07 - TeSys U Profibus...	Device	Telemecanique	1.00																																																																																								
6	<p>The selected GSD is displayed.</p>  <p>The screenshot shows the SIMATIC Manager interface with the 'System_3: Topology Explorer' window. The 'DTM Browser' is open, and the 'Properties of device' dialog box is displayed. The dialog box has tabs for General, Device information, DTM information, and Protocol information. The 'DTM information' tab is selected. The 'DTM name management' section shows the 'Alias name' as 'ATV71 Profibus DP from GSD' and the 'Tag name' as 'ATV71 Profibus DP from GSD'. The 'Default I/O vision management' section shows the 'Variable name' as 'ATV71_Profibus_DP_from_GSD_IN' and the 'Default output I/O vision' section shows the 'Variable name' as 'ATV71_Profibus_DP_from_GSD_OUT'.</p>																																																																																										

Step	Action
7	<p>Double-click and select the profile to be added. Click Add.</p> <p>NOTE: In ATV, only one profile exists.</p> 
8	<p>Selected profile appears in the Configured Modules tab.</p> 
9	Assign the required Profibus address and save the Configuration .
10	Complete the required settings like physical connections and close Topological Editor .

Creating a Project with Profibus Template

The following table describes the steps for creating a project with Profibus templates:

Step	Action
1	Instantiate the required control module template (for example, ATV71PB) in Application Editor .
2	Configure the instance as required.
3	<p>Click Projects Explorer > Assign > Generate > Create Executable > Map Controller and Communication.</p> <p>The PRM IO scanner line would be available for mapping.</p> 

Step	Action
4	<p>The following figure shows the Communication Mapping Editor after mapping:</p> 
5	<p>Map the hardware.</p> <p>The following figure shows the Hardware Mapping Editor before mapping:</p>  <p>NOTE: Check the device-related hardware template is mapped to device and the PRM hardware mappings are correct.</p>
6	<p>The PRMPortChx (where x= Q or M, that is, Quantum or M340) is mapped to the Ethernet port of Controller/NOE to which the PRM is communicating. The following figure shows the Hardware Mapping Editor after mapping:</p> 
7	Build all.

CANOpen Technology

What's in This Chapter

Advantys Configuration in EcoStruxure Process Expert..... 312

Overview

This chapter describes the configuration and setup details of the devices connected through CANOpen.

Advantys Configuration in EcoStruxure Process Expert

Overview

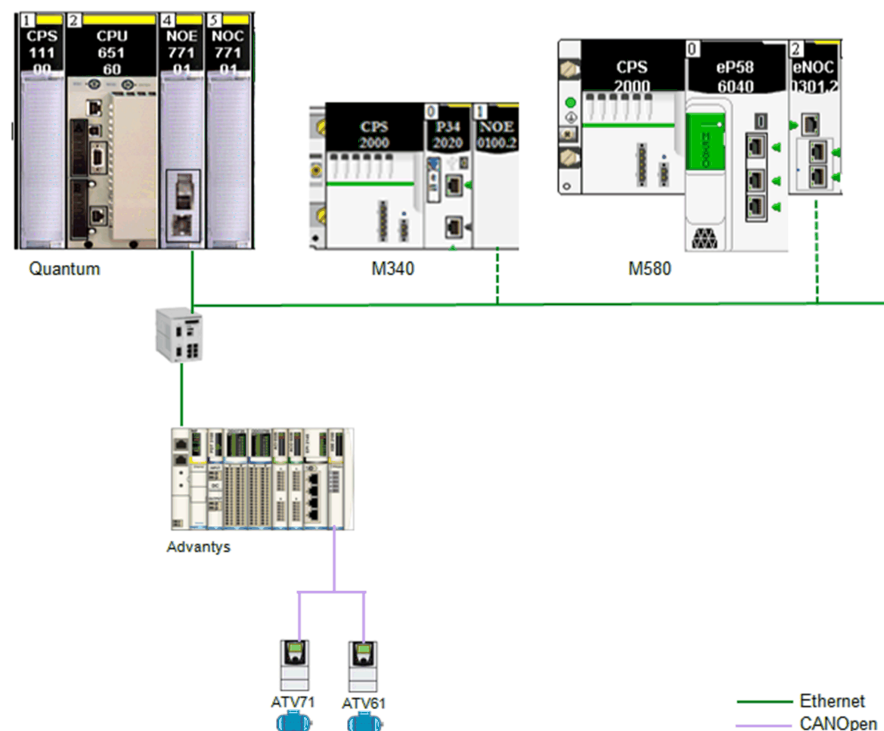
This section describes the procedure to configure Advantys island. In this example ATV71 is used as a reference. However, same procedure can be used for creating an application with other devices.

Advantys Configuration in EcoStruxure Process Expert

Overview

This document explains the steps to be followed for configuring Advantys island using IOScanning. ATV71 is used as a reference. However, same procedure can be used for creating an application with other devices

Architecture

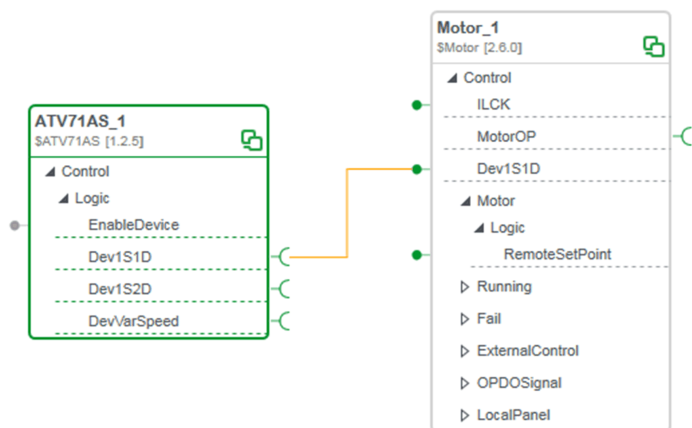


Prerequisites

Step	Action
1	In Project Explorer , create a project for Quantum controller.
2	In Topology, create PLC hardware configuration with <code>IOScanning</code> enabled for communication port.
3	Configure IP address in Advantys island. IP address is 192.168.1.114.

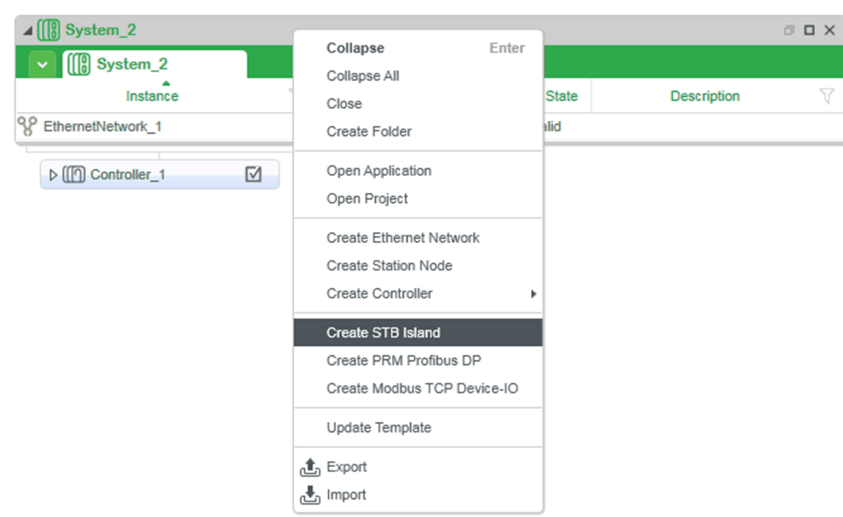
Application Explorer

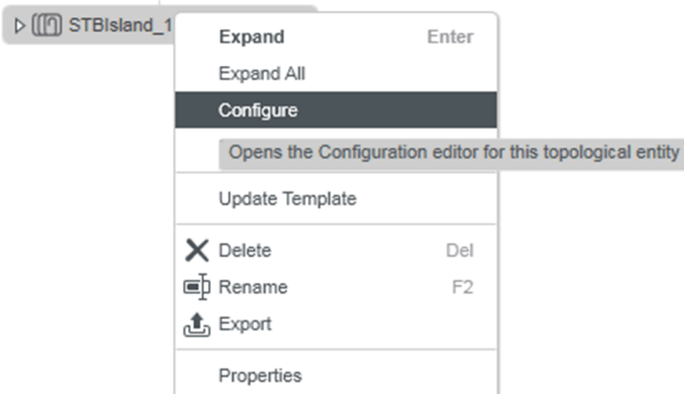
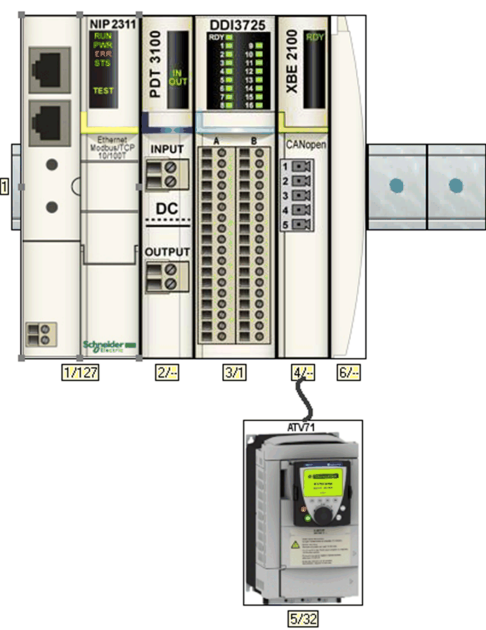
Step	Action
1	Instantiate <code>ATV71AS</code> and <code>Motor</code> templates in Application Explorer .
2	Right click on Instance of <code>ATV71AS</code> and select Edit Links . Link the objects as shown below.

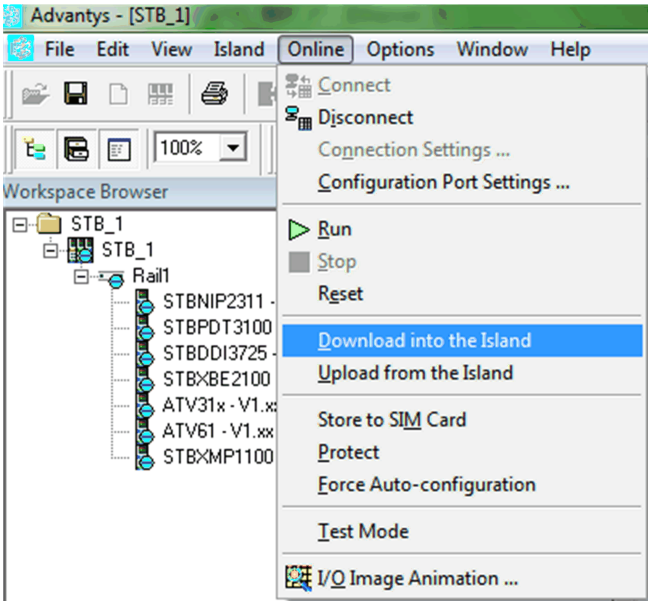

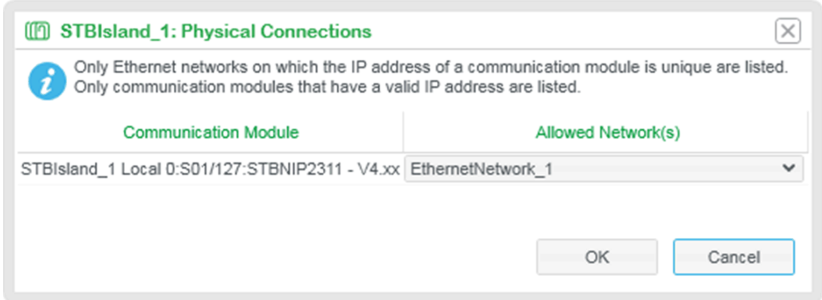


Topology Explorer

Step	Action
1	Navigate to Topology, right-click on System and select Create STB Island . New hardware instance, STBIsland_1 is created under topology explorer.
2	Configure the Advantys island by right-click as shown below.

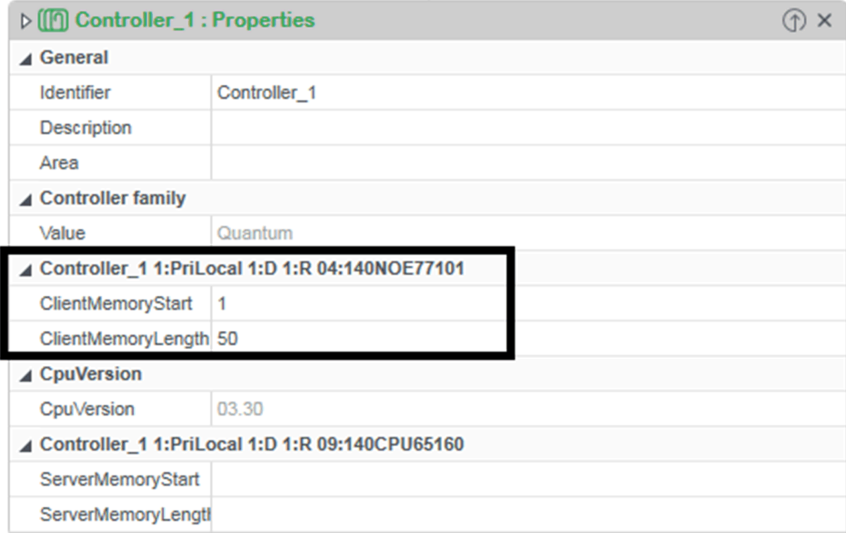


Step	Action
	
3	<p>The architecture shown below is configured with a CANOpen master, ATV71 drive is selected under CANOpen master. Check that required configurations are done in the ATV71 drive to communicate on CANOpen protocol.</p>  <p>NOTE: User should make the required configuration as per the application.</p>
4	<p>Perform Build All, once configuration is completed and download the same to the Advantys island as shown below. Island is healthy status appears after download is completed and starts working without any detected error. Once download is done successfully, close the window to save the configuration.</p>

Step	Action
	
5	<p>Right-click on STBIsland_1 and select Properties to configure the IP address. Set the required IP address and subnet mask of Advantys Island device as shown below.</p> 
6	<p>Right-click on STBIsland_1 instance and goto physical connections and select the Ethernet Network Address in Allowed Network(s) as shown.</p> 

IO Scanning

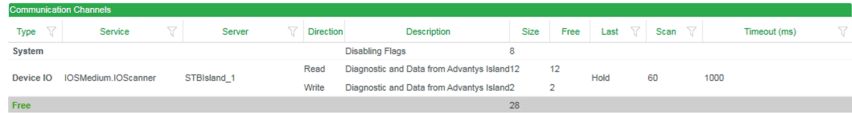
In this example, Advantys is connected to NOE module of the Quantum controller, so the configuration has to be made in the same module.

Step	Action
1	In Topology, right click on Controller instance and open Properties .
2	<p>Define ClientMemoryStart address and ClientMemoryLength to access under the NOE.</p>  <p>NOTE: ClientMemoryLength can be given as per the user requirement. For example, the above configuration occupies 14 ClientMemoryLength. User will get to know the required length when the Communication Mapping is done.</p>

Project Explorer

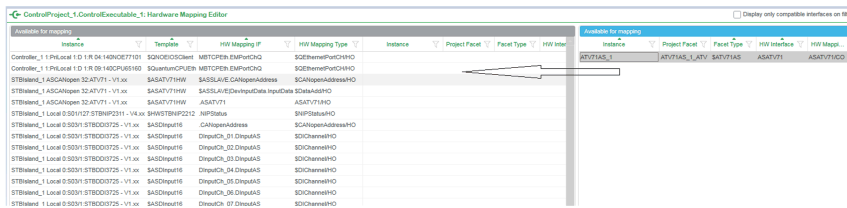
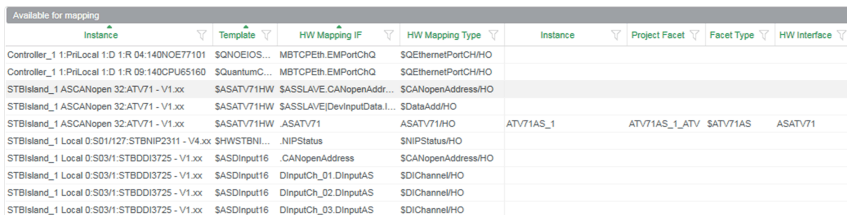
Map Communication

Step	Action
1	Navigate to executables section, right-click on Executables and select Map Communication .
2	Map the communication counterparts - IO device to communication channels.
3	A dialog box opens to allow the user to configure the scan rate, timeout, and so on. Select the required values and click OK .
4	Now, Advantys is available under Communication Channels and it occupies memory length required for this application.



Type	Service	Server	Direction	Description	Size	Free	Last	Scan	Timeout (ms)
System				Disabling Flags	8				
Device IO	IOSMedium.IOScanner	STBIsland_1	Read	Diagnostic and Data from Advantys Island12	12	Hold	60	1000	
			Write	Diagnostic and Data from Advantys Island2	2				
Free					28				

Map Hardware

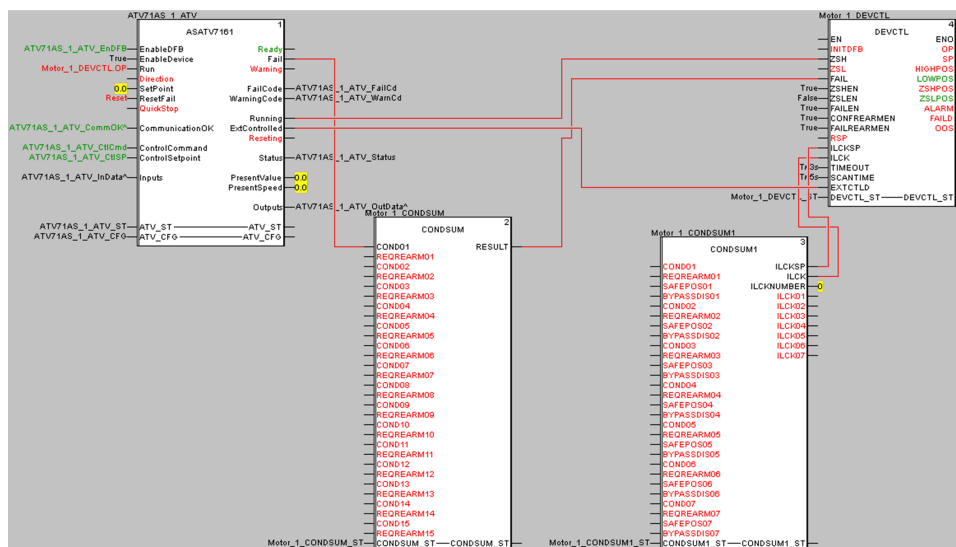
Step	Action
1	Navigate to Executables section, right-click on Executables and select Map Hardware .
2	Map the hardware interfaces by drag and drop of application interfaces, as shown below. <div>  <p>Map ATV71AS Data to ATV71/HO of \$ASATV71HW. This mapping is required to pass ATV71 data to DFB.</p> </div>
3	Once hardware mapping is done, ATV71 instance will be moved as shown below. <div>  </div>

Generation and Build

Step	Action
1	Assign the facet and generate and build the control project.
2	Perform build all and deploy the built project.

Read/Write Devices

Now, user should be able to read/write devices. If configurations are correct, *CommunicationOK* bit will be high. Now, user can perform *ResetFail* and make the DFB into *Ready* state and control the device using the DFB.



Index

A

Advantys Configuration in EcoStruxure Process Expert	
Advantys Configuration in EcoStruxure Process	
Expert	312
Altistart	
Modbus communication configuration	290
Altivar	
Modbus communication configuration	290
Altivar 71/61	
configuring Altivar71/61 Ethernet	
communication	249

C

circuit breakers	
MBCOMPACTNSX	24
communication	
Advantys Configuration in EcoStruxure Process	
Expert	312
configuring Altivar71/61 Ethernet	
communication	249
configuring device Ethernet communication	248
configuring device Modbus communication	287
configuring Ethernet communication with Modbus	
devices	259
configuring TeSys T Ethernet communication	256
Modbus Serial - Explicit	298
Modbus TCP/IP - Explicit	273
Modbus TCP/IP - Explicit with Gateway	278
Modbus TCP/IP - Implicit with NOC	268
Modbus TCP/IP - Implicit with NOE	264
Compact	
Modbus communication configuration	290
configuring the devices	
ATV 31, ATV 312 and ATV 12	290

D

devices	
Advantys Configuration in EcoStruxure Process	
Expert	312
configuring Altivar71/61 Ethernet	
communication	249
configuring device Ethernet communication	248
configuring device Modbus communication	287
configuring Ethernet communication with Modbus	
devices	259
configuring TeSys T Ethernet communication	256
Modbus Serial - Explicit	298
Modbus TCP/IP - Explicit	273
Modbus TCP/IP - Explicit with Gateway	278
Modbus TCP/IP - Implicit with NOC	268
Modbus TCP/IP - Implicit with NOE	264
digital protective relays	
SEPAM Profile	64

E

EcoStruxure Process Expert	
list of function blocks	17
EcoStruxure Process Expert automation system	
introduction	16
Ethernet	

configuring Altivar71/61 Ethernet	
communication	249
configuring device Ethernet communication	248
configuring Ethernet communication with Modbus	
devices	259
configuring TeSys T Ethernet communication	256
Modbus TCP/IP - Explicit	273
Modbus TCP/IP - Explicit with Gateway	278
Modbus TCP/IP - Implicit with NOC	268
Modbus TCP/IP - Implicit with NOE	264
Ethernet technology	
CommunicationOK Variable	248

G

gateway	
configuring Ethernet communication with Modbus	
devices	259
general	
device resources overview	17
general overview of device library	16
supervision services	20
general concepts	
function block interface	20
monitoring	22
set-point management	22

H

Hardwired Circuit Breaker	56
---------------------------------	----

M

Masterpact	
Modbus communication configuration	290
MASTERPACT	39
Modbus	
configuring device Ethernet communication	287
configuring Ethernet communication with Modbus	
devices	259
device configuration details	290
Modbus Serial - Explicit	298
Modbus Serial - Explicit	
Modbus Serial - Explicit	298
Modbus TCP/IP - Explicit	
Modbus TCP/IP - Explicit	273
Modbus TCP/IP - Explicit with Gateway	278
Modbus TCP/IP - Implicit with NOC	
Modbus TCP/IP - Implicit with NOC	268
Modbus TCP/IP - Implicit with NOE	
Modbus TCP/IP - Implicit with NOE	264
Modbus technology	
general	287
motor controllers and starters	
Tesy U Profile	107
Motor controllers and starters	
TeSys T Profile	85

O

OperationLimit	61
----------------------	----

P

Power Meter	
Modbus communication configuration	290

power monitoring devices	
ACCUSINE PCS Profile.....	150
PM profile.....	124
Smart UPS Profile.....	140
Profibus	
Advantys Configuration in EcoStruxure Process	
Expert	312
progressive starters	
ATS Progressive Starter	164

S

safety module	
MBXPSMC	235
Sepam	
Modbus communication configuration.....	290

T

TeSys T	
configuring TeSys T Ethernet communication	256
Modbus communication configuration.....	290
TeSys U	
Modbus communication configuration.....	290

V

variable speed drives	
ATV Profile	180

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

+ 33 (0) 1 41 29 70 00

www.se.com

As standards, specifications, and design change from time to time,
please ask for confirmation of the information given in this publication.

© 2023 Schneider Electric. All rights reserved.

EIO0000001309.16