



# Process Expert - General Purpose Library Classic

## Equipment Module Control Services Reference Manual

EIO0000003014.06  
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 .....	5
Qualification of Personnel .....	5
Proper Use .....	6
Before You Begin .....	6
Start-up and Test .....	7
Operation and Adjustments .....	7
About the Book .....	9
Overview .....	13
Equipment Module Templates .....	14
Delivering Control Services .....	14
List of Function Blocks .....	15
Equipment Module .....	16
Equipment Module .....	17
Description of Equipment Module .....	17
Definition of Statuses, Equipment Module State and Commands .....	19
Strategy Execution Status Matrix .....	21
Strategy Execution Status Controller .....	22
Operating Modes .....	22
EMCTL - Equipment Module Controller .....	23
Description .....	23
DFB Representation .....	23
Inputs .....	23
Inputs/Outputs .....	24
Public Variables .....	27
Auxiliary Functions of Equipment Module .....	32
EMTRANS .....	32
Description .....	32
Inputs .....	33
Inputs/Outputs .....	34
EMFAILED .....	34
Description .....	34
Inputs/Outputs .....	35
EMSTABLESTEP .....	35
Description .....	35
Inputs/Outputs .....	36
EMSTEPDESC .....	36
Description .....	36
Inputs .....	37
Inputs/Outputs .....	38
Public Variables .....	38
EMLOGIC - User Defined Process Logic .....	39
Description .....	39
Inputs .....	39
Inputs/Outputs .....	39
EMCTL - Example of Use .....	41
Description .....	41
DFB Representation .....	42
Reusable Equipment Module Logic .....	42

Pump Set Pattern.....	51
Pump Set Equipment Module .....	52
Description of Pump Set Equipment Module .....	52
Strategy Selection and Functionality.....	53
Disabling the Strategies Based on Equipment Module State.....	54
Input and Output Parameters .....	54
Input Parameters Applicable For Each Strategy .....	55
Pump Set Equipment Module State Machine .....	55
Pump Set Equipment Module Diagnosis State .....	56
Pump Set Pumping Asset Diagnosis State .....	57
Modifying the Pump Set Pattern .....	57
PUMPSETPATTERN - Pump Set Pattern Logic.....	58
Description .....	58
DFB Representation .....	58
Inputs .....	59
Outputs .....	60
Inputs/Outputs .....	60
Auxiliary Functions of Pump Set Equipment Module .....	62
PUMPINGASSETSEQ.....	62
Description .....	62
DFB Representation.....	63
Inputs .....	63
Outputs .....	64
Inputs/Outputs .....	64
Flow Control Pattern.....	65
Flow Control Equipment Module .....	66
Description of Flow Control Equipment Module .....	66
Strategy Selection and Functionality.....	67
Disabling the Strategies Based on Equipment Module State.....	68
Input and Output Parameters .....	68
Input Parameters Applicable For Each Strategy .....	69
Flow Control Equipment Module State Machine .....	69
Setpoint Management .....	70
Flow Control Equipment Module Diagnosis State .....	71
Flow Control Pumping Asset Diagnosis State .....	72
Modifying the Flow Control Pattern .....	72
FLOWCTLPATTERN - Flow Control Process Logic .....	73
Description .....	73
DFB Representation .....	73
Inputs .....	74
Outputs .....	75
Inputs/Outputs .....	76
Auxiliary Functions of Flow Control Equipment Module .....	78
PUMPINGASSETSEQ1.....	78
Description .....	78
DFB Representation.....	79
Inputs .....	80
Outputs .....	80
Inputs/Outputs .....	80
Index .....	81

# 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 function blocks (DFBs) and variables that are encapsulated in the Control facets referenced by the Equipment module templates to provide Control services.

For a list of templates and the services that they provide, refer to the user guides mentioned in this document.

This document does not cover any development procedures and internal functionality details of EcoStruxure Process Expert.

This document is for users with knowledge of EcoStruxure Process Expert, and of the Supervision and Control Participants.

## Validity Note

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

## Related Documents

This document describes the functionality and features of the control services, supervision services, and process templates.

This document covers the functional aspects of templates, control, and supervision services when engineering a system, using EcoStruxure™ Process Expert, and describes the dynamic objects visible from the runtime. It does not cover any operational aspects, nor does it provide information on how to use the supervision services to monitor and operate control systems.

Title of Documentation	Reference Number
EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Templates Reference Manual	EIO0000003016 (eng)
EcoStruxure™ Process Expert - General Purpose Library Classic Equipment Module Supervision Services Reference Manual	EIO0000003015 (eng)
EcoStruxure Process Expert User Guide	EIO0000001114 (eng)

## 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.<sup>1</sup>
- 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.**

**NOTE:** 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.

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

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

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

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

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



---

# Overview

## What's in This Part

Equipment Module Templates .....	14
----------------------------------	----

## Overview

The Process Expert software provides the resources that have been pre-configured and tested by Schneider Electric and that were specifically designed for automating a large variety of processes.

The control resources for process control provide the common required functions, facilitating the development of control systems.

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

This document describes the basic concepts and details behind each one of the function blocks (DFBs) for implementing the common cross-process and cross-market Control functions.

# Equipment Module Templates

## What's in This Chapter

Delivering Control Services .....	14
List of Function Blocks .....	15

## Overview

This chapter describes how the services of the Equipment Module templates are made available to you through EcoStruxure Process Expert and the embedded Participant.

## Delivering Control Services

### Introduction

Inside Equipment Module control module templates, the resources providing Control services are organized in composite and facet templates. They are *elements* of the control module template and its instances.

Interfaces link these elements within the control module and allow making links to instances of other templates.

### Control Facet Templates

The facet template is the smallest component of a Equipment Module template.

It encapsulates the constituents of the Control Participant, which are the function blocks and variables described in this document.

During generation (see EcoStruxure™ Process Expert, User Guide), these constituents create the logic of the Control (see EcoStruxure™ Process Expert, User Guide) Participant project that you have created in EcoStruxure Process Expert.

### Control Composite Templates

The Control composite template is a container for Control elements and/or other composite references.

They help organizing Control services inside the control module template.

### Configuring Control Services

Typically, Equipment Module templates contain core and optional services.

To customize the services provided by an instance that uses the control module template, the **Properties** window of the instance allows you to:

- Activate/deactivate optional resources.
- Configure the parameters of core resources and selected optional resources.

In addition, you can modify the logic of the Control Participant project during refinement.

## List of Function Blocks

### List of Families

The function blocks described in this document are grouped by family.

Each DFB is assigned to the same family as the Equipment Module template that references it.

### Description

The table lists the function blocks of each family:

Family Name	Function Blocks	Description
Equipment Module	EMCTL, page 16	Equipment Module Controller
Pump Set	PUMPSETPATTERN, page 51	Pump Set Pattern
Flow Control	FLOWCTLPATTERN, page 65	Flow Control Pattern

# Equipment Module

## What's in This Part

Equipment Module .....	17
EMCTL - Equipment Module Controller .....	23
Auxiliary Functions of Equipment Module .....	32
EMLOGIC - User Defined Process Logic .....	39
EMCTL - Example of Use .....	41

## Overview

This part provides a detailed description of the functions, pins, pin layout, and variables of the function blocks of the equipment module.

These function blocks do not reflect any specific installation.

**⚠ 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.<sup>1</sup>
- 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.**

<sup>1</sup> 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.



# Equipment Module

## What's in This Chapter

Description of Equipment Module .....	17
Definition of Statuses, Equipment Module State and Commands .....	19
Strategy Execution Status Matrix .....	21
Strategy Execution Status Controller.....	22
Operating Modes.....	22

## Overview

This chapter describes the equipment module management.

## Description of Equipment Module

### Overview

The operation of the function block that is available for equipment module is described in this chapter.

The **EMCTL** DFB belongs to the General Purpose Library and is used to monitor and manage finite number of specific minor processing activities (strategies). The module is based on the ISA-S88 Standard, Part 1 (S88.01) and implements the status controller defined in this standard for controlling procedure model components (phases, operations, and so on).

### Function Description

User can use this DFB to change states in the process. In compliance with ISA-S88.01I, you can use this DFB in continuous processes to initiate start, stop or emergency sequences.

The DFB implements equipment state, strategy, strategy execution state management and commands received from:

- The Supervision system (when sequences are manually controlled by the operator)
- Other control sequences

The table compares the functions that are available in the DFB:

Function	EMCTL
Management of strategy execution state machine and commands according to ISA-S88	x
Management of messages for the operator integrated with the sequence	—
Management of operating modes according to ISA-S88: Manual (enables to change from one sequence step to any other sequence step), Semi (step advance with confirmation from the operator), and Auto	x
Owner Management: Operator/Program	x
Step time automatically calculated by the control block	x
Automatic management of transitions through the <b>EMTRANS</b> function block	x
Automatic management of sequence detected failure through the <b>EMFAILED</b> function block	x
Automatic management of sequence steady-step marking through the <b>EMSTABLESTEP</b> function block	x

Function	EMCTL
Possibility of displaying the sequence graphically (active step and next level of transitions and steps) from the monitoring subsystem	x
<b>NOTE:</b> 'x' indicates that the functions are available in the DFB.	

Program the steps and transitions that the sequences implement (through ST or SFC language) while the component (EMCTL) allows the programmer to:

- Manage the statuses of the strategy execution sequence and
- Process the commands received from other levels as described above.

As a minimum, implement the strategy subsequence that implements normal operation (RUNNING), page 41.

Otherwise user can implement subsequences that attend to commands received from other levels of the control system or that react to detected failures in the process (HOLDING, RESTARTING, STOPPING, and ABORTING) detected by the strategy sequence itself.

The table describes the main functions of the DFB:

Function	Description
Processing strategy execution States and Commands	The DFB processes the commands received and determines the status of the strategy execution sequence with the objective of determining the subsequences (idle, hold, continue, abort, stop.) and which step needs to run in each cycle of the program.
Managing Entry Points in the strategy execution sequence	The DFB enables to retain the step number with which the execution of the normal sequence (Running) needs to be restarted (Restart) after being restarted from held status (Held).
Handling of Detected Failures	The component manages controller detected failures (without user intervention from the Supervision system). The component manages this detected failure and triggers the execution of the detected failure handling subsequence.
Strategy management	Strategy is any minor processing activity which an equipment module can perform, a maximum of 16 strategies are supported in equipment module. The DFB allows to select the strategy which need to be executed for the Equipment module. For an equipment module, strategy can be from 1 to 16. The Strategy codes and their interpretation is EM specific however the supervisory resources will allow the users to define associated text to the respected codes.  The supervisory resource will not allow the user from modifying the Strategy while the status is not <i>IDLE</i> .
Equipment state management	The DFB allows to calculate the Equipment module state based on user defined logic. This DFB allows to configure up to 32 Equipment module states, where state 0 to 12 is as per recommendation defined in NS88 standard, 13 to 15 is reserved and 16 to 31 can be used to interpret user defined equipment state.
Disable strategy	The DFB allows to disable the strategy based on user defined logic so that disabled strategy will not be available in supervision for selection. Hence, the execution for disable strategy can be avoided.
Auto reset disable	Automatic reset ( <i>STOPPED</i> → <i>IDLE</i> , automatically performs reset operation without consuming any extra execution cycle) can be disabled by setting the input pin (AUTORESETDIS) to a logical high.

# Definition of Statuses, Equipment Module State and Commands

## Statuses

The table describes the possible strategy execution statuses along with the description of each status:

Status	Description
Idle	Inactive operation. Waiting for command to start ( <b>Start</b> ) running.
Running	Normal operation.
Complete	Normal operation has finished. Requires an initialization order ( <b>Reset</b> ) to return to the Idle status.
Pausing	A pause ( <b>Pause</b> ) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
Paused	Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue ( <b>Resume</b> ), the sequence continues with the next step of the normal operation sequence (Running).
Holding	A hold ( <b>Hold</b> ) command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
HoldingForError	A detected failure is automatically detected in the process (for example, a detected failure in Equipment Module), and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (Heldforerror) status.
Held	Status reached after the pause sequence is completed and the next steady status is reached.
HeldForError	Operation put on hold due to detected fail condition. Waiting for command to Restart ( <b>Restart</b> ) to get operation to running.
Restarting	A restart command ( <b>Restart</b> ) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
Stopping	An order to stop ( <b>Stop</b> ) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
Stopped	The stop sequence (Stopping) is completed. Requires an initialization command ( <b>Reset</b> ) to go to inactive (Idle) status.
Aborting	An order to end ( <b>Abort</b> ) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
Aborted	An aborting sequence (Aborting) has been completed. Requires an initialization command ( <b>Reset</b> ) to go to inactive (Idle) status.

The (Held) status can also be reached (without receiving a command) if the equipment module itself detects a problem situation. The DFB has a signal that allows this situation to be communicated.

The DFB has the necessary means to be able to restart the Running sequence from the last steady step in which the initial run was held by a **Hold** command.

## Equipment Module State

The table describes the equipment module states along with codes and descriptions recommended by NS88 (ISA88 for Non-Stop production).

Equipment Module State	Code	Description
Off	0	The equipment is powered up, but is not yet ready to be operated.
Stopped	1	The equipment is ready to be started, is not performing its basic processing function, and equipment is not processing material.
Starting	2	The equipment has been commanded to start and is in the process of starting, but the equipment has not yet been commanded to process material.
Ready	3	The operation has finished starting and is ready for operation.
Standby	4	The equipment is running and is waiting for material to process.
Producing	5	The equipment is performing its basic processing function.
Switching	6	The equipment is finishing processing of a product and starting to process a new one.
Clearing	7	The equipment is finishing processing of the remaining product.
Holding	8	The equipment has been commanded to hold and stop processing and is performing the actions to go to held state.
Held	9	The equipment is not processing, it is waiting for resuming.
Stopping	10	The equipment has been commanded to stop processing and is performing to clear the material.
Aborting	11	The equipment has been commanded to abort (some emergency) processing and is performing to clear the material.
Aborted	12	The equipment has aborted processing and the material is cleared.
-	13...15	Reserved.
<User Defined>	16...31	These sixteen codes are reserved for general purpose states, so that the user can use as needed. In case of using them, the supervision resources have to be adjusted accordingly to interpret such codes and showing the right states to the operator in runtime.

## Commands

The table describes the commands that can be communicated to the equipment module along with the description of each command:

Command	Description
<b>Start</b>	Allows the normal operation of the sequence (Running) to be started. It is only valid if the status is inactive (Idle).
<b>Stop</b>	Allows the sequence execution to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
<b>Hold</b>	Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
<b>Restart</b>	Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
<b>Abort</b>	Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).
<b>Reset</b>	Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
<b>Pause</b>	Allows normal execution (Running) to be paused in the next steady Sequence status.
<b>Resume</b>	Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.

# Strategy Execution Status Matrix

## Description

The following table describes the finite status controller that the DFB implements:

Initial State	Command	Activity	Final State (without command)
Idle	Start	Running	–
Running	Stop	Stopping	Stopped
	Hold	Holding	Held
	Pause	Pausing	Paused
	Abort	Aborting	Aborted
Complete	Reset	Idle	–
Pausing	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Paused	Stop	Stopping	Stopped
	Resume		Running
	Abort	Aborting	Aborted
Holding	Stop	Stopping	Stopped
	Abort	Aborting	Aborted
Held	Stop	Stopping	Stopped
	Restart	Restarting	Running
	Abort	Aborting	Aborted
Restarting	Stop	Stopping	Stopped
	Hold	Holding	Held
	Abort	Aborting	Aborted
Stopping	Abort	Aborting	Aborted
Stopped	Reset	Idle	–
	Abort	Aborting	Aborted
Aborting	No action possible		
Aborted	Reset	Idle	–

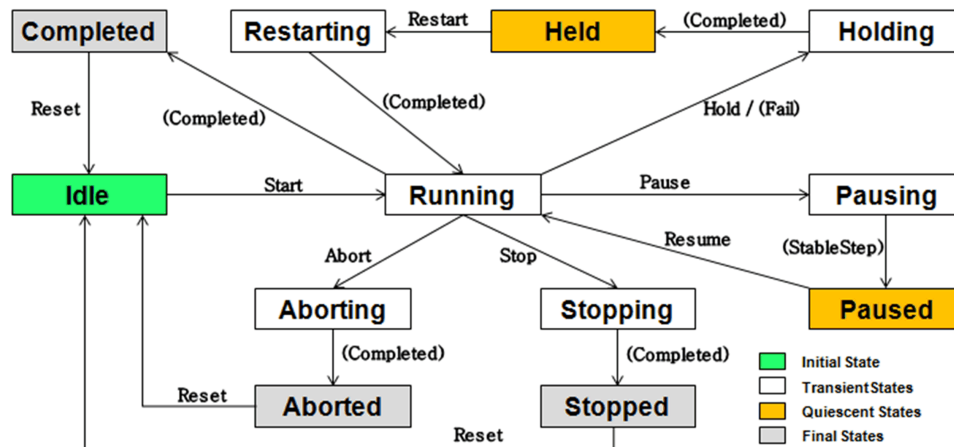
The **Held** status can also be reached (without receiving a command) if the equipment module itself encounters a detected error. The block has a signal that allows this situation to be communicated.

The block has the necessary means to be able to restart the **Running** sequence from the last steady step in which the initial run was held by a **Hold** command.

# Strategy Execution Status Controller

## Description

The following diagram describes the possible strategy execution statuses and commands that the Control Expert component manages according to the ISA-S88 standard:



Next to the transition arrows, the commands that the component receives (from the Supervision system or another higher level sequence) are shown in black, and status/commands that sequence obtains from other blocks (for example, user logic) are shown in parenthesis.

**NOTE:** This state transition diagram is derived from the first three initial states of the state transition matrix, page 21.

## Operating Modes

### Mode Table

The following table provides the description of the sequence operating modes of EMCTL DFB:

Operating mode	Implemented as:
Automatic	It is the normal running mode. The sequence of selected strategy runs automatically based on the transitions defined in the controller.
Semiautomatic	In this operating mode, after the conditions defined in the transitions are met, you need to confirm the move to the next step.
Manual	The sequence of selected strategy does not automatically advance to a step; user can manually change from one strategy execution step to another. Once the sequence step is selected, operating mode will change to semi-automatic and starts the strategy execution step selected in manual mode.

# EMCTL - Equipment Module Controller

## What's in This Chapter

Description .....	23
DFB Representation .....	23
Inputs .....	23
Inputs/Outputs .....	24
Public Variables .....	27

## Overview

This chapter provides a detailed description of the functions, pins, pin layout, and variables of the EMCTL DFB.

## Description

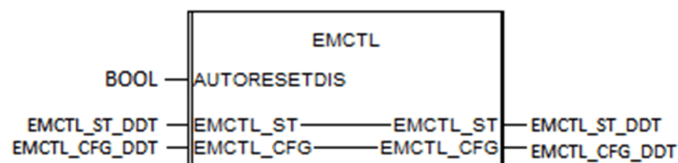
### Initialization

If the strategy execution status is unknown, it is initialized as Idle regardless of whether a controller has restarted or not. This initialization is done automatically.

## DFB Representation

### Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



## Inputs

### Input Parameter Description

Parameter	Type	Description
AUTORESETDIS	BOOL	Disabling the automatic reset of the strategy execution state.

# Inputs/Outputs

## Input/Output Parameter Description

Parameter	Type	Description
EMCTL_ST	EMCTL_ST_DDT	Provides the necessary data for monitoring and controlling equipment module.
EMCTL_CFG	EMCTL_CFG_DDT	Provides the necessary data for background monitoring.

## EMCTL\_ST\_DDT Type

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
ETIME	-	-	UDINT	The DFB calculates the execution time elapsed in tenths of a second for the current step of the control sequence. If execution of the program is stopped and resumed, the accumulated execution time is considered but not the time that the program was stopped.
STRATEGY	-	-	UINT	Selected strategy in equipment module.
DISSTRATEGY	-	-	WORD	Disabling the execution of strategy.
	DISSTRATEGY01	0		Disabling the execution of strategy 1.
	DISSTRATEGY02	1		Disabling the execution of strategy 2.
	DISSTRATEGY03	2		Disabling the execution of strategy 3.
	DISSTRATEGY04	3		Disabling the execution of strategy 4.
	DISSTRATEGY05	4		Disabling the execution of strategy 5.
	DISSTRATEGY06	5		Disabling the execution of strategy 6.
	DISSTRATEGY07	6		Disabling the execution of strategy 7.
	DISSTRATEGY08	7		Disabling the execution of strategy 8.
	DISSTRATEGY09	8		Disabling the execution of strategy 9.
	DISSTRATEGY10	9		Disabling the execution of strategy 10.
	DISSTRATEGY11	10		Disabling the execution of strategy 11.
	DISSTRATEGY12	11		Disabling the execution of strategy 12.
	DISSTRATEGY13	12		Disabling the execution of strategy 13.
	DISSTRATEGY14	13		Disabling the execution of strategy 14.
	DISSTRATEGY15	14		Disabling the execution of strategy 15.
	DISSTRATEGY16	15		Disabling the execution of strategy 16.
EMSTATE	-	-	UINT	Equipment module state.  For additional information refer Equipment state management, page 19.
STATE	-	-	UINT	Provides the status of the strategy execution. Read-only access to the data is contained in this bits word. Refer to the Definition of Statuses, Equipment Module State and Commands, page 19.
	IDLE	0		Inactive operation. Waiting for command to start (Start) running.
	RUNNING	1		Normal operation.
	HELD	2		Status reached after the pause sequence is completed and the next steady state is reached.



Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
	HELDFORERROR	3		Status reached after the pause sequence is completed and the next steady state is reached to a detected failure.
	STOPPED	4		The stop sequence (Stopping) is completed. Requires an initialization command (Reset) to go to inactive (Idle) status.
	PAUSED	5		Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue (Resume), the sequence continues with the next step of the normal operation sequence (Running).
	ABORTED	6		An aborting sequence (Aborting) has been completed. Requires an initialization command (Reset) to go to inactive (Idle) state.
	COMPLETE	7		Normal operation has finished. Requires an initialization order (Reset) to return to the Idle status.
	HOLDING	8		A hold (Hold) command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
	HOLDINGFORER- ROR	9		A detected failure is automatically detected in the process (for example, a detected failure in Equipment Module), and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (HeldForError) status.
	RESTARTING	10		Restart command (Restart) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
	PAUSING	11		A pause (Pause) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
	STOPPING	12		An order to stop (Stop) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
	ABORTING	13		An order to end (Abort) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
COMMAND	-		UINT	Enables commands to be sent to the strategy execution sequence usually from the Supervision system. Write access to the data is contained in this bits word. The DFB sets the command to 0 after processed (successfully or not). Refer to the Definition of States, Equipment Module State and Commands, page 19.
	START	0		Allows the normal operation of the sequence (Running) to be started.
	HOLD	1		Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	RESTART	2		Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
	STOP	3		Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	RESET	4		Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
	PAUSE	5		Allows normal execution (Running) to be paused in the next steady Sequence status.
	RESUME	6		Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.
	ABORT	7		Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description												
STW	-		WORD	Sequence status word. In this word, the <i>EMTRANS</i> auxiliary DFB is used to load the status of the transitions (maximum of six) from the current step of the sequence to the steps in the next level of the sequence, where 0 means that the conditions in the transition are not fulfilled and 1 means they are fulfilled.												
	STATUS1Transition	0		Status of the first transition												
	STATUS2Transition	1		Status of the second transition												
	STATUS3Transition	2		Status of the third transition												
	STATUS4Transition	3		Status of the fourth transition												
	STATUS5Transition	4		Status of the fifth transition												
	STATUS6Transition	5		Status of the sixth transition												
	AUTORESETDIS	15		Auto reset disable												
CFGW	-		WORD	Sequence configuration word.												
	OWNER	0		Read/write access. Enables to configure whether the sequence commands will come from the program (0) or the operator (1).												
	SEMI	1		Read/write access. Enables to configure the sequence in automatic (0) or semi-automatic (1) mode.												
	MANUAL	2		Read/write access. Enables to configure the strategy execution in automatic/semi-automatic (0) or manual (1) mode.												
				<table><tr><th>SEMI</th><th>MANUAL</th><th>Mode</th></tr><tr><td>OFF</td><td>OFF</td><td>Automatic</td></tr><tr><td>OFF</td><td>ON</td><td>Semi-automatic</td></tr><tr><td>ON</td><td>—</td><td>Manual</td></tr></table>	SEMI	MANUAL	Mode	OFF	OFF	Automatic	OFF	ON	Semi-automatic	ON	—	Manual
	SEMI	MANUAL		Mode												
	OFF	OFF		Automatic												
	OFF	ON		Semi-automatic												
	ON	—		Manual												
NEXTSTEP	3	Read/write access. Enables to confirm the change of steps to the next sequence step in Semi-Automatic mode. The DFB sets the signal to 0 after it is processed.														
SCROLL_UP	5	Read/write access. Enables to show the steps previous to the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.														
SCROLL_DOWN	6	Read/write access. Enables to show the steps following the one being shown on the current window in manual mode. The DFB sets the signal to 0 after it is processed.														
CSTEPD	-		String [122]	Current step description (three characters for step number concatenated with seventeen characters for description).												

## EMCTL\_CFG\_DDT Type

Name	Type	Description
NSTEP	INT	Number of the new step at which the subsequence needs to be positioned in manual mode. The DFB sets this variable to 0 after it is processed.
STEPD	STRING[122]	Description of steps. Limited to six steps. In automatic and semi-automatic mode, it shows the destination steps specified in the current step transitions. In manual mode, it shows the list of the steps defined in the strategy execution sequence. This variable format is: six steps, twenty characters each. Of the twenty characters of each step, the first three are for the step number and the next seventeen are for its description. The value of this variable is assigned for the auxiliary functions.
TRANSD	STRING[122]	Description of transitions. Limited to six transitions. This variable format is six transitions, twenty characters each. The twenty characters are for its description. The value of this variable is assigned for the auxiliary functions.

## Public Variables

### Public Variable Description

Variable	Type	Description
SC	EMCTL_SC_DDT	Provides the frequently needed data to monitor the equipment module status and control it from the sequential control.

## EMCTL\_SC\_DDT Type

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
ETIME	-		UDINT	Elapsed time in the current step (dsec).
STRATEGY	-		UINT	Current strategy of equipment module.
EMSTATE	-		UINT	Equipment module state. For additional information refer Equipment state management, page 17.
STATE	-		UINT	Read-only access. Refer to the <i>EMCTL_ST.STATE</i> input/output pin, page 24.
	IDLE	0		Inactive operation. Waiting for command to start (Start) running.
	RUNNING	1		Normal operation.
	HELD	2		Status reached after the pause sequence is completed and the next steady state is reached.
	HELDFOREERROR	3		Status reached after the pause sequence is completed and the next steady state is reached to a detected failure.
	STOPPED	4		The stop sequence (Stopping) is completed. Requires an initialization command (Reset) to go to inactive (Idle) status.
	PAUSED	5		Status reached after the pause sequence is completed and the next steady status is reached. Upon receiving the command to continue (Resume), the sequence continues with the next step of the normal operation sequence (Running).
	ABORTED	6		An aborting sequence (Aborting) has been completed. Requires an initialization command (Reset) to go to inactive (Idle) state.

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
	COMPLETE	7		Normal operation has finished. Requires an initialization order (Reset) to return to the Idle status.
	HOLDING	8		A hold (Hold) command is received and the sequence that allows the process to be changed to a known status is executed. After this sequence is finished, it automatically goes to the held (Held) status.
	HOLDINGFORError	9		A detected failure is automatically detected in the process (for example, a detected failure in Equipment Module), and the sequence that allows the process to be to a known status is executed. After this sequence is finished, it automatically goes to the held (HeldForError) status.
	RESTARTING	10		Restart command (Restart) is received while the sequence is being held (Held). The sequence runs to return to the normal operation sequence (Running). The normal operation status (Running) is set automatically after the sequence is complete.
	PAUSING	11		A pause (Pause) command is received for a short period while the normal operation was being executed (Running). The normal sequence runs until it automatically goes to paused (Paused) status.
	STOPPING	12		An order to stop (Stop) has been received. The sequence to finish the sequence in a controlled way is executed (it is not possible to continue with it). The stopped operation status (Stopped) is automatically activated after the sequence is complete.
	ABORTING	13		An order to end (Abort) has been received. A fast stop, which is not necessarily controlled is executed. The aborted operation status (Aborted) is automatically activated after the sequence is completed.
COMMAND	-		UINT	Sequence commands.
	START	0		Allows the normal operation of the sequence (Running) to be started.
	HOLD	1		Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	RESTART	2		Allows the sequence for continuing the operation (Restarting) and resume normal operation (Running).
	STOP	3		Allows the sequence to be stopped (Running, Pausing, Paused, Holding, Held, or Restarting) and the stopping sequence to be run (Stopping).
	RESET	4		Triggers the transition to idle status (in Complete, Aborted, or Stopped status).
	PAUSE	5		Allows normal execution (Running) to be paused in the next steady Sequence status.
	RESUME	6		Allows normal operation of the sequence (Running) to be resumed from the paused (Paused) status.
	ABORT	7		Allows the sequence to be aborted (in any status except Idle, Complete, Aborting, and Aborted) and the aborting sequence to be run (Aborting).
	COMPLETED	8		Write access.  1 = Sends a command to finish the current subsequence (Running, Holding, Restarting, Stopping, and so on).  0 = The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to  1 by using the EMTRANS auxiliary DFB is recommended.
	STABLESTEP	9		Write access.  1 = Sends a command to mark the current step as a steady step, that is: <ul style="list-style-type: none"> <li>The step with which the sequence will resume after its is held or</li> </ul>

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
				<ul style="list-style-type: none"> <li>The step on which the sequence will be held after the sequence is paused.</li> </ul> <p>0 = The signal is automatically set to 0 by the DFB after the signal is processed.</p> <p>As an alternative, forcing the value of this variable to 1 by using the EMSTABLESTEP auxiliary DFB is recommended.</p>
CSTEP	-		INT	Read-only access. Number of the active sequence current step (Running, Pausing, Holding, Restarting, Stopping, or Aborting). The value 0 corresponds to the initial step of each subsequence.
NEXTSTEP	-		INT	Write access. Enables the number of the next step to be run in the sequence to be set.
				The signal is automatically set to 0 by the DFB after the signal is processed. As an alternative, forcing the value of this variable to 1 by using the EMTRANS auxiliary DFB is recommended.
MODE	-		INT	Read-only access, page 24.
				(bits 1 and 2). Mode is encoded as:
			VALUE	MODE
			0	Automatic
			1	Semi-Automatic
			2	Manual
CURRENTITEM	-		INT	Read-only access.
				1 = Indicates the current position of the list of steps in Manual mode. The value -1 indicates that there has been a mode change and -2 indicates that the mode change has been identified.
DEST			ARRAY [1...6] of INT	Destinations steps after transitions.
DESTW			WORD	Transitions Conditions word.
PROPERTIES	-		WORD	Collective word of necessary boolean variables.
	OWNER	0		1 = Read-only access, page 24.
	ONENTRY	1		Read-only access.
				1 = Reports that a new sequence step has just been started.
				User can use this event in the logic of the sequence step actions to run any actions. The actions run only when the step starts.
	ONEXIT	2		Read-only access.
				1 = Reports that the execution of the sequence current step is about to finish.
				User can use this event in the logic of the sequence step actions to run any actions. The actions run only when the step ends.
	FAIL	4		Write access.
				1 = Sends a command to stop the sequence normal execution.
				0 = The signal is automatically set to 0 by the DFB after the signal is processed.
				With the activation of this signal, the Holding For Detected Error subsequence is run next if the Running, Pausing, Restarting, or Paused subsequence was running previously. As an alternative, forcing the value of this variable to 1 by using the EMFAILED auxiliary DFB is recommended.
	NEWCYCLE	5		Read-only access.
				1 = Indicates that a new sequence state machine execution cycle has started.

Name	BitRank Name (if any)	Bit Number (if any)	Type	Description
	UPDATESTEP	6		Read-only access. 1= Internal mark indicates the need for updating the steps descriptions.
	RUNNINGX	7		Read-only access. 1 = Indicates that the current state of the sequence is Holding or Holding for Detected Error.
	HOLDINGX	8		Read-only access. 1 = Indicates that the current state of the sequence is Holding or Holding for Detected Error.

The following elements (status) are described in the Status Controller section, page 22. The Definition of Statuses, Equipment Module State and Commands, page 19 shows the sequence finite state controller.

Status	Type	Description
Idle	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Idle.
Running	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Running.
Complete	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Complete.
Pausing	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Pausing.
Paused	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Paused.
Holding	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Holding.
HoldingFE	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Holding or Holding for Detected Error.
Held	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Held.
HeldFE	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Held for detected error.
Restarting	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Restarting.
Stopping	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Stopping.
Stopped	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Stopped.

Status	Type	Description
Aborting	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Aborting.
Aborted	BOOL	Read-only access. 1 = Indicates that the current state of the sequence is Aborted.

# Auxiliary Functions of Equipment Module

## What's in This Chapter

EMTRANS .....	32
EMFAILED .....	34
EMSTABLESTEP .....	35
EMSTEPDESC .....	36

## Overview

This section describes the function blocks that supplement the `EMCTL` DFB and that have the objective of facilitating the programming of the equipment module.

## EMTRANS

### Overview

This section describes the `EMTRANS` DFB.

## Description

### General

The main objective of `EMTRANS` DFB is to set up the control strategy sequence transitions based on the condition to be evaluated and the next step to be executed when this expression is fulfilled.

Additionally, the DFB manages the information necessary to graphically represent the state of the transitions and steps following the current step (`DEST` and `DESTW` fields of the `EMCTL_ST_DDT` equipment module state structure).

Include call to this function in the logic of the steps (normally in ST language) to program the transitions between the sequence steps.

### Function Call

The function call is carried out from the steps of the control subsequences (Running, Restarting, Holding, and so on) normally in ST language to program the transitions between these steps.

If several `EMTRANS` calls are included in the same step and if more than one of them is fulfilled in the same step execution cycle, the step corresponding to the transition that is fulfilled and was programmed first will run. The number of `EMTRANS` calls from one single step is limited to 6.

```
TRANS (CONDITION := (*BOOL*),
DESC := (*string[22]*),
NEXTSTEP := (*INT*),
EMCTL_SC := (*EMCTL_SC_DDT*),
EMCTL_ST := (*EMCTL_ST_DDT*),
EMCTL_CFG := (*EMCTL_CFG_DDT*))
```

A function call example is included below:



**4: (Wait for end of reactive dosing)**

```

if SSC.ONENTRY then

BISEN2_REACT.REM:=false;

BISEN2_REACT.LSP:=true;

end_if;

BISEN2_REACT.LSPSEL:=(OP01 < (IP01 - 10.0)); (Speed 1 or 2)

TRANS ((IP01 - COLA_REACT) <= OP01, 'Dosing finished?', SSC.
CSTEP+1, SSC, SST, SCFG);

```

In the previous example, the TRANS call included in step 4 of the sequence corresponds to a EMTRANS block instance to which the following parameters are passed:

- (IP01 - COLA\_REACT) <= OP01OS: Corresponds to the expression of the condition that needs to be fulfilled for the transition to be executed. You can include complex expressions that result in a Boolean value (true or false).
- Dosing finished?: Corresponds to the description of the transition.
- SSC.CSTEP+1: Corresponds to the next step that needs to be executed when the transition is fulfilled. In this case, the next step (step 5) is executed after the current step (step 4). You can include any expression jumping to an existing step in the sequence of the selected strategy.
- SSC: Corresponds to the SC public variable of the EMCTL block instance that is controlling the strategy sequence execution.
- SST: Corresponds to the DDT\_ST variable of the EMCTL block instance that is controlling the strategy sequence execution.
- SCFG: Corresponds to the DDT\_CFG variable of the EMCTL block instance that is controlling the strategy sequence execution.

## Inputs

### Input Parameter Description

Parameter	Type	Description
CONDITION	BOOL	1 = Indicates that the condition for executing the transition is fulfilled.
DESC	STRING[22]	Provides the description of the transition to be represented in the monitoring program. The character string is limited to 20 characters.
NEXTSTEP	INT	Subsequence step number to which a jump needs to be made when the transition is fulfilled. Refer to the CONDITION input. The permissible values for the steps go from 1 to 999 with the exception of NEXTSTEP to -1. This indicates that when the transition is fulfilled, the corresponding subsequence (to go, for example, from Running to Complete, from Holding to Held, and so on) will be completed.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
EMCTL_SC	EMCTL_SC_DDT	Corresponds to the SC public variable of the EMCTL block instance that is controlling the sequence execution.
EMCTL_ST	EMCTL_ST_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the equipment module. Refer to the EMCTL block, page 23 for more details.
EMCTL_CFG	EMCTL_CFG_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the equipment module. Refer to the EMCTL block, page 23 for more details.

## EMFAILED

### Overview

This section describes the EMFAILED DFB.

### Description

#### General

The main objective of the EMFAILED DFB is to report detected errors to the equipment module management block (EMCTL) during strategy sequence execution so that the *HoldingFE* subsequence is triggered.

### Function Call

The function call is made from the Running, Restarting, Pausing, or Paused control subsequence steps normally in ST language with the purpose of reporting detected errors in the process that is being controlled. The detected error requires the hold sequence to be executed (*HoldingFE*).

A function call example is included below:

(Detected failure condition monitoring after initializations)

```
if EMCTL.CSTEP > 0 and CONDFALLO then
  FAILED (EmCtl) ; (Trigger execution of the Hold sequence)
end_if;
```

In the previous example, the *FAILED* call that is included in the logic monitors the detected error conditions in each execution cycle (in this particular case, but could be associated with a specific step) corresponds to a EMFAILED DFB block instance to which the following parameters are passed:

- *EmCtl*: Corresponds to the name of the SC public variable of the EMCTL block instance that is controlling the execution of the sequence.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
EMCTL_SC	EMCTL_SC_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL block, page 23 for more details.

## EMSTABLESTEP

### Overview

The section describes the EMSTABLESTEP DFB.

### Description

#### General

The main objective of the EMSTABLESTEP DFB is to report to the equipment module management block (EMCTL) that the current running sequence step corresponds to a step, that is, the re-entry point in the normal execution (Running) sequence executes after restarting execution (Restarting).

### Function Call

The function call is performed from the Running control subsequence steps normally in ST language for the purpose of reporting. The current step is the entry point to which the program should return when the Running subsequence is resumed after finishing the Restarting subsequence. Basically, it enables to resume the sequence execution that verifies the appropriate actions are performed or not.

For example, suppose a valve has to be opened. After the valve is opened, the next step involves turning on a pump. While waiting for the valve to open, a detected failure that makes the sequence stop occurs (Holding subsequence execution). You can resume execution by running the restart (Restarting) subsequence. After this, the program returns to the normal running subsequence (Running). After the program goes back to this Running subsequence, the intention is to make another attempt at opening the valve (because most likely, it has been closed during the Holding subsequence) not to start the pump in proceeding directly. Therefore, in the valve opening step, the step is marked and does not call the function EMSTABLESTEP in the pump starting step.

The following is an example of a function call:

0: (End Water dosing)

```
if EmCtl.ONENTRY then  
RAMP_AGUA.LSTART:=false;  
FIC_AGUA.REM:=false;  
FIC_AGUA.MAN:=true;  
FIC_AGUA.LOP:=0.0;  
VALV_AGUA.LSP:=false;
```

```

STABLESTEP (EmCtl1);

end_if;

TRANS (VALV_AGUA.LOWPOS, EMCTL.CSTEP+1, EmCtl1);

```

In the previous example, the STABLESTEP call included in the logic that monitors detected failure conditions in each execution cycle (in this particular case, but it could be associated with a specific step) corresponds to a `EMSTABLESTEP` DFB block instance to which the following parameters are passed:

- `Emctl1`: Corresponds to the name of the SC public variable of the `EMCTL` block instance that is controlling the sequence execution.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
EMCTL_SC	EMCTL_SC_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL block, page 23 for more details.

## EMSTEPDESC

### Overview

This section describes the `EMSTEPDESC` DFB.

### Description

#### General

The main objective of the `EMSTEPDESC` DFB is to assign descriptions to the strategy execution steps that are defined in the strategy execution sequence to allow you to monitor them from the Supervision system.

### Function Calls

The function call is made from the control subsequence steps (Running, Restarting, Holding, and so on) at the beginning of the code to provide the description of the subsequence steps for their representation in the monitoring program. It is normally written in ST language.

Include as many `EMSTEPDESC` calls as there are steps defined in the subsequence or as there are steps that need to be represented in the Supervision system.

The syntax of the function is the following:

```

STEPDESC (STEPNUMBER := (INT),
DESC := (string[18]),
EMCTL_SC := (EMCTL_SC_DDT),
EMCTL_ST := (EMCTL_ST_DDT),
EMCTL_CFG := (EMCTL_CFG_DDT),

```

The following is an example of a function call:

(Running or Pausing Subsequence (302 / 314))

(Step descriptions)

```
STEPDESC (1, 'Init', SSC, SST, SCFG);
STEPDESC (2, 'Valves ON', SSC, SST, SCFG);
STEPDESC (3, 'Wait Valves ON', SSC, SST, SCFG);
STEPDESC (4, 'Dosing Reactive', SSC, SST, SCFG);
STEPDESC (5, 'Motor OFF', SSC, SST, SCFG);
STEPDESC (6, 'Waiting for queue', SSC, SST, SCFG);
```

(Steps and Transitions)

```
case SSC.CSTEP of
```

1: (#RunningStep0# Initial Step)

(Check initial conditions)

```
if IC_FAIL then
FalloCI := true;
FAILED (SSC); (Initial conditions not satisfied)
else
FalloCI := false;
```

(Initialization)

In the above example, the STEPDESC calls included at the beginning of the subsequence correspond to a `EMSTEPDESC` DFB block instance to which the following parameters are passed in case of the first call:

- 1: Corresponds to step number 1.
- Init: Corresponds to the definition of the step.
- SSC: Corresponds to the `SC` public variable of the `EMCTL` block instance that is controlling the sequence execution.
- SST: Corresponds to the `DDT_ST` variable of the `EMCTL` block instance that is controlling the sequence execution.
- SCFG: Corresponds to the `DDT_CFG` variable of the `EMCTL` block instance that is controlling the sequence execution.

## Inputs

### Input Parameter Description

Parameter	Type	Description
STEPNUMBER	INT	Number of the subsequence step. The values allowed for the steps are from 1 to 999.
DESC	STRING[22]	Description of the number of the subsequence step. The character string is limited to 17 characters. This is the value that will be shown in the sequence monitoring program.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
EMCTL_SC	EMCTL_SC_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL1, page 23 block for more details.
EMCTL_ST	EMCTL_ST_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL1, page 23 block for more details.
EMCTL_CFG	EMCTL_CFG_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL1, page 23 block for more details.

## Public Variables

### Public Variable Description

Parameter	Type	Description
ORDERNUMBER	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.
MAXORDERNUMBER	INT	Read-only public variable reserved for building the list of steps in manual mode that will be shown in the monitoring program.

# EMLOGIC - User Defined Process Logic

## What's in This Chapter

Description .....	39
Inputs .....	39
Inputs/Outputs .....	39

## Overview

This chapter describes the *EMLOGIC* DFB.

## Description

### General

It is an unprotected function block in which user can add the logic to manage various equipment states, strategies and strategy execution states depending upon the process controlled by the user. Sample code is available in the function block for the user reference.

## Inputs

### Input Parameter Description

Parameter	Type	Description
INITCONDFAIL	BOOL	The initial conditions should be satisfied for the equipment module operation. The result of the conditional summary block (CONDSUM) can be connected to this pin if user has multiple conditions.
FAILCOND	BOOL	The detected failure conditions during the operation of equipment module can be connected to this pin. The result of the conditional summary block (CONDSUM) can be connected to this pin if user has multiple conditions.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description	
IOPAR	EM_IOPAR_DDT	Manages the parameters required for Equipment Module logic.  <b>NOTE:</b> The parameter and its type can be modified according to specific requirements.  Below table displays the information about the data types to be followed for Input/Output parameters.	
		Data Type (user)	Control Expert Data Type
		Boolean	BOOL
		Duration	TIME

Parameter	Type	Description	
		Date/Time	DINT
		Enumeration	INT
		Numeric (Real)	REAL
		Numeric (Signed Integer)	INT
		Numeric (Double Signed Integer)	DINT
EMCTL_SC	EMCTL_SC_DDT	Corresponds to the SC public variable of the EMCTL block instance that is controlling the sequence execution.	
EMCTL_ST	EMCTL_ST_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL block, page 23 for more details.	
EMCTL_CFG	EMCTL_CFG_DDT	Provides the data necessary to communicate with the EMCTL block that is controlling the sequence. Refer to the EMCTL block, page 23 for more details.	



# EMCTL - Example of Use

## What's in This Chapter

Description .....	41
DFB Representation .....	42
Reusable Equipment Module Logic.....	42

## Overview

This chapter describes the example use of the `EMCTLDFB`.

## Description

### General

This chapter gives an example of an advanced equipment module controlled by the `EMCTL`, `EMTRANS`, `EMSTABLESTEP`, `EMFAILED`, and `EMSTEPDESC` DFBs.

The objective of the sequence is to control the loading of water in a process unit. For this purpose, a reusable component (DFB) is created so that you can use this sequence in different process units by only changing the connections of the component pins.

## Reusable Sequence Definition

The reusable DFB of the user, `DISTILL_EM`, declares the `EMCTL` `EMCTL_` Instance variable to manage the state of the sequence.

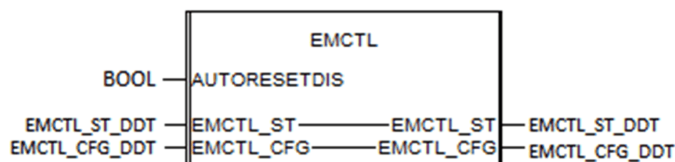
Name	no.	Type	Value	Comment	R/W Rights of Referenced Variable
<b>DISTILL_EM</b>		<DFB>		User sequence	
<inputs>					
<outputs>					
<inputs/outputs>					
FeedingRamp	9	ARAMP_SC_DDT			
FeedingPID	10	PIDCTL_SC_DDT			
FeedingPump	11	DEVCTL_SC_DDT			
BottomLevelPID	12	PIDCTL_SC_DDT			
BottomPump	13	DEVCTL_SC_DDT			
HeadTempPID	14	PIDCTL_SC_DDT			
VolatileTempPID	15	PIDCTL_SC_DDT			
DistillatePump	16	DEVCTL_SC_DDT			
ChilledWaterValve	17	DEVCTL_SC_DDT			
CoolingWaterValve	18	DEVCTL_SC_DDT			
SSC	22	EMCTL_SC_DDT			
SST	23	EMCTL_ST_DDT			
SCFG	24	EMCTL_CFG_DDT			
<public>					
<private>					
TRANS		EMTRANS			
STABLESTEP		EMSTABLESTEP			
FAILED		EMFAILED			
Running		BOOL			
Holding		BOOL			
Restarting		BOOL			
Aborting		BOOL			
Stopping		BOOL			
STEPDESC		EMSTEPDESC			
<sections>					
Init		<ST>			
RUNNING		<ST>			
HOLDING		<ST>			
RESTARTING		<ST>			
ABORTING		<ST>			
STOPPING		<ST>			
Common		<ST>			

**NOTE:** Declaration of the auxiliary components manages the transitions (`EMTRANS`), steady-step flags (`EMSTABLESTEP`), and notifications to trigger the holding subsequence (`EMFAILED`).

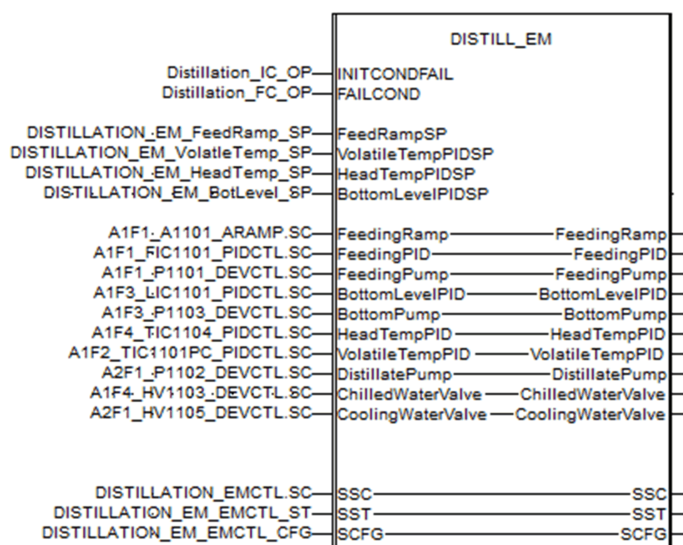
## DFB Representation

### Representation

The following diagrams include the calls to the sequence created by the user as well as to the control block that will control it (EMCTL).



**NOTE:** The call to the block controls the execution of the equipment module controller.



**NOTE:** The call to the functional block implements the control sequence of the user and will reuse for different process units if needed.

## Reusable Equipment Module Logic

### Equipment Module Logic With Description

The basic structure of the sequence as well as the steps that are defined for the sequence objective are included: Distillation. The parts of the code that make up the basic structure of the sequence are marked in bold.

(Initialization)

The following common actions have to be executed before the sequence execution:

```
Running := SSC.RUNNINGX;

Holding := SSC.HOLDINGX;

Restarting := SSC.RESTARTINGX;

Aborting := SSC.ABORTINGX;

Stopping := SSC.STOPPINGX;
```

**NOTE:** The default actions to be taken by DFB are defined in this section. However, if any section has to be added by user and actions need to be taken before execution of that section, it has to be added here.

(Running Subsequence)

In the running sequence, different strategies are implemented such as Start distillation, Stop distillation, Quick Stop.

In the running sequence, Distillation process started step by step in sequence.

Case SSC.Strategy of

1: (Start Distillation Strategy)

```
STEPDESC (1, 'Initialization', SSC, SST, SCFG);
STEPDESC (10, 'Refrigeration ON', SSC, SST, SCFG);
STEPDESC (20, 'Feeding ON', SSC, SST, SCFG);
STEPDESC (30, 'Distilling', SSC, SST, SCFG);
```

(\* Blocking from restarting of the strategy \*)

```
if SST.EMSTATE = 5 then
SSC.CSTEP := 30;
End_if;
```

**NOTE:** Definition of the descriptions for the subsequence steps.

(Steps and Transitions)

case SSC.CSTEP of

1: (Initial Step)

**NOTE:** First step with a value of 1. The allowed values for steps are 1 to 999.

(Check initial conditions)

```
If INITCONDFAIL then
FAILED (SSC); (Initial conditions not satisfied)
end_if;
```

**NOTE:** Report detected failure to trigger the holding subsequence.

```
if (SST.EMSTATE = 16) then
TRANS (true, 'Jump to Step 20', 20, SSC, SST, SCFG); (* Transition to
Step 20 *)
elsif (SST.EMSTATE = 17) then
TRANS (true, 'Jump to Step 30', 30, SSC, SST, SCFG); (* Transition to
Step 30 *)
else
TRANS (true, 'Jump to Step 10', 10, SSC, SST, SCFG); (* Transition to
Step 10 *)
end_if;
```

**NOTE:** Unconditional transition to the next step.

10: (Refrigeration ON)

```
if SSC.ONENTRY then
```

**NOTE:** Execute the actions only for the first time.

```
ChilledWaterValve.REM := false;
ChilledWaterValve.LSP := true;
CoolingWaterValve.REM := false;
```

```
CoolingWaterValve.LSP := true;
```

```
end_if;
```

```
TRANS (ChilledWaterValve.HIGHPOS and CoolingWaterValve.  
HIGHPOS and SSC.ETIME >= 30, 'Valves open', 20, SSC, SST, SCFG;
```

**NOTE:** The evaluation priority for these transitions is the order in which they are shown (up down; these transitions are not functional, and have only been included as an example in this case). You can define a maximum of 6 transitions.

## 20: (Feed ON)

```
if SSC.ONENTRY then
```

```
FeedingRamp.REM := false;
```

```
FeedingRamp.LTARGETSP := FeedRampSP;
```

```
FeedingRamp.LSTART := true;
```

```
FeedingPID.AUTO := true;
```

```
FeedingPump.REM := false;
```

```
FeedingPump.LSP := true;
```

```
BottomLevelPID.REM := false;
```

```
BottomLevelPID.LSP := BottomLevelPIDSP;
```

```
BottomLevelPID.AUTO := true;
```

```
BottomPump.LSP := true;
```

```
DistillatePump.REM := true;
```

```
end_if;
```

```
TRANS (BottomLevelPID.PV >= 50.0, 'Level >= 50%', 30, SSC, SST,  
SCFG);)
```

## 30: (Distilling Started)

```
if SSC.ONENTRY then
```

```
HeadTempPID.REM := false;
```

```
HeadTempPID.LSP := HeadTempPIDSP;
```

```
HeadTempPID.AUTO := true;
```

```
VolatileTempPID.REM := false;
```

```
VolatileTempPID.LSP := VolatileTempPIDSP;
```

```
VolatileTempPID.AUTO := true;
```

```
end_if;
```

```
TRANS (True, 'Return to IDLE', -1, SSC, SST, SCFG);
```

```
end_case;
```

## 2: ( Stop distillation Strategy)

```
STEPDESC (1, 'Initialization', SSC, SST, SCFG);
```

```
STEPDESC (10, 'Heating OFF', SSC, SST, SCFG);
```

```
STEPDESC (20, 'Discharge C11', SSC, SST, SCFG);
```

```
STEPDESC (30, 'Discharge D11', SSC, SST, SCFG);
```

```
STEPDESC (40, 'Close Valves', SSC, SST, SCFG);
```

(\* Blocking from restarting of the strategy \*)

```
if (SST.EMSTATE = 1 or SST.EMSTATE = 3) then
SSC.CSTEP := 40;
end_if;
```

#### **(Steps and Transitions)**

```
case SSC.CSTEP of
```

```
1: (* Initial Step. DO NOT REMOVE *)
```

```
(* Check initial conditions *)
```

```
IF INITCONDFAIL then (* To check the Initial Conditions *)
```

```
FAILED (SSC) ; (* Execute 'Holding for error' sequence *)
```

```
end_if;
```

```
TRANS (true, 'Jump to Step 10', 10, SSC, SST, SCFG) ; (* Transition to  
Step 10 *)
```

```
10: (Heating OFF)
```

```
FeedingRamp.REM := false;
```

```
FeedingRamp.LTARGETSP := 0.0;
```

```
FeedingRamp.LSTART := true;
```

```
FeedingPID.MAN := true;
```

```
FeedingPID.LOP := 0.0;
```

```
FeedingPump.REM := false;
```

```
FeedingPump.LSP := false;
```

```
HeadTempPID.REM := false;
```

```
HeadTempPID.MAN := true;
```

```
HeadTempPID.LOP := 0.0;
```

```
VolatileTempPID.MAN := true;
```

```
VolatileTempPID.LOP := 0.0;
```

```
TRANS (SSC.ETIME >= 30, 'Wait 3 sec', 10, SSC, SST, SCFG) ;
```

```
20: (Discharge C11)
```

```
if SSC.ONENTRY then (* To be done only the first time the step is executed *)
```

```
BottomLevelPID.REM := false;
```

```
BottomLevelPID.MAN := true;
```

```
BottomLevelPID.LOP := 100.0;
```

```
BottomPump.REM := false;
```

```
BottomPump.LSP := true;
```

```
end_if;
```

```
TRANS (BottomLevelPID.PV <= 20.0 and SSC.ETIME >= 30, 'C11  
Empty', 20, SSC, SST, SCFG) ;
```

```
if SSC.ONEXIT then (* To be done only when step exists *)
```

```
BottomLevelPID.REM := false;
```

```
BottomLevelPID.MAN := true;
```

```
BottomLevelPID.LOP := 0.0;
```

```

BottomPump.REM := false;
BottomPump.LSP := false;
end_if;
30: (Discharge D11)
IF SSC.ONENTRY THEN (* To be done only the first time the step is executed *)
DistillatePump.REM := false;
DistillatePump.LSP := true;
end_if;
TRANS (DistillatePump.ILCKD and SSC.ETIME >= 30, 'D11 Empty',
30, SSC, SST, SCFG); (End of sequence)
if SSC.ONEXIT then (* To be done only when step exists *)
DistillatePump.REM := false;
DistillatePump.LSP := false;
end_if;
40: (* Close Water Valves *)
ChilledWaterValve.REM := false;
ChilledWaterValve.LSP := false;
CoolingWaterValve.REM := false;
CoolingWaterValve.LSP := false;
TRANS (((ChilledWaterValve.LOWPOS and CoolingWaterValve.
LOWPOS and SSC.ETIME >= 30) or (SST.EMSTATE = 1)), 'Valves
Closed', -1, SSC, SST, SCFG);
end_case;
3: (Quick Stop)
(* Step descriptions *)
(Steps and Transitions)
case SSC.CSTEP of
1: (* Transitions should not be included for emergency situations *)
FeedingRamp.REM := false;
FeedingRamp.LTARGETSP := 0.0;
FeedingRamp.LSTART := true;
FeedingPID.MAN := true;
FeedingPID.LOP := 0.0;
FeedingPump.REM := false;
FeedingPump.LSP := false;
BottomLevelPID.REM := false;
BottomLevelPID.MAN := true;
BottomLevelPID.LOP := 0.0;
BottomPump.REM := false;
BottomPump.LSP := false;
HeadTempPID.REM := false;

```

```
HeadTempPID.MAN := true;
HeadTempPID.LOP := 0.0;
VolatileTempPID.MAN := true;
VolatileTempPID.LOP := 0.0;
DistillatePump.REM := true;
DistillatePump.LSP := false;
ChilledWaterValve.REM := false;
ChilledWaterValve.LSP := false;
CoolingWaterValve.REM := false;
CoolingWaterValve.LSP := false;
TRANS (true, 'End', -1, SSC, SST, SCFG); (* To 'Aborted' state *)
(Stopping subsequence)
case SSC.CSTEP of
1:
TRANS (SSC.ETIME >= 10, 'End', -1, SSC, SST, SCFG); (* To 'Stopped'
state after 5 seconds *)
End_case;
(Common Sequence)
(* Checks detected fail conditions *)
if FAILCOND then
FAILED (SSC); (* Forces to Hold the sequence*)
SST.EMSTATE := 1; (* Stopped *)
end_if;
(* Manual mode - based on the process step selected, EMSTATE is adapted *)
If SSC.Mode = 2 then
If SSC.Strategy = 1 then (* Start Distillation Strategy *)
if SSC.CSTEP = 1 then
SST.EMSTATE := 3; (* Ready *)
elsif SSC.CSTEP = 10 then
SST.EMSTATE := 2; (* Starting *)
elsif SSC.CSTEP = 1=20 then
SST.EMSTATE := 16; (* Cooling ON *)
elsif SSC.CSTEP = 1=30 then
SST.EMSTATE := 17; (* Feeding ON *)
else;
end_if;
elsif SSC.Strategy = 2 then (* Stop Distillation Strategy *)
If SSC.CSTEP = 10 then
SST.EMSTATE := 3; (* Ready *)
else;
```

```

SST.EMSTATE := 10; (* Stopping *)
end_if;
else;
end_if;
end_if;

```

In the common sequence, logic for disabling the strategy, EMSTATE switching logic is based on the process states that are implemented.

```

Case SST.EMSTATE of 0: (* Off state *)
SST.DISSTRATEGY := 0; (* OFF state - Stop distillation strategy is disabled *)
SST.DISSTRATEGY02 := TRUE;
If SSC.Strategy <> 0 then
SST.DISSTRATEGY := 0;
SST.EMSTATE := 3; (* Ready *)
end_if;
1: (* Stopped *)
SST.DISSTRATEGY := 0; (* Stopped state - Stop distillation strategy is disabled *)
SST.DISSTRATEGY02 := TRUE;
If (SSC.Strategy = 1 and not SST.HeldForError) then
SST.EMSTATE := 3; (* Ready *)
end_if;
2: (* Starting *)
SST.DISSTRATEGY := 0; (* Starting state - Start distillation strategy is disabled *)
SST.DISSTRATEGY01 := TRUE;
SST.EMSTATE := 3; (* Ready *)
If (ChilledWaterValve.HIGHPOS and CoolingWaterValve.HIGHPOS
and SSC.ETIME >= 30) then
SST.EMSTATE := 16; (Feeding ON)
end_if;
3: (* Ready *)
If (not SST.HeldForError) then
If ChilledWaterValve.LSP = true and CoolingWaterValve.LSP = true
then
SST.EMSTATE := 2; (* Stopping *)
end_if;
SST.DISSTRATEGY02 := TRUE;
else
SST.EMSTATE := 1; (* Stopped *)
end_if;
5: (* Producing *)

```



```
If (FeedingRamp.LTARGETSP = 0.0 and (HeadTempPID.MAN = true and
HeadTempPID.LOP = 0.0)) then

SST.EMSTATE := 10; (* Stopping *)

end_if;

SST.DISSTRATEGY := 0; (* Producing state - Start distillation strategy is
disabled *)

SST.DISSTRATEGY01 := TRUE; (* Stopping *)

10: (* Stopping *)

If DistillatePump.LowPOS and FeedingPump.LOWPOS and
ChilledWaterValve.LOWPOS and CoolingWaterValve.LOWPOS and
BottomPump.LowPOS then

SST.EMSTATE := 1; (* Stopped *)

end_if;

If Not SST.Running then

SST.DISSTRATEGY := 0; (* Stopping state - Stop distillation strategy is disabled
*)

SST.DISSTRATEGY01 := TRUE;

end_if;

16: (* Cooling ON *)

If not (SSC.Strategy = 3) then

If FeedingPump.HIGHPOS and BottomPump.HIGHPOS then

SST.EMSTATE := 17; (* Feeding ON *)

end_if;

If Not SST.Running then

SST.DISSTRATEGY := 0; (* Cooling ON state - Stop distillation strategy is
disabled *)

SST.DISSTRATEGY02 := TRUE;

end_if;

else

If SST.Running then

SST.EMSTATE := 10; (* Stopping *)

end_if;

end_if;

17: (* Feeding ON *)

If not (SSC.Strategy = 3) then

If (HeadTempPID.LOP > 5.0 and VolatileTempPID.LOP > 5.0 and
BottomLevelPID.PV >= 50.0) then

SST.EMSTATE := 5; (* Producing *)

end_if;

If Not SST.Running then

SST.DISSTRATEGY := 0; (* Feeding ON state - Stop distillation strategy is
disabled *)

SST.DISSTRATEGY02 := TRUE;
```

```
end_if;  
else  
If SST.Running then  
SST.EMSTATE := 10; (* Stopping *)  
end_if;  
end_if;  
end case;
```

# Pump Set Pattern

## What's in This Part

Pump Set Equipment Module .....	52
PUMPSETPATTERN - Pump Set Pattern Logic .....	58
Auxiliary Functions of Pump Set Equipment Module.....	62

## Overview

This part provides a detailed description of the functions, pins, pin layout, and variables of the function blocks of the Pump Set Pattern.

These function blocks do not reflect any specific installation.

### **⚠ 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.<sup>1</sup>
- 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.**

<sup>1</sup> 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.

# Pump Set Equipment Module

## What's in This Chapter

Description of Pump Set Equipment Module .....	52
Strategy Selection and Functionality .....	53
Disabling the Strategies Based on Equipment Module State .....	54
Input and Output Parameters .....	54
Input Parameters Applicable For Each Strategy .....	55
Pump Set Equipment Module State Machine .....	55
Pump Set Equipment Module Diagnosis State .....	56
Pump Set Pumping Asset Diagnosis State .....	57
Modifying the Pump Set Pattern .....	57

## Overview

This chapter describes the Pump Set Equipment Module management.

## Description of Pump Set Equipment Module

### Overview

It is an unprotected function block which is derived from `EMLOGIC` user-defined DFB. The `PUMPSETPATTERN` DFB checks the status and controls the operation of control modules (Motors/Valves/Sensors, etc.) and services (Initial Conditions/ Detected Failure Conditions, etc.) to form a Pump Set equipment module.

A pump, an inlet valve, an outlet valve, and a drain valve are collectively referred to as a pumping asset. To configure a pumping asset, connecting a pump is mandatory whereas the valves are optional.

The `PUMPSETPATTERN` DFB can be used to run common pumping applications like Sump management, Transfer and Circulation pump, etc. The maximum number of pumping assets supported for operation is five. The number of pumping assets can be modified according to the users application requirement.

## Function Description

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

Function	Description
Incorrect Configuration	<p>Determines the total number of pumps connected to the Pump Set from the connections to the pump input pins of the DFB.</p> <p>If the pump connections are not in order, it detects an incorrect configuration. For example, If <code>PUMP01</code>, <code>PUMP02</code>, <code>PUMP04</code> input pins are connected, the incorrect configuration is detected since the pumps are not connected sequentially. When there is a incorrect configuration is detected, <code>PUMPSETPATTERN</code> equipment is not available for operation.</p>
Order Sequence	<p>The order sequence arranges the pumping assets in the order of minimum operation time/ highest priority first to maximum operation time/ least priority last based on the type of strategy run. In case of priority type, lower priority value corresponds to higher priority.</p> <p>If a balance strategy is selected, the order sequence is arranged by operation time. If there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher preference.</p> <p>If a priority strategy is selected, the order sequence is arranged by priority. If there are multiple pumping assets with the same priority, then for each set of same priority pumping assets, lesser operation time is considered as higher priority. Furthermore, in a set of pumping assets with same priority, if there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher priority.</p>

Function	Description
Switchover	<p>If an undesirable condition is detected in an active pumping asset in the field, a Stop command is issued from the Pump Set to the inoperable pumping asset and the next available pumping asset in the order sequence is started.</p> <p>When the detected fault status of a pumping asset is reset, it automatically goes to Idle mode and waits for the command from the next calculation/ switching caused by the following factors:</p> <ul style="list-style-type: none"> <li>Change in the number of requested pumping assets.</li> <li>Detected failure of an active pump and switch over to an Idle one.</li> <li>Recalculation is requested or forced.</li> </ul> <p><b>NOTE:</b> Switchover does not apply if a user has paused/stopped/held a strategy during its execution.</p>
Operation	<p>The Pump Set can be run in <i>Operator</i> or <i>Program</i> owner. Strategies are used to run and stop the Pump Set, and once it achieves the target (or target not achieved), the Strategy is completed and the equipment module state reflects the reality of the field.</p> <ul style="list-style-type: none"> <li>Incorrect configuration or a detected failure condition will force the Pump Set to Stopped state.</li> <li>If the Pump Set is in <i>Stopped</i> state, and the initial conditions are not satisfied (<i>INITCONDFAIL</i> is <i>TRUE</i>), then the Pump Set will go to <i>Off</i> state, and no command is possible, until the initial conditions are satisfied.</li> <li>In <i>Stopped</i> state, the Pump Set can be started by selecting a strategy. Once a strategy is completed, if the Pump Set does not achieved the desired set point, it will take corrective action when a pumping asset becomes available for it to control.</li> <li>Recalculation Strategy will capture the Priority/ operation time of the pumping assets, refresh the order sequence and evaluate the assets which need to run and those that can be stopped.</li> <li>The pumping assets are started as per the order sequence.</li> <li>The execution of Stop Strategy, or Abort command during a strategy execution, or a detected failure condition will stop the pumping assets in parallel.</li> </ul>
Recalculation	<p>Recalculation of the order sequence can be triggered either by the execution of the Recalculate strategy, or by the rising edge of the input pin. If the equipment module state was of <i>Program</i> type prior to recalculation, then the input pin <i>ACTIVEPUMPSSP</i> is considered for set point. After recalculating the order sequence, the Pump Set will try to achieve the set point (by starting the higher preference assets or stopping the lower preference assets). Once set point is achieved, Pump Set evaluates whether switch over is possible, by checking if an asset with higher preference in the order sequence is available and is idle, and pumping asset with lesser preference is available and active. This way, the Pump Set always tries to achieve the set point but does not exceeds it.</p>
Detected Failure Condition	<p>If a detected failure occurs then the Pump Set follows the Stop strategy execution matrix (Stopping to Stopped).</p>

**NOTE:** For all the *DEVCTL* instances connected to the Pump Set, the *FAILREARMEN*, *CONFREARMEN*, and *FAILEN* pins should be *TRUE*. Also, the *TIMEOUT* value should be greater than 0 seconds.

**NOTE:** PumpSetPattern do not check for interlock conditions of individual asset for availability of the pumping assets. However, user can still connect interlock of one asset to another asset (e.g -Inlet valve must be open before starting of the pump) as PumpSetPattern will be starting pumping assets in sequence. If user wants other process interlocks on pumping assets then he should apply it as a *failure* condition of PumpSetPattern or he can modify pattern to cater his requirement.

## Strategy Selection and Functionality

### Description

The given below table explains the selection of Strategy and its functionality.

Strategy	Strategy Number	Action
Program SP/Balance	1	The number of active pumps set point is defined by input pin <i>ACTIVEPUMPSSP</i> . This strategy is only enabled in case that the <i>ACTIVEPUMPSSP</i> input pin has been connected. In this strategy, the order sequence is determined by pump operation time.
Operator SP/Balance	2	The number of active pumps is defined by the <i>ACTIVEPUMPSSP</i> Input Parameter. In this strategy, the order sequence is determined by pump operation time.

Strategy	Strategy Number	Action
Program SP/Priority	3	The number of active pumps is defined by input pin <code>ACTIVEPUMPSSP</code> of the DFB. This strategy is only enabled in case that the <code>ACTIVEPUMPSSP</code> input pin has been connected. In this strategy, the order sequence is determined by pump priority.
Operator SP/Priority	4	The number of active pumps is defined by the <code>ACTIVEPUMPSSP</code> input parameter. In this strategy, the order sequence is determined by pump priority.
Recalculate	5	Forces the <code>PUMPSETPATTERN</code> DFB to refresh the order sequence and perform the recalculation function.
Stop Equipment	6	Stop operation of all the pumping assets simultaneously.

## Disabling the Strategies Based on Equipment Module State

### Description

For each equipment module state, the disabled strategies are shown below:

Equipment Module State	Code	Disabled Strategies
Off	0	1,2,3,4,5,6
Stopped	1	5,6
Stopping	10	1,2,3,4,5
Running Program SP/Balance	16	-
Running Operator SP/Balance	17	-
Running Program SP/Priority	18	-
Running Operator SP/Priority	19	-
Configuring	20	If strategy execution state is Idle, then all strategies are disabled. If not Idle, then all are disabled except the current strategy.
Running Externally	21	5,6

## Input and Output Parameters

### Description

The Input parameters are as shown below:

Variable Name	Parameter Number	Data Type	Description
<code>IOPAR.ACTIVEPUMPSSP</code>	1	INT	Number of Active Pumps
<code>IOPAR.RESETOPERATIONTIME</code>	2	BOOL	Reset Pumps Operation Time
<code>IOPAR.PUMPPRIORITY[0]</code>	3	INT	Priority of Pumping Asset 01
<code>IOPAR.PUMPPRIORITY[1]</code>	4	INT	Priority of Pumping Asset 02
<code>IOPAR.PUMPPRIORITY[2]</code>	5	INT	Priority of Pumping Asset 03
<code>IOPAR.PUMPPRIORITY[3]</code>	6	INT	Priority of Pumping Asset 04
<code>IOPAR.PUMPPRIORITY[4]</code>	7	INT	Priority of Pumping Asset 05

The Output parameters are as shown below:

Variable Name	Parameter Number	Data Type	Description
IOPAR.EMDIAGSTATE	1	ENUM	Pump Set Diagnosis State
IOPAR.REQUESTEDPUMPS	2	INT	Number of requested Pumps
IOPAR.ACTIVEPUMPS	3	INT	Number of active pumps
IOPAR.PUMPOPERATIONTIME[0]	4	DURATION	Operation Time for Pump 01
IOPAR.ASSETDIAGSTATE[0]	5	ENUM	Diagnosis state of Pumping Asset 01
IOPAR.PUMPOPERATIONTIME[1]	6	DURATION	Operation Time for Pump 02
IOPAR.ASSETDIAGSTATE[1]	7	ENUM	Diagnosis state of Pumping Asset 02
IOPAR.PUMPOPERATIONTIME[2]	8	DURATION	Operation Time for Pump 03
IOPAR.ASSETDIAGSTATE[2]	9	ENUM	Diagnosis state of Pumping Asset 03
IOPAR.PUMPOPERATIONTIME[3]	10	DURATION	Operation Time for Pump 04
IOPAR.ASSETDIAGSTATE[3]	11	ENUM	Diagnosis state of Pumping Asset 04
IOPAR.PUMPOPERATIONTIME[4]	12	DURATION	Operation Time for Pump 05
IOPAR.ASSETDIAGSTATE[4]	13	ENUM	Diagnosis state of Pumping Asset 05

## Input Parameters Applicable For Each Strategy

### Description

The below table provides the information about the input parameters that are applicable for each strategy.

Strategy	Input Parameters Applicable
Program SP/Balance	2
Operator SP/Balance	1,2
Program SP/Priority	2,3,4,5,6,7
Operator SP/Priority	1,2,3,4,5,6,7
Recalculate	2
Stop Equipment	2

## Pump Set Equipment Module State Machine

### Description

The below given table provides the list of applicable strategies for each Equipment Module State, the transient state when the strategy is started, and the final equipment module state once the strategy is completed:

Initial EM State	Strategy Command	Transient EM State	Final EM State
Off	No action possible	-	Stopped
			Running Externally
Stopped	Program SP/Balance	Configuring	Running Program SP/Balance
	Operator SP/Balance	Configuring	Running Operator SP/Balance

Initial EM State	Strategy Command	Transient EM State	Final EM State
	Program SP/Priority	Configuring	Running Program SP/Priority
	Operator SP/Priority	Configuring	Running Operator SP/Priority
	-	-	Off
	-	-	Running Externally
<ul style="list-style-type: none"> <li>Running Program SP/Balance</li> <li>Running Operator SP/Balance</li> <li>Running Program SP/Priority</li> <li>Running Operator SP/Priority</li> </ul>	Recalculate	Configuring	RunningX*
	Program SP/Balance	Configuring	Running Program SP/Balance
	Operator SP/Balance	Configuring	Running Operator SP/Balance
	Program SP/Priority	Configuring	Running Program SP/Priority
	Operator SP/Priority	Configuring	Running Operator SP/Priority
	Stop Equipment	Stopping	Stopped
	-	-	Running Externally
Configuring	No action possible	-	-
Stopping	No action possible	-	-
Running Externally	Program SP/Balance	Configuring	Running Program SP/Balance
	Operator SP/Balance	Configuring	Running Operator SP/Balance
	Program SP/Priority	Configuring	Running Program SP/Priority
	Operator SP/Priority	Configuring	Running Operator SP/Priority
	-	-	Stopped
*: X represents initial Equipment Module State			

## Pump Set Equipment Module Diagnosis State

### Description

The Pump Set equipment module diagnosis state is evaluated as given below:

- **Normal:** State is valid if below condition is true:
  - There are no detected failure and alert conditions.
- **Warning:** State is valid if any of below condition is true:
  - When total number of active pumping assets are more than number of assets requested by pump set pattern.
  - If stop strategy could not stop all the pumping assets.
  - If requested number of assets is equal to fully active assets and there is a problem in any pumping asset.
- **Failure:** State is valid if any of below condition is true:
  - When there is an incorrect configuration.
  - When there is a detected failure condition.
  - When the number of active pumping assets is lesser than number of pumping assets required by pump set pattern.

**NOTE:** The Pump Set equipment module diagnosis state is not evaluated when the Pump Set is in transient state (Configuring or Stopping) to avoid unnecessary alert and detected failure generation. However, the Pump Set detected failure and incorrect configuration are always evaluated.



# Pump Set Pumping Asset Diagnosis State

## Description

Each individual pumping asset has its own diagnosis state and each of them are evaluated as given below:

- **Normal:** The set of Valves and Pump is operating properly.
- **Warning:** The set of Valves and Pump caused a problem during its last operation. The last time that the algorithm tried to execute the complete sequence for starting/stopping the Pumping Asset for some reason (for example, the operating mode was not correct, it triggered an alarm, the confirmations were not reached, etc.). Switching from alert to a detected failure does not reset the alert condition, the only way that the alert is reset is when the same Pumping Asset is operated again for starting or stopping it.
- **Failure:** Any one of the Valves or Pump is in inoperable condition (The `FAILED`, `ALARM`, `REARMR` or `OOS` value is `TRUE`).

## Modifying the Pump Set Pattern

### Description

The `PUMPSETPATTERN` handles 5 pumping assets, which is the maximum number of assets supported. If the required number of pumping assets in the application is less than 5, then the below steps can be followed to optimize the `PUMPSETPATTERN` DFB.

- Remove the input/output pins that are not required. For example, if only 4 pumping assets are required, then remove `PUMP05`, `INLETVALVE05`, `OUTLETVALVE05`, and `DRAINVALVE05`.
- In the private variables, the size of arrays with the description `User modifiable internal array` has to be reduced to reflect the expected number of pumping assets. For example, if only 3 pumping assets are required, reduce the array size to `[0...2]`.
- In the `Init` section of the DFB, modify the mapping of the input/output pins to the arrays by removing the references of the deleted pins.
- In the equipment module state section, modify the mapping of the array elements to the input/output pins by removing the references of the deleted pins.
- In the equipment module state section, remove the unwanted instances of the `PUMPINGASSETSEQ1` DFB. For example, if only 3 pumping assets are required, delete the instances of `PumpingAsset3` and `PumpingAsset4` from the **EMState** section.

# PUMPSETPATTERN - Pump Set Pattern Logic

## What’s in This Chapter

Description .....	58
DFB Representation .....	58
Inputs .....	59
Outputs .....	60
Inputs/Outputs .....	60

## Overview

This chapter provides a detailed description of the functions, pins, pin layout, and variables of the PUMPSETPATTERN DFB.

## Description

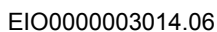
### General

The PUMPSETPATTERN DFB can be used to run common pumping applications like Sump management, Transfer and Circulation pump, etc. The maximum number of pumping assets supported for operation is five. The number of pumping assets can be modified according to the users application requirement.

## DFB Representation

### Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



## EIO0000003014.06

## EIO0000003014.06

EIO0000003014.06

## PumpSet\_ENGPAR\_DDT Type

Name	Type	Description
SWITCHONDELAY	TIME	Delay between the end of the starting sequence of a pumping asset and the start of the starting sequence of another pumping asset.
DRAININGTIME	TIME	Time for which Drain Valve will be open when stopping the pumping asset.
DELAYAFTERVALVEOPEN	TIME	Delay time before next action after opening valves.
DELAYAFTERVALVE-CLOSE	TIME	Delay time before next action after closing valves.
DELAYAFTERPUMPSTART	TIME	Delay time before next action after starting pumps.
DELAYAFTERPUMPSTOP	TIME	Delay time before next action after stopping pumps.
ENABLEDSTRATEGIES	INT	Determines strategies to be enabled. <ul style="list-style-type: none"> <li>1 = Priority only</li> <li>2 = Balance only</li> <li>Any other value = All enabled</li> </ul>
OUTVALVEFIRST	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.

## Outputs

### Output Parameter Description

Parameter	Type	Description
CONFIGERROR	INT	Configuration alert code.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
IOPAR	PUMPSET_IOPAR_DDT	Input and output parameters
PUMP01	DEVCTL_SC_DDT	Asset 01 pump
INLETVALVE01	DEVCTL_SC_DDT	Asset 01 inlet valve
OUTLETVALVE01	DEVCTL_SC_DDT	Asset 01 outlet valve
DRAINVALVE01	DEVCTL_SC_DDT	Asset 01 drain valve
PUMP02	DEVCTL_SC_DDT	Asset 02 pump
INLETVALVE02	DEVCTL_SC_DDT	Asset 02 inlet valve
OUTLETVALVE02	DEVCTL_SC_DDT	Asset 02 outlet valve
DRAINVALVE02	DEVCTL_SC_DDT	Asset 02 drain valve
PUMP03	DEVCTL_SC_DDT	Asset 03 pump
INLETVALVE03	DEVCTL_SC_DDT	Asset 03 inlet valve
OUTLETVALVE03	DEVCTL_SC_DDT	Asset 03 outlet valve
DRAINVALVE03	DEVCTL_SC_DDT	Asset 03 drain valve

Parameter	Type	Description
PUMP04	DEVCTL_SC_DDT	Asset 04 pump
INLETVALVE04	DEVCTL_SC_DDT	Asset 04 inlet valve
OUTLETVALVE04	DEVCTL_SC_DDT	Asset 04 outlet valve
DRAINVALVE04	DEVCTL_SC_DDT	Asset 04 drain valve
PUMP05	DEVCTL_SC_DDT	Asset 05 pump
INLETVALVE05	DEVCTL_SC_DDT	Asset 05 inlet valve
OUTLETVALVE05	DEVCTL_SC_DDT	Asset 05 outlet valve
DRAINVALVE05	DEVCTL_SC_DDT	Asset 05 drain valve
EMSC	EMCTL_SC_DDT	Equipment module sequence control structure, page 27.
EMST	EMCTL_ST_DDT	Provides the necessary data for monitoring and controlling Pump Set Equipment module, page 24.
EMCFG	EMCTL_CFG_DDT	Provides the necessary data for background monitoring, page 61.

**NOTE:** For additional information of those parameters that belong to DEVCTL\_SC\_DDT type (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual).

## PUMPSET\_IOPAR\_DDT Type

Name	Type	Description
PUMPSETOPERATIONTIME	ARRAY[0...4] OF TIME	Operation time for pumps as output parameter.
ASSETDIAGSTATE	ARRAY[0...4] OF INT	Pumping asset diagnosis state as output parameter.
PUMPPRIORITY	ARRAY[0...4] OF INT	Pumping asset priority as input parameter. The minimum value of the array elements is 1.
ACTIVEPUMPSSP	INT	Number of active pumps set point as input parameter. Minimum value of the ACTIVEPUMPSSP is 1 and the maximum value is the number of configured pumps.
REQUESTEDPUMPS	INT	Number of currently requested pumps to be run as output parameter.
ACTIVEPUMPS	INT	Number of pumps currently active as output parameter.
EMDIAGSTATE	INT	PumpSet Diagnosis state as output parameter.
RESETOPERATIONTIME	BOOL	Reset pumps operating Time as input parameter.

# Auxiliary Functions of Pump Set Equipment Module

## What's in This Chapter

PUMPINGASSETSEQ .....	62
-----------------------	----

## Overview

This section describes the function blocks that supplement the `PUMPSETPATTERN` DFB and that have the objective of facilitating the programming of the equipment module.

## PUMPINGASSETSEQ

## Overview

This section describes the `PUMPINGASSETSEQ` DFB.

## Description

### General

It is an unprotected function block, that is used to control the start and stop sequence of a pumping asset. It evaluates the alert condition of the pumping asset diagnosis state.

## Pumping Asset Sequence Execution

The Operation sequence to start a pumping asset is as follows:

- If used, close the Drain valve, wait for the Drain Valve to get closed (plus the delay time given by `DELAYAFTERVALVECLOSE`) or a detected failure.
- If `OUTVALVEFIRST` engineering parameter is `True`:
  - If used, open the Inlet valve, wait for the inlet valve to open (plus the delay time for each given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
  - If used, open the Outlet valve, wait for the outlet valve to open (plus the delay time for each given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by `DELAYAFTERPUMPSTART`) or for a detected failure.
- If `OUTVALVEFIRST` engineering parameter is `False`:
  - If used, open the Inlet Valve, wait for the Inlet Valve to open (plus the delay time given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by `DELAYAFTERPUMPSTART`) or for a detected failure.
  - If used, open the Outlet Valve, wait for the Outlet Valve to open (plus the delay time given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
- Wait for the delay time in case if it is needed to start more pumps.

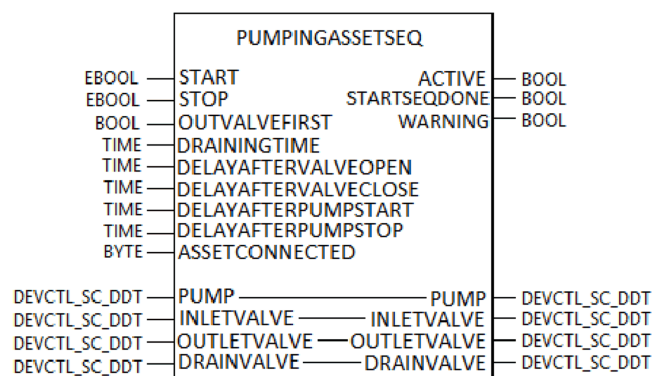
The Operation sequence to stop a pumping asset is as follows:

- If `OUTVALVEFIRST` engineering parameter is `True`:
  - Stop the pump, wait for the pump to stop (plus the delay time given by `DELAYAFTERPUMPSTOP`) or for a detected failure.
  - If used, close the Inlet valve, wait for the inlet valve to close (plus the delay time for each given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
  - If used, close the Outlet valve, and wait for the outlet valve to close (plus the delay time for each given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
  - If used, open the Drain and wait for the Drain Valve to open (plus the delay time given by `DRAININGTIME`) or for a detected failure. Wait for the configured `DRAININGTIME` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
- Else, if `OUTVALVEFIRST` engineering parameter is `False`:
  - If used, close the Outlet Valve, wait for the Outlet Valve to get closed (plus the delay time given by `DELAYAFTERVALVECLOSE`) or a detected failure.
  - Stop the Pump, and wait for the Pump to stop (plus the delay time given by `DELAYAFTERPUMPSTOP`) or for a detected failure.
  - If used, open the Drain valve, and wait for the Drain Valve to open (plus the delay time given by `DRAININGTIME`) or a detected failure. Wait for the configured `DRAININGTIME` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
  - If used, close the Inlet Valve, wait for the Inlet Valve to close (plus the delay time given by `DELAYAFTERVALVECLOSE`) or for a detected failure.

## DFB Representation

### Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



## Inputs

### Input Parameter Description

Parameter	Type	Description
<code>START</code>	<code>EBOOL</code>	Trigger the start operation.
<code>STOP</code>	<code>EBOOL</code>	Trigger the stop operation.

Parameter	Type	Description
OUTVALVEFIRST	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.
DRAININGTIME	TIME	Time for which drain valve will be open when stopping the pumping asset.
DELAYAFTERVALVEOPEN	TIME	Delay time before next action after opening valves.
DELAYAFTERVALVECLOSE	TIME	Delay time before next action after closing valves.
DELAYAFTERPUMPSTART	TIME	Delay time before next action after starting pumps.
DELAYAFTERPUMPSTOP	TIME	Delay time before next action after starting pumps.
ASSETCONNECTED	BYTE	Indicates whether the components of a pumping asset are connected or not.

## Outputs

### Output Parameter Description

Parameter	Type	Description
ACTIVE	BOOL	Indicates the status of the pumping asset sequence: <ul style="list-style-type: none"> <li>• True when start sequence is started.</li> <li>• False only when stop sequence is fully executed.</li> </ul>
STARTSEQDONE	BOOL	Indicates that the start sequence has executed completely.
WARNING	BOOL	Alerts based on each equipment availability/ detected failure/ alarm / owner / interlock.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
PUMP	DEVCTL_SC_DDT	Pump
INLETVALVE	DEVCTL_SC_DDT	Inlet valve
OUTLETVALVE	DEVCTL_SC_DDT	Outlet valve
DRAINVALVE	DEVCTL_SC_DDT	Drain valve

**NOTE:** For additional information of those parameters that belong to DEVCTL\_SC\_DDT type (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual).



# Flow Control Pattern


## What's in This Part

Flow Control Equipment Module .....	66
FLOWCTLPATTERN - Flow Control Process Logic .....	73
Auxiliary Functions of Flow Control Equipment Module .....	78

## Overview

This part provides a detailed description of the functions, pins, pin layout, and variables of the function blocks of the Flow Control Pattern.

These function blocks do not reflect any specific installation.

 **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.<sup>1</sup>
- 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.**

<sup>1</sup> 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.

# Flow Control Equipment Module

## What's in This Chapter

Description of Flow Control Equipment Module .....	66
Strategy Selection and Functionality .....	67
Disabling the Strategies Based on Equipment Module State .....	68
Input and Output Parameters .....	68
Input Parameters Applicable For Each Strategy .....	69
Flow Control Equipment Module State Machine .....	69
Setpoint Management .....	70
Flow Control Equipment Module Diagnosis State .....	71
Flow Control Pumping Asset Diagnosis State .....	72
Modifying the Flow Control Pattern .....	72

## Overview

This chapter describes the Flow Control Equipment Module management.

## Description of Flow Control Equipment Module

### Overview

It is an unprotected function block which is derived from `EMLOGIC` user-defined DFB. The `FLOWCTL_PATTERN` DFB checks the status and controls the operation of control modules (Motors/Valves/Sensors, etc.) and services (Initial Conditions/ Detected Failure Conditions, etc.) to form a Flow Control Equipment Module.

A variable speed pump, an inlet valve, an outlet valve, and a drain valve are collectively referred to as a pumping asset. To configure a pumping asset, connecting a pump is mandatory whereas the valves are optional.

### Function Description

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

Function	Description
Incorrect Configuration	<p>Determines the total number of pumps connected to the <code>FLOWCTL_PATTERN</code> DFB from the connections to the pump input pins</p> <p>If the pump connections are not in order, it detects an incorrect configuration. For example, If <code>PUMP01</code>, <code>PUMP02</code>, <code>PUMP04</code> input pins are connected, the incorrect configuration is detected since the pumps are not connected sequentially. When there is a incorrect configuration is detected, Flow Control equipment module is not available for operation.</p>
Order Sequence	<p>The order sequence arranges the pumping assets in the order of minimum operation time/ highest priority first to maximum operation time/ least priority last based on the type of strategy run. In case of priority type, lower priority value corresponds to higher priority.</p> <p>If a balance strategy is selected, the order sequence is arranged by operation time. If there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher preference.</p> <p>If a priority strategy is selected, the order sequence is arranged by priority. If there are multiple pumping assets with the same priority, then for each set of same priority pumping assets, lesser operation time is considered as higher priority. Furthermore, in a set of pumping assets with same priority, if there are multiple pumping assets with same operation time, then the first connected pumping asset is considered to have a higher priority.</p>
Switchover	<p>If an undesirable condition is detected in an active pumping asset in the field, a Stop command is issued from the <code>FLOWCTL_PATTERN</code> DFB to the inoperable pumping asset and the next available pumping asset in the order sequence is started.</p>

Function	Description
	<p>When the detected fault status of a pumping asset is reset, it automatically goes to Idle mode and waits for the command from the next calculation/ switching caused by the following factors:</p> <ul style="list-style-type: none"> <li>• Change in flow setpoint requires a change in number of pumps.</li> <li>• Detected failure of an active pump and switch over to an Idle one.</li> <li>• Recalculation is requested or forced.</li> </ul> <p><b>NOTE:</b> Switchover does not apply if a user has paused/stopped/held a strategy during its execution.</p>
Operation	<p>The <b>FLOWCTLPATTERN</b> can be run in Operator or Program owner. Strategies are used to run and stop the pumping asset, and once it achieves the target (or target not achieved), the Strategy is completed and the equipment module state reflects the reality of the field.</p> <ul style="list-style-type: none"> <li>• Incorrect configuration or detected failure condition will force the <b>FLOWCTLPATTERN</b> to Stopped state.</li> <li>• If the <b>FLOWCTLPATTERN</b> is in Stopped state, and the initial conditions are not satisfied (<b>INITCONDFAIL</b> is <b>TRUE</b>), then the <b>FLOWCTLPATTERN</b> will go to Off state, and no command is possible, until the initial conditions are satisfied.</li> <li>• When <b>FLOWCTLPATTERN</b> is in <b>STOPPED/ OFF</b> state, if any pumping asset is running outside the control of Equipment Module, then <b>FLOWCTLPATTERN</b> will be in Running Externally state.</li> <li>• In Stopped state, the <b>FLOWCTLPATTERN</b> can be started by selecting a strategy. Once a strategy is completed, if the <b>FLOWCTLPATTERN</b> is not able to achieve the desired flow based on the required number of pumps, then it will generate a detected failure and then it will take corrective action when a pumping asset becomes available for it to control.</li> <li>• Recalculation Strategy will capture the Priority/ operation time of the pumping assets, refresh the order sequence and evaluate the assets which need to run and those that can be stopped.</li> <li>• Whenever a change in flow setpoint requires a change in number of pumps, recalculation will be executed.</li> <li>• The pumping assets are started as per the order sequence.</li> <li>• The execution of Stop Strategy, or Abort command during a strategy execution, or detected failure condition will stop the pumping assets simultaneously.</li> <li>• Pumps running outside the control of Flow Control Equipment Module are not considered for setpoint management.</li> </ul>
Recalculation	<p>Recalculation of the order sequence can be triggered either by the execution of the Recalculate strategy, or by the rising edge of the input pin <b>RECALCULATE</b> or change in the flow set point. If the equipment module state was of Program type prior to recalculation, then based on the change of Flow Set Point, required number of pumps changes thus leads to execution of Recalculation strategy. After recalculating the order sequence, it will try to achieve the set point (by starting the higher preference assets or stopping the lower preference assets). Once set point is achieved, it evaluates whether switch over is possible, by checking if an asset with higher preference in the order sequence is available and is idle, and pumping asset with lesser preference is available and active.</p>
Detected Failure Condition	<p>If a detected failure occurs then the <b>FLOWCTLPATTERN</b> follows the Stop strategy execution matrix (Stopping to Stopped).</p>

**NOTE:** For all the **DEVCTL** and **SDDEVCTL** instances connected to the Flow Control, the **FAILREARMEN**, **CONFREARMEN**, and **FAILEN** pins should be **TRUE**. Also, the **TIMEOUT** value should be greater than 0 seconds.

**NOTE:** PumpSetPattern do not check for interlock conditions of individual asset for availability of the pumping assets. However, user can still connect interlock of one asset to another asset (e.g -Inlet valve must be open before starting of the pump) as PumpSetPattern will be starting pumping assets in sequence. If user wants other process interlocks on pumping assets then he should apply it as a failure condition of PumpSetPattern or he can modify pattern to cater his requirement.

## Strategy Selection and Functionality

### Description

The given below table explains the selection Strategy and its functionality.

Strategy	Strategy Number	Action
Program SP/Balance	1	The required number of pumps is calculated from the value of input pin <code>FLOWSP</code> . This strategy is only enabled in case that the <code>FLOWSP</code> input pin has been connected. In this strategy, the order sequence is determined by pump operation time.
Operator SP/Balance	2	The required number of pumps are calculated from the value of <code>IOPAR.FLOWSP</code> input parameter. In this strategy, the order sequence is determined by pump operation time.
Program SP/Priority	3	The required number of pumps are calculated from the value of input pin <code>FLOWSP</code> . This strategy is only enabled in case that the <code>FLOWSP</code> input pin has been connected. In this strategy, the order sequence is determined by pump priority.
Operator SP/Priority	4	The required number of pumps are calculated from the value of <code>IOPAR.FLOWSP</code> input parameter. In this strategy, the order sequence is determined by pump priority.
Recalculate	5	Forces the <code>FLOWCTL_PATTERN</code> DFB to refresh the order sequence and perform the recalculation function.
Stop Equipment	6	Stop operation of all the pumping assets simultaneously.

## Disabling the Strategies Based on Equipment Module State

### Description

The disable strategies for Flow Control Equipment Module is same as Pump Set Pattern Equipment Module. For additional information, page 54.

## Input and Output Parameters

### Description

The Input parameters are as shown below:

Reference Number	Variable Name	Data Type	Description
1	<code>IOPAR.FLOWSP</code>	REAL	Flow setpoint
2	<code>IOPAR.RESETOPERATIONTIME</code>	BOOL	Reset pumps operation time
3	<code>IOPAR.PUMPPRIORITY[0]</code>	INT	Priority of pumping asset 01
4	<code>IOPAR.PUMPPRIORITY[1]</code>	INT	Priority of pumping asset 02
5	<code>IOPAR.PUMPPRIORITY[2]</code>	INT	Priority of pumping asset 03
6	<code>IOPAR.PUMPPRIORITY[3]</code>	INT	Priority of pumping asset 04
7	<code>IOPAR.PUMPPRIORITY[4]</code>	INT	Priority of pumping asset 05

The Output parameters are as shown below:

Reference Number	Variable Name	Data Type	Description
1	<code>IOPAR.PUMPOPERATIONTIME[0]</code>	DURATION	Operation time for pump 01.
2	<code>IOPAR.PUMPOPERATIONTIME[1]</code>	DURATION	Operation time for pump 02.
3	<code>IOPAR.PUMPOPERATIONTIME[2]</code>	DURATION	Operation time for pump 03.
4	<code>IOPAR.PUMPOPERATIONTIME[3]</code>	DURATION	Operation time for pump 04.
5	<code>IOPAR.PUMPOPERATIONTIME[4]</code>	DURATION	Operation time for pump 05.

Reference Number	Variable Name	Data Type	Description
6	IOPAR.PUMPSPEEDSP[0]	REAL	Speed setpoint of the pump 01.
7	IOPAR.PUMPSPEEDSP[1]	REAL	Speed setpoint of the pump 02.
8	IOPAR.PUMPSPEEDSP[2]	REAL	Speed setpoint of the pump 03.
9	IOPAR.PUMPSPEEDSP[3]	REAL	Speed setpoint of the pump 04.
10	IOPAR.PUMPSPEEDSP[4]	REAL	Speed setpoint of the pump 05.
11	IOPAR.CURRENTFLOWSP	REAL	Current flow setpoint.
12	IOPAR.FLOWPV	REAL	Flow process value.
13	IOPAR.ASSETDIAGSTATE[0]	ENUM	Pumping asset 01 diagnosis state.
14	IOPAR.ASSETDIAGSTATE[1]	ENUM	Pumping asset 02 diagnosis state.
15	IOPAR.ASSETDIAGSTATE[2]	ENUM	Pumping asset 03 diagnosis state.
16	IOPAR.ASSETDIAGSTATE[3]	ENUM	Pumping asset 04 diagnosis state.
17	IOPAR.ASSETDIAGSTATE[4]	ENUM	Pumping asset 05 diagnosis state.
18	IOPAR.EMDIAGSTATE	ENUM	Flow control equipment module diagnosis state.
19	IOPAR.REQUIREDPUMPS	INT	Number of pumps required to meet the flow setpoint.
20	IOPAR.ACTIVEFLOWPUMPS	INT	Number of active pumps to meet the flow setpoint.
21	IOPAR.TOTALACTIVEPUMPS	INT	Total number of active pumps.

## Input Parameters Applicable For Each Strategy

### Description

The below table provides the information about the input parameters that are applicable for each strategy.

Strategy	Input Parameters Applicable
Program SP/Balance	2
Operator SP/Balance	1,2
Program SP/Priority	2,3,4,5,6,7
Operator SP/Priority	1,2,3,4,5,6,7
Recalculate	2
Stop Equipment	2

## Flow Control Equipment Module State Machine

### Description

Flow Control follows the same Equipment Module State Machine as Pump Set Equipment Module. For additional information, page 55.

# Setpoint Management

## Description

Based on the flow setpoint value and the engineering parameters (*MINIMUMSP*, *MAXIMUMSP*, and *OPTIMALSP*), the Flow Control Equipment Module calculates the number of pumps and the speed required to achieve the desired flow setpoint.

The applicable range of values for the above mentioned items is given below:

- Flow setpoint: 0.0 - 100.0%
- Pump speed: 0.0 - 100.0%
- Maximum setpoint: 0.0 - 100.0%
- Minimum setpoint: 0.0 - Maximum setpoint value
- Optimal setpoint: Minimum setpoint to Maximum setpoint value

The calculation for number of pumps and their speed is governed by the following rules:

- Pump speed should never go under minimum setpoint, unless it is necessary to achieve flow setpoint.
- Pump speed should never go over maximum setpoint, unless it is necessary to achieve flow setpoint.
- Pumps running in control of flow control equipment module should always operate at same speed.
- Pump speed should go up to maximum setpoint before starting the next pump.
- Pump should stop when the remaining can work at optimal setpoint.

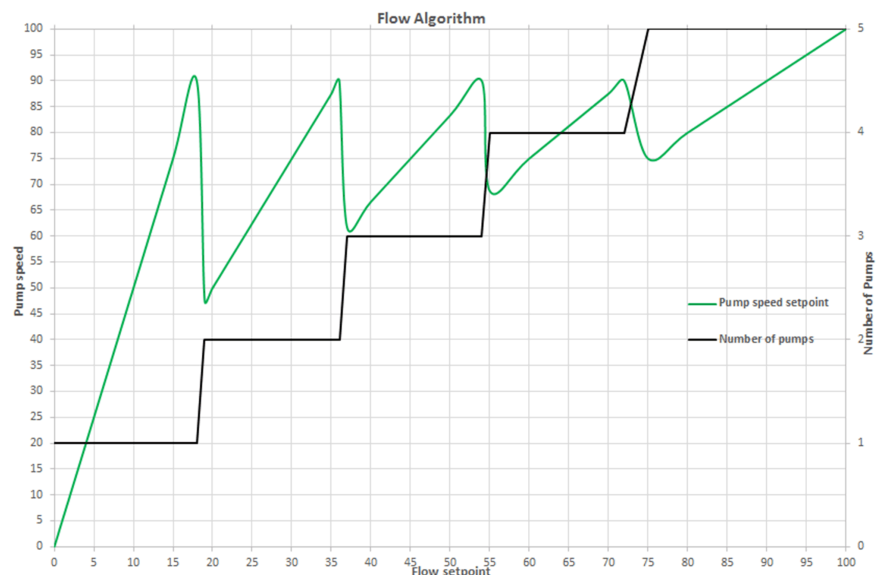
### NOTE:

- This calculation takes place when the equipment module state is either running or configuring.
- If the optimal setpoint value is 0.0%, then pumps once started will not be stopped, since the remaining pumps cannot run at this optimal setpoint and achieve the flow setpoint.
- If the flow setpoint is 0.0%, then the calculation will result in one pump running at 0.0% speed.

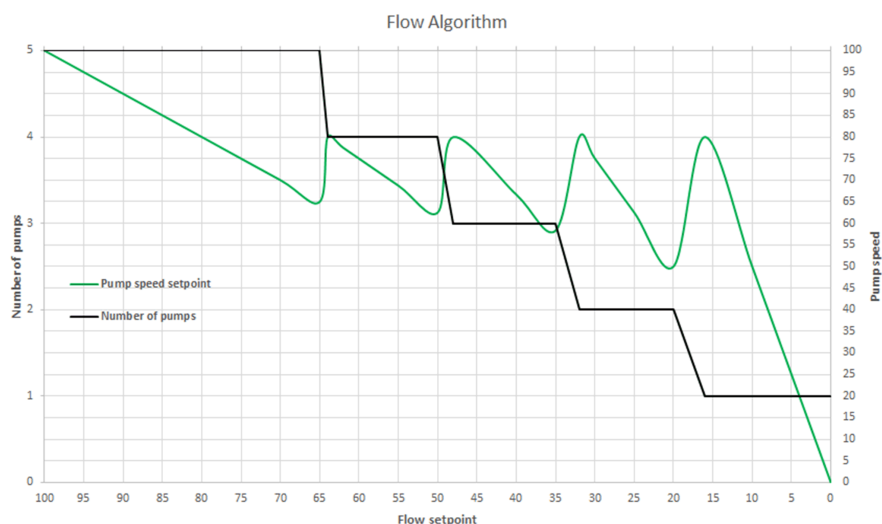
The below given graphs show the calculation result of increasing and decreasing flow setpoint. These graphs are based on the below configuration:

- Maximum setpoint – 90%
- Minimum setpoint – 40%
- Optimal setpoint – 80%
- Configured pumps – 5

**Case 1:** Increasing flow setpoint



**Case 2: Decreasing flow setpoint**



## Flow Control Equipment Module Diagnosis State

### Description

The Flow Control equipment module diagnosis state is evaluated as given below:

- **Normal:** State is valid if all below conditions are true:
  - When no pumping asset is running in external control mode or interlocked or remote Set point.
  - There are no detected failure and alert conditions.
- **Warning:** State is valid if any of below condition is true:
  - When total number of running pumping assets (inclusive of the pumps that are not in control of `FLOWCTLPATTERN`) are running more than number of assets requested by flow control pattern based on the current flow set point.
  - If stop strategy could not stop all the pumping assets.
  - If requested number of assets is equal to fully active assets for Equipment Module and there is a problem in any pumping asset.

- **Failure:** State is valid if any of below condition is true:
  - When there is an incorrect configuration.
  - When there is a detected failure condition.
  - When the number of active flow pumping asset is lesser than number of pumping assets required by flow control pattern to meet the flow setpoint.

## Flow Control Pumping Asset Diagnosis State

### Description

Each individual pumping asset has its own diagnosis state and each of them are evaluated as given below:

- **Normal:** The set of Valves and Pump is operating properly.
- **Warning:** The set of Valves and Pump caused a problem during its last operation. The last time that the algorithm tried to execute the complete sequence for starting/stopping the Pumping Asset caused a problem for some reason (for example, the operating mode was not correct, it triggered an alarm, the confirmations were not reached, etc...). Switching from alert to a detected failure does not reset the alert condition, the only way that the alert is reset is when the same Pumping Asset is operated again for starting or stopping it.
- **Failure:** Any one of the Valves or Pump is in inoperable condition (The `FAILD`, `ALARM`, `REARMR` or `OOS` value is `TRUE`).

## Modifying the Flow Control Pattern

### Description

The `FLOWCTLPATTERN` handles 5 pumping assets, which is the maximum number of assets supported. If the required number of pumping assets in the application is less than 5, then the below steps can be followed to optimize the `FLOWCTLPATTERN` DFB.

- Remove the input/output pins that are not required. For example, if only 4 pumping assets are required, then remove `PUMP05`, `INLETVALVE05`, `OUTLETVALVE05`, and `DRAINVALVE05`.
- In the private variables, the size of arrays with the description `User modifiable internal array` has to be reduced to reflect the expected number of pumping assets. For example, if only 3 pumping assets are required, reduce the array size to `[0...2]`.
- In the **Init** section of the DFB, modify the mapping of the input/output pins to the arrays by removing the references of the deleted pins.
- In the equipment module state section, modify the mapping of the array elements to the input/output pins by removing the references of the deleted pins.
- In the equipment module state section, remove the unwanted instances of the `PUMPINGASSETSEQ1` DFB. For example, if only 3 pumping assets are required, delete the instances of `PumpingAsset3` and `PumpingAsset4` from the **EMState** section.



# FLOWCTLPATTERN - Flow Control Process Logic

## What's in This Chapter

Description .....	73
DFB Representation .....	73
Inputs .....	74
Outputs .....	75
Inputs/Outputs .....	76

## Overview

This chapter provides a detailed description of the functions, pins, pin layout, and variables of the FLOWCTLPATTERN DFB.

## Description

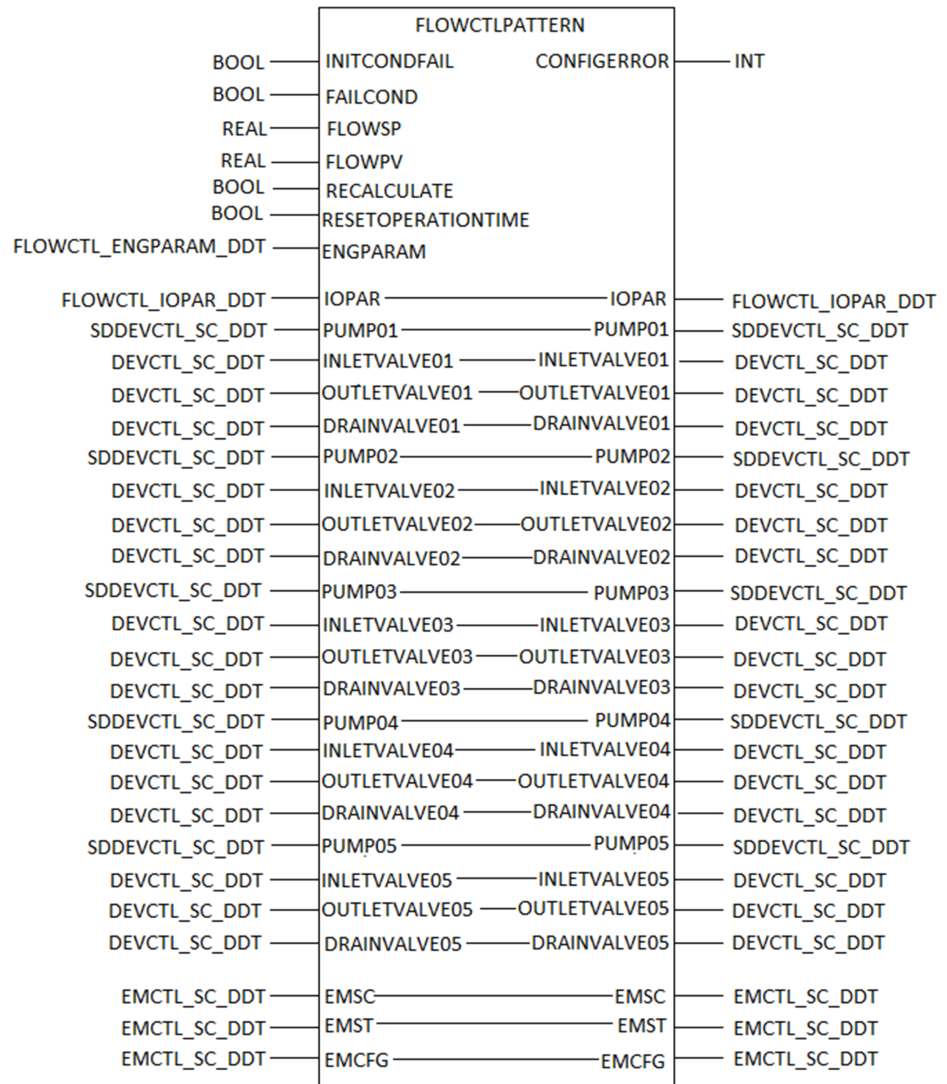
### General

The FLOWCTLPATTERN DFB can be used to run common pumping applications like Sump management, Transfer and Circulation pump, etc. The maximum number of pumping assets supported for operation is five. The number of pumping assets can be modified according to the users application requirement.

## DFB Representation

### Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



## Inputs

### Input Parameter Description

Parameter	Type	Description
INITCONDFAIL	BOOL	Detected failure in initial condition.
FAILCOND	BOOL	Detected failure condition.
FLOWSP	REAL	Flow setpoint.
FLOWPV	REAL	Flow present value.
RECALCULATE	BOOL	Recalculate ON/OFF status of pumps based on current equipment module state.

Parameter	Type	Description
RESETOPERATIONTIME	BOOL	Reset the operation time of all the pumps.
ENGPARG	FLOWCTL_ENGPARG_DDT	<p>Flow Control engineering parameters.</p> <p><b>NOTE:</b></p> <p>If this input pin is not connected, then the following default values are considered by the DFB:</p> <ul style="list-style-type: none"> <li>• MINIMUMSP: 40%</li> <li>• MAXIMUMSP: 90%</li> <li>• OPTIMALSP: 80%</li> <li>• SWITCHONDELAY: 3s</li> <li>• DRAININGTIME: 5s</li> <li>• DELAYAFTERVALVEOPEN: 3s</li> <li>• DELAYAFTERVALVECLOSE: 3s</li> <li>• DELAYAFTERPUMPSTART: 3s</li> <li>• DELAYAFTERPUMPSTOP: 3s</li> <li>• ENABLEDSTRATEGIES: 0</li> <li>• OUTVALVEFIRST: FALSE</li> </ul>

## FLOWCTL\_ENGPARG\_DDT Type

Name	Type	Description
MINIMUMSP	REAL	Minimum pump setpoint.
OPTIMALSP	REAL	Optimal pump setpoint.
MAXIMUMSP	REAL	Maximum pump setpoint
SWITCHONDELAY	TIME	Delay between the end of the starting sequence of a pumping asset and the start of the starting sequence of another pumping asset.
DRAININGTIME	TIME	Time for which Drain Valve will be open when stopping the pumping asset.
DELAYAFTERVALVEOPEN	TIME	Delay time before next action after opening valves.
DELAYAFTERVALVE-CLOSE	TIME	Delay time before next action after closing valves.
DELAYAFTERPUMPSTART	TIME	Delay time before next action after starting pumps.
DELAYAFTERPUMPSTOP	TIME	Delay time before next action after stopping pumps.
ENABLEDSTRATEGIES	INT	<p>Determines strategies to be enabled.</p> <ul style="list-style-type: none"> <li>• 1 = Priority only</li> <li>• 2 = Balance only</li> <li>• Any other value = All enabled</li> </ul>
OUTVALVEFIRST	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.

## Outputs

### Output Parameter Description

Parameter	Type	Description
CONFIGERROR	INT	Configuration alert code.

# Inputs/Outputs

## Input/Output Parameter Description

Parameter	Type	Description
IOPAR	FLOWCTL_IOPAR_DDT	Input and output parameters
PUMP01	SDDEVCTL_SC_DDT	Asset 01 pump
INLETVALVE01	DEVCTL_SC_DDT	Asset 01 inlet valve
OUTLETVALVE01	DEVCTL_SC_DDT	Asset 01 outlet valve
DRAINVALVE01	DEVCTL_SC_DDT	Asset 01 drain valve
PUMP02	SDDEVCTL_SC_DDT	Asset 02 pump
INLETVALVE02	DEVCTL_SC_DDT	Asset 02 inlet valve
OUTLETVALVE02	DEVCTL_SC_DDT	Asset 02 outlet valve
DRAINVALVE02	DEVCTL_SC_DDT	Asset 02 drain valve
PUMP03	SDDEVCTL_SC_DDT	Asset 03 pump
INLETVALVE03	DEVCTL_SC_DDT	Asset 03 inlet valve
OUTLETVALVE03	DEVCTL_SC_DDT	Asset 03 outlet valve
DRAINVALVE03	DEVCTL_SC_DDT	Asset 03 drain valve
PUMP04	SDDEVCTL_SC_DDT	Asset 04 pump
INLETVALVE04	DEVCTL_SC_DDT	Asset 04 inlet valve
OUTLETVALVE04	DEVCTL_SC_DDT	Asset 04 outlet valve
DRAINVALVE04	DEVCTL_SC_DDT	Asset 04 drain valve
PUMP05	SDDEVCTL_SC_DDT	Asset 05 pump
INLETVALVE05	DEVCTL_SC_DDT	Asset 05 inlet valve
OUTLETVALVE05	DEVCTL_SC_DDT	Asset 05 outlet valve
DRAINVALVE05	DEVCTL_SC_DDT	Asset 05 drain valve
EMSC	EMCTL_SC_DDT	Equipment module sequential control structure, page 27.
EMST	EMCTL_ST_DDT	Equipment module status structure.
EMCFG	EMCTL_CFG_DDT	Equipment module configuration structure.

**NOTE:** For additional information of those parameters that belong to DEVCTL\_SC\_DDT type (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual) and SDDEVCTL\_SC\_DDT type (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual).

## FLOWCTL\_IOPAR\_DDT Type

Name	Type	Description
PUMPOPERATIONTIME	ARRAY[0...4] OF TIME	Operation time for pumps as output parameter.
PUMPSPEEDSP	ARRAY[0...4] OF REAL	Speed setpoint of the pumps as output parameter.
FLOWSP	REAL	Flow setpoint as input parameter.
CURRENTFLOWSP	REAL	Current flow setpoint as output parameter.
FLOWPV	REAL	Flow process value as output parameter.

Name	Type	Description
ASSETDIAGSTATE	ARRAY[0 . . . 4] OF INT	Pumping asset diagnosis state as output parameter.
PUMPPRIORITY	ARRAY[0 . . . 4] OF INT	Pumping asset priority as input parameter.
EMDIAGSTATE	INT	Flow control equipment module diagnosis state as output parameter.
REQUIREDPUMPS	INT	Number of pumps required to meet the flow setpoint as output parameter.
ACTIVEFLOWPUMPS	INT	Number of active pumps to meet the flow setpoint as output parameter.
TOTALACTIVEPUMPS	INT	Total number of active pumps as output parameter.
RESETOPERATIONTIME	BOOL	Reset pumps operating time as input parameter.

# Auxiliary Functions of Flow Control Equipment Module

## What's in This Chapter

PUMPINGASSETSEQ1 .....	78
------------------------	----

## Overview

This section describes the function blocks that supplement the `FLOWCTLPATTERN` DFB and that have the objective of facilitating the programming of the equipment module.

## PUMPINGASSETSEQ1

## Overview

This section describes the `PUMPINGASSETSEQ1` DFB.

## Description

### General

It is an unprotected function block, that is used to control the start and stop sequence of a pumping asset. It evaluates the alert condition of the pumping asset diagnosis state.

## Pumping Asset Sequence Execution

The Operation sequence to start a pumping asset is as follows:

- If used, close the Drain valve, wait for the Drain Valve to get closed (plus the delay time given by `DELAYAFTERVALVECLOSE`) or a detected failure.
- If `OUTVALVEFIRST` engineering parameter is `True`:
  - If used, open the Inlet valve, wait for the inlet valve to open (plus the delay time for each given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
  - If used, open the Outlet valve, wait for the outlet valve to open (plus the delay time for each given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by `DELAYAFTERPUMPSTART`) or for a detected failure.
- If `OUTVALVEFIRST` engineering parameter is `False`:
  - If used, open the Inlet Valve, wait for the Inlet Valve to open (plus the delay time given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
  - Start Pump, wait for the Pump running (plus the delay time given by `DELAYAFTERPUMPSTART`) or for a detected failure.
  - If used, open the Outlet Valve, wait for the Outlet Valve to open (plus the delay time given by `DELAYAFTERVALVEOPEN`) or for a detected failure.
- Wait for the delay time in case if it is needed to start more pumps.

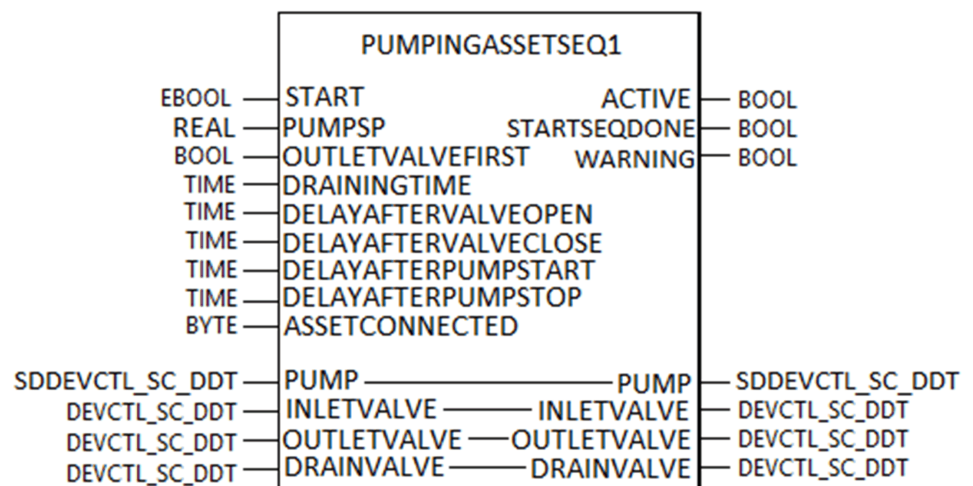
The Operation sequence to stop a pumping asset is as follows:

- If `OUTVALVEFIRST` engineering parameter is `True`:
  - Stop the pump, wait for the pump to stop (plus the delay time given by `DELAYAFTERPUMPSTOP`) or for a detected failure.
  - If used, close the Inlet valve, wait for the inlet valve to close (plus the delay time for each given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
  - If used, close the Outlet valve, and wait for the outlet valve to close (plus the delay time for each given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
  - If used, open the Drain and wait for the Drain Valve to open (plus the delay time given by `DRAININGTIME`) or for a detected failure. Wait for the configured `DRAININGTIME` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
- Else, if `OUTVALVEFIRST` engineering parameter is `False`:
  - If used, close the Outlet Valve, wait for the Outlet Valve to get closed (plus the delay time given by `DELAYAFTERVALVECLOSE`) or a detected failure.
  - Stop the Pump, and wait for the Pump to stop (plus the delay time given by `DELAYAFTERPUMPSTOP`) or for a detected failure.
  - If used, open the Drain valve, and wait for the Drain Valve to open (plus the delay time given by `DRAININGTIME`) or a detected failure. Wait for the configured `DRAININGTIME` and then close the Drain Valve, wait for the Drain Valve to close (plus the delay time given by `DELAYAFTERVALVECLOSE`) or for a detected failure.
  - If used, close the Inlet Valve, wait for the Inlet Valve to close (plus the delay time given by `DELAYAFTERVALVECLOSE`) or for a detected failure.

## DFB Representation

### Representation

The DFB that is used in the program has the following aspect at the section level. You can use it in any of the programming languages, although it is designed for use with the FBD language.



## Inputs

### Input Parameter Description

Parameter	Type	Description
START	EBOOL	Trigger the start/stop operation.
PUMPSP	REAL	Pump setpoint.
OUTLETVALVEFIRST	BOOL	1 = Outlet valve is opened before starting the pump and closed after stopping the pump.
DRAININGTIME	TIME	Time for which drain valve will be open when stopping the pumping asset.
DELAYAFTERVALVEOPEN	TIME	Delay time before next action after opening valves.
DELAYAFTERVALVECLOSE	TIME	Delay time before next action after closing valves.
DELAYAFTERPUMPSTART	TIME	Delay time before next action after starting pumps.
DELAYAFTERPUMPSTOP	TIME	Delay time before next action after starting pumps.
ASSETCONNECTED	BYTE	Indicates whether the components of a pumping asset are connected or not.

## Outputs

### Output Parameter Description

Parameter	Type	Description
ACTIVE	BOOL	Indicates the status of the pumping asset sequence: <ul style="list-style-type: none"> <li>• True when start sequence is started.</li> <li>• False only when stop sequence is fully executed.</li> </ul>
STARTSEQDONE	BOOL	Indicates that the start sequence has executed completely.
WARNING	BOOL	Alerts based on each equipment availability/ detected failure/ alarm / owner / interlock.

## Inputs/Outputs

### Input/Output Parameter Description

Parameter	Type	Description
PUMP	SDDEVCTL_SC_DDT	Pump
INLETVALVE	DEVCTL_SC_DDT	Inlet valve
OUTLETVALVE	DEVCTL_SC_DDT	Outlet valve
DRAINVALVE	DEVCTL_SC_DDT	Drain valve

**NOTE:** For additional information of those parameters that belong to DEVCTL\_SC\_DDT type (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual) and SDDEVCTL\_SC\_DDT type (see EcoStruxure™ Process Expert - General Purpose Library Classic Process Control Services Reference Manual).



---

# Index

## A

auxiliary functions of equipment module	
EMFAILED .....	34
EMTRANS .....	32
auxiliary functions of sequential control	
EMSTABLESTEP .....	35
EMSTEPDESC .....	36

## C

commands	
sequential control .....	20

## E

EMCTL	
DFB representation .....	23
inputs .....	23
inputs/outputs .....	24
public variables .....	27
EMCTL - example of use	
reusable sequence definition .....	41
EMCTL1	
description .....	23
EMCTL1 - example of use	
DFB representation .....	42
reusable equipment module logic .....	42
EMCTL1-example of use	
general description .....	41
EMFAILED	
function call .....	34
general description .....	34
inputs/outputs .....	35
EMLOGIC	
inputs .....	39
inputs/outputs .....	39
EMSTABLESTEP	
function call .....	35
general description .....	35
inputs/outputs .....	36
EMSTEPDESC	
description .....	36
function calls .....	36
inputs .....	37
inputs/outputs .....	38
public variables .....	38
EMTRANS	
function call .....	32
general description .....	32
inputs .....	33
inputs/outputs .....	34
equipment module	
auxiliary functions of equipment module .....	32
EMCTL .....	23
EMCTL - example of use .....	41
equipment module overview .....	17
Strategy Execution Status controller .....	22
Equipment Module	
definition of statuses and commands .....	19
overview .....	17
Strategy Execution Status Matrix .....	21
Equipment Module control templates	
description .....	14

## F

FlowCtlPattern	
inputs/outputs .....	76

## P

process Control services	
list of function blocks .....	15
PumpSetPattern	
inputs/outputs .....	60

## S

sequential control	
function description .....	17
operating modes .....	22
statuses	
equipment module .....	19

Schneider Electric  
35 rue Joseph Monier  
92500 Rueil Malmaison  
France

+ 33 (0) 1 41 29 70 00

[www.se.com](http://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.

EIO0000003014.06