

Premium and Atrium using Unity Pro

Process Control User Manual

07/2012

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

No part of this document may be reproduced in any form or by any means, electronic or mechanical, including photocopying, without express written permission of Schneider Electric.

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

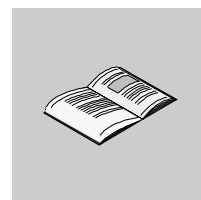
When devices are used for applications with technical safety requirements, the relevant instructions must be followed.

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

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

© 2012 Schneider Electric. All rights reserved.

Table of Contents



	Safety Information	9
	About the Book	11
Part I	Introduction to the process control application ...	13
Chapter 1	Process Control in Automated Systems	15
	Place of the Process Control Application in an Automation Application ..	15
Chapter 2	Hardware and software introduction	17
2.1	Introduction to the Process Control Processors	18
	Features of Processors with Built-in Process Control	18
2.2	Overview of Process Control Software Tools	19
	Configuring a Task Type for the Processor	20
	How to Access Process Control Application Configuration	21
	Description of Loop Controller Configuration Screens	22
	Runtime Screen Configuration Tools	26
	Auto-tuning Control Loops	27
	Setpoint Programmer	28
Chapter 3	Procedure for installing a process control	29
	How To Set Up a Process Control Using PL7	29
Part II	Installing the process control application	31
Chapter 4	Introduction to Loop Controllers	33
4.1	Definition and structure of a loop controller	34
	Structure of Loop Controllers	34
4.2	Description of the Types of Loop Controllers	36
	Types of Process Control	37
	Introduction to the Process Loop	38
	Introducing the Single Loop	39
	Introduction to the Cascade Loop	40
	Introduction to the Auto Selector Loop	41
4.3	Description of the processing branches	42
	Introduction to Built-in Functions	43
	Process Value Processing Branch	47
	Setpoint Processing Branch	49
	Feed Forward Processing Branch	51
	Loop Controller and Command Branch	52

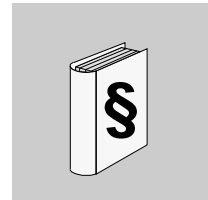
	The ON OFF 2 or 3 States Loop Controller	53
	The PID or IMC Loop Controller	54
	The Split Range or Hot/Cold (PID or IMC) Loop Controller	55
	Output Processing Branch	57
	Servo Drive Output Branch	58
	PWM Branch Output	60
	Loops Summary Table	61
4.4	The Setpoint Programmer	62
	Description of Setpoint Programmer	63
	Linking a Setpoint Programmer to a Control Loop	65
	Guaranteed Dwell Time of Setpoint Programmer	66
	Control Outputs	68
	Bumpless Start	70
	Executing a Profile	72
	Parameters for Setpoint Programmer	74
	Operation Initialization and Monitoring	75
4.5	Global parameters of control loops	76
	Description of Global Parameters for Process Control Loops	76
Chapter 5	Calculation Functions	79
5.1	Functions of the process value branch	80
	Input Format	81
	Filtering of First Order	83
	Square Root	85
	Function Generator	86
	Scaling	89
	Scale Limiter	90
	Level Alarms	91
	Totaling	93
5.2	Functions of the setpoint branch	96
	Ratio	97
	Selection	99
	Scaling	100
	Setpoint Limiter	102
	Tracking Setpoint	104
	Speed Limiter	106
5.3	Feed forward branch functions	108
	Scaling	109
	Lead Lag	111
	Alarm on Deviation	113
5.4	Functions of the Loop Controller Branch	115
	2-State ON/OFF Loop Controller	116
	3-State ON/OFF Loop Controller	119
	PID	122
	PID Parameters	125
	Detailed PID Equations	128

	Model Loop Controller	131
	Model Loop Controller Parameters	135
	Autotuning	137
	Auto-Tuning Parameters	140
	Autotuning Process	142
	Auto-Tuning Operating Modes	144
	Auto-Tuning Diagnostics Parameters	145
	Aborting Auto-Tuning	146
	Split Range	151
	Hot/Cold	154
5.5	Functions of the output branch	157
	Servo	158
	Operating Process Examples of the Servo Function.	161
	PWM	164
	Output Scaling.	166
	Output Limiter	168
	Output Format	170
Chapter 6	Control Loop Configuration	171
6.1	Loop and I/O Configuration	171
	Configuring a Process Control Loop	172
	Configuring the Inputs and Outputs Associated to the Process Control Loops	173
Chapter 7	Control Loop Adjustment	175
7.1	Adjust Feed Forward	176
	Gain Adjustment	177
	Leadlag Adjustment.	178
7.2	PID Adjustment	181
	PID Parameter Adjustment Method.	182
	Role and Influence of the Parameters of a PID during a Loop Adjustment	185
7.3	Model Loop Controller Adjustment	189
	Steps to Follow to Adjust the Model Loop Controller	190
	Instructions for Regulating Ks Static Gain.	191
	Instructions for Adjusting Dead Time or Delay T_DELAY.	192
	How to Regulate the Time Constant	194
Chapter 8	Debugging a Process Control Loop	197
	Description of the Debug Screen	198
	Modifying Parameters for Each Loop	200
	Functional Modification of Each Loop	201
	Debugging the Setpoint Programmer	202
	Data Storage	204

Chapter 9	Operating Modes	205
9.1	Operating process control channels	206
	Distribution of Process Control Handling	207
	Synchronizing Pre- and Post- Processing	208
	Multitasking Application	209
9.2	Handling process control operations according to PLC operating mode.	210
	Handling Process Control Operations According to the PLC Operating Mode	210
9.3	Common operating modes for process control loops	212
	Loop Execution in Manual and Automatic Modes	213
	Autotuning and Tracking Mode	214
	Auto / Manual and Manual / Auto Switching	216
	Behavior of Loops During I/O Errors	217
9.4	Operating Modes for Each Process Control Loop	218
	Process Loop Operating Modes	219
	Single Loop Operating Modes (3 Single Loops)	220
	Cascade Loop Operating Modes	221
	Auto Selector Loop Operating Modes	224
Chapter 10	Process Control Language Objects and IODDT	227
10.1	Process Control Language and IODDT Objects	228
	Language Objects Associated with the Process Control Channels	228
10.2	Language Objects Associated with the Process Control Channels	229
	Introduction	230
	Control Loops Command Word	232
	Setpoint Programmer Command Word CMD_ORDER (%MW.r.m.c.7)	234
	Summary of Selection and Command Words	235
	The "Command Order" Double Word	236
10.3	Language Objects Associated with the Process Loop	238
	Detailed Description of Default Language Objects for the T_PROC_PLOOP-Type IODDT	239
	Detailed Description of Diagnostic Language Objects for the T_PROC_PLOOP-Type IODDT	243
	Detailed Description of Process Control Language Objects for the T_PROC_PLOOP-Type IODDT	245
	Configuration Language Objects	249
10.4	Language Objects Associated with the 3 Single Loops	253
	Detailed Description of Default Language Objects for the T_PROC_3SING_LOOP-Type IODDT	254
	Detailed Description of Diagnostic Language Objects for the T_PROC_3SING_LOOP-Type IODDT	263
	Detailed Description of Process Control Language Objects for the T_PROC_3SING_LOOP-Type IODDT	265
	Configuration Language Objects	270

10.5	Language Objects Associated with the Single Loop Cascade	274
	Detailed Description of Default Language Objects for the	
	T_PROC_CASC_LOOP-Type IODDT	275
	Detailed Description of Diagnostic Language Objects for the	
	T_PROC_CASC_LOOP-type IODDT	281
	Detailed Description of Process Control Language Objects for the	
	T_PROC_CASC_LOOP-Type IODDT	284
	Configuration Language Objects.	289
10.6	Language Objects Associated with the Self-selective Loop	294
	Detailed Description of Default Language Objects for the	
	T_PROC_CONST_LOOP-Type IODDT	295
	Detailed Description of Diagnostic Language Objects for the	
	T_PROC_CONST_LOOP-Type IODDT	301
	Detailed Description of Process Control Language Objects for the	
	T_PROC_CONST_LOOP-Type IODDT	304
	Configuration Language Objects.	309
10.7	Language Objects Associated with the Setpoint Programmer	315
	Detailed Description of the T_PROC_SPP-type IODDT's Implicit	
	Exchange Language Objects	316
	Detailed Description of Status and Default Language Objects for the	
	T_PROC_SPP-Type IODDT	317
	Detailed Description of Process Control Language Objects for the	
	T_PROC_SPP-Type IODDT	322
	Configuration Language Objects.	326
Index		333

Safety Information



Important Information

NOTICE

Read these instructions carefully, and look at the equipment to become familiar with the device before trying to install, operate, 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 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 an imminently hazardous situation which, if not avoided, **will result in** death or serious injury.



WARNING

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

CAUTION

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

<i>NOTICE</i>

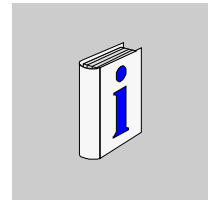
<i>NOTICE</i> is used to address practices not related to physical injury.

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.

About the Book



At a Glance

Document Scope

This manual describes the software installation of the control application on Premium and Atrium PLCs using Unity Pro software.

Validity Note

This documentation is valid for Unity Pro from version 7.0.

Product Related Information

⚠ WARNING
UNINTENDED EQUIPMENT OPERATION The application of this product requires expertise in the design and programming of control systems. Only persons with such expertise should be allowed to program, install, alter, and apply this product. Follow all local and national safety codes and standards. Failure to follow these instructions can result in death, serious injury, or equipment damage.

User Comments

We welcome your comments about this document. You can reach us by e-mail at techcomm@schneider-electric.com.

Introduction to the process control application



In This Part

This part introduces the principles of process control and the associated software and hardware solutions.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
1	Process Control in Automated Systems	15
2	Hardware and software introduction	17
3	Procedure for installing a process control	29

Process Control in Automated Systems

1

Place of the Process Control Application in an Automation Application

General Introduction

A process control application may be programmed using either:

- an Elementary Function Block (EFB),
- a control loop.

Presentation of Function Block Libraries

There are three process control function block libraries:

- CONT_CTL (continuous process library (*see Unity Pro, Control, Block Library*)),
- CLC (*see Unity Pro, Obsolete, Block Library*) and CLC_PRO (*see Unity Pro, Obsolete, Block Library*) (obsolete library),
- CLC_INT (in the general library (*see Unity Pro, Standard, Block Library*)).

The CLC, CLC_PRO and CLC_INT libraries are deemed to be 'obsolete'. When importing an application, they enable you to retrieve process control functions programmed in PL7 from the CLC_INT library and those programmed using Concept from the CLC and CLC_PRO libraries, without errors.

It is therefore preferable to use the CONT_CTL library for a new project incorporating process control. This is the standard for all processor types (Premium, Atrium and Quantum).

Presentation of Control Loops

The processors with process control functionality use an application-specific channel called the Loop channel. On this application-specific channel, loop controllers are available and are represented schematically by algorithms which correspond to the type of process control necessary for the supervision of industrial processes.

These loop controllers can be of the following types:

- process loop,
- cascade loop,
- auto-selector loop,
- single control loop,
- setpoint programmer.

Control loop parameters (*see page 21*) are set during processor configuration via the application screens.

Presentation of Application Screens

The Human Machine Interface (HMI) required for all process control is provided by:

- the Unity Pro debugging and adjustment screens,
- supervisors.

Hardware and software introduction



Subject of this chapter

This chapter introduces the hardware solutions, which allow the implementation of the process control application, built-in to the Premium programmable PLCs.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
2.1	Introduction to the Process Control Processors	18
2.2	Overview of Process Control Software Tools	19

2.1 Introduction to the Process Control Processors

Features of Processors with Built-in Process Control

At a Glance

Processors used for process control display the same technical features as those described in the basic hardware manual. Features specific to process control are the following:

- Number of process control channels,
- Supported process control functions.

Processor Features

Number of "application-specific" channels supported:

- Premium (*see Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual*)
- Atrium (*see Premium and Atrium using Unity Pro, Processors, racks and power supply modules, Implementation manual*)

2.2 Overview of Process Control Software Tools

Subject of this Section

This section introduces the software tools running under Unity Pro and run-time software required to install the process control application.

What Is in This Section?

This section contains the following topics:

Topic	Page
Configuring a Task Type for the Processor	20
How to Access Process Control Application Configuration	21
Description of Loop Controller Configuration Screens	22
Runtime Screen Configuration Tools	26
Auto-tuning Control Loops	27
Setpoint Programmer	28

Configuring a Task Type for the Processor

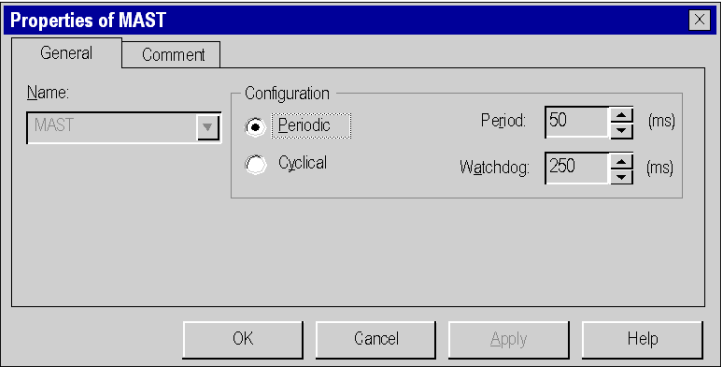
At a Glance

Description of the Project Task Selection Process

NOTE: using process control in the context of a project **requires** a periodic MAST task.

Processor Configuration

The following table explains how to configure periodic MAST tasks.

Step	Derivative
1	From the Project Browser , select the Program directory.
2	From the MAST sub-directory, accessible through the TASK directory, select the Properties command from the context-sensitive menu. Results: The following window appears. 
3	From the Configuration area, check the Periodic box, and provide a time value for the period, expressed in ms.
4	Validate the selection by clicking on the OK button.
5	Validate the changes by clicking on the Confirmation icon on the toolbar.

How to Access Process Control Application Configuration

At a Glance

The software setup for process control channels is identical to that used for Unity Pro application-specific functions.

Application-specific functions have the following characteristics:

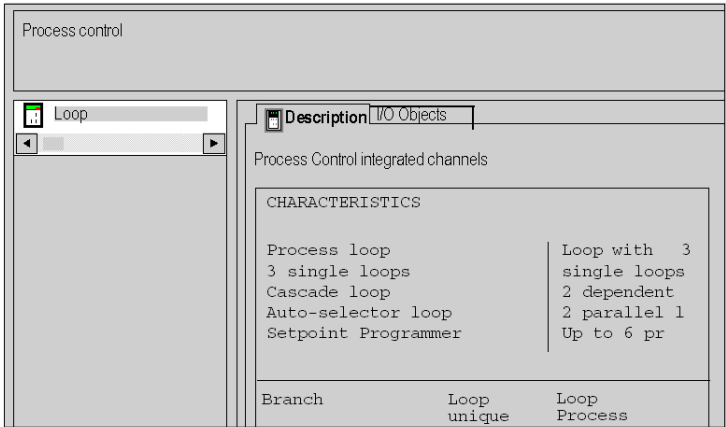
- dedicated screens,
- specific instructions,
- language objects (*see page 227*).

Setting up a process control begins by accessing the configuration screens, consisting of:

- configuration screens (off-line and online modes),
- debugging screens (online mode).

Instructions

Proceed with the following steps:

Step	Action
1	From the project browser, select the Configuration directory.
2	Press + , the tree corresponding to the configuration is expanded.
3	From the X-Bus directory pop-up menu, select Open to display the default configuration.
4	Double click on the Loop area of the processor. Result: the following screen appears: 

Description of Loop Controller Configuration Screens

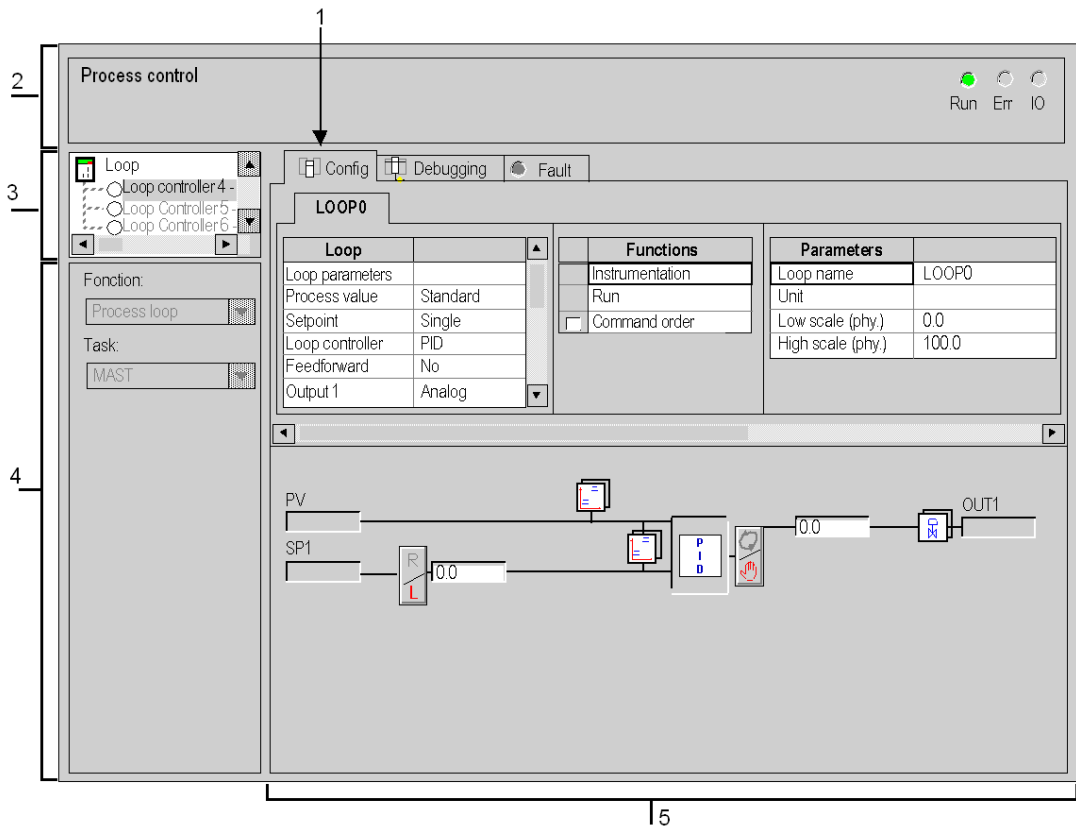
At a Glance

There are two different types of loop control configuration screens:

- configuration screens,
- debugging screens.

Illustration

The following figure shows a configuration screen.



Description

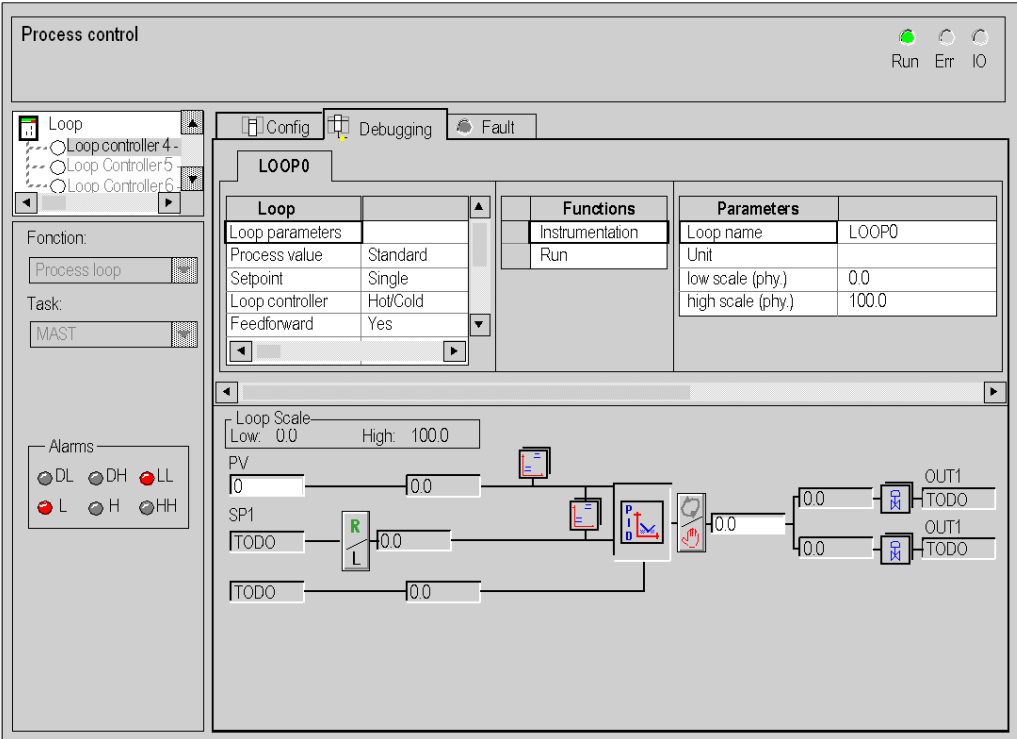
The following table lists all components of the configuration screen, as well as their respective functions.

Address	Component	Function
1	Tabs	The foremost tab indicates the current mode (in this instance: Configuration). Select each mode by clicking on the corresponding tab. The following modes are available: ● Configuration, ● Debugging, accessible only in Online mode, ● Diagnostics, (Default) accessible only in Online mode.
2	Module area	Displays the abridged module indicator. There are three indicators in the same zone that provide the status of the PLC in online: ● RUN indicates the status of the PLC, ● ERR indicates a fault on the processor or embedded devices. ● I/O indicates a fault from a module, channel or configuration.
3	Channel area	Is used: ● By clicking on the reference number, to display the tabs: ● Description which gives the characteristics of the device. ● I/O Objects (<i>see Unity Pro, Operating Modes</i>) which is used to presymbolize the input/output objects. ● Fault which shows the device faults (in online mode). ● To choose the loop controller ● To display the Symbol, the name of the channel defined by the user (using the variable editor).
4	Global Parameters area	Use this area to select the global parameters associated with to the channel: ● Function: Functions associated with the channel and available are the following: ● a process loop, ● a cascade loop, ● an auto selector loop, ● three single loops, ● a setpoint programmer. ● Task: defines the MAST or FAST task through which the channel's implicit exchange objects will be exchanged. This parameter is invariable.

Address	Component	Function
5	Configuration area	Use this area to set configuration parameters applicable to the channel. Some options may be grayed-out to indicate that they are not available. This area comprises two sections: - the function selection area, - the visualization and configuration area for the loop controller block diagram.
Legend		
(1) : This function is available for versions of Unity Pro > V2.1.		

Illustration

The figure below represents a debug screen.



Utilities Accessible through the Debugging Screen

The following utilities are accessible through the debugging screen:

- input value simulation (process value, Feed forward, etc.),
- block diagram animation,
- editing parameters for calculating functions,
- modifying and saving all parameters,
- sending self-tuning and manual commands, etc.

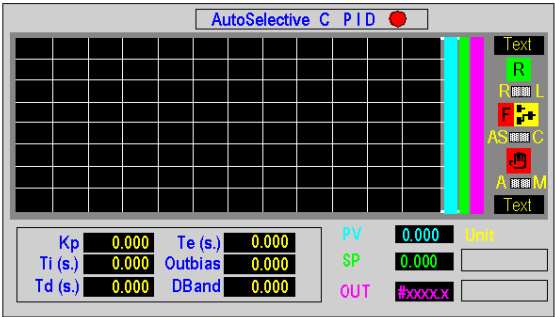
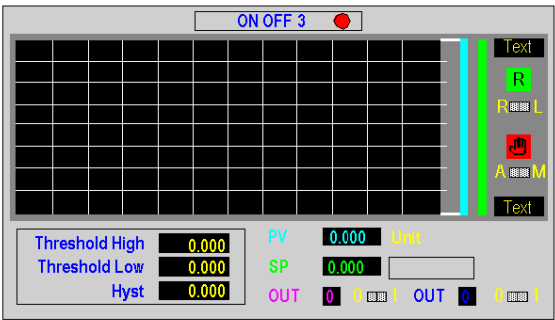
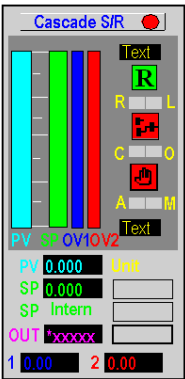
Runtime Screen Configuration Tools

At a Glance

The runtime screens editor allows the use of bar charts and specific library (see *Unity Pro, Operating Modes*) objects for the operation of control loops. The block animation process is automatic and very straightforward.

Sample Configuration Tool

This example shows the process control interfaces accessible through the runtime screens editor.



Auto-tuning Control Loops

At a Glance

Auto-tuning is applicable to most processes, such as temperature regulation, flow regulation, pressure regulation, etc.

Loop controllers built into these control loops are used to calculate a set of configuration parameters (K_p , T_i , T_d), upon receiving a request for auto-tuning.

These parameters are accessible through:

- Unity Pro debugging screens,
- an animation table.

Setpoint Programmer

At a Glance

All process control channels can be configured using the setpoint programmer.

The features of a setpoint programmer are as follows:

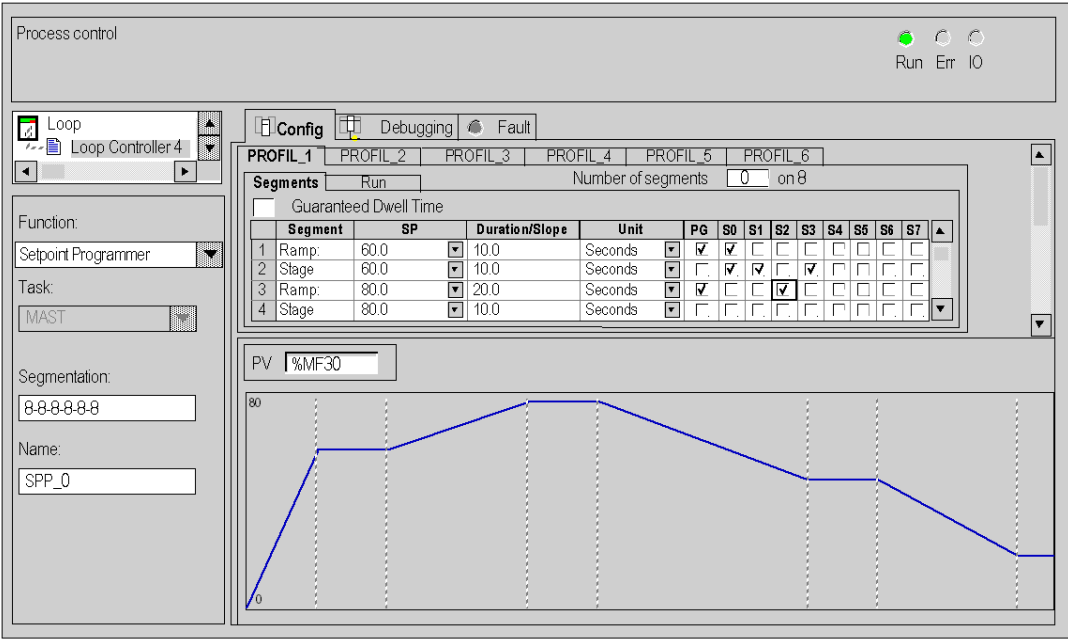
- 1 to 6 profiles,
- a maximum of 48 segments distributed over the configured profiles,
- Profiles may be assigned to several control loops.

NOTE: The maximum duration of a segment is 2 hours.

NOTE: The PV measurement tracking function may be used when the setpoint programmer is interfaced to a single loop.

Setpoint Programmer Configuration Screen

A configuration screen allows the user to assign profiles to a process control loop.



Procedure for installing a process control



How To Set Up a Process Control Using PL7

At a Glance

Setting up a process control involves a certain number of steps as described in the following procedure.

Setup Fundamentals with Processor

The table below lists the various setup steps involving the processor.

Step	Description	Mode
Declaring variables	Declaration of variable of the IODDT type for the project process control loops and variables.	Local (1)
Programming	Project programming.	Local (1)
Configuration	Declaring modules	Local
	Configuring process control channels	
	Entering configuration parameters	
Linkage	Linking IODDT's to the configured modules (Variables Editor).	Local (1)
Generating	Project generation (analyzing and editing links).	Local
Transfer	Transfer of the project to the PLC.	On-line
Adjusting/Debugging	Project debugging from debug screens, animation tables.	On-line
	Editing the program and adjustment parameters	
Documentation	Building the documentation file and printing the different data relating to the project.	On-line (1)
Operation/Diagnostics	Displaying the different data needed for running the project.	On-line
	Project / module diagnostics.	
Legend		
(1)	The steps above may also be implemented in the other operating mode.	

Installing the process control application



In This Part

This part describes the different control loops and functions, as well as their installation, from configuration to debugging.

What Is in This Part?

This part contains the following chapters:

Chapter	Chapter Name	Page
4	Introduction to Loop Controllers	33
5	Calculation Functions	79
6	Control Loop Configuration	171
7	Control Loop Adjustment	175
8	Debugging a Process Control Loop	197
9	Operating Modes	205
10	Process Control Language Objects and IODDT	227

Introduction to Loop Controllers

4

Subject of this Chapter

This chapter introduces the various loop controllers:

- types,
- structure,
- processing branches,
- parameters.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
4.1	Definition and structure of a loop controller	34
4.2	Description of the Types of Loop Controllers	36
4.3	Description of the processing branches	42
4.4	The Setpoint Programmer	62
4.5	Global parameters of control loops	76

4.1 Definition and structure of a loop controller

Structure of Loop Controllers

At a Glance

"Loop controller" is a generic term used to designate a processor control channel. Any loop controller may consist of several control loops.

Example: a master loop and a slave loop.

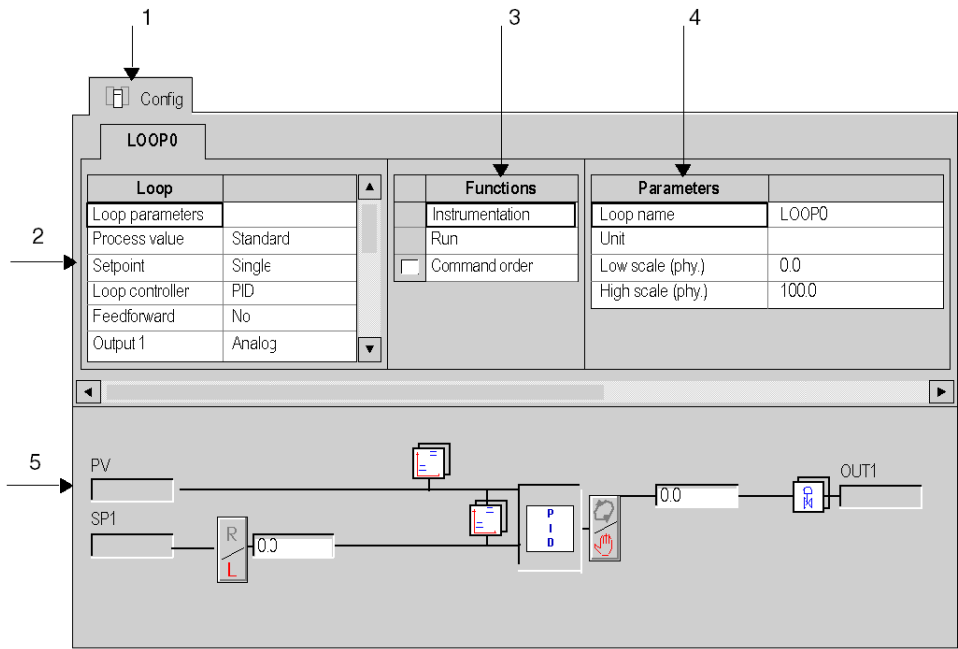
A loop controller therefore consists of several control loops, themselves including:

- branches or blocks (process value processing, etc.) consisting of:
 - calculating functions (gain, filtering, square root, etc.) which are defined by any number of parameters.

The Unity Pro Configuration screen allows the user to define the appropriate hierarchical structure.

Illustration

This figure represents a configuration screen, which allows this hierarchical breakdown, as well as the configuration of process control channels.



Description

The following table describes the hierarchical structure of a process control channel.

Address	Description
1	Loops Tab
2	Branches List
3	Functions List
4	Parameters List
5	Process Control Loop Block Diagram

4.2 Description of the Types of Loop Controllers

Subject of this Section

This section introduces the different loop controllers as well as their structure.

What Is in This Section?

This section contains the following topics:

Topic	Page
Types of Process Control	37
Introduction to the Process Loop	38
Introducing the Single Loop	39
Introduction to the Cascade Loop	40
Introduction to the Auto Selector Loop	41

Types of Process Control

At a Glance

For a process control channel, it is possible to select from 5 predefined profiles:

- a process loop,
- three single loops,
- a cascade loop,
- an auto selector loop,
- a setpoint programmer (*see page 62*).

Each loop offers default parameters, with the exception of the setpoint programmer. The use of the different functions built into the algorithms (square root, function generator, etc.) is predefined, as is the initial value of each parameter.

Description of the Control Loops

The control loops are made up 5 processing branches which will create the required algorithm:

- the process value processing branch,
- the Feed forward processing branch,
- the setpoint processing branch,
- the loop controller branch,
- the output processing branch.

The operation of each processing branch (*see page 42*) is the same, irrespective of the process control type which is selected.

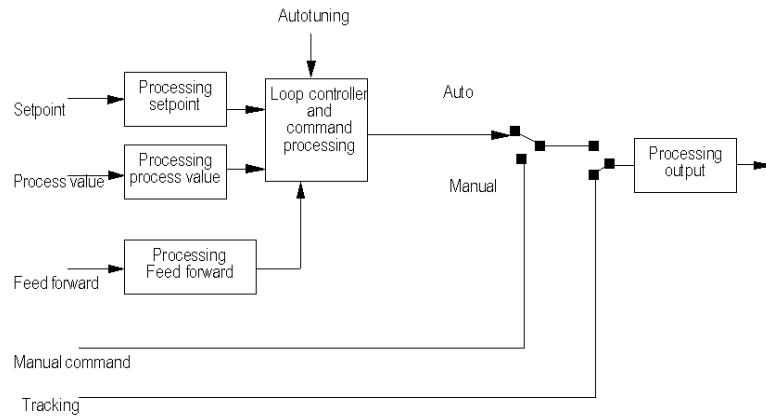
Introduction to the Process Loop

At a Glance

The process loop is a loop with only one loop controller.

Loop Diagram

This schema illustrates the process paths of the process loop.



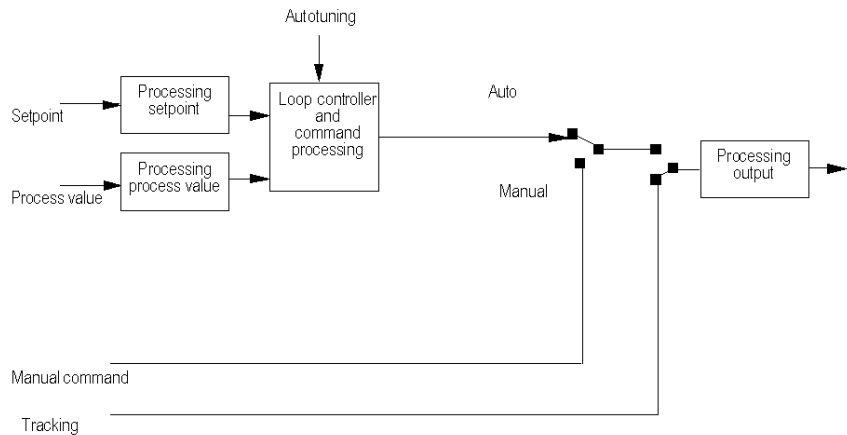
Introducing the Single Loop

At a Glance

The **single loop** profile automatically links three single loops to the loop controller, in order to increase the number of loops. These loops operate independently of each other.

Loop Diagram

This diagram illustrates the process branches for the single loop.



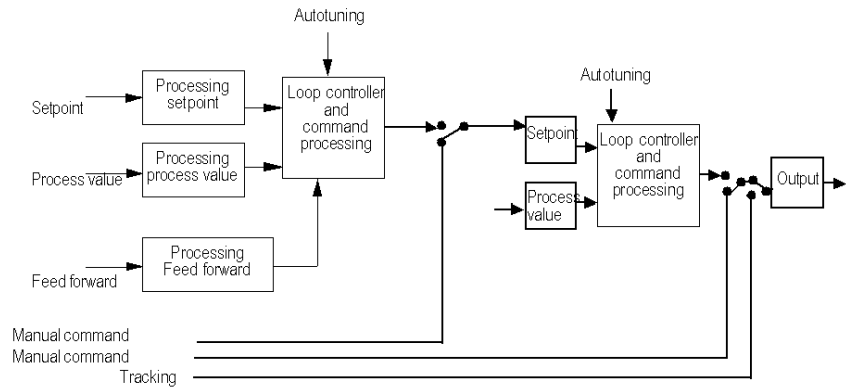
Introduction to the Cascade Loop

At a Glance

The cascade loop is made up of two dependent loops: a master loop and a slave loop. The master loop output is the setpoint for the slave loop.

Loop Diagram

This diagram illustrates the process paths of the cascade loop.



Introduction to the Auto Selector Loop

At a Glance

The **Auto Selector** loop, which is also called the **restricted** loop is made up of two parallel loops:

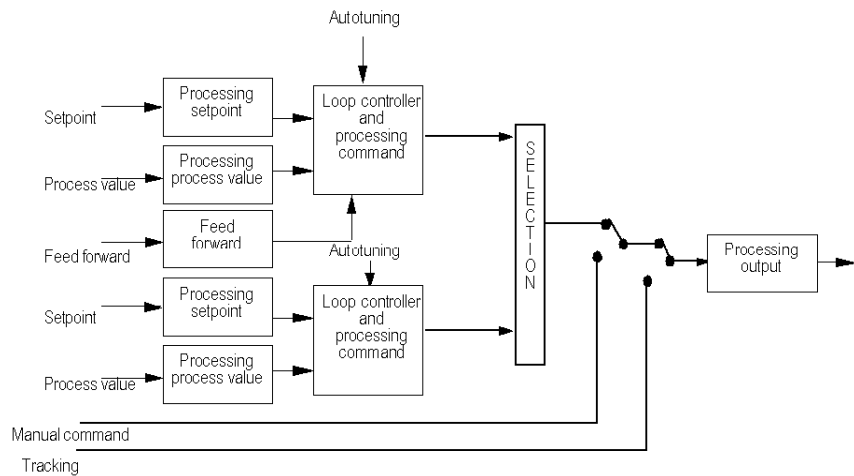
- the main loop,
- the secondary loop.

The output is selected according to a precise algorithm.

The secondary loop is a single loop.

Loop Diagram

This diagram illustrates the process paths of the Auto Selector loop.



4.3 Description of the processing branches

Subject of this Section

This section introduces the different processing branches that make up the loop controller loops.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction to Built-in Functions	43
Process Value Processing Branch	47
Setpoint Processing Branch	49
Feed Forward Processing Branch	51
Loop Controller and Command Branch	52
The ON OFF 2 or 3 States Loop Controller	53
The PID or IMC Loop Controller	54
The Split Range or Hot/Cold (PID or IMC) Loop Controller	55
Output Processing Branch	57
Servo Drive Output Branch	58
PWM Branch Output	60
Loops Summary Table	61

Introduction to Built-in Functions

At a Glance

Each processing branch has its own built-in calculation functions. Each calculation function is represented by an icon in the block diagram. Each type of branch has specific calculation functions.

Process Value Branch

List of calculation functions.

Icon	Description
	First order filtering
	Square Root
	Function Generator
	Scale Limiter
	Alarms on level
	Totaling
	Scaling

Setpoint Branch

List of calculation functions.

Icon	Description
	Selection
	
	
	Ratio
	Setpoint Limiter
	Process Value Setpoint Tracker
	Speed Limiter
	Scaling

Loop Controller

List of calculation functions.

Icon	Description
	2-state ON/OFF
	3-state ON/OFF
	PID
	Internal model loop controller
 	Hot/Cold
 	Split Range

Feed Forward Branch

List of calculation functions.

Icon	Description
	Scaling
	Lead Lag

Output Loop Controller

List of calculation functions.

Icon	Description
	Scaling
	Analog Output
	Servo Drive Output
	Pulse Output
	Output Limiter

Process Value Processing Branch

At a Glance

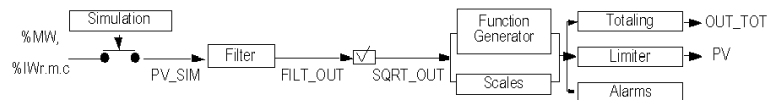
There are two types of process values:

- Standard process value,
- External process value.

The external process value allows you to input to the loop controller a process value **PV** which has been processed outside the control loop. This solution is offered for cases where calculation of the process value requires specific or customized functions which are not available in standard process value processing.

Standard Process Value

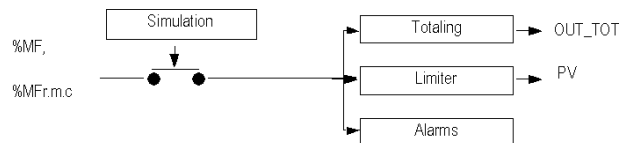
Block diagram of the standard process value branch.



- The first order filtering function has a gain coefficient.
- There are four thresholds for the **Alarms on process value** block with a hysteresis of 1% of the full scale.
- Two input formats may be used: unipolar or bipolar.
- There is no bump when switching to simulated mode; the initial simulation value retained is the last process value read.
- The function generator has scaling built-in.
- It is possible to limit the process value to the scale limits.

External Process Value

Block diagram of the external process value branch.



Initialization Overview

- On start-up, the associated data is first updated before the launch of the first process of this branch.
- If the process value input address is not set, processing is performed on the simulated value initially set to zero.
- On initialization, there is a consistency check on the configuration entered. If there is a problem with the configuration, the loop remains in initialization state.

Runtime Control Operation

The two serious faults checked by process value processing are parameter and internal calculation errors (division by zero, overflow, etc.).

If	Then
a serious error has been detected	the loop processing switches to fallback state : • the value of the calculated process value PV is frozen, • the control loop outputs are frozen.
the error disappears	the loop resumes in the previous operating mode, without bumps on the outputs.
there is a serious error on process value processing during a cold start	the loop remains in its initialization position and does not start.
scale values are incorrect (non-integer value) during a cold start	the loop remains in initialization position yet does not start.
due to an error, the loop remains in its initialization position and does not start.	the problem must be eliminated for the loop to restart automatically.
there is an runtime error in the scale values.	processing is performed with the old valid scale values which were overwritten in the current scale parameters. Scale parameters are updated when the check proves to be correct.

Setpoint Processing Branch

At a Glance

There are four types of setpoint:

- the ratio setpoint,
- the selection setpoint,
- the single setpoint (remote with scaling),
- the setpoint programmer.

Only the two last setpoints can be used with the 3 single loops or the Auto Selector loop.

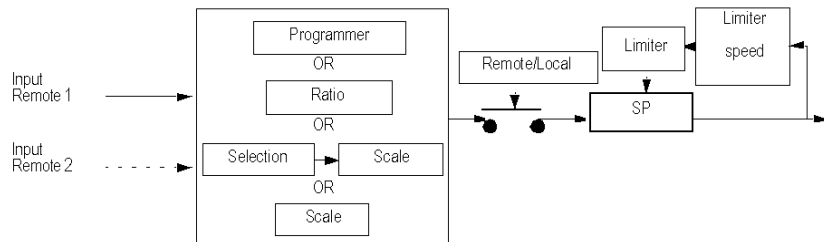
Definition

A **local setpoint** is a setpoint written by a Man-Machine Interface.

A **remote setpoint** is a setpoint produced by a process.

Block Diagram of the Branch

Block diagram of the setpoint processing branch.



- The local value tracks the value of the **remote setpoint** in order to avoid bumps on the change of operation.
- If the address of the **remote** setpoint is not given, local mode is **forced**.
- To avoid changes which are too abrupt, the speed of the setpoint can be limited.
- By default, the setpoint is limited to the loop scale. A more restrictive limit can be fixed.
- When the loop controller is in manual mode, the setpoint can track the process value.

How the Initialization Works

- On start-up, the associated data is first updated before the launch of the first process of this branch.
- If the process value input address is not set, processing is performed on the simulated value initially set to zero.
- On initialization, there is a consistency check on the configuration entered. If there is a problem with the configuration, the loop remains in a state of initialization.

Operation of Execution Check

The two faults checked by setpoint processing are parameter errors (not written in floating point format) and internal calculation errors (division by zero, overflow, etc.). When such a fault appears:

- The result of the SP setpoint processing is frozen.
- Warnings are displayed.
- These errors are not considered to be serious at the level of the control loop; loop controller and output value calculation is performed with the value of the frozen setpoint.
- Calculation of the **SP** setpoint restarts as soon as the fault disappears.
- Other faults associated with the built-in setpoint functions are indicated. They are detailed in the description of each function (*see page 79*).

Feed Forward Processing Branch

At a Glance

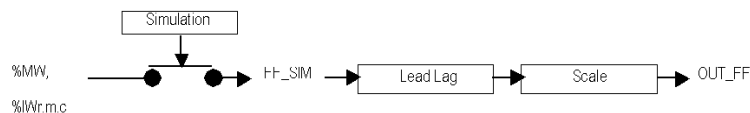
In a classic process control using **PID**, the loop controller reacts to the process output variations (closed loop process control). Therefore, in the occurrence of a disruption, the loop controller starts reacting only when the process value moves away from the setpoint.

Use the Feed Forward function to compensate for a measurable disruption as soon as it appears.

This open loop function anticipates the disruption's effects: this procedure is referred to as anticipating action (or Feed Forward).

Block Diagram of the Branch

Block Diagram for Feed Forward Processing Branch



How the Initialization Works

If the Feed Forward branch address is undefined, processing is performed on the simulated value initially set to zero.

Operation of Execution Check

Setpoint processing checks for two types of errors: parameter errors (not written in floating point format) and internal calculation errors (division by zero, overflow, etc.). When such a fault appears:

- The result of processing this setpoint is frozen.
- These errors are not considered to be serious at the control loop level; loop controller and output value calculation is performed with the frozen **Feed Forward** value.
- Specific warnings are displayed.
- The OUT_FF value, on the loop controller input, is updated as soon as the error is resolved.
- Other faults associated with the built-in setpoint functions are indicated. They are detailed in the description of each function (*see page 79*).

Loop Controller and Command Branch

At a Glance

There are 8 types of loop controller:

- Autotuning PID loop controller,
- Discrete mode loop controller:
 - 2 states,
 - 3 states,
- Heat/Cool PID loop controller,
- Split Range PID loop controller,
- Internal model loop controller,
- Heat/Cool internal model loop controller,
- Split Range internal model loop controller,

The ON OFF 2 or 3 States Loop Controller

At a Glance

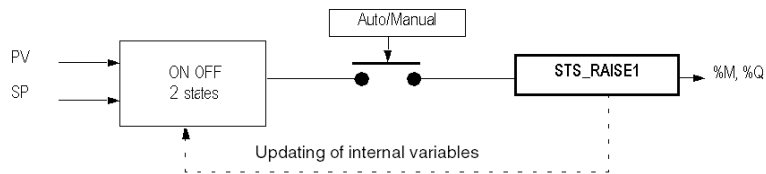
This type of branch contains only the ON OFF function with 2 or 3 states according to the selection. It is available in the **process loops** and **3 single loops**. When this type of branch is selected, there is neither an output branch, nor a Feed forward branch.

The loop controller output is copied into the status bit STS_RAISE1 for the ON OFF 2 states loop controller and also into the status bit STS_LOWER1 for the ON OFF 3 states loop controller.

The internal variables are updated by recognizing the previous value of the command.

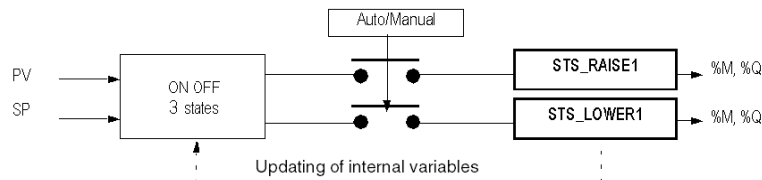
Block Diagram of the 2 States Loop Controller

Block diagram of the branch with an ON OFF 2 states loop controller.



Block Diagram of the 3 States Loop Controller

Block diagram of the branch with an ON OFF 3 states loop controller.



The PID or IMC Loop Controller

At a Glance

The diagram below illustrates the standard **PID** branch. There are variants, depending on the loops. Each variant is dealt with in the description of the different loops (*see page 218*).

For certain functions, the internal variables are updated by recognizing the previous value of the command. This enables switching to be made without bumps and enables integral action overflow to be avoided by recognizing the output limitations.

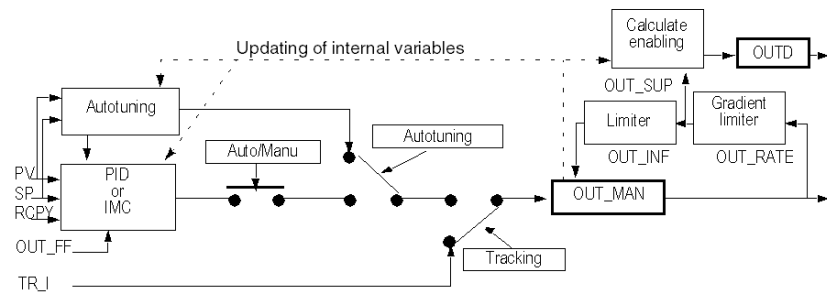
The output limitations apply in all loop controller operating modes.

NOTE: The RCPY input (external input address) only exists on the model loop controller.

NOTE: In manual mode, if the gradient limit is used, the OUT_MAN value (target value before limitation) may temporarily be different from the command applied on output.

Block Diagram of the PID or IMC Loop Controller

Block diagram of the PID or IMC loop controller.



The Split Range or Hot/Cold (PID or IMC) Loop Controller

At a Glance

The IMC function is identical to the IMC single loop controller.

The PID function is identical to the PID single loop controller.

The only differences are:

- the absence of autotuning,
- the limiting of OUT_MAN between 0 and 100,
- the value of OUT_BIAS is fixed at 0 and cannot be parameterized.

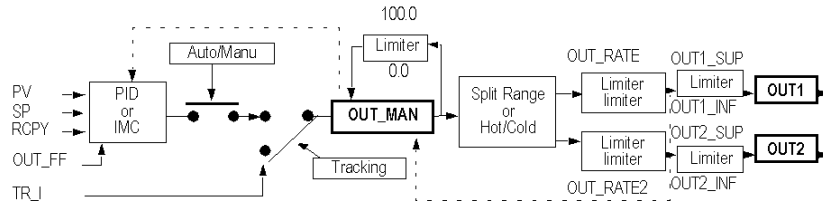
The OUT_MAN command is recalibrated according to the limitations which are applied (as with a single PID), in order to avoid problems with overflow of the integral action and with the operation of the Split Range or Heat/Cool function.

Each output of the Split Range or Heat/Cool function has its own limitations with regard to level and gradient.

The operating mode acts on the output of the OUT_MAN loop controller.

Block Diagram of the Split Range or Heat/Cool Loop Controller

Block diagram of the Split Range or Heat/Cool loop controller (PID or IMC).



Initialization Overview

- There is a consistency check on the configuration selected. If there is a problem with the configuration:
 - the loop remains in initialization state,
 - the error is indicated in the status words.
- On cold start-up, the parameters and input values PV, SP, etc. associated with this branch are first updated before the launch of the first loop controller process.
- It is possible to select the initial loop controller operating modes, by creating parameters in the configuration screen or by sending commands in a sequential program. The loop can thus start in manual or automatic mode.

Runtime Control Operation

Setpoint processing checks for two types of errors: parameter errors (not written in floating point format) and internal calculation errors (division by zero, overflow, etc.). Should such an error occur:

- Loop controller processing goes into a fallback position:
 - the calculated value of the OUT command is frozen,
 - the control loop outputs are maintained.
- The state returns to normal as soon as the fault disappears. The loop then starts off again on the outputs without bumps.
- The input and output scale values are checked.
- In case of inconsistency on a cold start, the loop switches to serious error mode.
- Warnings are generated in the status words.

Output Processing Branch

At a Glance

There are three types of output:

- analog output,
- Servo drive output,
- PWM output (pulse modulation).

Regardless of the type of output selected, the OUT_MAN command calculated by the loop controller goes through a limiter whose lower OUTi_INF and upper OUTi_SUP limits define the range of output variation. These limits define the output scale.

Analog Output

With the exception of the aforementioned limiter, no function is specific to this process. The calculated floating point value is converted into an integer which is then transmitted to an analog channel (%QW) or in a memory word (%MW).

There are two different integer formats:

- unipolar [0, 10000], default format,
- bipolar [-10000, 10000].

Servo Drive Output Branch

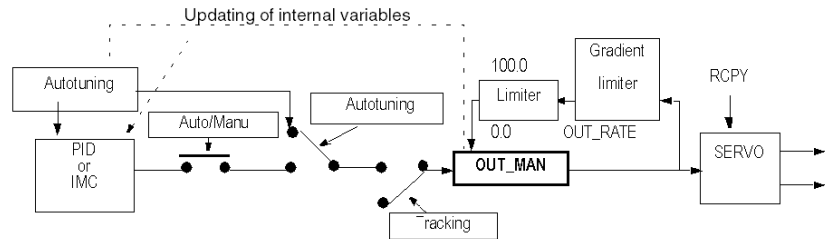
At a Glance

This output is made up of a SERVO function with or without a copy of the actuator's position.

- After a Split Range or Heat/Cool PID, only the SERVO with copy position is available.
- With this type of output, the loop controller output scale must be OUT_i_INF and OUT_i_SUP in the interval $[0, 100]$.
- Its output period is the task period.
- If the loop controller is in automatic mode, the SERVO output takes into account the new loop controller output value in each sampling period. In the other modes, it is in each cycle of the task.

Block Diagram of a SERVO Output with Position Copy

Block diagram of a Servo drive output branch with position copy (RCPY).



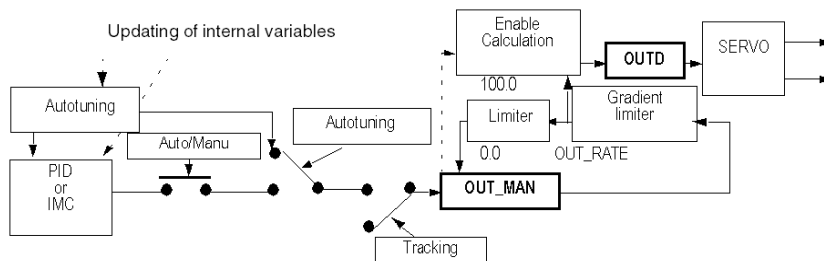
Information

The SERVO function inputs are the output OUT_MAN of the loop controller and the position copy of the actuator $RCPY$.

When the SERVO output follows a Split Range or Heat/Cool function, the copy input is indispensable. SERVO function inputs are then $OUT1$ or $OUT2$ and $RCPY$.

Block Diagram of a SERVO Output without Position Copy

Block diagram of a Servo drive output branch without position copy (RCPY).



Information

The Servo function entry is a variation of the OUTD command of the PID.

NOTE: OUTD is not affected by limiting output OUT_MAN. This can be used to perform a floating process control - the command calculated by the PID has no direct link with the actual position of the actuator. In particular, it is possible to continue opening or closing a motorized valve even when OUT_MAN is saturated.

PWM Branch Output

At a Glance

This output consists of a PWM function in which the input is the command:

- OUT_MAN in the case of a PID loop controller,
- OUT1 or OUT2 in the case of Heat/Cool or Split Range loop controllers.

With this type of output branch, the output scale must be OUTi_INF and OUTi_SUP in the interval [0 , 100].

Its output period is the task period and is independent of the operating mode of the loop controller.

Initialization Overview

- On startup the parameters and output branch input value are updated before the first processing.
- If the output address is not filled in , the processing is carried out but the output conversion is not performed.
- There is a consistency check on the configuration entered. If there is a problem with the configuration, the loop remains in initialization state.

Runtime Control Operation

Setpoint processing checks for two types of errors: parameter errors (not written in floating point format) and internal calculation errors (division by zero, overflow, etc.). Should such an error occur:

- The result of the output is frozen.
- When an error disappears, the status reverts to normal and the output is recalculated without bumps.
- During a cold start, if the scale values are invalid (non-integer value), the loop remains in the initialization position and does not restart. Therefore, the outputs retain their initial value.
- The loop restarts when the error is eliminated.

Loops Summary Table

Table

This table summarizes the elements associated with each type of loop.

Branch	Single loop	Process loop	Cascade loop		Auto selector loop	
			Master	Slave	Main	Secondary
Process standard process value	No	Yes	Yes (1)(2)	No	Yes	No
Single standard process value	Yes	Yes	Yes	Yes	Yes	Yes
Feed Forward	No	Yes	Yes	No	Yes	No
Single setpoint	Yes	Yes	Yes	-	Yes	Yes
SPP profile setpoint	Yes	Yes	Yes	-	Yes	Yes
Selection setpoint	No	Yes	Yes	-	Yes	No
Setpoint with ratio	No	Yes	Yes	-	Yes	No
ON OFF 2 states loop controller	Yes	Yes	No	No	-	-
ON OFF 3 states loop controller	Yes	Yes	No	No	-	-
PID loop controller	Yes	Yes	Yes	Yes	Yes	Yes
Heat / Cool PID loop controller (2)(3)	No	Yes	-	Yes	No	-
Split Range PID loop controller	No	Yes	-	Yes	No	-
IMC loop controller	Yes	Yes	Yes (1)(3)	Yes (1)	Yes (1)	Yes (1)
Heat / Cool IMC loop controller (2)(3)	No	Yes	Yes (1)	Yes (1)	Yes (1)	Yes (1)
Split Range IMC loop controller	No	Yes	Yes (1)	Yes (1)	Yes (1)	Yes (1)
Analog output	Yes	Yes	-	Yes	Yes	-
SERVO output	Yes	Yes	-	Yes	Yes	-
PWM output	Yes	Yes	-	Yes	Yes	-

(1) One loop controller, either master or slave.

(2) The totalizing function does not exist on the process value branch of the master loop.

(3) The autotuning function is not built into the PID/IMC Heat/Cool loop controllers.

4.4 The Setpoint Programmer

Subject of this Section

 This section describes the operation of the setpoint programmer.

What Is in This Section?

 This section contains the following topics:

Topic	Page
Description of Setpoint Programmer	63
Linking a Setpoint Programmer to a Control Loop	65
Guaranteed Dwell Time of Setpoint Programmer	66
Control Outputs	68
Bumpless Start	70
Executing a Profile	72
Parameters for Setpoint Programmer	74
Operation Initialization and Monitoring	75

Description of Setpoint Programmer

At a Glance

The setpoint programmer generates a setpoint. The setpoint evolves over time according to a profile consisting of various parameters:

- a maximum of 6 profiles may be configured, comprising a total of
- 48 segments (i), numbered from 1 to 48.

Segments are defined by the following parameters:

- SP_i (%MF), target setpoint,
- VAL_i (%MF), duration or slope of segment,

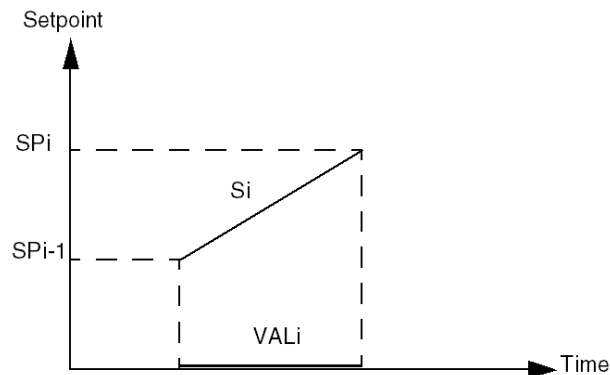
Any segment may be configured as:

- a ramp,
- a step (in this instance: $SP_i = SP_{i-1}$).

Any segment may be configured in seconds, minutes or hours.

Figure

This diagram illustrates the introductory remarks.



Mapping Profiles/Segments

The following configurations are available:

- 1 profile with a maximum of 48 segments,
- 1 profile with a maximum of 32 segments and 1 profile with a maximum of 16 segments,
- 2 profiles with max. 24 segments each,
- 3 profiles with max. 16 segments each,
- 4 profiles with max. 12 segments each,
- 6 profiles with max. 8 segments each,
- 1 profile with 24 segments, 1 profile with 16 segments and 1 profile with 8 segments.

Memory Mapping for Parameters and Setpoints

The 48 target setpoints and their respective durations are stored, respectively, in the `%MFxy.i.50` to `%MFxy.i.240` words. The number of configured profiles does not affect this mapping.

For 3 profiles of 16 segments each, mapping is as follows:

- 1st setpoint of the 1st profile is found in `SP1 (%MFr.m.c.50)`,
- 1st setpoint of 2nd profile is found in `SP17 (%MFr.m.c.114)`,
- 1st setpoint of 3rd profile is found in `SP33 (%MFr.m.c.178)`.

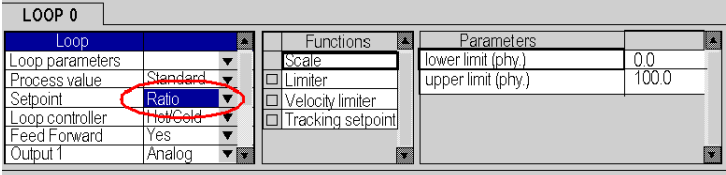
For 2 profiles of 24 segments each, mapping is as follows:

- 1st setpoint of 1st profile is found in `SP1 (%MFr.m.c.50)`,
- 1st setpoint of 2nd profile is found in `SP25 (%MFr.m.c.146)`.

Linking a Setpoint Programmer to a Control Loop

Procedure

The following table lists the steps required to associate a setpoint programmer to a control loop.

Step	Derivative
1	Use the Project Browser to access the Hardware Configuration.
2	Double-click on the Loop area of the processor.
3	Select a Loop Controller corresponding to a process control loop.
4	Select Choose the required function. Result: the system displays the following (or a similar) screen: 
5	Select Programmer as the setpoint type.
6	Enter as the setpoint input address (SP1 in block diagram) the output address of the setpoint programmer, that is SP (%MFr.m.c.20), where c indicates the number of the loop controller selected as setpoint programmer.

Guaranteed Dwell Time of Setpoint Programmer

At a Glance

A process' reaction to a setpoint change may be more or less rapid. The reaction does not necessarily follow the setpoint variation as calculated by the programmer. However, it is still possible to follow the evolution of a process value and to guarantee a dwell time at the selected setpoint: the dwell time countdown starts when the deviation between the setpoint and the process value is smaller than a previously defined threshold (THLD).

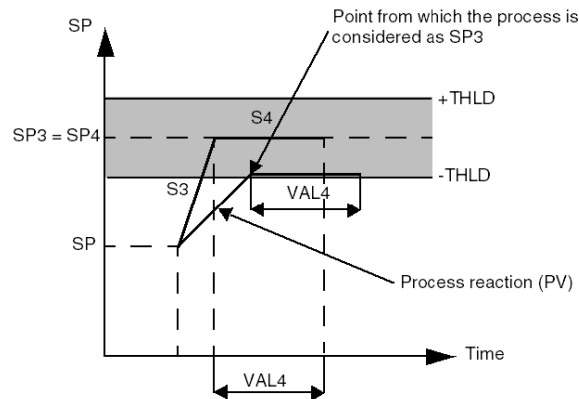
There are several manners to obtain the guaranteed dwell time:

- high deviation overflow,
- low deviation overflow,
- high and low setpoint deviation overflow.

In such instances, the dwell period countdown is "frozen" (interrupted) at each overflow.

Figure

The following figure illustrates the guaranteed dwell time.



Configuring a Guaranteed Dwell Time

Proceed as follows to configure a guaranteed dwell time.

Step	Action
1	Select a loop controller.
2	Select the Setpoint Programmer function.
3	Enable the function by checking the Guaranteed Dwell Time box.
4	Fill in the threshold value THLD and the type of overflow required: ● high deviation, $PV > dwell + THLD$ ● low deviation, $PV < dwell - THLD$ ● high deviation, $PV > dwell + THLD$ or $PV < dwell - THLD$ ● input deviation, deviation only at start of dwell.
5	In the GDT column, check the dwell time to which the function will be affected.
6	Enter the address of the process value you wish to monitor in the PV text area. Note: Should you omit to provide the address of the process value, the Guaranteed Dwell Time function will not be used, even if you have configured it. During execution, it is possible to inhibit this function by issuing a command (<i>see page 72</i>).

Control Outputs

At a Glance

The setpoint programmer has 8 `STOR0` to `STOR7` logic bit outputs (`%MWxy.i.3:X0` to `X7`), which can be associated to the segments in order to generate discrete actions.

These outputs are called `S0` to `S7` on the Unity Pro Configuration Screen.

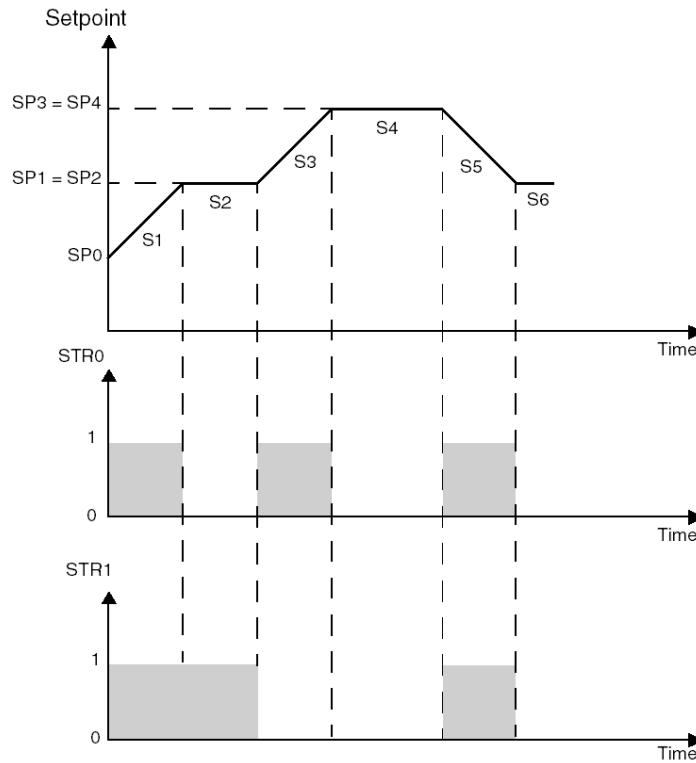
Sample Configuration Screen

This illustration shows a sample configuration for logic outputs associated with segments from a setpoint programmer.

	Segment x	SP x	VAL x	Unit	Pq	S0	S1	S2	S3	S4	S5	S6	S7
1	Ramp ▼	50.0	40.0	Second ▼		✓	✓						
2	Stage ▼	50.0	20.0	Second ▼			✓	✓					
3	Ramp ▼	80.0	40.0	Second ▼		✓			✓			✓	
4	Stage ▼	80.0	40.0	Second ▼				✓		✓			
5	Ramp ▼	40.0	35.0	Second ▼		✓	✓				✓		
6	Stage ▼	40.0	40.0	Second ▼				✓					
7	Ramp ▼	10.0	20.0	Second ▼		✓							✓

Runtime Timing Diagram

This timing diagram describes the position of the outputs in relation to the current segment.



Bumpless Start

At a Glance

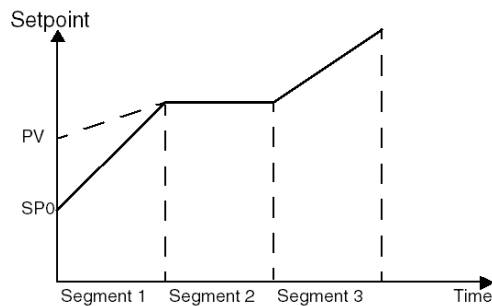
A setpoint profile starts on an initial setpoint value SP0, defined in the **Execution** tab. To avoid a bump on starting, the profile may start from measured value PV and reach setpoint SP1 in agreement with the 1st segment's characteristics.

Before initializing a profile, the calculated setpoint is, by default, set to SP0 (or PV depending on the profile configuration). Should the selected profile be modified, the calculated setpoint may be updated by performing the **Reset** command on this profile.

This functionality is also used to carry out profile loop iterations. To select these functions, simply check the appropriate boxes in the **Execution** tab window.

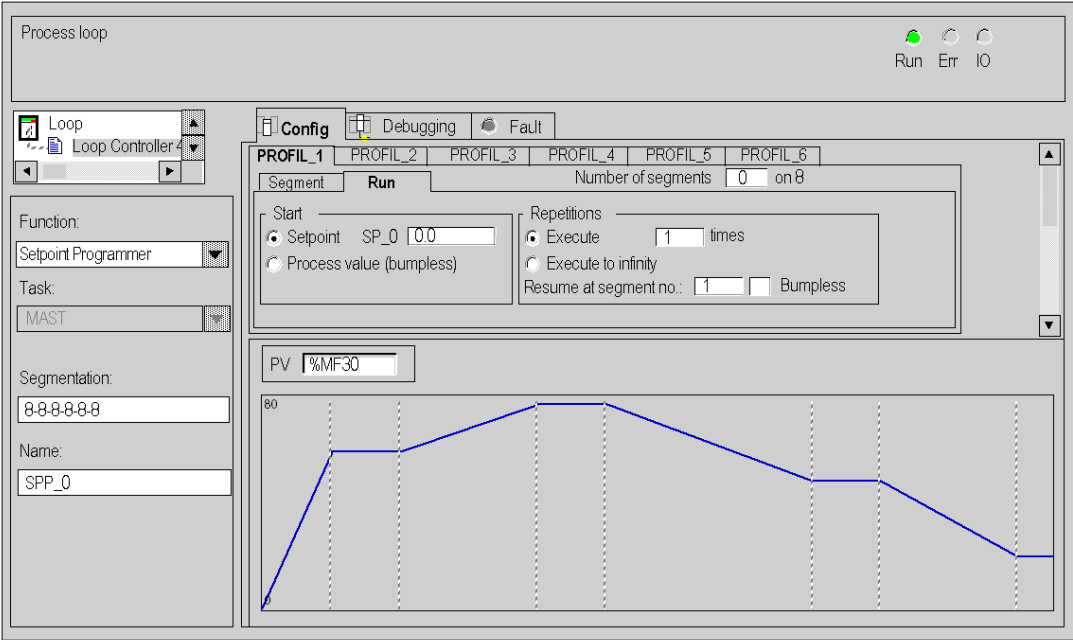
Runtime Timing Diagram

This timing diagram describes the operation of a bumpless start.



Execution Configuration Window

This figure shows a sample configuration screen used to configure the bumpless initialization and loop iteration function.



Executing a Profile

At a Glance

The following operations are applicable to any profile:

- one-time execution,
- reiteration of some number of instances,
- continuous looping.

The number of iterations is specified in the NB_RT_PFi word, and may vary from 1 to 32,767.

NOTE: the first loop iteration segment is not necessarily the first segment in the profile. The segment in question is configured in the **\$\$\$Execution** tab of the Configuration screen. The advantage of this method lies in the capacity to define, for any given profile, segments which will execute only once upon startup (initialization segments).

How to Launch a Profile Execution Command

The following table lists the steps that trigger the execution of a profile using a software command.

Step	Action
1	Assign the command value to the CMD_ORDER (%MW _r .m.c. 7) word. Note: The following table lists all available commands.
2	Include the number of the desired profile in the CMD_PARAM (%MD _r .m.c. 8 (1 to 6)) double word.
3	Execute the WRITE_CMD (LOOP_i) command. with LOOP_i designating an IODDT-type variable corresponding to the i channel to which the command is applied. Sample LOPP_i of the T_PROC_SPP type. Note: This command is transmitted through a method called explicit exchange (see page 230).

Profile Execution Commands

It's possible to launch and apply only one profile at any given moment. The following table lists the commands associated with the hexadecimal value assigned to the CMD_ORDER (%MW_{r.m.c.7}) word.

Command	Hexadecimal value	Comment
START	16#0002	Triggers the execution of the selected profile.
STOP	16#0003	Stops the execution of the selected profile.
RESET	16#0001	Reinitializes the setpoint programmer, and readies it to wait for another START.
NEXT	16#0006	Jumps to the following segment.
BACK	16#0007	Jumps to the previous segment.
HOLD	16#0004	Freezes the setpoint evolution and stops time.
DEHOLD	16#0005	Releases the profile in progress.
HOLD_PG	16#0008	Inhibits the guaranteed dwell time function for the current profile.
DEHOLD_PG	16#0009	Activates the guaranteed dwell time function for the current profile.

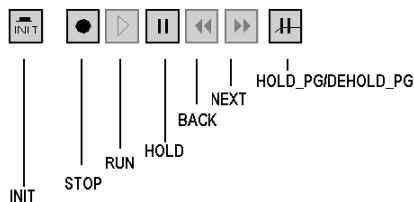
Prerequisite Conditions

The execution of commands is subject to certain conditions:

- the RESET command is always accepted,
- the START command is accepted only if the programmer is running,
- the NEXT and BACK commands are rejected if the profile isn't frozen,
- the STOP command is rejected if the programmer is in launch mode,
- the HOLD_PG and DEHOLD_PG commands are rejected when the function is unused.

Monitoring Through the Master Screen

Each profile may be controlled from its master screen, using the following controls.



Parameters for Setpoint Programmer

Internal Parameters

Table of parameters built-in the function.

Meaning	Symbol	Data Type	Variation Range	Default Value	R/W
Number of reiterations of profile # i	NB_RT_PFi	INT	0/32767	1	R/W
Value of guaranteed dwell time threshold for profile # i	THLD_PFi	REAL	0.0/3.4 E38	0.0	R/W
Value of initial setpoint for profile # i	SPO_PFi	REAL	-3.4 E38/3.4 E38	0.0	R/W
Target setpoint for segment # i	SPi	REAL	-3.4 E38/3.4 E38	0.0	R/W
Time or speed value for segment # i	VALi	REAL	-3.4 E38/3.4 E38	0.0	R/W

Output Parameters

Table of output parameters.

Meaning	Symbol	Data Type	Variation Range	Default Value	R/W
Current profile number	CUR_PF	INT	0/32767	0	R
Current segment number	SEG_OUT	INT	0/32767	0	R
Number of current iteration	CUR_ITER	INT	0/32767	0	R
Value of calculated setpoint	SP	REAL	-3.4 E38/3.4 E38	-	R
Value of total time elapsed (including freeze time)	TOTAL_TIME	REAL	0.0/3.4 E38	-	R
Value of time elapsed on current segment (including freeze time)	CUR_TIME	REAL	0.0/3.4 E38	0.0/3.4 E38	R

Operation Initialization and Monitoring

Initialization Overview

Upon startup, the system performs a consistency check of the predefined configuration. In case of problem with the configuration, the setpoint programmer indicates the error and remains in initialization state.

Runtime Control Operation

Setpoint processing checks for two types of critical errors: parameter errors (not written in floating point format) and internal calculation errors (division by zero, overflow, etc.).

If	Then
an error in the setpoint calculation has been detected	the setpoint result is frozen
the error disappears	normal status is resumed.
a ramp includes two identical setpoints	a warning is displayed, while the system proceeds with the setpoint calculation. The system immediately switches to the next segment if the programmer is on the faulty segment.
a ramp has a rising or falling speed of zero (0.0)	a warning is displayed, while the system proceeds with the setpoint calculation. The calculated setpoint is frozen if the programmer is on the faulty segment.
a dwell has two distinct setpoints	a warning is displayed, while the system proceeds with the setpoint calculation. There is a rise or a fall depending on the setpoint values if the programmer is on the faulty segment.
the guaranteed dwell time is configured with a threshold value of 0.0	a warning is displayed, while the system proceeds with the setpoint calculation. Processing of the guaranteed dwell time is disabled if the programmer is on the faulty segment.

If the setpoint programmer is on the faulty segment, an additional warning displays the following message: "Error on current segment."

NOTE: For these warnings to be displayed, the execution time for each segment must necessarily be greater than the PLC task processing time.

4.5 Global parameters of control loops

Description of Global Parameters for Process Control Loops

At a Glance

Some parameters associated with control channels may be separated in three categories:

- the Task parameter linked to the loop,
- instrumentation parameters which characterize the process control loop,
- parameters linked to loop execution.

The Task parameter can be modified in the general parameters area of the configuration screen (*see page 22*), whereas the execution and instrumentation parameters can be modified in the configuration area.

PLC Task

This table describes the PLC task parameter:

Parameter	Description
Task	The process control channels must be assigned to a PLC task, either MAST or FAST. The MAST task is selected by default. Select the FAST task if you wish to improve the sampling speed and execution priority.

Instrumentation Parameters

This table describes the instrumentation parameters for the process control loop:

Parameter	Description
Name	This name is included in the constants (%KW); it may be assigned to each loop. The name may consist of a maximum of eight characters.
Unit	This parameter, with a maximum of 6 characters, is included in the constants (%KW). These characters define the process control loop unit (Example: DEGREE).
Application ID	This parameter identifies the process control loop configuration. To check their authenticity, these application IDs may be compared to each other. They numerically represent the loop's configuration constants. Any modification to parameters for the KP unit significantly affects the application ID. One should note, however, the following specificities of cascade and auto-selector loops: ● cascade loop: Any modification to parameters for the KP unit on one of the loops (master or slave) is instantly reflected on this parameter for both loops. However, the application ID is modified only for the slave loop. ● auto-selector loop: Any modification to parameters for the KP unit on one of the loops (main or auto-selector) is instantly reflected on this parameter for both loops. However, the application ID is modified only for the secondary loop.
Upper Scale Lower Scale	These thresholds define the physical scale within which the loop controls the process. The upstream branch calculations (process value and setpoint) are both set to the same scale. Note: the setpoint branch includes a scaling function which returns the variable scale range, as previously defined during setpoint input. This function is useful when process loops are chained together to form a cascade. By default, this scale must be identical to the physical scale defined by the loop. Note: it is also possible to set the output branch (<i>see page 166</i>) to a specific scale.
"Command Order" double word	Use the Command Order double word (<i>see page 236</i>) to manage loop operation. It replaces the WRITE_CMD function. For the first 16 bits, this double word is identical to the process control loop status word.

Execution Parameters

This table describes the control loop execution parameters.

Parameter	Description	
Sampling period	This is the loop controller's processing period, in automatic mode. The default value is 0.3 seconds. This value must be a modulo of the task period. Otherwise, the process control cyclical processing will default to the nearest modulo. Example: T_MAST = 0.1 s, T_ECH = 0.124 s T_ECH true = 0.1 s	
	If	Then
	the sampling period is less than the task period	the true sampling period is automatically fixed to that of the task period.
	Period calculations are made taking into account the true value of T_ECH. If the task processing execution time exceeds the theoretical period, the error is reflected in the <code>OVERRUN</code> bit %S19.	
Trigger method	The Process Control Loop can be triggered automatically	

Calculation Functions

5

Subject of this Chapter

This chapter describes the calculation functions of the different processing branches:

- Functions of the process value branch,
- Functions of the setpoint branch,
- Functions of the Feed forward branch,
- Functions of the loop controller branch,
- Functions of the output branch.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
5.1	Functions of the process value branch	80
5.2	Functions of the setpoint branch	96
5.3	Feed forward branch functions	108
5.4	Functions of the Loop Controller Branch	115
5.5	Functions of the output branch	157

5.1 Functions of the process value branch

Subject of this Section

This section describes the calculation functions of the process value processing branch:

- Input format,
- Filter,
- Square root,
- Function generator,
- Scaling,
- Scale limiter,
- Alarms on level,
- Totalizer.

What Is in This Section?

This section contains the following topics:

Topic	Page
Input Format	81
Filtering of First Order	83
Square Root	85
Function Generator	86
Scaling	89
Scale Limiter	90
Level Alarms	91
Totaling	93

Input Format

Description

The "input format" function is used solely with standard process values. It provides the gross value of the loop's analog input. In order to do this, you must configure the format in compliance with the type of analog input channel.

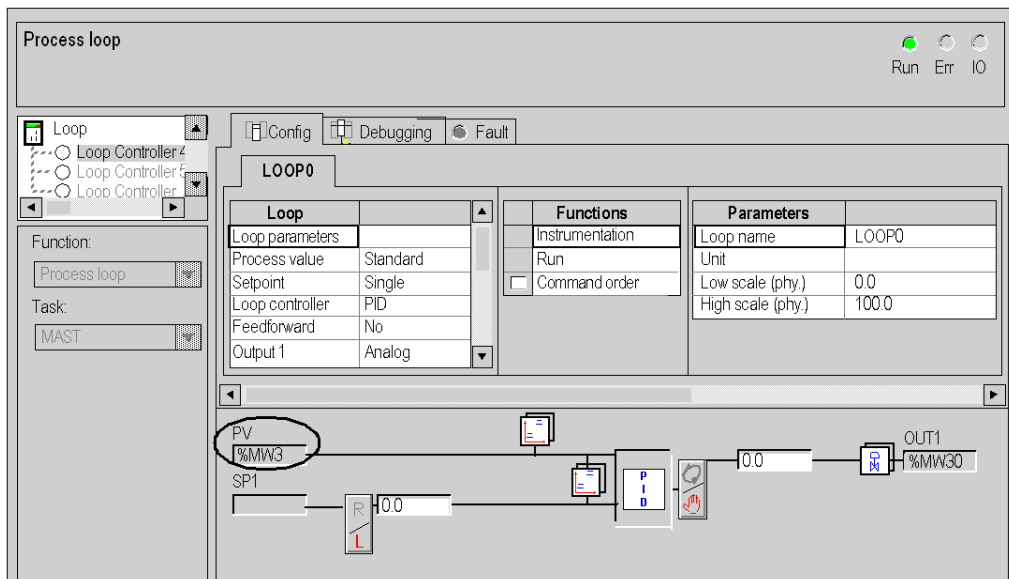
Two range formats are available:

- Unipolar: 0 to 10000 (default selection),
- Bipolar: -10000 to +10000.

Process Value Input Address

You must define the process value input address on the Unity Pro process control configuration screen. In the case of a standard process value, the address is either an %IW input word or an internal %MW word.

The input address is entered in the graphic section of the screen:



Function Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process Value Input	/	%IW %MW	-32768 / 32767	/	R

Internal Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Range	PV_UNI_BIP	%KW bit	/	0 (Unipolar)	R

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Input enabled	PV_SIM	INT	-32768 / 32767	0	R/W

Operation Monitoring

Should you omit to enter the input address, the software automatically reads in the simulation value. This value is initially set at 0.

When Using an External Process Value

In the case of an external process value, the "input format" function is not required, since the value is already an integer. This input is then transferred into the process value floating variable (PV) upon entering the loop controller.

Filtering of First Order

Description

This function filters the first order with a time constant T. The filter transfer function is as follows:

$$\text{FILT\_OUT} = \text{GAIN\_FILT} \times \frac{1}{1 + pT\_FILT} \times \text{PV\_SIM}$$

where:

PV_SIM: Function input value

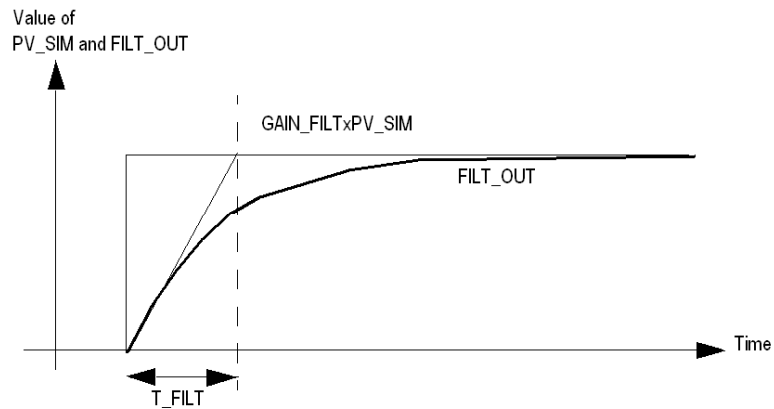
T_FILT: Time constant

FILT_OUT: Result of the function

p: Laplace operator

GAIN_FILT: Proportional Coefficient

This first order filter is directly applied to the input process value.



Function Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Input Enabled	PV_SIM	INT	-32768 / 32767	0	R/W

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Filtering time (ms)	T_FILT	REAL	0.0 / 3.4E38	0.0	R/W
Gain	GAIN_FILT	REAL	-3.4E38 / 3.4E38	1.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Output value	FILT_OUT	REAL	-3.4 E38 / 3.4E38	/	R

NOTE: If the function is not selected, the filter output value is the same as its input value.

Execution Monitoring

Monitoring the parameters of this function is integrated into the error management of the process value branch. If the time constant is negative, its value is set at 0.0.

Square Root

Description

This function calculates the square root of a numerical size. The square root extract is typically used to linearize an output process value produced by a deprimogenous unit

The function performs the following calculation:

$$\begin{aligned} \text{OUT} &= 100 \sqrt{\text{FILT\_OUT}} && \text{if FILT\_OUT} \geq 0 \\ \text{OUT} &= 0 && \text{if FILT\_OUT} < 0 \end{aligned}$$

Function Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default value	R/W
Function input	FILT_OUT	REAL	-3.4E38 / 3.4E38	/	R

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default value	R/W
Output value	SQRT_OUT	REAL	-3.4 E38 / 3.4E38	/	R

NOTE: If the function is not selected, the value of its output is the same as its input value.

Operation Monitoring

There is no specific check dedicated to this function. The monitoring of the parameters is integrated into the error management of the process value branch.

Function Generator

Description

The generator corrects the non-linear readings of the process value input signal. Correcting this non-linearity is done via 7 linear adjacent segments, in variable increments, defined by the coordinates of their points.

The function also performs a scaling which is exclusive to the process value scaling function, described below.

The output is calculated by the linear interpolation between the 2 points whose abscissae contain the input parameter value.

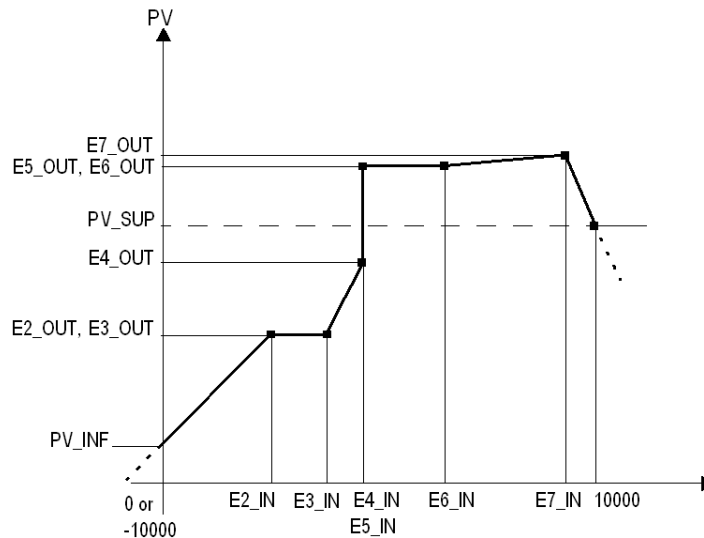
$$PV = f(x) = \{(X1, Y1), \dots, (X7, Y7)\}$$

where:

$X1 = 0$ or -10000 and $Y1 = PV\_INF$ (lower limit of loop scale),

$X7 = 10000$ et $Y7 = PV\_SUP$ (upper limit of loop scale).

Function generation diagram:



Outside the input scale, it is possible using configuration to extrapolate or limit the value of the measurement value calculated on the process value scale.

Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Function input	SQRT_OUT	REAL	-3.4E38 / 3.4E38	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Extrapolation	EXTRAPOL	Bit constant	/	0 (no)	R
Abscissa 1	/	REAL	-3.4E38 / 3.4E38	0.0 or -1000.0	/
Abscissa 2	E2_IN	REAL	-3.4E38 / 3.4E38	1428.0	R/W
Abscissa 3	E3_IN	REAL	-3.4E38 / 3.4E38	2857.0	R/W
Abscissa 4	E4_IN	REAL	-3.4E38 / 3.4E38	4285.0	R/W
Abscissa 5	E5_IN	REAL	-3.4E38 / 3.4E38	5714.0	R/W
Abscissa 6	E6_IN	REAL	-3.4E38 / 3.4E38	7143.0	R/W
Abscissa 7	E7_IN	REAL	-3.4E38 / 3.4E38	8571.0	R/W
Abscissa 8	/	REAL	-3.4E38 / 3.4E38	10000.0	/
Coordinate 1	PV_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Coordinate 2	E2_OUT	REAL	-3.4E38 / 3.4E38	14.28	R/W
Coordinate 3	E3_OUT	REAL	-3.4E38 / 3.4E38	28.57	R/W
Coordinate 4	E4_OUT	REAL	-3.4E38 / 3.4E38	42.85	R/W
Coordinate 5	E5_OUT	REAL	-3.4E38 / 3.4E38	57.14	R/W
Coordinate 6	E6_OUT	REAL	-3.4E38 / 3.4E38	71.43	R/W
Coordinate 7	E7_OUT	REAL	-3.4E38 / 3.4E38	85.71	R/W
Coordinate 8	PV_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process value	PV	REAL	-3.4E38 / 3.4E38	/	R

NOTE: Parameters PV_INF and PV_SUP are defined within the global parameters of the loop.

Execution Monitoring

The monitoring of the parameters is integrated into the error management of the process value branch.

The coordinates of the abscissae must be ascending. For example, if $E_{j+1\_IN} < E_{j\_IN}$, a warning message is displayed. Nonetheless the calculation is done with the current parameters.

Scaling

At a Glance

The process value branch is automatically scaled using the loop's global parameters: PV_INF and PV_SUP (*see page 76*).

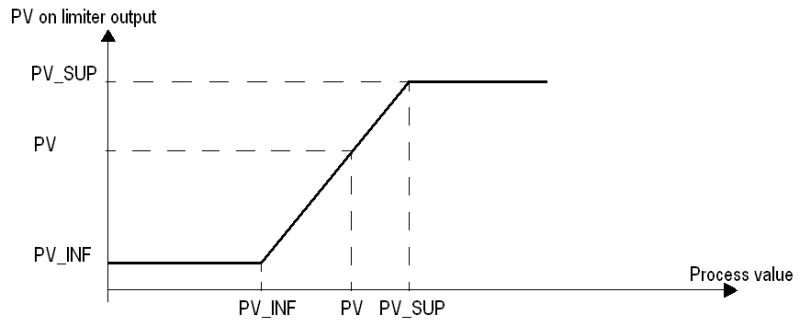
Scale Limiter

Description

This function is used to limit the process value in relation to the physical scale defined for the process control loop.

If this function is activated, the scaling is only done in the range PV_INF to PV_SUP. Outside of this range, the output is limited to the scale values.

Value on limiter output:



Level Alarms

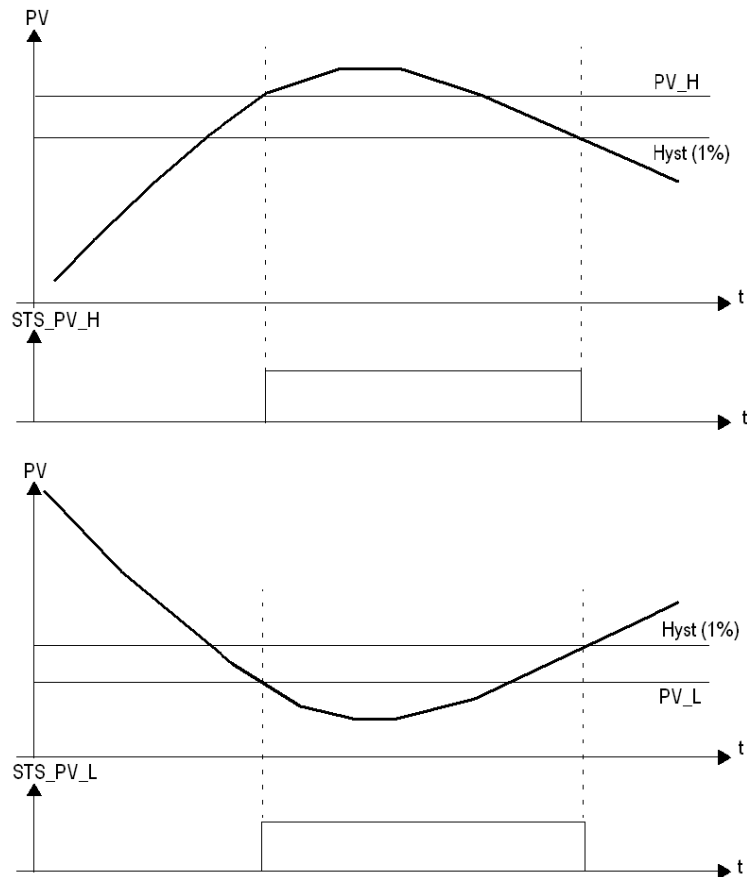
Description

This function monitors the progress of the process value by comparing its value to the 4 following thresholds: PV_LL, PV_L, PV_H and PV_HH.

Each alarm is associated to a status bit.

These alarms are controlled by a fixed hysteresis of 1% in relation to the scale defined in the loop's global parameters.

Alarms for crossing threshold PV_H or PV_L:



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process value	PV	REAL	-3.4 E38 / 3.4E38	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Very low threshold	PV_LL	REAL	-3.4 E38 / 3.4E38	5.0	R/W
Low threshold	PV_L	REAL	-3.4 E38 / 3.4E38	5.0	R/W
High threshold	PV_H	REAL	-3.4 E38 / 3.4E38	95.0	R/W
Very high threshold	PV_HH	REAL	-3.4 E38 / 3.4E38	95.0	R/W

Output parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Very low limit	STS_PV_LL_LIM	EBOOL	/	/	R
Low limit	STS_PV_L_LIM	EBOOL	/	/	R
High limit	STS_PV_H_LIM	EBOOL	/	/	R
Very high limit	STS_PV_HH_LIM	EBOOL	/	/	R
Or alarms (*)	STS_ALARMS	EBOOL	/	/	R

(*) Or bit logic of alarms upon crossing threshold and alarms on deviation.

Execution Monitoring

The monitoring of the parameters is integrated into the error management of the process value branch.

Totaling

Description

This function includes the input value (typically a negative) according to time, and returns a cumulative (typically a volume).

For this, it uses a partial internal accumulator (Acc) that includes the PV value and is automatically reset to 0 each time it reaches an adjustable threshold THLD.

The number of resets is recorded in order to enable the recomposing of the global cumulative OUT_TOT.

Function Principle

On each execution the accumulator Acc and the cumulative OUT_TOT are calculated using the algorithm:

```
Acc(new) = Acc(old) + PV . DT
SI Acc(new) >= THLD ALORS
Acc(new) = Acc(new) - THLD
CptInit = CptInit + 1
FINSI
OUT_TOT = CptInit x THLD + Acc(new)
```

where:

CptInit = number of resets

DT = task period

ACC (old) = value of accumulator Acc on the previous cycle

Adjusting the Integration Threshold THLD

Generally, integration threshold value corresponds to an easily determined process characteristic (the capacity of a tank, for example). During a cycle, a status bit is set to 1 each time the partial accumulator reaches the integration threshold.

The function can also be used to integrate small input value, even when the result of the integration is very large. In this case, the values to be integrated may become negligible to the accumulated value and are therefore no longer taken into consideration.

To avoid this, we advise you to limit the accumulator to a threshold THLD, so that the value to be integrated is never negligible in relation to this partial accumulator. When the threshold THLD is equal to 0, the function does not integrate a value and the function remains fixed.

Time Base

In order to integrate the new process value, the relationship between the cumulative value and the PV value must be less than 10^9 .

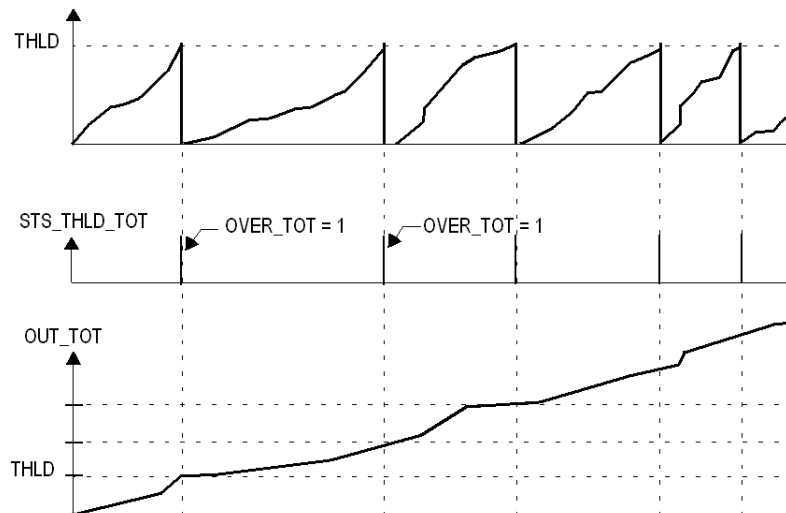
Associated Commands

Two commands are associated to the function:

- Reset: the OUT_TOT function output is set to 0, as are all the internal variables (following a phase change in production, for example).
- Hold: integration is paused. The function output retains its former value. In this mode the user can modify the cumulative value OUT_TOT, prompting a new calculation of the internal variables. This allows the cumulative value to be recalibrated (after an automation stop, for example).

Timing Graph

Totalizer Function:



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process value	PV	REAL	-3.4E38 / 3.4E38	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Totaling Threshold	THLD	REAL	0.0 / 3.4E38	1.0E38	R/W
Time Base (h)	/	%KW bit	/	/	R

Output parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Totaling Value	OUT_TOT	REAL	0.0 / 3.4E38	0.0	R
Threshold reached	STS_THLD_TOT	EBOOL	/	/	R

Execution Monitoring

The monitoring of the parameters is integrated into the error management of the process value branch.

5.2 Functions of the setpoint branch

Subject of this Section

This section describes the calculation functions of the setpoint processing branch:

- Ratio,
- Selection,
- Scaling,
- Setpoint limiter,
- Tracking setpoint,
- Speed limiter

What Is in This Section?

This section contains the following topics:

Topic	Page
Ratio	97
Selection	99
Scaling	100
Setpoint Limiter	102
Tracking Setpoint	104
Speed Limiter	106

Ratio

Description

The Ratio function is used to carry out the report process control; in other words, to assign a size to the value of the external input (control value).

The Ratio function calculates the loop controller setpoint according to the control value, by applying the following formula:

$$SP = \text{RATIO} \times (\text{SP Remote 1}) + \text{RATIO\_BIAS}$$

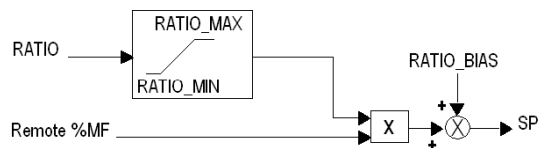
where:

SP Remote 1 = control value.

It is possible to define maximum and minimum limitations on the ratio reports.

NOTE: The size connected to the SP Remote 1 input is an external process value rather than a setpoint.

Functional diagram of the Ratio function:



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint input	/	%MFi	-3.4E38 / 3.4E38	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Ratio value	RATIO	REAL	-3.4E38 / 3.4E38	1.0	R/W
Min ratio value	RATIO_MIN	REAL	-3.4E38 / 3.4E38	0.0	R/W
Max ratio value	RATIO_MAX	REAL	-3.4E38 / 3.4E38	100.0	R/W
Ratio bias	RATIO_BIAS	REAL	-3.4E38 / 3.4E38	0.0	R/W

Output parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint value	SP	REAL	-3.4E38 / 3.4E38	/	R
Scaling Error	RATIO_WARN	EBOOL	/	/	R

Execution Monitoring

Monitoring the parameters is integrated into the error management of the setpoint branch.

Selection

Description

The selection function is used to select a setpoint by comparing two numerical inputs. This selection can be:

- Max selection: the Remote 1 setpoint input is greater than the Remote 2 setpoint input,
- Min selection: the Remote 1 setpoint input is smaller than the Remote 2 setpoint input,
- "Switch" selection: the input is selected by an explicit command.
The switch is straightforward and is done without hysteresis.

Operation Monitoring

Monitoring this function is integrated into the error management of the setpoint branch.

Scaling

Description

This function is used to express a setpoint value in the process value scale defined by PV_INF and PV_SUP (loop parameters).

It takes into account the input range of the setpoint (INP_INF_{Ri}, INP_SUP_{Ri}) and is applied to the Remote 1 and Remote 2 setpoints.

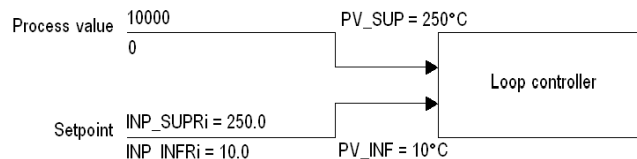
The Scaling function performs the following calculation:

$$SP = (IN - INP\_INF_{Ri}) \times \frac{PV\_SUP - PV\_INF}{INP\_SUP_{Ri} - INP\_INF_{Ri}} + PV\_INF$$

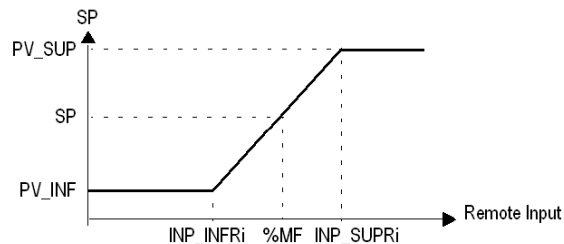
This function is optional and can be used to track two loops (to cascade two process loops, for example).

In the absence of this function: INP_INF_{Ri} = PV_INF and INP_SUP_{Ri} = PV_SUP.

Functional diagram of the Scaling function:



Changing the setpoint according to Remote setpoint input



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint input	/	%MFi	-3.4E38 / 3.4E38	/	R/W

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Low input scale	INP_INFRI	REAL	-3.4E38 / 3.4E38	0.0	R/W
High input scale	INP_SUPRI	REAL	-3.4E38 / 3.4E38	100.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint value	SP	REAL	-3.4E38 / 3.4E38	/	R

Operation Monitoring

Monitoring the parameters of this function is integrated into the error management of the setpoint branch.

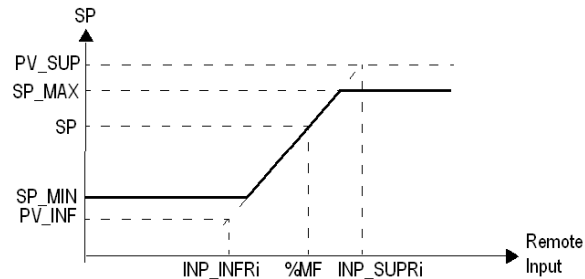
Setpoint Limiter

Description

When this function is activated, the scaling is only done in the range limits, defined by the parameters SP_MIN and SP_MAX. When this function is not activated, the setpoint value is limited to the physical scales of the regulation loop.

The interval (SP_MIN / SP_MAX) must be included in the interval (PV_INF / PV_SUP).

The interval [SP_MIN / SP_MAX] must be in the interval (PV_INF / PV_SUP).



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint Value	SP	REAL	-3.4E38 / 3.4E38	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Low setpoint limit	SP_MIN	REAL	-3.4E38 / 3.4E38	0.0	R/W
High setpoint limit	SP_MAX	REAL	-3.4E38 / 3.4E38	100.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint Value	SP	REAL	-3.4E38 / 3.4E38	/	R

Execution Monitoring

Monitoring the parameters of this function is integrated into the error management of the setpoint branch.

If $SP\_MIN \geq SP\_MAX$, $SP\_MIN < PV\_INF$ or $SP\_MAX > PV\_SUP$, the setpoint output remains unchanged and a bit is set to state 1 in the status words.

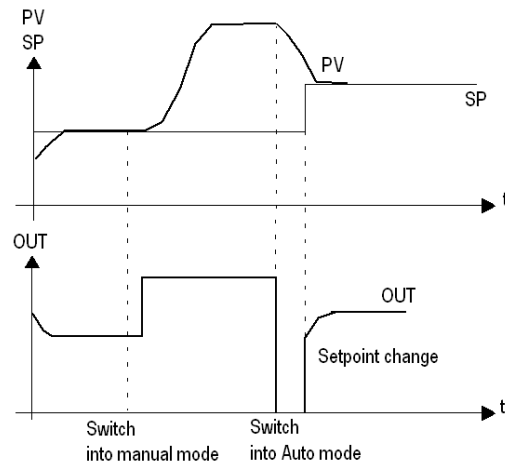
Tracking Setpoint

Description

In local setpoint mode and when the corrector is not in automatic mode, this function prompts the tracking of the measurement by the local setpoint. This avoids bumps on the corrector output when it returns to automatic mode.

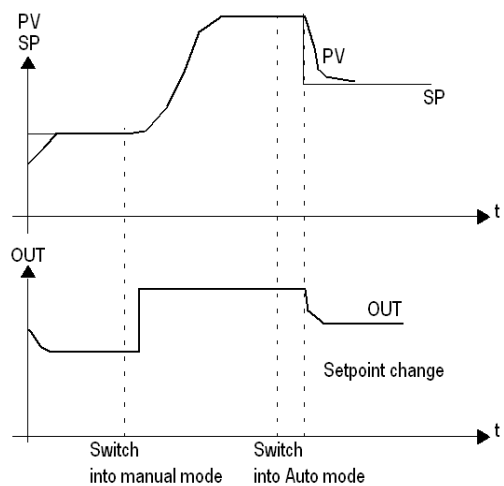
Function Not Configured

Change of the output when the function is not configured:



Function Configured

Change of the output when the function is configured:



Speed Limiter

Description

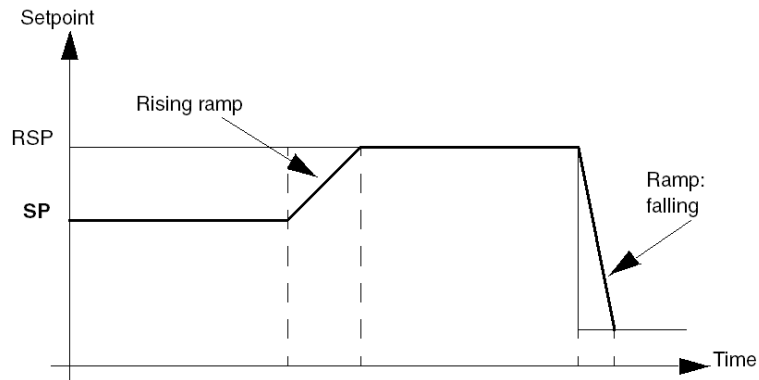
This function is used, when changing setpoint, to reach the new value while respecting a speed limit. Rising and descending speed limits can be different.

When the requested value on input is greater than the current value of the output SP, the function increases the value of this output at the speed R_RATE, until the value SP is equal to that requested.

If the value R_RATE is nil, there is no slope and SP is a direct copy of the input value.

When the input value changes as the slope is being generated the function attempts to reach this new target.

Function diagram:



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Setpoint Value	SP	REAL	-3.4E38 / 3.4E38	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Rising speed limit	R_RATE	REAL	0.0 / 3.4E38	0.0	R/W
Descending speed limit	D_RATE	REAL	0.0 / 3.4E38	0.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Limited Setpoint Value	SPEED_LIM_OUT	REAL	-3.4E38 / 3.4E38	/	R/W

NOTE: R_RATE and D_RATE at 0.0 mean there is no limit.

This function can be applied to the Remote setpoint and the local setpoint or only to the local setpoint, according to the chosen configuration.

Execution Monitoring

Monitoring the parameters of this function is integrated into the error management of the setpoint branch.

5.3 Feed forward branch functions

Subject of this Section

This section describes the calculation functions of the Feed forward processing branch:

- Scaling,
- Leadlag,
- Alarm on deviation.

What Is in This Section?

This section contains the following topics:

Topic	Page
Scaling	109
Lead Lag	111
Alarm on Deviation	113

Scaling

Description

This function is used to change the scale of the digital value of the Feed Forward input.

The Scaling function performs the following transfer function:

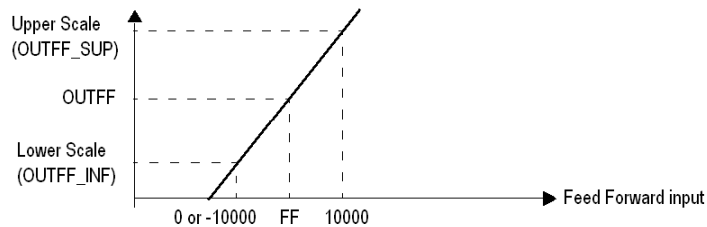
$$\text{OUTFF} = (\text{FF} - \text{in\_min}) \times \frac{(\text{OUTFF\_SUP} - \text{OUTFF\_INF})}{(\text{in\_max} - \text{in\_min})} + \text{OUTFF\_INF}$$

where:

in_min = 0 or -10000

in_max = 10000

Feed Forward value:



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Feed Forward input	/	%IW %MW	-32768 / 32767	/	R

Internal parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower Scale	OUT_FF_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Upper Scale	OUT_FF_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Feed Forward value	OUT_FF	REAL	-3.4E38 / 3.4E38	/	R/W

Execution Monitoring

Monitoring the parameters of this function is integrated into the error management of the Feed Forward branch.

There is no monitoring on the scale parameter order. The lower limit can have a greater value than that of the upper limit.

Lead Lag

Description

The Lead Lag (or Leadlag) function performs a step advance/delay-type transfer function. This allows you to model the influence of disruptions and, in turn, to perform an anticipated process control in an open loop.

The Lead Lag function performs the following transfer function:

$$\text{OUTFF} = \frac{1 + p \times T1\_FF}{1 + p \times T2\_FF} \times \text{FF}$$

where:

FF = size of analog input: internal variable (disruption process value)

T1_FF = time constant corresponding to forward motion of phase

T2_FF = time constant corresponding to phase delay

p = Laplace operator

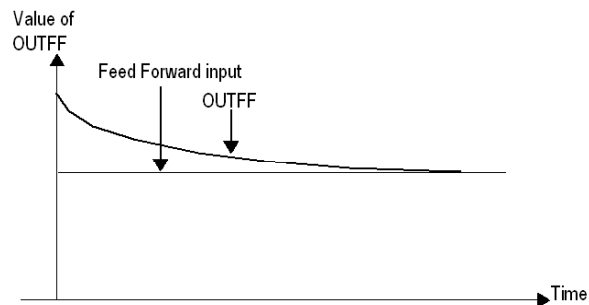
OUTFF = calculated size

For a graded input, the output response is a function of T1_FF and T2_FF (step advance or delay):

- If T1_FF > T2_FF, a step advance is performed.
- If T1_FF < T2_FF, a step delay is performed.

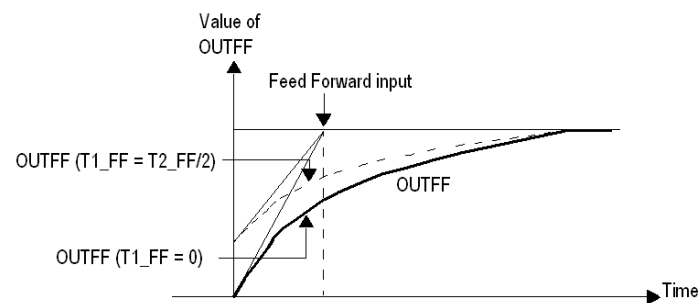
Step Advance

The Lead Lag function is configured in step advance (T1_FF > T2_FF): the OUTFF output is ahead of the input.



Step Delay

The Lead Lag function is configured in step delay ($T1\_FF < T2\_FF$): The OUTFF output is behind the input.



Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Feed Forward input	/	REAL	-3.4E38 / 3.4E38	/	R

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Time 1 (s)	T1_FF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Time 2 (s)	T2_FF	REAL	-3.4E38 / 3.4E38	0.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Feed Forward value	OUTFF	REAL	-3.4E38 / 3.4E38	/	R

Execution Monitoring

Monitoring the parameters of this function is integrated into the error management of the Feed forward branch.

Alarm on Deviation

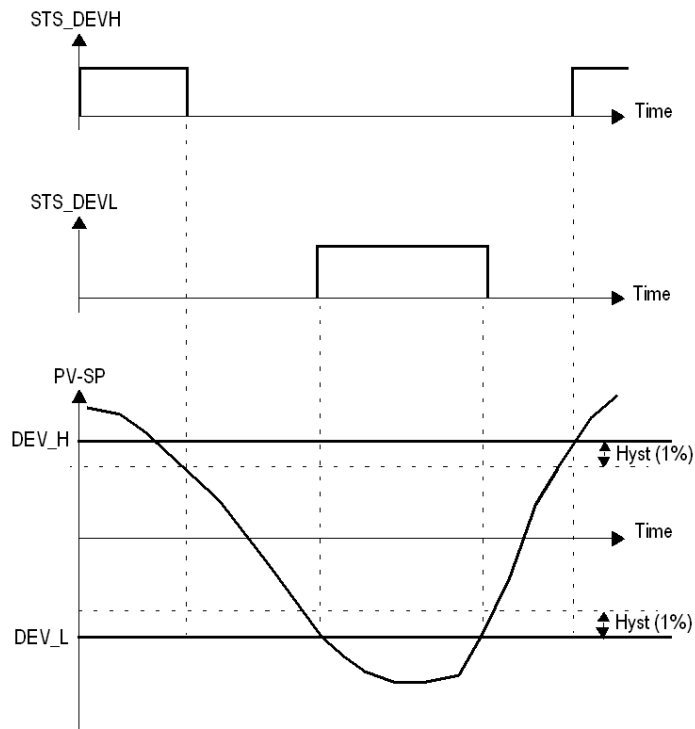
Description

This function monitors changes in the deviation between the process value (PV) and the setpoint (SP), by comparing these 2 values to 2 thresholds (high deviation threshold and low deviation threshold).

These alarms are controlled by a fixed hysteresis of 1% of the full scale of the loop.

NOTE: The threshold values must be greater than the hysteresis (1%), otherwise the alarms will still be active.

Alarm on deviation function diagram:



Parameters

Input parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process Value	PV	REAL	-3.4E38 / 3.4E38	/	R
Setpoint Value	SP	REAL	-3.4E38 / 3.4E38	/	R

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
High deviation threshold	DEV_H	REAL	0.0 / 3.4E38	5.0	R/W
Low deviation threshold	DEV_L	REAL	-3.4E38 / 0.0	-5.0	R/W

Output Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Upper Limit	STS_DEVH	EBOOL	/	/	R
Lower Limit	STS_DEVL	EBOOL	/	/	R
Or alarms (*)	STS_ALARMS	EBOOL	/	/	R

NOTE: (*) Or alarms = Or bit logic on level alarms and alarms on deviation.

Execution Monitoring

The monitoring of parameters is integrated into the error management of the Process Value branch.

5.4 Functions of the Loop Controller Branch

Subject of this Section

This section describes the calculation functions of the loop controller branch:

- ON OFF 2 states,
- ON OFF 3 states,
- PID,
- Model loop controller,
- Autotuning,
- Split Range,
- Heat/Cool.

What Is in This Section?

This section contains the following topics:

Topic	Page
2-State ON/OFF Loop Controller	116
3-State ON/OFF Loop Controller	119
PID	122
PID Parameters	125
Detailed PID Equations	128
Model Loop Controller	131
Model Loop Controller Parameters	135
Autotuning	137
Auto-Tuning Parameters	140
Autotuning Process	142
Auto-Tuning Operating Modes	144
Auto-Tuning Diagnostics Parameters	145
Aborting Auto-Tuning	146
Split Range	151
Hot/Cold	154

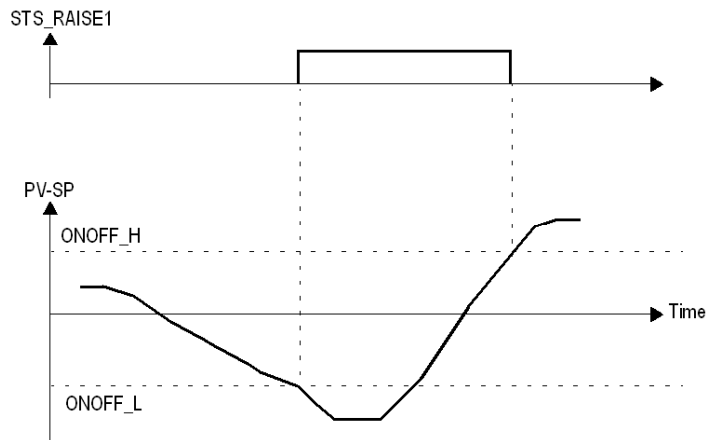
2-State ON/OFF Loop Controller

Description

The 2-state ON/OFF Loop Controller processes simple process controls, for which discrete 2 position monitoring is adequate.

Actuator control is performed according to process value/setpoint deviation in relation to 2 thresholds (upper and lower).

Functional Diagram:



Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process Value	PV	REAL	-3.4E38 / 3.4E38	/	R
Setpoint Value	SP	REAL	-3.4E38 / 3.4E38	/	R

Internal Parameters:

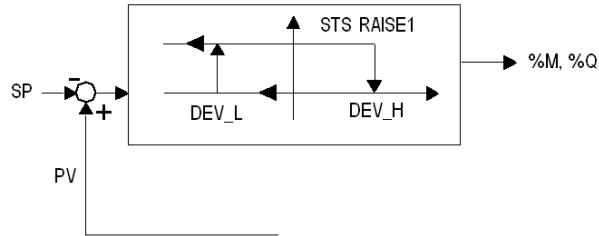
Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower Threshold	ONOFF_L	REAL	-3.4E38 / 3.4E38	-5.0	R/W
Upper Threshold	ONOFF_H	REAL	-3.4E38 / 3.4E38	5.0	R/W

Output Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Status	STS_RAISE1	EBOOL	/	/	R
Auto_Manu State	STS_M_A	EBOOL	/	/	R
Command	/	EBOOL	/	/	R
Process Value / Setpoint Deviation	DEV	REAL	-3.4E38 / 3.4E38	/	R

Process Value / Setpoint Deviation

As soon as the Process Value/Setpoint Deviation ($DEV = PV - SP$) becomes smaller than the lower ONOFF_L threshold, the logic output switches to 1 state. If the deviation starts to increase again, it must exceed the threshold ONOFF_H for the output to switch to 0 state.



Operating Modes

The 2-state ON/OFF Loop Controller has two operating modes:

- Automatic mode: the output is calculated by the loop controller.
- Manual mode: the loop controller does not set the output. You can directly modify the value of the variable connected to the output.
In case of cold start, the manual mode output is set to the 0 state.

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem appears in the floating-point calculation.
- Lower threshold > 0 .
- Upper threshold < 0 .

In any case, the error is considered to be critical. The regulation loop output is frozen. Errors are reflected in the status words.

3-State ON/OFF Loop Controller

Description

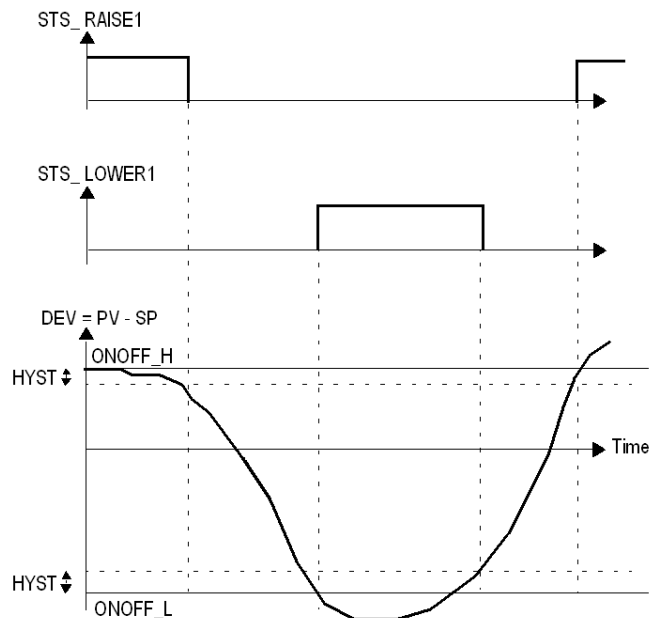
The 3-state ON/OFF Loop Controller processes simple process controls, for which discrete 3 position monitoring is adequate.

Control of the 2 actuators is performed according to the position of process value/setpoint deviation in relation to 2 thresholds (upper and lower).

Threshold management integrates a configurable hysteresis. It is possible, for instance, to use the loop controller to regulate a hot/cold process as discrete.

For more complex process controls, we recommend using a PID loop controller.

Functional Diagram:



Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process Value	PV	REAL	-3.4E38 / 3.4E38	/	R
Setpoint Value	SP	REAL	-3.4E38 / 3.4E38	/	R

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower Threshold	ONOFF_L	REAL	-3.4E38 / 3.4E38	-5.0	R/W
Upper Threshold	ONOFF_H	REAL	-3.4E38 / 3.4E38	5.0	R/W
Hysteresis	HYST	REAL	ONOFF_L / ONOFF_H	0.0	R/W

Output Parameters:

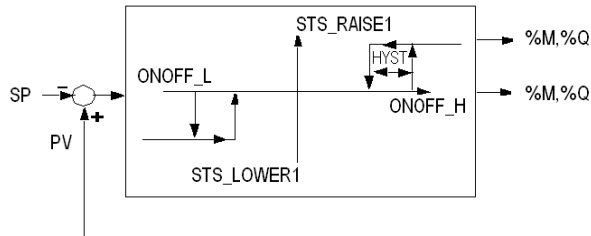
Parameter	Symbol	Type	Thresholds	Default Value	R/W
Process Value / Setpoint Deviation	DEV	REAL	-3.4E38 / 3.4E38	/	R
OUT1 command status	STS_RAISE1	EBOOL	/	/	R
OUT2 command status	STS_LOWER1	EBOOL	/	/	R
Auto_Man	STS_M_A	EBOOL	/	/	R
Command value	OUT1	EBOOL	/	/	R
Command value	OUT2	EBOOL	/	/	R

Operating Modes

The 3-state ON/OFF Loop Controller has two operating modes:

- Automatic mode: the STS_LOWER1 and STS_RAISE1 outputs are calculated by the loop controller.
- Manual Mode: the loop controller does not set the outputs. You may directly modify the value of the variable connected to the STS_LOWER1 et STS_RAISE1 outputs.

Operating modes and associated commands:



Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem arises in a calculation in floating point mode.
- Lower threshold > 0.
- Upper threshold < 0.

In each case, the error is considered to be serious. The regulation loop output is frozen. Errors are reflected in the status words.

PID

Description

The PID function executes a PID algorithm with a mixed (serial / parallel) or parallel structure. It has the following functions:

- Calculation of proportional, integral and derivative actions in incremental or absolute format.
- Antisaturation of the integral action.
- Direct or inverse action.
- Derivative from the process value or from the deviation.
- Parametering of the transient gain of the derivative.
- Integral band
- Feed forward action to compensate for interference.
- Dead band on deviation.
- High and low limitation of the output signal.
- Limit of the gradient of the output signal.
- Output shifting, also called manual integrated.
- Selecting Automatic / Manual operating mode.
- Tracking Mode.
- Autotuning of the principle coefficients.

Transfer Function

The PID transfer function depends on the structure used (mixed or parallel structure):

Mixed structure

$$OUT = kp \cdot \left(1 + \frac{1}{n \cdot p} + \frac{td \cdot p}{1 + \left(\frac{td}{kd} \right) \cdot p} \right) \cdot IN$$

Parallel structure

$$OUT = \left(kp + \alpha \cdot \frac{1}{n \cdot p} + \alpha \cdot \frac{td \cdot p}{1 + \left(\frac{td}{kd} \right) \cdot p} \right) \cdot IN$$

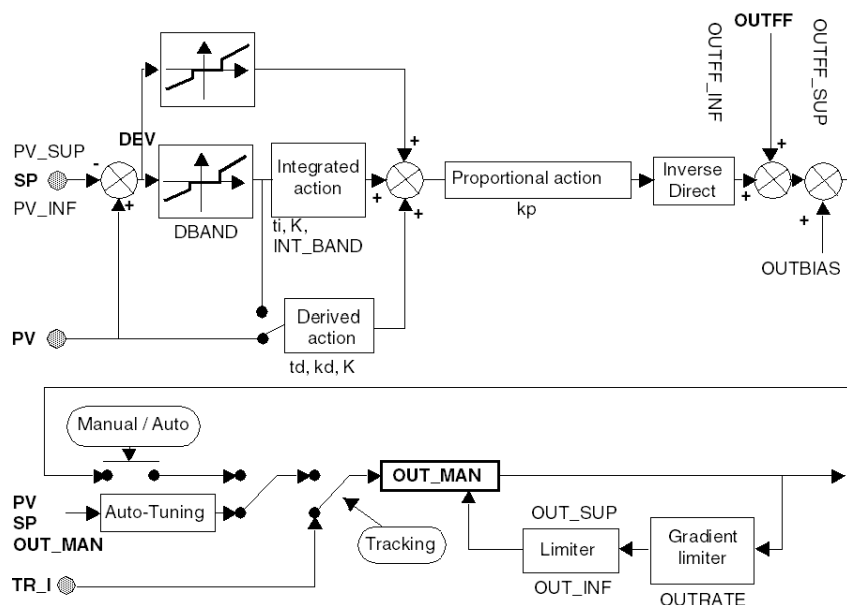
$$\alpha = \text{scale factor} = \frac{\text{OUT\_SUP} - \text{OUT\_INF}}{\text{PV\_SUP} - \text{PV\_INF}}$$

Note: the user parameter k_p may be specified :

- either in physical scale, then $k_p = KP$,
- or in standardized scale, then $k_p = \alpha KP$

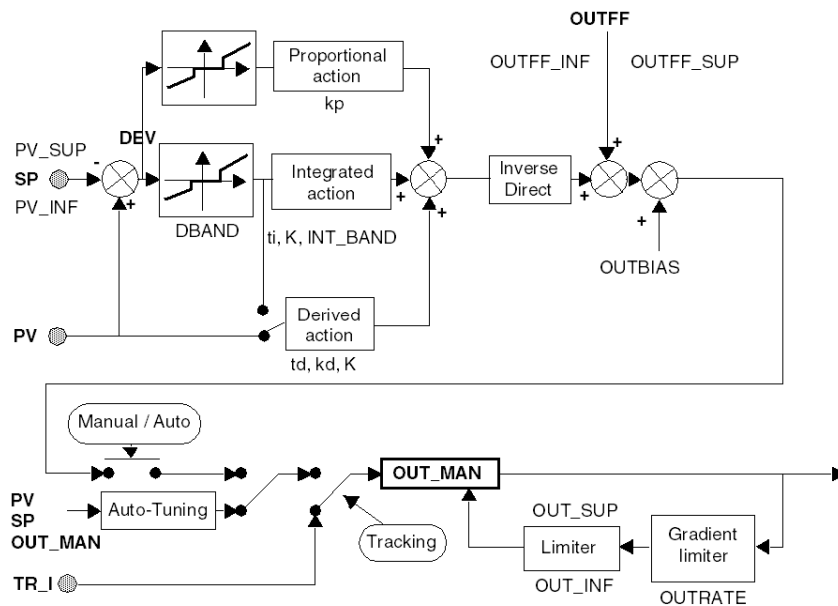
Functional Diagram of the Mixed PID

This diagram illustrates the principle of the mixed structure PID. It does not represent the implementation of the algorithm in incremental format.



Function Diagram of the Parallel PID

This diagram illustrates the principle of the parallel structure PID. It does not represent the implementation of the algorithm in incremental format.



PID Parameters

Mixed or Parallel Structure

- If the loop controller structure is mixed (default configuration), the proportional action is applied downstream of the integral and derivative actions. The K gain applied to these actions is therefore equal to k_p (*see page 122*).
- If the loop controller structure is parallel, the proportional action is applied parallel to the integral and derivative actions. In this case the k_p gain is not applied to the integral and derivative actions. The K gain is simply equal to the relation between the output scale and the process value scale.

Direct or Inverse Action

The direction of the PID Loop Controller may be adapted to that of the actuator/process pair. The action can be defined in the opposite direction (default configuration) or in the same direction.

If the action is direct, a positive gap ($PV - SP$) leads to an output increase.

If the action is inverse a positive gap ($PV - SP$) leads to an output decrease.

Derivative Action

The derivative action can act upon the process value or the gap.

Manual / Auto Switching

The absolute form of the algorithm allows Manu / Auto switching without bumps (*see page 212*).

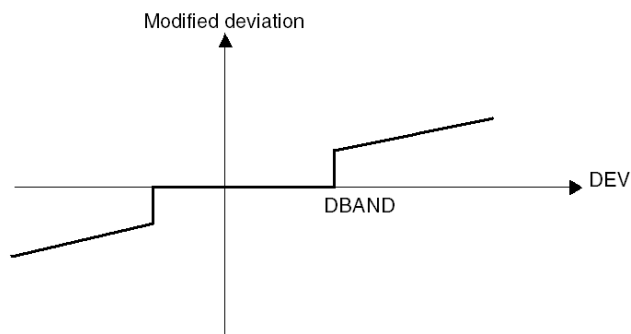
Adjustment Parameters

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower limit of the process value scale	PV_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Upper limit of process value scale	PV_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Lower limit of output scale	OUT_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Upper limit of output scale	OUT_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Proportional gain	KP	REAL	0.0 / 3.4E38	1.0	R/W
Integration time (s)	TI	REAL	0.0 / 3.4E38	0.0	R/W
Derivation time (s)	TD	REAL	0.0 / 3.4E38	0.0	R/W
Derivative gain	KD	REAL	1,0 / 3.4E38	10.0	R/W
Dead band on deviation	DBAND	REAL	0.0 / 3.4E38	0.0	R/W
Manual compensation of the static gap	OUTBIAS	REAL	-3.4E38 / 3.4E38	0.0	R/W
Limiting the output variation, in units s	OUTRATE	REAL	0.0 / 3.4E38	0.0	R/W
Integral band	INT_BAND	REAL	0.0 / 3.4E38	0.0	R/W

Dead band on deviation

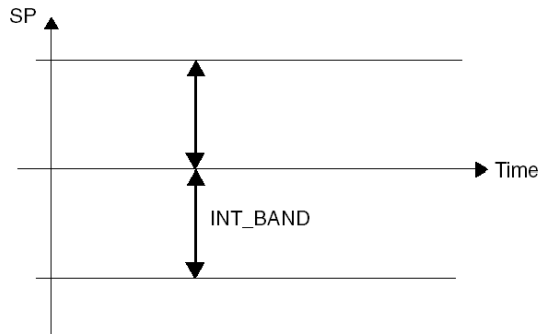
The dead band on deviation (DBAND) is used to limit small recovery bumps to actuator to the function point. As long as the gap remains smaller than DBAND (in absolute value), the loop controller assumes it to be null for its calculations.



Integral Band

The integral band defines a zone (around the setpoint) in which the integral action is calculated. When the process value / setpoint gap is greater than this band the integral action is frozen.

The integral band is limited to the proportional band ($100 / K_p$).



BIAS on the Command

If the integral action is not used ($T_i = 0$), the use of BIAS on the PID command (OUTBIAS), is used to guarantee precision of the function point.

If T_i is different from 0, OUTBIAS is not taken into account.

Detailed PID Equations

Introduction

The PID algorithm uses the loop controller parameters (*see page 125*) as well as the intermediate variables and the following functions.

Intermediate variables and functions:

Variable/Function	Description
TermP	Proportional action value.
TermI	Integral action value.
TermD	Derived action value.
TermFF	Feed forward action value (disruption compensation).
(new)	Indicates a calculated value during the current execution of the algorithm.
(old)	Indicates a value calculated during the previous execution of an algorithm.
kp	Proportional gain. This user parameter can be specified either on a physical scale or on a standardized scale: - Physical scale: $kp = KP$ - Standardized scale: $kp = \alpha \cdot KP$
K	Integral and derived action gains. The gain varies according to the loop controller structure (mixed or parallel) and the presence of the proportional action: - If the structure is mixed and $kp \neq 0$, then $K = kp$ - If the structure is parallel and $kp = 0$, then $K = \alpha = \text{scale factor} = \frac{OUT\_SUP - OUT\_INF}{PV\_SUP - PV\_INF}$
VAR	Variable used in the derived action formula. Its value depends on the "Derived action" parameter: - VAR = PV if the derived action is on the process value. - VAR = DEV if the derived action is in the deviation.
Direction	- Direction = +1 if action is direct. A positive deviation (PV - SP) leads to an output increase. - Direction = -1 if action is inverse. A positive gap (PV - SP) leads to an output decrease.

Variable/Function	Description
T_ECH	Sampling Period
Function Δ	$\Delta(x(t)) = x(t) - x(t-1)$
Limit function	Corrector output limit function.

Absolute Form of the Algorithm

If **Ti = 0**, the absolute form of the algorithm is used. The loop controllers are of type P or PD.

$$\text{OUT} = \text{TermP} + \text{TermD} + \text{TermFF} + \text{OUTBIAS}$$

$$\text{OUTD} = \text{OUTP}(\text{new}) - \text{OUTP}(\text{old})$$

$$\text{OUT} = \text{limit}(\text{OUT}) \text{ (not implemented)}$$

$$\text{TermP} = \text{sens} \cdot k_p \cdot \text{DEV}$$

$$\text{TermD} = \frac{td \cdot \text{TermD}(\text{old}) + \text{sens} \cdot K \cdot td \cdot kd \cdot (\text{VAR}(\text{new}) - \text{VAR}(\text{old}))}{kd \cdot dt + td}$$

Incremental Form of the Algorithm

If **Ti <> 0**, the incremental form of the algorithm is used. The loop controllers are of type PID.

$$\text{OUTD} = \Delta \text{TermP} + \text{TermI} + \Delta \text{TermD} + \Delta \text{TermFF}$$

$$\text{OUT} = \text{OUT}(\text{old}) + \text{OUTD}(\text{new}), \text{ default mode}$$

$\text{OUT} = \text{RCPY} + \text{OUTD}(\text{new})$, actuator position copy mode. This mode is used in certain special cases where the actuator position can be different from the calculated output of the PID (SERVO output, cascaded loop or auto selector loop).

$$\text{OUT} = \text{limit}(\text{OUT})$$

$$\Delta \text{TermP} = \text{sens} \cdot k_p \cdot \Delta(\text{DEV})$$

$$\Delta \text{TermD} = \Delta \left(\frac{td \cdot \text{TermD}(\text{old}) + \text{sens} \cdot K \cdot td \cdot kd \cdot (\text{VAR}(\text{new}) - \text{VAR}(\text{old}))}{kd \cdot dt + td} \right)$$

Antisaturation

The integral antisaturation mechanism is implicit in the algorithm.

Pure Integral Mode

The loop controller is able to work in pure integral mode ($k_p = 0$). In this case the equations are as follows:

$$\text{OUTD} = \text{TermI} + \text{TermFF}$$

$$\text{OUT} = \text{OUT}(\text{old}) + \text{OUTD}(\text{new}), \text{ default mode}$$

$$\text{OUT} = \text{RCPY} + \text{OUTD}(\text{new}), \text{ actuator position copy mode}$$

$$\text{OUT} = \text{limit}(\text{OUT})$$

$$\text{TermI} = \text{sens} \cdot \alpha \cdot \text{T\_ECH/TI} \cdot \text{DEV}$$

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem arises in a calculation in floating point mode.
- The output scale is inconsistent at the time of the cold start of the controller ($\text{OUT\_INF} \geq \text{OUT\_SUP}$).

In each case, the error is considered to be serious. The loop output is frozen, while errors are signaled in the status words.

Model Loop Controller

Description

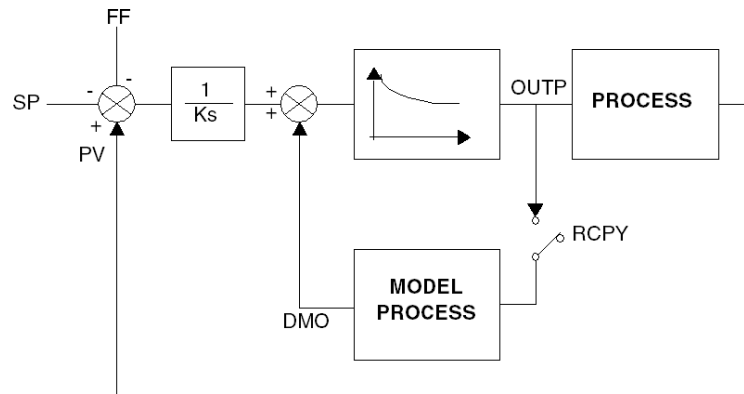
The model loop controller is used where there are significant pure delays in relation to the time constant of the process, a case which cannot be processed satisfactorily by classic PID process control. The model loop controller is also useful for regulating a non-linear process.

The model is first order + delay. Nevertheless, this loop controller can process any stable or aperiodic process, whatever its order. The parameters to be given are:

- The static gain (delta process value/delta command ratio in open loop).
- The equivalent time constant (response time/3).
- The process pure delay value (estimated value).
- The time constant in open loop/time constant in closed loop ratio.

Diagram of the Principal

The diagram of principal of the model loop controller algorithm is as follows:



Installing the Loop Controller

Installing the model loop controller is similar to installing a PID loop controller. The adjustment of PID parameters KP, TI and TD being replaced by gain adjustment, the time constant, the pure delay of the process model and the ratio of the time constants in open and closed loop.

The model loop controller has the same inputs and outputs as a PID (PV, RSP, FF, OUTP). It also has the RCPY optional input (model external input) which can be used to input the process's real input (the flow measured on output from a valve, for example).

NOTE: The DMO model output is not directly comparable to the PV process value. At this level, the model does not take into account Ks static gain and possible compensation (BIAS).

Functions

The functions other than the command calculation are identical to those of the PID:

- Direct or inverse action.
- Feed forward action to compensate for disruptions.
- Dead band on deviation.
- High and low limit of the output signal.
- Limit of the output gradient.
- Selecting Automatic / Manual operating mode.
- Tracking Mode.
- Autotuning of the main coefficients.

Delay Management

In the processs used by this loop controller, the delay is either:

- Variable (transfer of matter according to the flow in a circuit, speed of the transport base, for example).
- Very large.

These two cases are processed using a register (buffer) of size which can be parameterized. According to the size of this register, it will be possible to sample either all the sampling periods, one period in two, or one period in three etc.

It is possible to increase or decrease the delay T_DELAY during the execution of the program. The new delay is applied instantly, as long as it is compatible with the size of the register. The sampling period of the delay remains unchanged.

If the value of T_DELAY becomes too large compared to the size of the register it becomes impossible to store enough input values to reach the required delay, if the sampling is done in the same period. Therefore, the sampling period of the delay is recalculated and the output is only valid after a time equal to the new delay. To avoid this problem, you are advised to set the size of the register, taking into account the possible increases in the delay T_DELAY.

If the delay decreases by default the sampling does not change. All the same, it is possible to order a new sampling calculation if necessary.

In the case of a dynamic modification of the task time or the sampling period, the output is only valid after a time equal to the delay.

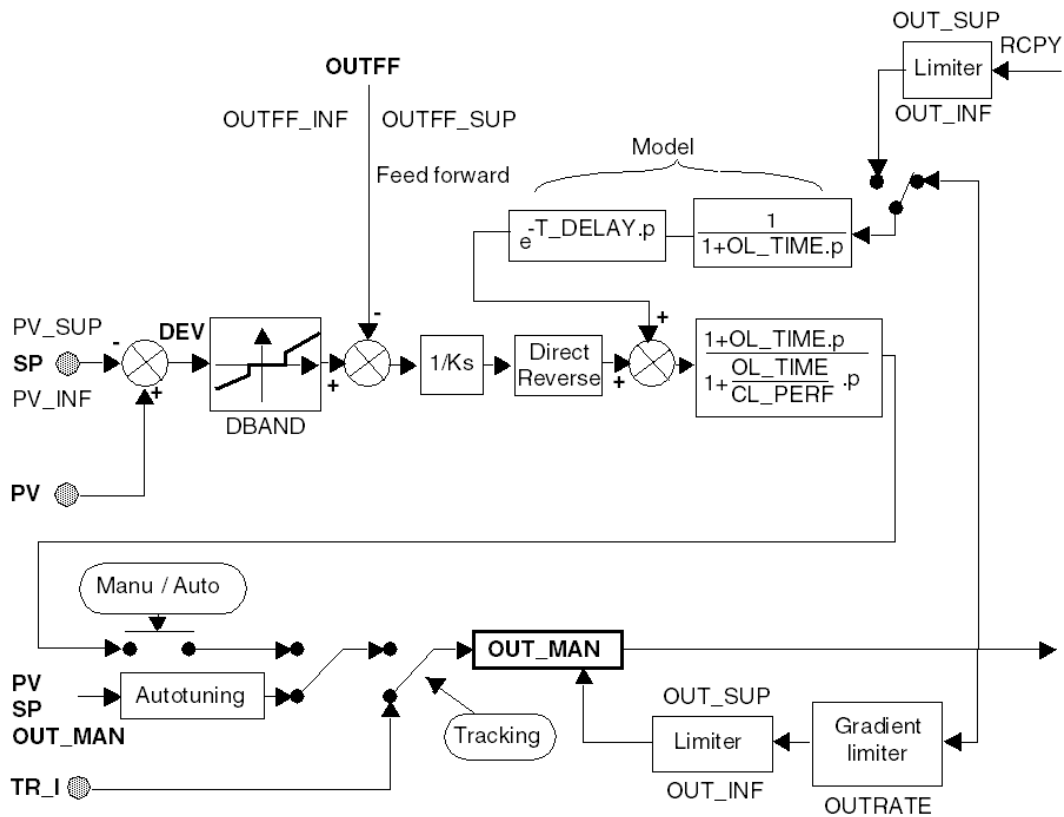
All dynamic modifications of T_DELAY between 0 s and 30 s is taken into account immediately without changing the sampling of the register.

Example

Sampling period	T_ECH = 300 ms
Size of delay register	50
Delay	T_DELAY = 25 s
Therefore, the delay register is sampled every 2 T_ECH	$50 \times 2 \times 0,3 = 30 \text{ s} > 25 \text{ s}$

Function Diagram

The function diagram of the model loop controller is as follows:



Model Loop Controller Parameters

Direct or Inverse Action

The direction of the PID loop controller can be adapted to that of the actuator torque / process. The action can be defined in the opposite direction (default configuration) or in the same direction.

Adjustment Parameters

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower limit of output scale	OUT_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Upper limit of output scale	OUT_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Static gain of the process in open loop	KS	REAL	0.0 / 3.4E38	1.0 (*)	R/W
Time constant of the process in open loop	OL_TIME	REAL	0.0 / 3.4E38	1.0 (*)	R/W
Ratio of natural time constant (open loop) to desired time constant (closed loop)	CL_PERF	REAL	0,1 / 3.4E38	1.0	R/W
Current pure delay of the process(s)	T_DELAY	REAL	0.0 / 3.4E38	0.0	R/W
Dead band on deviation	DBAND	REAL	0.0 / 3.4E38	0.0	R/W
Command copy input (optional)	RCPY	%IW, %MW	-32768 / 32767	/	R
Model output (optional)	DMO	%MF	-3.4E38 / 3.4E38	/	R
Delay register (obligatory)	/	%MF:n (**)	/	/	R

(*) KS and OL_TIME cannot take the value 0 (inconsistent value). They will be forced to the value 1.0.

(**) n corresponds to the size of the register and must be greater than 0.

Dead Band on Deviation

Dead band on deviation is identical to the PID loop controller (*see page 125*).

Output Parameters

You can access the value of the command OUT_MAN, but also the delayed output value of the DMO model.

Limits

The integrating processes are not managed by the model loop controller.

You can use a Servo drive without output copy because the model loop controller does not implement an incremental algorithm (the command value is calculated and then the command variation).

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem appears in the floating-point calculation.
- The output scale is inconsistent on cold start of the PLC (OUT_INF >= OUT_SUP).

In each case, the error is considered to be serious. The loop output is frozen, while errors are signaled in the status words.

Autotuning

Description

The autotuning function is used to gain time during the start-up of an installation by guaranteeing a stable adjustment.

The autotuning algorithm is based on a method of Ziegler-Nichols type:

- Analysis of the procedure for checking if this is first order with delay. The length of this analysis is 2.5 times the response time of open loops.
- Calculation of PID adjustment parameters (KP, TI, TD) or those of the model loop controller (KS, T1, T_DELAY). The range of set parameters is modulated by performance criteria in order to prioritize the response time to disruptions or stability.

Process Types

The algorithm processes the following process types:

- Single input and single output processes.
- Naturally stable processes or integrators.
- Asymmetric processes in the limits allowed by the PID algorithm.

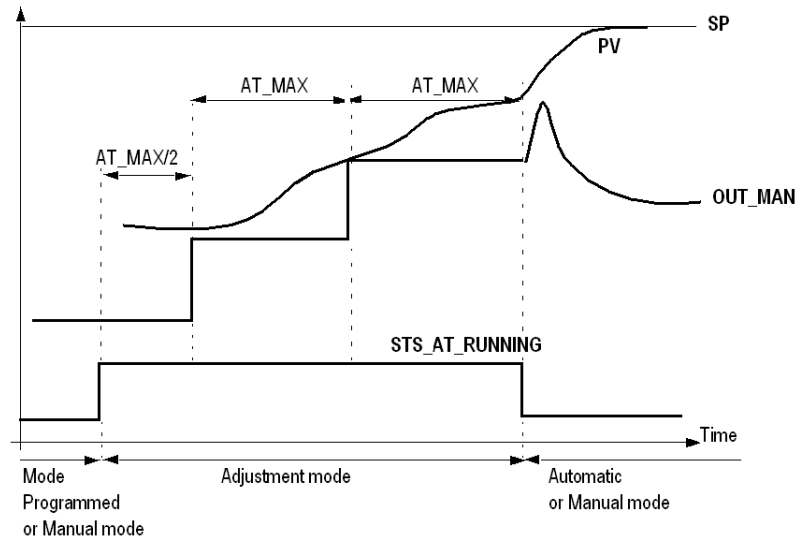
Autotuning Types

There are two possible autotuning types: warm or cold autotuning. The first phase of autotuning is the same for each of these: Noise and stability test of the process of length $0.5 \times AT\_TMAX$ during which the outputs remain constant. The following phases depend on the type of autotuning. The selection is made automatically by the algorithm.

Cold Autotuning

Cold autotuning is performed if the deviation between process value and setpoint exceeds 40% and if the process value is less than 30%. Two scales in the same direction are then applied to the loop controller output (OUT_MAN). Each scale has a length of AT_MAX.

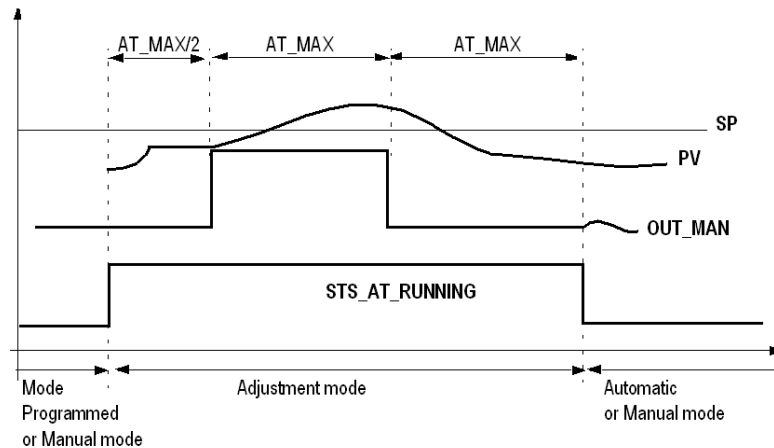
When autotuning is complete, the loop assumes its previous operating mode.



Warm Autotuning

If cold autotuning conditions are not fulfilled, warm autotuning is performed. A scale is applied to the loop controller output (OUT_MAN), then an inverse scale. Each scale has a length of AT_MAX.

When autotuning is complete, the loop assumes its previous operating mode.



Auto-Tuning Parameters

Internal Parameters

Internal parameters of the auto-tuning function are as follows:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Size of command division (%)	AT_STEP	REAL	-100.0 / 100.0	10.0	R/W
Duration of the scale (s)	AT_TMAX	REAL	4.0 / 3.4E38	100.0	R/W
Criteria for auto-tuning performance	AT_PERF	REAL	0.0 / 1.0	0.5	R/W
Proportional gain	KP	REAL	0.0 / 3.4E38	1.0	R/W
Integral time (s) (*)	TI	REAL	0.0 / 3.4E38	0.0	R/W
Derivation time (s) (**)	TD	REAL	0.0 / 3.4E38	0.0	R/W
Model gain (***)	KS	REAL	0.0 / 3.4E38	1.0	R/W
Model time constant (s) (***)	T1	REAL	0.0 / 3.4E38	0.0	R/W
Model delay (s) (***)	T_DELAY	REAL	0.0 / 3.4E38	0.0	R/W

(*) According to the loop controller used (PID or model loop controller).

(**) See PID loop controller (*see page 122*).

(***) See model loop controller (*see page 131*).

Output Parameters

For the output parameters, the range of variation (limits) and default value have no object. Therefore, these columns have not been given in the table in order to make it more readable.

Output parameters of the auto-tuning function are as follows:

Parameter	Symbol	Type	R/W
Value before proportional coefficient auto-tuning or model gain	KP_PREV	REAL	R
Value before integral coefficient auto-tuning or time constant model	TI_PREV	REAL	R
Value before derived coefficient auto-tuning or model delay	TD_PREV	REAL	R

Parameter	Symbol	Type	R/W
Auto-tuning in progress	STS_AT_RUNNING	EBOOL	R
Auto-tuning failed	AT_FAILED	EBOOL	R
Auto-tuning diagnostics interrupted	AT_ABORTED	EBOOL	R
Auto-tuning diagnostics parameter error	AT_ERR_PARAM	EBOOL	R
Auto-tuning diagnostics system error or power failure	AT_ERR_PWF_OR_EFB_FAIL	EBOOL	R
Auto-tuning diagnostics process value or actuator saturation	AT_ERR_SATUR	EBOOL	R
Auto-tuning diagnostics insufficient process value deviation	AT_DV_TOO_SMALL	EBOOL	R
Auto-tuning Diagnostics sampling period too large	AT_TSAMP_HIGH	EBOOL	R
Auto-tuning diagnostics inconsistent response	AT_INCONSIST_RESP	EBOOL	R
Auto-tuning diagnostics process value initially unstable	AT_NOT_STAB_INIT	EBOOL	R
Auto-tuning diagnostics scale period too small	AT_TMAX_TOO_SMALL	EBOOL	R
Auto-tuning diagnostics process value noise too high	AT_NOISE_TOO_HIGH	EBOOL	R
Auto-tuning diagnostics division period too large	AT_TMAX_TOO_HIGH	EBOOL	R
Auto-tuning diagnostics overflow > 10%	AT_OVERSHOOT	EBOOL	R
Auto-tuning diagnostics step minimum too high	AT_UNDERSHOOT	EBOOL	R
Auto-tuning diagnostics procedure not symmetrical enough	AT_UNSYMETRICAL_PT	EBOOL	R
Auto-tuning diagnostics integrating process	AT_INTEGRATING_PT	EBOOL	R

Autotuning Process

Adjustment Mode

Analysis of the process is divided into 3 stages:

- Noise and stability analysis of the process.
- First analysis of the response to a step function which provides a first identification model. From this first estimate a filter is calculated to be used in the second analysis.
- A second analysis of the response to a second step function, refined by using a data filter. A full process model is thus obtained.

After each of these two analyses, a set of loop controller parameters to be adjusted is calculated. The equations giving these loop controller parameters are based on the gain and the ratio of response time to process delay.

If the difference between these two analyses is too great, the model's estimate is rejected and autotuning fails. The loop controller output is set back to its value before the start of autotuning.

On a robust level, the algorithm must be capable of supporting the gain and time constant changes in a ratio 2, without losing its stability. The asymmetrical processes are supported as long as they follow these constraints. If this is not the case, an error is signaled by the diagnostics.

Step Function Parametering

The 2 step functions applied to the output during the autotuning process are characterized by 2 parameters:

- The duration of the step function AT_TMAX, which must be greater than 4 s.
- The amplitude of the step function AT_STEP, which must be greater than 1% of the output scale (OUT_INF, OUT_SUP).

The function also checks that the output does not go beyond the output scale limits. This checking is done as the autotuning starts.

Maximum duration of the step function

The maximum value of AT_TMAX is limited by the task cycle time into which the loop is configured. This value, **in seconds**, can be configured to a maximum of **65.5 x task cycle time (en ms)**.

For example, if the task MAST = 50 ms, AT_TMAX = 65.5 x 50 = **3275 seconds** maximum.

Parameter Values

As an indication, the following table gives the parameter values for some classic process control types:

Scheme type	AT_TMAX (s)	AT_STEP (%)
Liquid flow or pressure	5 - 30	10 - 20
Gas pressure	60 - 300	10 - 20
Level	120 - 600	20
Steam temperature or pressure	600 - 3600	30 - 50
Composition	600 - 3600	30 - 50

Performance Criterion

The loop controller adjustment can be done according to the performance criterion value AT_PERF.

The AT_PERF parameter varies between 0 and 1, which means it is possible to prioritize stability for AT_PERF close to 0 or to obtain a more dynamic adjustment (and therefore to optimize response time to disruptions) by pushing AT_PERF towards 1.

NOTE: The parameters AT_PERF, AT_TMAX et AT_STEP are unique to each process control channel. There is therefore more than one set of parameters for the 3 single, cascade or auto-selector loop controllers. Consequently, for one process control channel, one autotuning operation can be activated and executed at a given instant.

Auto-Tuning Operating Modes

Auto-Tuning Commands

Different commands are used to drive the auto-tuning function:

- Starting auto-tuning (ORDER_CMD (%MW_r.m.c.11) = 16#000E)
This command starts the auto-tuning process. It can be directly activated from the auto-tuning function grid.
- Stopping auto-tuning (ORDER_CMD (%MW_r.m.c.11) = 16#000F)
This command stops the auto-tuning process. In this case, PID parameters are left unchanged and a diagnostic is performed.
- Restore previous adjustment (ORDER_CMD (%MW_r.m.c.11) = 16#0010)
This command swaps the current loop controller parameters with the previous ones (KP_PREV, TI_PREV, TD_PREV). This command is not allowed if auto-tuning event is in progress.

Auto-Tuning Operating Modes

At the start of auto-tuning, the loop controller may be in automatic or manual mode. When auto-tuning begins, it switches to adjustment mode, and the output retains the last imposed or calculated value. At the end of auto-tuning:

- If auto-tuning is successful, the loop reverts to its previous mode (automatic or manual).
- If auto-tuning fails, the output is initialized at the pre-auto-tuning value, adjustments are not made and the loop reverts to its previous mode (automatic or manual).

The direction of loop controller action is checked and compared with the model gain sign. The system displays an error in case of incompatibility.

Auto-Tuning Diagnostics Parameters

Auto-Tuning Diagnostics

For a variety of reasons, the auto-tuning process may

- not be started,
- be aborted during execution,
- fail by setting forth (or not) a set of parameters determined by the cause of failure.

NOTE: It is not possible to perform 2 auto-tuning operations simultaneously on the same channel.

Acknowledgment

The diagnostics message acknowledgement function is accessible from the Unity Pro screen or with the acknowledge commands.

NOTE: The diagnostics message is unique to each process control channel. There is only one message for the 3 loops of the 3 single-loop controllers, or for the 2 loops of the Cascade and Auto-selector loop controllers.

Reasons Why the Auto-Tuning May Not Start

The following errors prevent auto-tuning from starting:

- Parameter error (Bit 2: AT_ERR_PARAM)
Possible causes of a parameter error are as follows:
 - Step duration too brief ($AT\_TMAX < 4$ s).
 - Amplitude too small ($AT\_STEP < 1\%$ of output scale).
 - Impossible test protocol. If the current output + n times the scaling amplitude ($n = 1$ for hot auto-tuning, and $n = 2$ for a cold auto-tuning) falls outside the output scale (OUT_INF , OUT_SUP), the test protocol is no longer applicable. $STEP\_AMPL$ must be set to a value compatible with the current function point.
- Incorrect sampling period (Bit 6: AT_TSAMP_HIGH)
If the sampling period is too large in relation to the step duration (greater than $AT\_TMAX/25$), the response analysis will be too imprecise, which in turn inhibits auto-tuning. This case is specific to very fast process controls (AT_TMAX increasing the process stabilization time by a few seconds). $TMAX$ may therefore be increased, as the algorithm is not very sensitive to this parameter (in a ratio from 1 to 3). You may also adjust the sampling period.

Reasons Why Auto-Tuning May Be Aborted

Several causes may lead to auto-tuning abortion (*see page 146*).

Aborting Auto-Tuning

Interruption Due to System Error

Bit 3: AT_ERR_PWF_OR_EFB_FAIL

Auto-tuning is aborted if a system PLC event arises, thus preventing the sequence from unfolding completely. For example, a power outage will automatically stop the auto-tuning function, upon power restoration.

Process Value Saturation

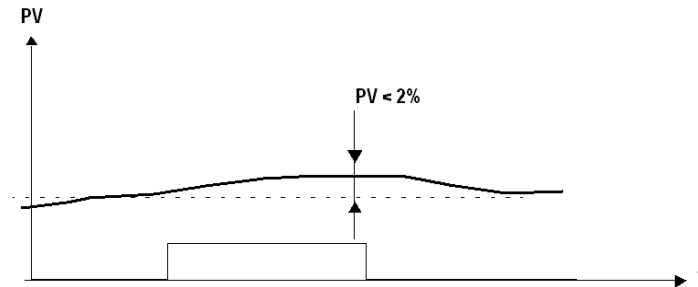
Bit 4: AT_ERR_SATUR

If the process value leaves the interval of the full scale (PV_INF, PV_SUP), autotuning is aborted and the loop controller reverts to its previous mode. Predicting the future process value even allows the autotuning to be stopped before the overflow occurs (when a first model has been identified).

Insufficient Variation

Bit 5: AT_DV_TOO_SMALL

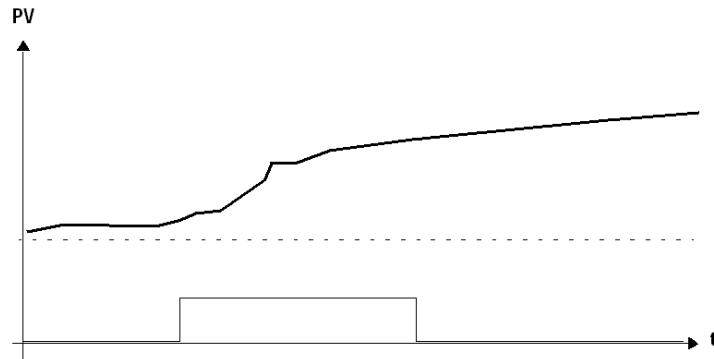
The amplitude of the step function is too brief to cause a significant process reaction. Therefore it is possible to increase AT_STEP.



Inconsistent Response

Bit 7: AT_INCONSIST_RESP

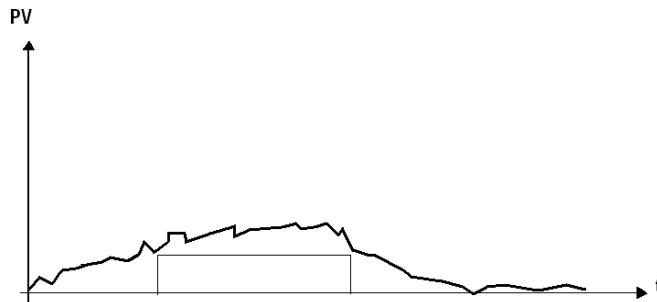
The process response is inconsistent (gains are not both + or -). This may be due to a significant disruption, coupling with other loops, etc. Auto-tuning is stopped and a diagnostic is generated.



Noise Too High

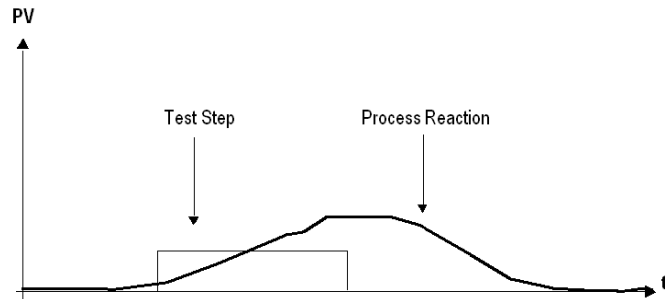
Bit 10: AT_NOISE_TOO_HIGH

The process reaction to the step function is not large enough in relation to noise. Filter the process value or increase AT_STEP.

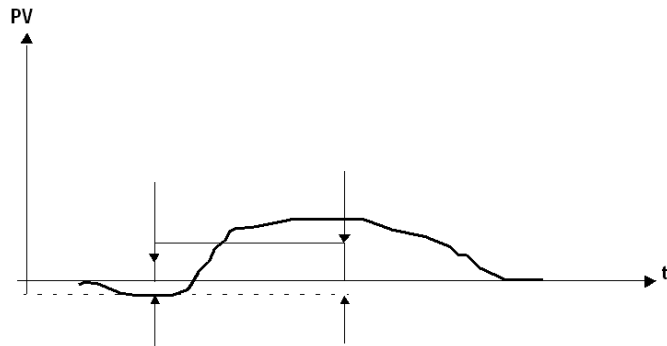


Step Duration (AT_TMAX) Too Short**Bit 9: AT_TMAX_TOO_SMALL**

The response is not stabilized until the initial command returns. The calculated parameters are therefore false.

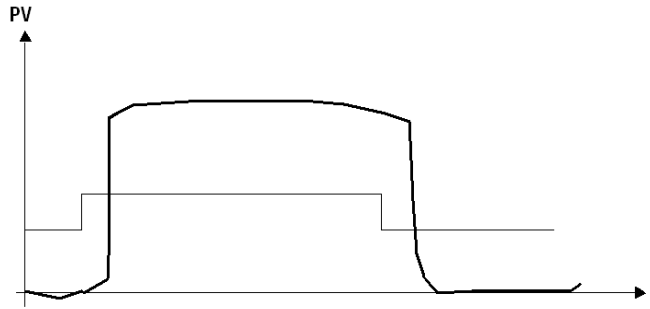
**Process Value Initially Not Stabilized****Bit 8: AT_NOT_STAB_INIT**

Auto-tuning was launched when the process value was not stabilized. If the process value deviation is significant in relation to the division, the test results will be false.



Step Duration (AT_TMAX) Too Long**Bit 11: AT_TMAX_TOO_HIGH**

AT_TMAX determines the sampling frequency of process values used to calculate coefficients. AT_TMAX must be between 1 and 5 times the process ascent time.

**Large Overshoot Process****Bit 12: AT_OVERSHOOT**

This bit is positioned in state 1 if the reaction to a command division prompts an excessive process value overshoot (greater than 10%). Process does not match the models treated by the algorithm.

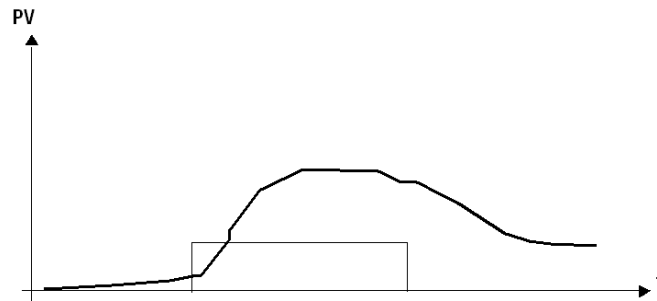
Non-Minimal Step Process**Bit 13: AT_UNDERSHOOT**

This bit is positioned in state 1 if the reaction to a command scale prompts a response inversion in its initial phase (undershoot greater than 10%). Process does not match the models treated by the algorithm.

Asymmetrical Process

Bit 14: AT_UNSYMETRICAL_PT

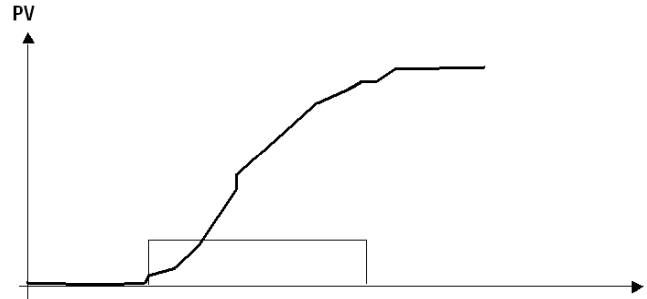
The process is asymmetrical (not symmetrical).



Integrating Process

Bit 15: AT_INTEGRATING_PT

Either it is an integrating process or AT_TMAX is too small and the process is asymmetrical. The calculated coefficients match the integrating process. If this is not the case, restart auto-tuning after increasing AT_TMAX.



Split Range

Description

This function is useful if two actuators are used to cover the whole print range of the adjustment spectrum. It is placed downstream of the loop controller.

The Split Range function also has the following functions:

- It manages overlaps as well as dead bandwidths between the two actuators.
- Manual command and a manual order (consistent with a single PID) can be used.

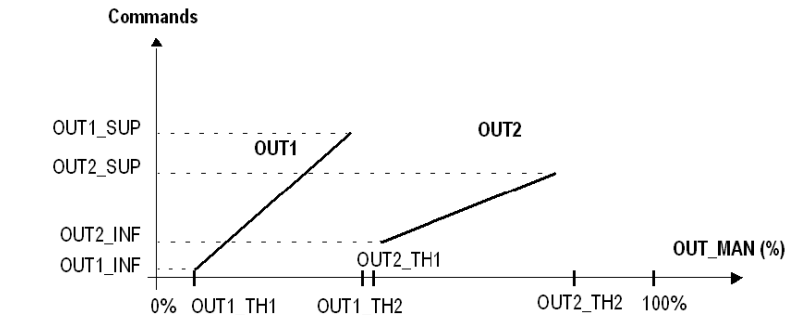
The Split Range function is used to manage analog outputs as well as servo drives with copy. It cannot be used to manage servo drives without copy.

When this function is used, the loop controller output scale must be (0.100).

Function Parametering

Function parametering defines the characteristics of each actuator, in other words, the way in which the two outputs should vary between the two thresholds.

The output value varies in a linear fashion. Outside the two thresholds, output is limited to pre-set thresholds.



with OUT_i_TH_j: output i threshold j

Function Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command value	OUT_MAN	REAL	0.0 / 100.0	/	R/W

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Value of OUT1 for OUT_MAN = OUT1_TH1	OUT1_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Value of OUT1 for OUT_MAN = OUT1_TH2	OUT1_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Value of OUT2 for OUT_MAN = OUT2_TH1	OUT2_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Value of OUT2 for OUT_MAN = OUT2_TH2	OUT2_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Input value for which OUT1 = OUT1_INF	OUT1_TH1	REAL	0.0 / 100.0	0.0	R/W
Input value for which OUT1 = OUT1_SUP	OUT1_TH2	REAL	0.0 / 100.0	50.0	R/W
Input value for which OUT2 = OUT2_INF	OUT2_TH1	REAL	0.0 / 100.0	50.0	R/W
Input value for which OUT2 = OUT2_SUP	OUT2_TH2	REAL	0.0 / 100.0	100.0	R/W

Output Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Limiting the output 1 variation in %/s	OUTRATE	REAL	0.0 / 3.4E38	0.0	R/W
Limiting the output 2 variation in %/s	OUTRATE2	REAL	0.0 / 3.4E38	0.0	R/W

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem arises in a calculation in floating point mode.

In any case, the error is considered to be critical. The loop output is frozen and the faults are signaled in the status words. A (warning) fault is also signaled if the OUT1_TH1, OUT1_TH2, OUT2_TH1 and OUT2_TH2 thresholds are not between 0 and 100%.

Hot/Cold

Description

This function is useful if two opposing actuators are used to cover the whole range of the adjustment spectrum. It is placed downstream from the Loop Controller.

The Hot/Cold function also includes the following operating processes:

- It manages overlaps as well as dead bandwidths between the two actuators.
- Manual command and a manual order (consistent with a single PID) can be used.

The Hot/Cold function is used to manage analog outputs as well as servo drives with copy. It does not allow managing servo drives without copy.

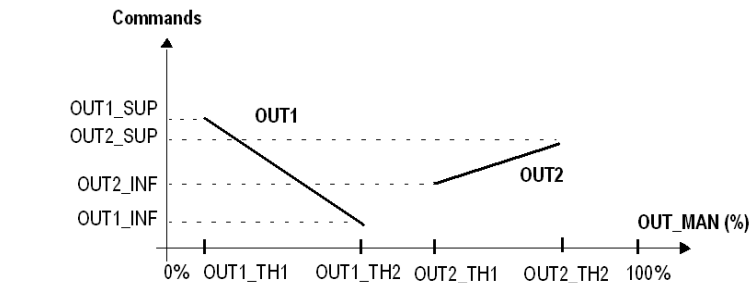
When using this function, you must set the loop controller output scale on (0, 100).

Function Parameters

Function parameters define the characteristics of each actuator, that is to say the manner in which both vary between both thresholds.

The output value varies in a linear fashion. Outside the two thresholds, the output is limited to the preset thresholds.

Output 1 manages the "cold," while Output 2 manages the "heat."



with OUT_i_TH_j: output *i* threshold *j*

Function Parameters

Input Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Value	OUT_MAN	Floating	0.0 / 100.0	/	R/W

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Value of OUT1 for OUT_MAN = OUT1_TH1	OUT1_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Value of OUT1 for OUT_MAN = OUT1_TH2	OUT1_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Value of OUT2 for OUT_MAN = OUT2_TH1	OUT2_INF	REAL	-3.4E38 / 3.4E38	0.0	R/W
Value of OUT2 for OUT_MAN = OUT2_TH2	OUT2_SUP	REAL	-3.4E38 / 3.4E38	100.0	R/W
Input value for which OUT1 = OUT1_INF	OUT1_TH1	REAL	0.0 / 100.0	50.0	R/W
Input value for which OUT1 = OUT1_SUP	OUT1_TH2	REAL	0.0 / 100.0	0.0	R/W
Input value for which OUT2 = OUT2_INF	OUT2_TH1	REAL	0.0 / 100.0	50.0	R/W
Input value for which OUT2 = OUT2_SUP	OUT2_TH2	REAL	0.0 / 100.0	100.0	R/W

Output Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Limiting the output 1 variation in %/s	OUTRATE	REAL	0.0 / 3.4E38	0.0	R/W
Limiting the output 2 variation in %/s	OUTRATE2	REAL	0.0 / 3.4E38	0.0	R/W

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem arises in a calculation in floating point mode.

In any case, the error is considered to be critical. The loop output is frozen, while errors are signaled in the status words. An error (warning) is also displayed if the OUT1_TH1, OUT1_TH2, OUT2_TH1 and OUT2_TH2 thresholds do not fall between 0 and 100%.

5.5 Functions of the output branch

Subject of this section

This section describes the calculation functions of the output processing path:

- Servo,
- PWM,
- Output scaling,
- Output format.

What Is in This Section?

This section contains the following topics:

Topic	Page
Servo	158
Operating Process Examples of the Servo Function	161
PWM	164
Output Scaling	166
Output Limiter	168
Output Format	170

Servo

Description

This function is used to perform process control with the electric servo drives with or without copy position. It uses the loop controller's digital output to generate 2 logic outputs, RAISE and LOWER.

When this function uses the copy position, it performs a loop control of the actuator position. When the copy position is not used, the loop controller and the associated servo function perform a floating point process control.

If the servo function is used, the loop controller output scale must be (0, 100).

Function Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Value	OUT_MAN	REAL	0.0 / 100.0	/	R
Command value (*)	OUTi	REAL	0.0 / 100.0	/	R
Upper Stop	/	EBOOL	/	/	R
Lower Stop	/	EBOOL	/	/	R
Copy position	/	REAL	0.0 / 3.4E38	/	R
Command variation value	OUTD	REAL	-100.0 / 100.0	/	R

(*) Hot/Cold or Split Range case

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Opening time (s)	T_MOTORi	REAL	0.0 / 3.4E38	10.0	R/W
Minimum time (s)	T_MINIi	REAL	0.0 / 3.4E38	0.0	R/W

Output Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Open command status	STS_RAISEi	EBOOL	/	/	R
Close command status	STS_LOWERi	EBOOL	/	/	R

Servo with Copy Position (RCPY)

The Servo function behaves in different manners depending on whether the copy position is used or not.

When the copy position is used, the Servo function generates a binary RAISE or LOWER command for each new OUT_MAN output value produced by the loop controller. The duration of this command is proportional to the deviation between the loop controller command and the copy position value, which then performs a loop control proportional to the actuator position.

NOTE: When the calculated period exceeds the loop sampling period (in automatic mode) or the task cycle time (in other operating modes), this is not stored for the next cycles.

Servo without Copy Position

If copy position is not used, the Servo function generates a binary RAISE or LOWER command for each new command variation value produced by the loop controller. The duration of this command is proportional to the output variation from the OUTD loop controller.

NOTE: When the calculated period exceeds the loop sampling period (in automatic mode) or the task cycle time (in other operating modes), the remaining application period is added onto the new period calculation. This enables it to be processed on several cycles.

The Servo function associated to the loop controller is used to perform a floating point process control. The algorithm does not use the loop controller's absolute output, but the output variation. The RAISE output (or LOWER, depending on the variation sign) is set at 1 for a length of time proportional to the valve opening time (T_MOTOR) and the OUTD variation value.

Pulse Period

The pulse period (T_IMP) to be applied to the output is calculated with the following principle:

- An initial theoretical value is given by the formula:

$$T\_IMP = (OUT\_MAN - RCPY) (\%) \times T\_MOTOR \text{ (with copy position)}$$

$$T\_IMP = (T\_IMP + OUTD) (\%) \times T\_MOTOR \text{ (without copy position)}$$
- In order not to generate pulses that are too short, pulses are limited to a minimum time period T_MINI.
- When the pulse time calculation gives a value less than T_MINI, the Servo function does not generate a pulse but stores the value for the next calculation. This enables correct processing where the output variations of a loop controller are insignificant but lasting.
- Without copy position, it is advisable to connect and use position stops in order to prevent the algorithm from becoming saturated.

Actuator Opening Time

T_MOTOR actuator opening time enables the function to adapt to different servo drives.

Pulse time to be applied to RAISE or LOWER is proportional to the full scale actuator opening time.

Minimum Pulse Period

The minimum pulse period T_MINI is used to avoid generating pulses that are too short, and generally harmful to actuators.

When the calculated pulse time to be applied to RAISE or LOWER is less than T_MINI, the function does not generate a pulse. In all cases, every pulse started lasts for at least the T_MINI period.

Position Stops

When position stop is reached, the RAISE and LOWER outputs are set to 0. The algorithm no longer takes into account the actions going in the direction of the stop.

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem arises in a calculation in floating point mode.

In any case, the error is considered to be critical. The loop output is frozen, while errors are signaled in the status words.

If T_MOTOR and T_MINI time parameters are negative, their value is forced to 0.0.

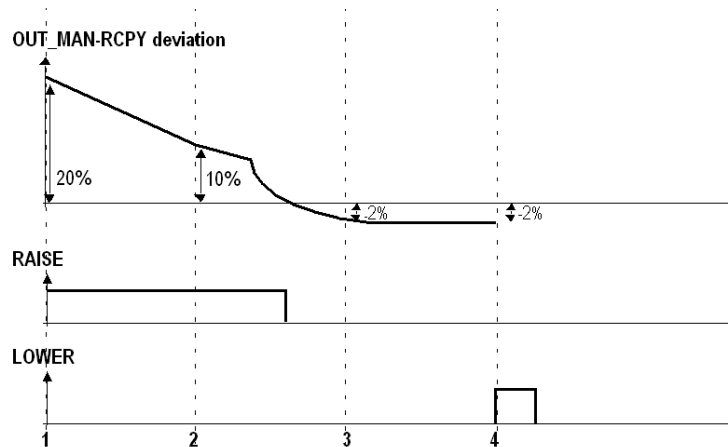
When the loop controller is in manual mode, its OUT_MAN output also controls the Servo function outputs.

Operating Process Examples of the Servo Function

Automatic Mode with Position Copy

Operation in **automatic mode with copy position**.

$T\_MOTOR = 25\text{ s}$, $T\_MINI = 1\text{ s}$ and sampling period = 4 s



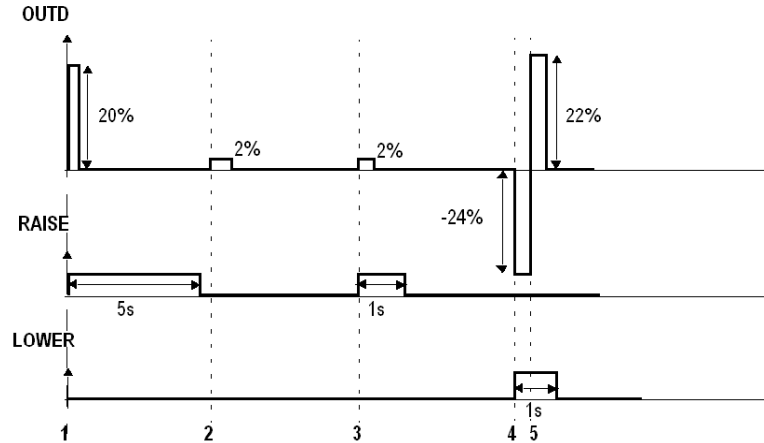
For each step, the operating principle is as follows.

Step	Operating principle
1	The deviation between OUT_MAN and RCPY is 20%. A pulse of 5 s ($25\text{ s} \times 20\%$) is generated on the RAISE output.
2	The deviation is 10%. A pulse of 2.5 s ($25\text{ s} \times 10\%$) is generated on the Raise output, without accounting for the second which remained from the previous pulse.
3	The deviation is -2%, which would correspond to a pulse of 0.5 s ($25\text{ s} \times 2\%$) on the LOWER output. As T_MINI equals 1 s, no pulse is generated (in contrast, the 0.5 s duration is stored).
4	The deviation is always -2%. The corresponding pulse (0.5 s) is added to the previously stored pulse (0.5 s) to give 1 s. As this duration is at least equal to T_MINI , the pulse is therefore applied to the LOWER output.

Automatic Mode without Position Copy

Operating in **automatic mode without position copy**.

$T\_MOTOR = 25\text{ s}$ and $T\_MINI = 1\text{ s}$



In this case, the command variation value is taken into account each time the Servo function is operated.

For each step, the operating principle is as follows.

Step	Operating principle
1	The PID output variation is +20%. A pulse of 5 s ($25\text{ s} \times 20\%$) is generated on the RAISE output.
2	The PID variation is +2%, which would correspond to a pulse of 0.5 s. As this pulse is less than $T\_MINI (=1\text{ s})$, it does not affect the outputs.
3	This second PID output variation is +2%. In order to be calculated, the function adds this variation to the previous variation (which was below the minimum value), corresponding to a global variation of +4%. A pulse of 1 s ($25\text{ s} \times 4\%$) is generated on the RAISE output.
4	The PID output variation is -24%. A pulse of 6 s ($25\text{ s} \times 24\%$) is generated on the LOWER output.
5	Before completion of the following second, a further variation of +22% on the PID output restores the system to a global variation of -2%. As this variation corresponds to a pulse of 0.5 s (below T_MINI), the function ends by carrying out the minimum pulse of 1 s.

Manual Mode without Position Copy

In the case of operating in **manual mode without copy position**, the opening or closing bit moves to 1 during a period of time which corresponding to the command difference (proportional to period of opening time).

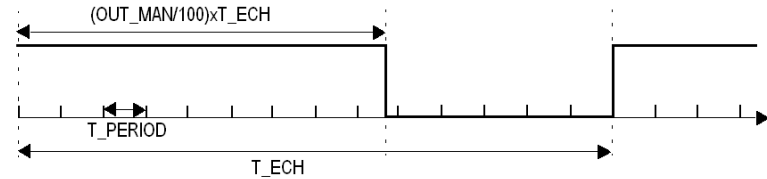
For example, $T\_MOTOR = 10\text{ s}$. If the command moves from 30% to 40% then the RAISE output = 1 during 1 s ($10\text{ s} \times (40\% - 30\%)$).

PWM

Description

This function is used to control a discrete actuator in period modulation. The logic output is set at 1 following a period proportional to the command calculated by the PID and to the given period modulation. The cyclic ratio of this type of output is defined as being the output activity rate, meaning the time ratio when the output is active throughout the whole period. The cyclic ratio (expressed in %) of a PMW output is therefore equal to the command calculated by the loop controller (expressed in %).

When the PMW function is used, the loop controller output scale must be (0.100).



Function Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Value	OUT_MAN	REAL	0.0 / 100.0	/	R
Command value (*)	OUTi	REAL	0.0 / 100.0	/	R

(*) Hot/Cold or Split Range case

Internal Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Minimum time (s)	T_MINli	REAL	0.0 / 3.4E38	0.0	R/W

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Status	STS_RAISE1	EBOOL	/	/	R

Function Period

The function period must be chosen according to the characteristics of the actuator. Therefore it is logical that the function period is equal to the loop controller sampling period because the actuator would not be in a position to take faster account of a sampling command.

Time Base

The time base used for modulation is the MAST task period or the FAST task period. In other words, the smallest available pulse lasts for the task period. However, the user is able to define a minimum pulse that is greater using the T_MINI parameter , so that the actuator constraints may be respected.

Resolution

The larger the PMW resolution function, the more precise the command carried out will be. The resolution is defined by the relationship between: sampling period / task period. A minimum of 10 is recommended.

For example, if the sampling period = 2 s (chosen according to actuator characteristics), the task period must not exceed 200 ms.

Execution Monitoring

The system indicates a processing error in the following cases:

- A non-integer input data item is detected on one of the parameters.
- A problem arises in a calculation in floating point mode.

In each case, the error is considered to be serious. The loop output is frozen, while errors are signaled in the status words.

Output Scaling

Description

This function is used to scale the output to the calculated command.

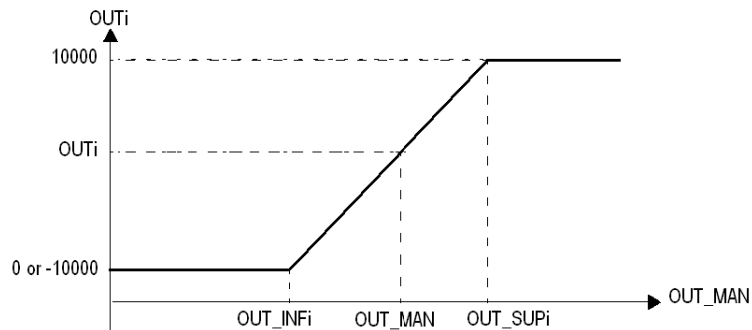
The scaling function is optional. It is used to select the scale according to specific outputs.

If this function is used, it introduces a scale factor. It performs the following calculation.

$$OUT = (OUT\_MAN - OUT\_INF) \times \frac{OUT\_MAX - OUT\_MIN}{OUT\_SUP - OUT\_INF} + OUT\_MIN$$

where: $OUT\_MAN = 0$ or -10000

and: $OUT\_MAX = 10000$



Function Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Value	OUT_MAN	REAL	-3.4E38 / 3.4E38	/	R
Command value (*)	OUTi	REAL	-3.4E38 / 3.4E38	/	R

(*) Hot/Cold or Split Range case

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower Scale	OUT_INFi	REAL	-3.4E38 / 3.4E38	0.0	R/W
Upper Scale	OUT_SUPi	REAL	-3.4E38 / 3.4E38	100.0	R/W

NOTE: In the configuration editor, the symbols OUT_INF and OUT_SUP are called Lower limit (%) and Upper limit (%).

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command value	OUT_MAN	REAL	-3.4E38 / 3.4E38	/	R

Execution Monitoring

The monitoring the parameters of this function is built into the error management of the output branch.

Output Limiter

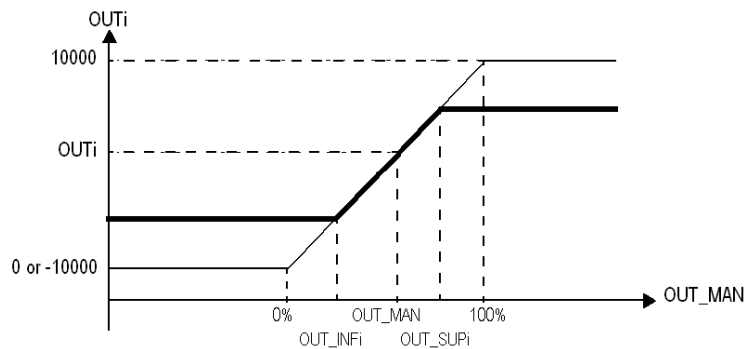
Description

This function is exclusive to the scale output function.

When this function is selected, it enables the scaling of output between the limits of the range defined by the parameters `OUT_INFi` and `OUT_SUPi`. In this case, the output scale is between 0% and 100%.

If this function is not activated, the value of the output is limited to the scale of the output defined by `OUT_INFi` and `OUT_SUPi`.

The limiter is selected by default with the values for the low limit of 0% and for the high limit of 100%.



Function Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Value	OUT_MAN	REAL	-3.4E38 / 3.4E38	/	R
Command value (*)	OUTi	REAL	-3.4E38 / 3.4E38	/	R

(*) Hot/Cold or Split Range case

Internal Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Lower Scale	OUT_INFi	REAL	-3.4E38 / 3.4E38	0.0	R/W
Upper Scale	OUT_SUPi	REAL	-3.4E38 / 3.4E38	100.0	R/W

NOTE: In the Configuration Editor, symbols OUT_INF and OUT_SUP are called Low limit (%) and High limit (%), with:

$-5\% < \text{OUT\_INFi} < 105\%$ and $-5\% < \text{OUT\_SUPi} < 105\%$.

Output Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command value	OUT_MAN	REAL	-3.4E38 / 3.4E38	/	R

Execution Monitoring

Monitoring of this function's parameters is built into the error management of the output branch.

Output Format

Description

This function is used to set the analog output value. 2 formats (or ranges) are possible:

- Unipolar: 0 / 10000 (default selection).
- Bipolar: -10000 / 10000.

Output Address Assignment

The output address is defined in the graphic section of the configuration screen. You should enter a word type variable (%QW of analog output or %MW).

Function Parameters

Input Parameters:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Command Value	OUT_MAN	REAL	-3.4E38 / 3.4E38	/	R
Command value (*)	OUTi	REAL	-3.4E38 / 3.4E38	/	R

(*) Hot/Cold or Split Range case

Internal Parameter:

Parameter	Symbol	Type	Thresholds	Default Value	R/W
Range	/	%KW bit	/	/	R/W

Control Loop Configuration



6.1 Loop and I/O Configuration

Subject of this Section

This section describes the methods that should be deployed to create a process control project with Unity Pro, as well as the configuration procedure. It also includes instructions for using the inputs/outputs associated with the loop controllers.

What Is in This Section?

This section contains the following topics:

Topic	Page
Configuring a Process Control Loop	172
Configuring the Inputs and Outputs Associated to the Process Control Loops	173

Configuring a Process Control Loop

At a Glance

A process control loop is configured according to a precise method that ensures nothing is overlooked.

Procedure

Observe the following steps for each process control loop to implement.

Step	Action
1	Give an exact definition of the process control structure to be used (single loop, process loop, cascade loop, self-selective loop).
2	Define the algorithm of the various processing branches (process value, setpoint, loop controller, etc.).
3	Select the various functions and parameters for each processing branch.
4	Enter the input and output interfaces.
5	Fix the initial values of the adjustment parameters.
6	Associate an IODDT to the selected loop.
7	If necessary, configure the exchanges for level 2 (for example: supervision).
8	Confirm the global configuration.

Configuring the Inputs and Outputs Associated to the Process Control Loops

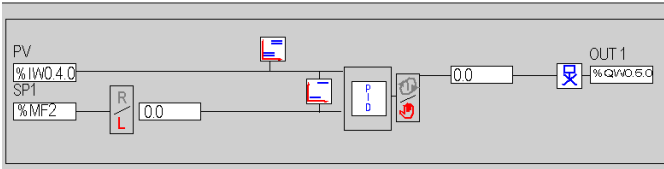
At a Glance

A process control loop must have inputs and outputs in order to operate. Inputs are used to obtain the process values, while outputs allow you to act upon the procedure to be regulated. Inputs are generally analog inputs, while outputs may be either analog or discrete (SERVO or PWM function). Inputs and outputs always belong to modules configured in the PLC.

NOTE: It is also possible to use internal words and bits which in turn are written into output words and bits, as well as internal words onto which an input value has previously been written.

Assigning Inputs and Outputs to a Process Control Loop

The following steps explain how to assign inputs and outputs to a process control loop.

Step	Action
1	Configure the required I/O modules.
2	From the configuration screen, enter addresses for the inputs and outputs into the process control loop diagram. Result: The figure below shows a sample assignment. 

Checks

- The following checks are performed when inputs and outputs are assigned:
- In order to validate the assignment of a module input or output to a block diagram, the corresponding module configuration must previously be confirmed.
 - No consistency check is performed at the level of I/O assignment to a specific task (MAST or FAST). We recommend assigning all inputs and outputs of a single loop to the same single task.
 - Should an I/O module be moved, be aware that its new address is not automatically reflected in the process control loop screens.
 - If a language object previously used by a process control loop no longer exists (if the module is eliminated, for instance), the software displays an error message during global validation.

Valid Types of Interface

The following table lists the valid language objects accepted by each type of interface.

Type of Interface	Valid Language Objects	Object Type
Standard input process value	%IW, %MW	INT
External input process value	%MF	REAL
Setpoint input Remote 1	%MF	REAL
Setpoint input Remote 2	%MF	REAL
Feed forward input	%IW, %MW	INT
Analog output	%QW, %MW	INT
Servo, PWM output	%Q, %M	EBOOL

Control Loop Adjustment

7

Subject of this Chapter

This chapter describes the adjustment methods for:

- Feed forward
- PID
- model loop controller

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
7.1	Adjust Feed Forward	176
7.2	PID Adjustment	181
7.3	Model Loop Controller Adjustment	189

7.1 Adjust Feed Forward

Subject of this Section

This section describes the procedures necessary for adjusting a Feed forward:

- gain adjustment
- Leadlag adjustment

What Is in This Section?

This section contains the following topics:

Topic	Page
Gain Adjustment	177
Leadlag Adjustment	178

Gain Adjustment

At a Glance

Debugging a Feed forward branch firstly requires a Gain adjustment The following procedure describes the procedure to be respected.

Procedure

This table describes the steps required in order to adjust the gain of a Feed forward branch.

Step	Action
1	Put the loop controller into manual mode.
2	Position T1_FF and T2_FF at zero.
3	Establish a disruption scale and stabilize the gain so that the disruption measured when Feed forward is inputted is compensated entirely. Note: The gain on the Feed forward branch occurs according to scale values.

Example of Gain Adjustment

If one says that:

- the PV variation is at -10% for a disruption variation of 5%
- the process value variation is at 7% for a command variation of 5%

then the gain deducted will be $(-10/5) / (7/5)$ which is -1.4.

For a Feed forward input between 0 and 10000 and FF_INF = 0.0 then FF_SUP = -140.0, for a command scale where OUT_INF = 0.0 and OUT_SUP=100

Leadlag Adjustment

At a Glance

Debugging a Feed forward branch requires Leadlag adjustment. The following procedure and example describe the procedure to be respected.

Procedure

This table describes the steps required in order to adjust the Leadlag of a Feed forward branch.

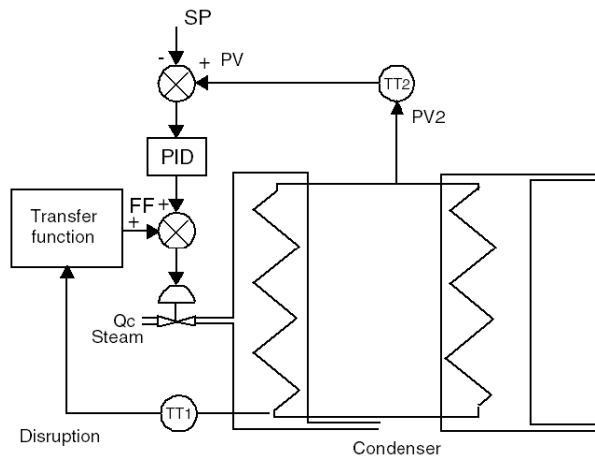
Step	Action
1	Put the loop controller into manual mode.
2	Position T1_FF at the time constant value for the process.
3	Position T2_FF at the time constant value for the disruption.
4	Carry out a disruption grade: • if the overshoot is positive, reduce T1_FF, and similarly, if the overshoot is negative, increase T1_FF • if the overshoot starts positively, increase T2_FF, and similarly, if the overshoot starts negatively, reduce T2_FF
5	Repeat step 4 until the overshoot is cancelled out.

Example of Leadlag Adjustment

An adjustment of the PV2 output temperature on the secondary circuit of the exchanger is required. A PID controls the incoming hot air valve according to PV2 and the SP setpoint. The cold water temperature acts as a measurable disruption in relation to this process control.

The Feed forward function can be used to react as soon as the cold water temperature fluctuates and not if PV2 has decreased.

Illustration of the example:



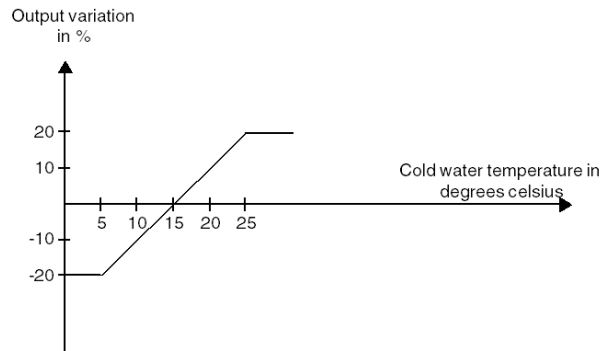
The working hypotheses are as follows:

- The output temperature of the condenser (temperature of the cold water) fluctuates between 5 and 25 degrees Celsius, with an average value of 15 degrees.
- A DT variation in this temperature affects the exchanger output temperature entirely.
- In order to compensate for an increase or decrease of 5 degrees Celsius in the exchanger output temperature, the controlling steam valve must be opened or closed by 10%.

Adjustments to the Feed forward input parameters must be such that the contribution of the cold water temperature to the valve controlling the steam flow rate is:

- zero to 15 degrees Celsius,
- in a ratio of 10% /5 degrees Celsius between 5 and 25 degrees.

The following figure illustrates the adjustment.



7.2 PID Adjustment

Subject of this Section

This section describes the methods for adjusting a PID:

- closed loop adjustment
- open loop adjustment

As well as the role and the effects of each parameter: K_p , T_i et T_d .

What Is in This Section?

This section contains the following topics:

Topic	Page
PID Parameter Adjustment Method	182
Role and Influence of the Parameters of a PID during a Loop Adjustment	185

PID Parameter Adjustment Method

Introduction

There are numerous approaches to adjusting a PID's parameters. We recommend Ziegler & Nichols' method, of which there are two variants:

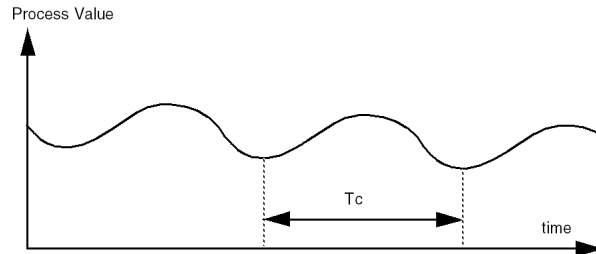
- closed loop adjustment
- open loop adjustment

Before implementing either method, one must determine the direction of the PID's action:

- if increasing the OUT output triggers an increase of the PV value, reverse the PID's direction ($K_P > 0$)
- Inversely, if increasing the OUT output provokes a decrease of the PV value, use the PID in the forward direction ($K_P < 0$)

Closed Loop Adjustment

This method consists in issuing a proportional command ($T_i = 0$, $T_d = 0$) in order to excite the process by increasing the gain until reaching oscillation, after having applied a step to the PID loop controller setpoint. Then, it's only a matter of measuring the critical gain value (K_{pc}) that triggered the unabsorbed oscillation, as well as the oscillation period (T_c) in order to derive the values ensuring optimal adjustment of the loop controller.



Depending on the loop controller type (PID or PI), coefficients are adjusted with the following values:

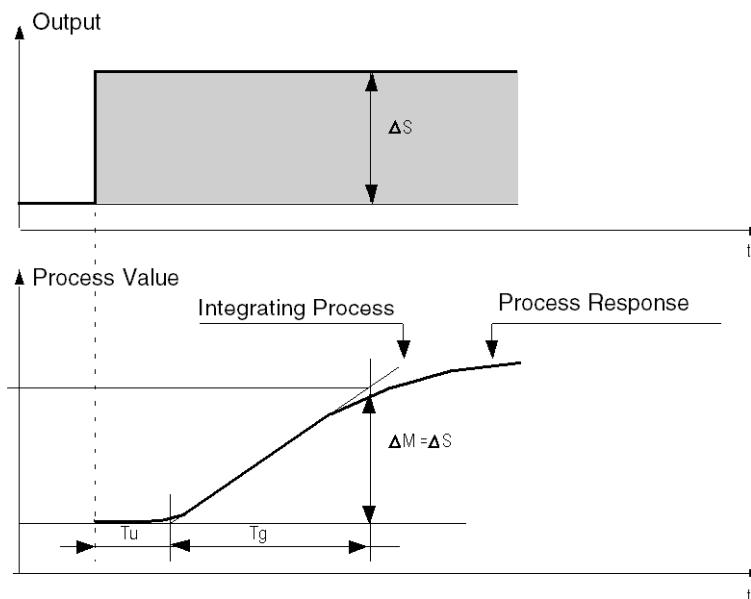
-	K_p	T_i	T_d
PID	$K_{pc}/1,7$	$T_c/2$	$T_c/8$
PI	$K_{pc}/2,22$	$0,83 \times T_c$	-

where K_p = proportional gain, T_i = integration time and T_d = derivative time.

NOTE: This adjustment method produces an extremely dynamic command which may provoke undesired overflows when changing the setpoints. If so, decrease the gain value until you obtain the proper behavior.

Open Loop Adjustment

After setting the Loop Controller in Manual mode, apply a step on its output and assimilate the beginning of the process result with an integrating process with pure delay..



The T_u time value is determined by intersecting the line representing the integrating process with the time axis. T_g time is then defined as being the time required by the controlled variable (process value) to fluctuate with the same amplitude (expressed as a scaling percentage) as the loop controller output.

Depending on the loop controller type (PID or PI), coefficients are adjusted with the following values:

-	K_p	T_i	T_d
PID	$-1.2 T_g/T_u$	$2 \times T_u$	$0.5 \times T_u$
PI	$-0.9 T_g/T_u$	$3.3 \times T_u$	-

where K_p = proportional gain, T_i = integration time and T_d = derivative time.

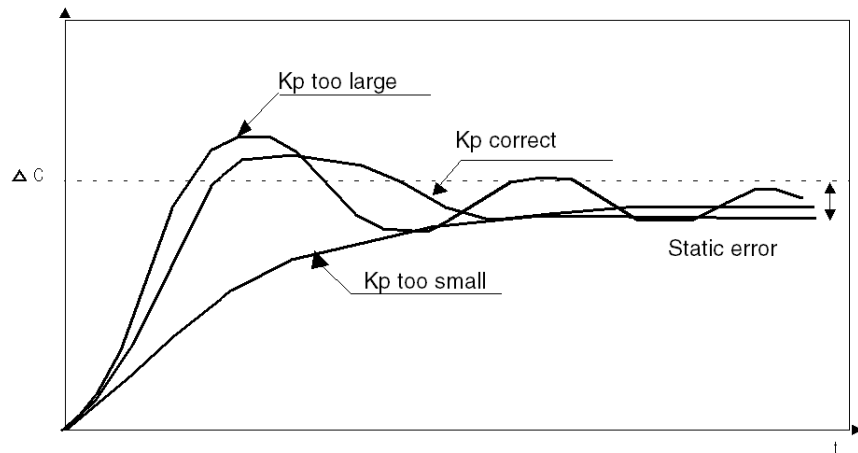
NOTE: Be mindful of measuring units. If the adjustment is done in Unity Pro, multiply the K_P value by 100.

This adjustment method also produces an extremely dynamic command which may provoke undesired overflows when changing the setpoint. If so, decrease the gain value until you observe the desired behavior. The advantage of this method lays in that it does not require any assumption as to the nature and order of the process. It applies as well to stable processes as to truly integrating processes. This method proves particularly useful in the context of slow processes (i.e. the glass industry), since the user needs only the beginning of the response to adjust the K_p , T_i and T_d coefficients.

Role and Influence of the Parameters of a PID during a Loop Adjustment

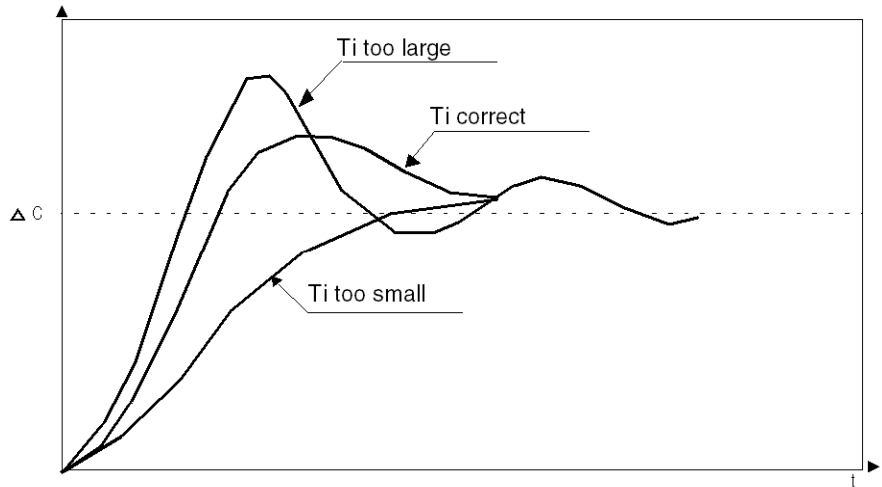
Proportional Action Influence

Proportional action enables the process response speed to be matched. The higher the gain, the more the response will accelerate and the static error will reduce (in purely proportional terms), but the greater the deterioration of stability will be. A compromise between speed and stability must be found. The following is the influence of integral action on the process response.



Influence of Integral Action

Integral action ensures that static error (the deviation between the process value and setpoint) is canceled out. The higher the integral action (T_i small), the greater the acceleration of the response and the more stability deteriorates. A compromise must be found between speed and stability. The influence of integral action on process response at a grade is as follows.

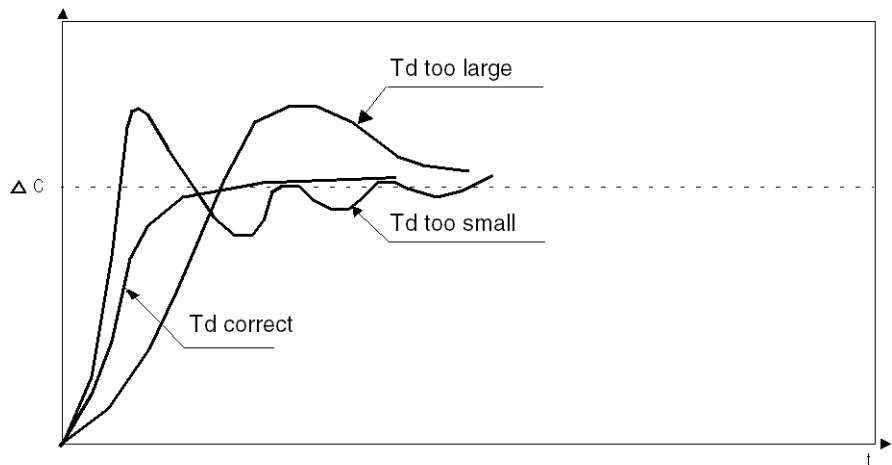


NOTE: Small T_i represents a high integral action.

K_p = proportional gain, T_i = integration time and T_D = derivative time.

Influence of Derivative Action

Derivative action is anticipatory. It adds a term that accounts for the speed variation of the difference, which enables anticipation by accelerating the process response when the deviation increases, and slowing it down when the deviation decreases. The higher the derivative action (large T_d), the more the response accelerates. Here again, a compromise between speed and stability must be found. The influence of derivative action on the process response at a division is as follows.



Limits of PID Process Control

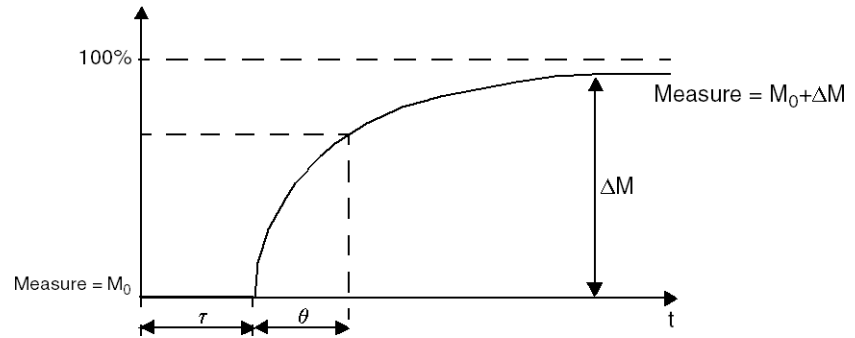
If the process is assimilated to a first command with pure time delay, of transfer function:

$$H(p) = K \frac{e^{-\tau p}}{1 + \theta p}$$

with:

τ = model delay,

θ = model time constant,



The process control performances are dependent on the ratio $\frac{\theta}{\tau}$

PID process control is suitable in the following area: $2 < \frac{\theta}{\tau} < 20$

For $\frac{\theta}{\tau} < 2$, meaning fast loops (θ small) or for processes with large delays (τ large)

the PID process control is not suitable and more advanced algorithms must be used.

For $\frac{\theta}{\tau} > 20$, a threshold process control with hysteresis is sufficient.

7.3 Model Loop Controller Adjustment

Contents of this Section

This section describes the adjustment principles of a model loop controller and develops the following points:

- static gain adjustment
- dead time or delay adjustment
- time constant adjustment

What Is in This Section?

This section contains the following topics:

Topic	Page
Steps to Follow to Adjust the Model Loop Controller	190
Instructions for Regulating Ks Static Gain	191
Instructions for Adjusting Dead Time or Delay T_DELAY	192
How to Regulate the Time Constant	194

Steps to Follow to Adjust the Model Loop Controller

Procedure

The steps in this table describe the action to be taken in order to adjust a model loop controller.

Step	Action
1	Use a graphic method based on an indexed response, such as the Broida method. This directly provides the parameters for a first order model with pure delay, in order to identify the process model.
2	Enhance the adjustment by running the IMC loop controller in automatic mode.
3	To check whether the model is adapted to the process, set CL_PERF to 1.0 (time constant of closed loop = time constant of open loop).
4	Proceed to the point of operation and run the loop controller in automatic mode.
5	Carry out a setpoint step function ΔC . Result: If the model parameters are correct, the measurement should rejoin the setpoint without overflow and the OUT_MAN command signal should be almost one division. If this is not the case, a correction must be made, which means: ● adapting the static gain ● adapting the dead time ● adapting the time constant

Instructions for Regulating Ks Static Gain

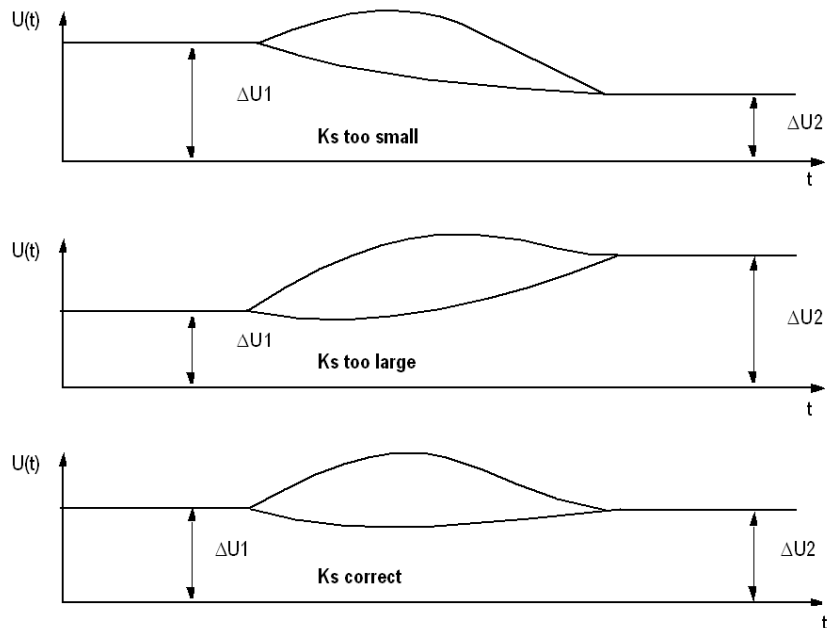
At a Glance

If the static gain is correct, at the time of the setpoint scale the size of variation $\Delta U1$ must be equal to $\Delta U2$. If this is not the case, correct the gain by applying the formula:

correct $K_s = K_s \text{ of test} \times \Delta U1 / \Delta U2$

Illustration of the Adjustment

The following figure illustrates the static gain adjustment.



Instructions for Adjusting Dead Time or Delay T_DELAY

At a Glance

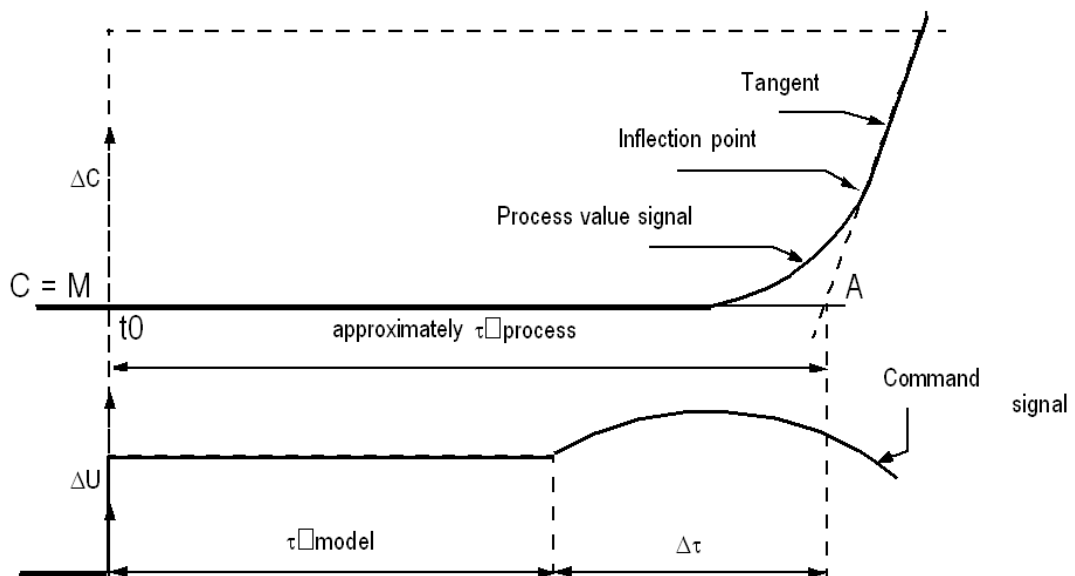
Observe the adjustment unit measurement and control signals for a recording. If we call τ the model delay, two illustrations appear:

- τ model is lesser than τ process
- τ model is greater than τ process

NOTE: Gain and delay adjustments can be carried out during the same test.

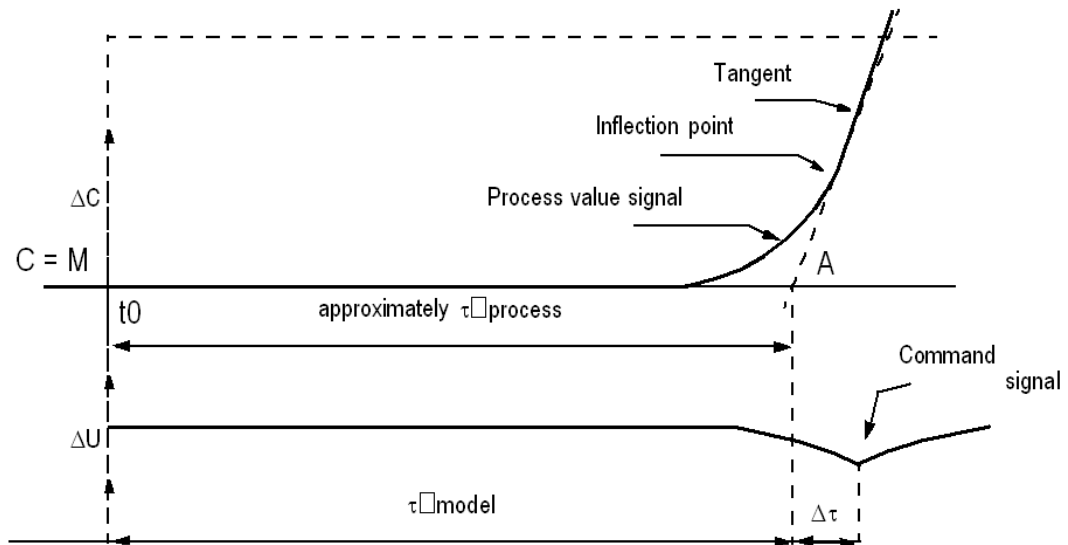
τ Model Lower Than τ Process

The following figure describes this scenario, T_DELAY takes on the value of A.



τ Model Greater Than τ Process

The following figure describes this scenario, T_{DELAY} takes on the value of A .



How to Regulate the Time Constant

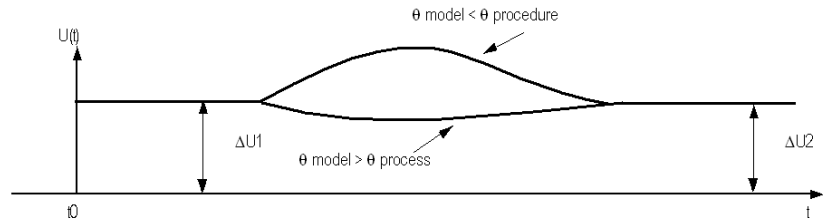
At a Glance

Having regulated dead time and static gain, the model time constant must be successively regulated by observing the command signal $U(t)$ record.

If we call θ the model delay, two illustrations appear:

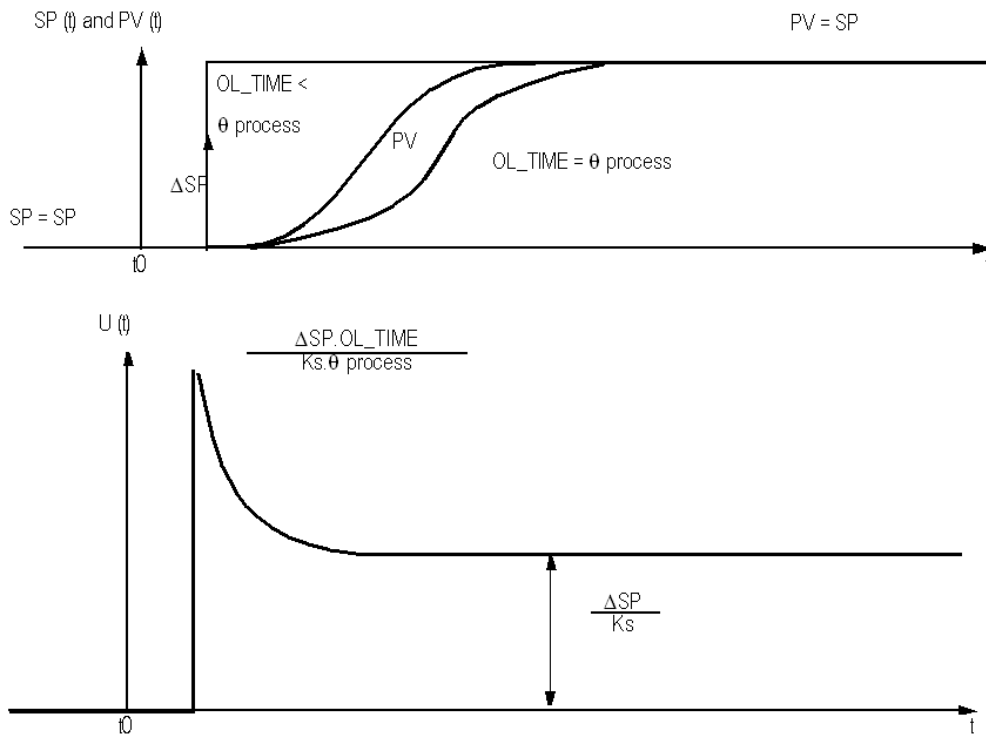
- θ model is lesser than θ practice
- θ model is greater than θ practice

This figure illustrates the two cases.



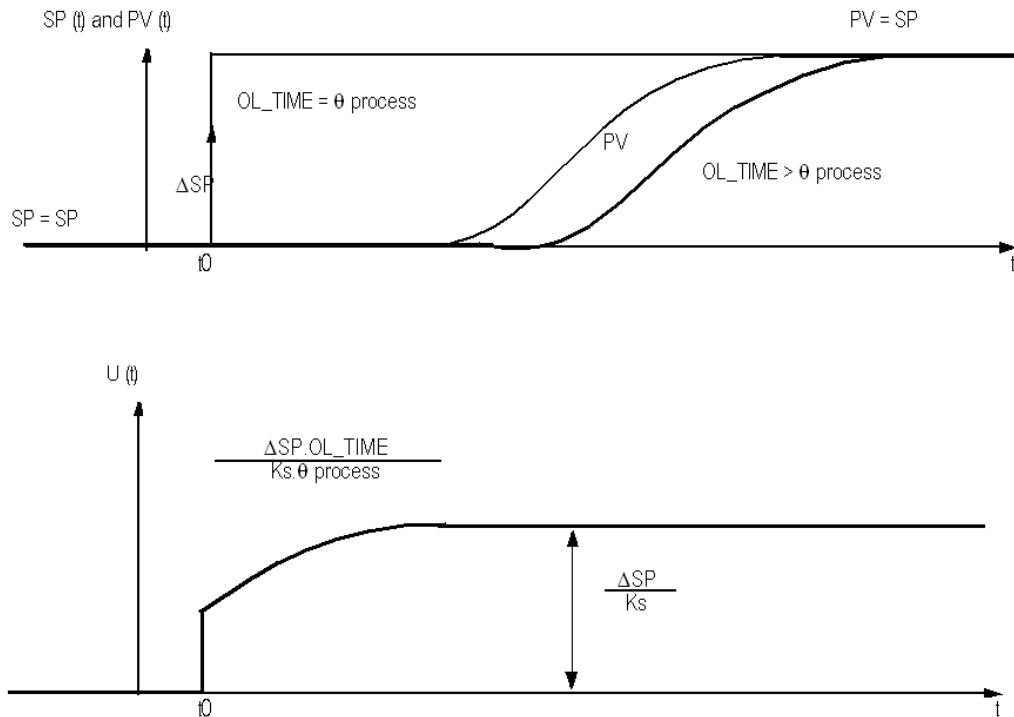
Signal Speed When $OL\_TIME < \theta$ Process

The following figure describes the signal speed.



Signal Speed During $OL\_TIME > \theta$ Process

The following figure describes the signal speed.



Closed Loop Time Constant Selection

Having determined the model to be used, the closed loop time constant selection remains to be made. Its value is dependent on the response speed of the required closed loop.

For processes that correspond to a first order model with delay, by choosing a time constant CL_PERF relation between 1.05 and 1.15, the system response is improved without running the risk of destabilising the process.

$CL\_PERF = OL\_TIME / \text{required closed loop time constant}$.

All CL_PERF increases correspond to an increase in response speed (it is a more important action of the adjustment unit), but also to an increased sensitivity to modeling errors.

Debugging a Process Control Loop

8

Subject of this Chapter

This chapter describes the process control loop debugging screens as well as the associated operational processes.

What Is in This Chapter?

This chapter contains the following topics:

Topic	Page
Description of the Debug Screen	198
Modifying Parameters for Each Loop	200
Functional Modification of Each Loop	201
Debugging the Setpoint Programmer	202
Data Storage	204

Description of the Debug Screen

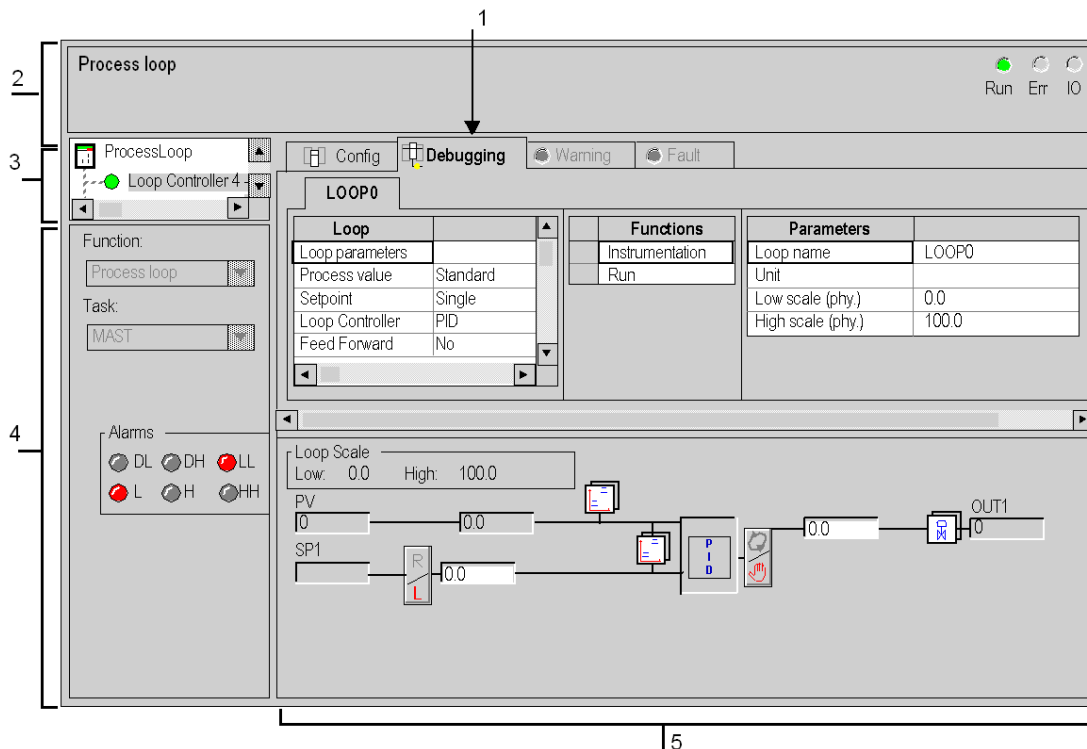
At a Glance

In Online mode, process control debugging screens are used to:

- display and animate the loop diagram
- display process alarms and channel faults
- modify adjustment parameters for each function
- simulate input interface values
- add, delete, and replace calculation functions
- modify configuration parameters of each PID reverse/forward mode-type function
- modify loop controller operating modes

Illustration

The following figure shows a debugging screen.



Description

The following table describes the main components of the Process Control Debugging screen, as well as their respective function.

Address	Component	Function
1	Tabs	The foremost tab indicates the current mode (Debug, in this instance). Select each mode by clicking on the corresponding tab. The following modes are available: ● Configuration, ● Debugging, accessible only in Online mode, ● WARNING displays warning messages pertaining to the process control loop status, ● Diagnostics, (Default) accessible only in Online mode.
2	Module area	Displays the abridged module indicator. There are three indicators in the same zone that provide the status of the PLC in online: ● RUN indicates the status of the PLC, ● ERR indicates a fault on the processor or embedded devices. ● I/O indicates a fault from a module, channel or configuration.
3	Channel area	Is used: ● By clicking on the reference number, to display the tabs: ● Description which gives the characteristics of the device. ● I/O Objects (see <i>Unity Pro, Operating Modes</i>) which is used to presymbolize the input/output objects. ● Fault which shows the device faults (in online mode). ● To select the loop controller, ● To display the Symbol, name of the channel defined by the user (using the variable editor).
4	Global Parameters area	Lists the general settings (or parameters) associated to the channel : ● Function: Process Control function associated to the loop. This parameter is invariable. ● Task: defines the MAST or FAST task through which the channel's implicit exchange objects will be exchanged. This parameter is invariable. The Alarms area. If the alarm functions are defined, displays all loop-associated alarms.
5	Debug area	Allows viewing of configured functions and block diagram. This area comprises two sections: ● one or several tabs, showing those functions already configured. Their associated parameter values are animated. They may be modified in Online mode only, ● The block diagram shows the intermediary calculation values (loop controller input measurement, for example). Click once on the mouse button to make changes to Auto/Manu and Remote/Local settings. Associated command values or setpoint values are captured directly in the input fields. Grayed-out input fields are inactive.

Modifying Parameters for Each Loop

Fundamentals

Adjustment parameters may be modified in online or offline mode from Unity Pro software:

- task screens
- data editor
- runtime screens
- variable tables
- UNITE server
- ...

Modifications do not require global re-configuration. The save mechanism (initial and current parameters) is applied to the parameters associated with each process control loop.

Functional Modification of Each Loop

At a Glance

Functions may be added to or deleted from the process control loops in **online mode**.

Fundamentals

The PLC may be in RUN mode during modifications to the process control loop functions.

WARNING

UNINTENDED EQUIPMENT OPERATION

Perform a channel reconfiguration after:

- modification of process loop functions
- modification of configuration parameters
(ex: extrapolating the generator function from the function, limiter looping)

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

Once the modifications are made, the loop starts in a determined state:

- If modifications are made at the Measurement, Setpoint, and Feed forward branch level, the loop performs a warm start.
- If modifications are made at the corrector or output level, the loop restarts using the initial operating modes, as previously defined during configuration.

NOTE: any addition or modification of I/O addresses or memory words is prohibited in Online mode. Modifying some branches—such as replacing a simple setpoint by a ratio setpoint—is also prohibited.

NOTE: the totaling function may not be added in Online mode. Its output is an %MF address.

It is not necessary to switch to Configuration mode to edit the initial adjustment parameter values Kp, Ti, and Td.

Operating mode:

- In Debugging mode, any change to the adjustment parameters from the task screen affects the current and initial values of said parameters.
- Only in the case of the **3 simple loops** loop controller, any modification to the initial value will be ignored if the same change is not applied to the configuration parameter also (for example, when adding a function).

Debugging the Setpoint Programmer

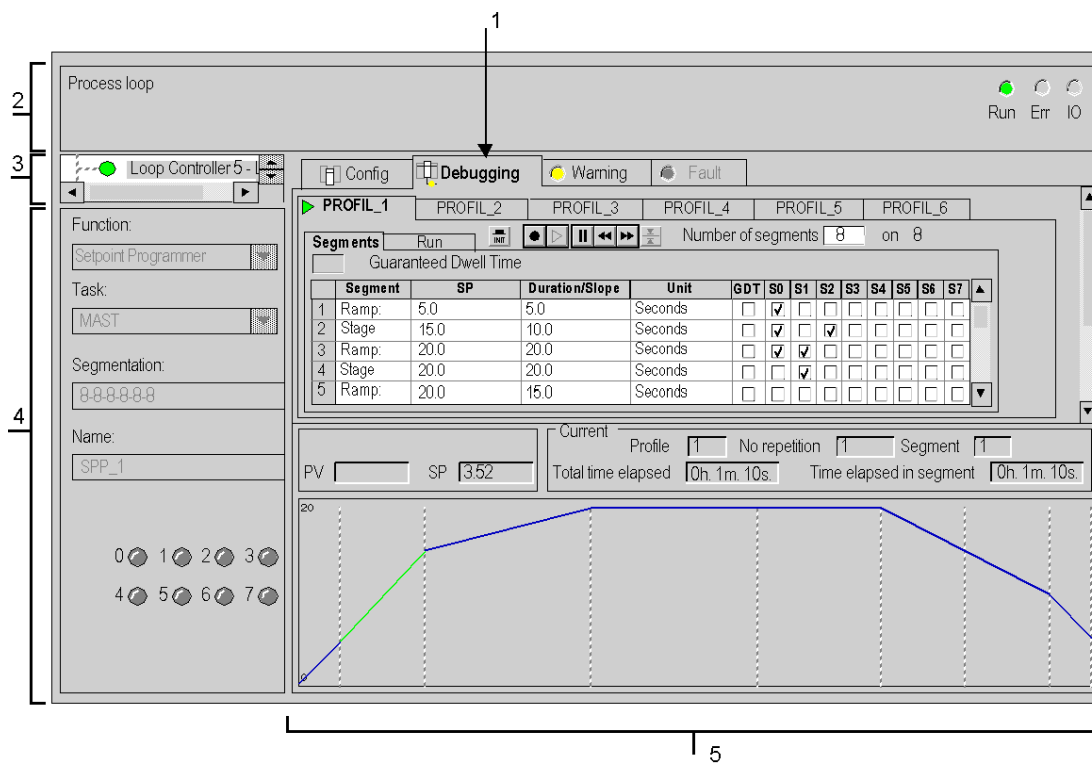
At a Glance

Setpoint programmers have their own debugging screen. When in Online mode, they have access to all available reconfiguration and data saving services.

NOTE: Any reconfiguration in Online mode immediately prevents the setpoint programmer from functioning.

Illustration

The following diagram shows a setpoint programmer in Debugging mode.



Operating Principle

Profile execution is displayed dynamically. The following information is provided:

- Number of current segment (SEG_OUT),
- Number of current iteration (CUR_ITER),
- Execution time of the current segment (TIME_SEG),
- Total time (TIME_TOTAL).

Notes on Operation:

- The TIME_SEG and TIME_TOTAL times change even if the profile is frozen.
- The state of monitoring outputs is directly displayed in the channel area.
- You may control each profile directly from the Command button located in the tab.
- When the profile is frozen, the SP (%MFr.m.c.20) setpoint values and STR0 to STR7 (%MW.r.m.c.3.0 to %MW.r.m.c.3.7) control outputs are not updated during execution of NEXT and BACK commands. They are refreshed as soon as profile unfreezes.
- Neither the target setpoint value (SPi) nor the duration of the current segment may be modified.

Data Storage

At a Glance

Two data save solutions are provided:

- saving adjustment parameters
- the backup application

Saving Adjustment Parameters

- All modifications of the adjustment parameters from the Unity Pro process control screens update the current and initial values.
- Modifying an adjustment parameter from the application or from an animation table affects the current value but does not modify the initial value. The SAVE_PARAM explicit instruction must be used to save this new value.
- On cold restart (%S0) and project loading, the current parameters are replaced by the initial parameters.

Backup Application

Premium PLCs provide the option of saving the project (program and data memory) on a Backup card. The content of this card can reload the RAM memory.

NOTE: A backup card cannot be used if the Premium PLC is already using a PCMCIA card.

Operating Modes



Subject of this Chapter

This chapter describes the process control operating modes: PLC and control loop operating modes.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
9.1	Operating process control channels	206
9.2	Handling process control operations according to PLC operating mode	210
9.3	Common operating modes for process control loops	212
9.4	Operating Modes for Each Process Control Loop	218

9.1 Operating process control channels

Subject of this Section

This section describes how process control handling is distributed and synchronized for the best possible results in processor loading.

What Is in This Section?

This section contains the following topics:

Topic	Page
Distribution of Process Control Handling	207
Synchronizing Pre- and Post- Processing	208
Multitasking Application	209

Distribution of Process Control Handling

MAST Task and Loop Sampling

The processing task period and the control loop sampling periods are different.

By default, the MAST task lasts 20 ms, while the process control channel sampling times are 300 ms.

Optimizing Processor Loading

To optimize processor loading, the periodic processing of different process control channels is distributed over several task cycles.

Distribution of the processing in this way is an entirely automatic operation and does not require programming.

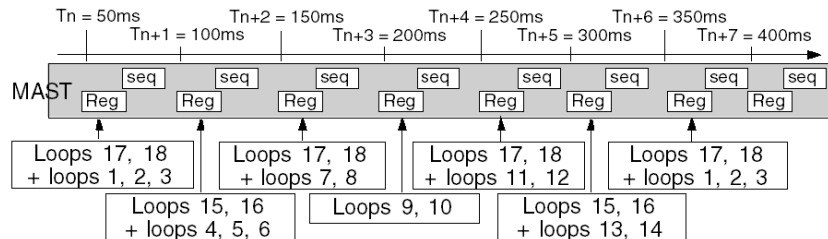
The order of loop distribution over the task cycles follows the order in which the loops were created.

Example

Let there be 18 loops configured in the following manner:

- 14 loops configured to 300 ms (loops 1 to 14)
- 2 loops configured to 200 ms (loops 15 and 16)
- 2 loops configured to 100 ms (loops 17 and 18)

Process distribution:



Reg: Processing of control loops.

Seq: Processing of user application.

NOTE: Control loops are processed (executed) during "Reg" time.

Synchronizing Pre- and Post- Processing

Trigger Bits

If you wish to closely synchronize the sequential processing with the periodical execution of each control loop, for each loop with 2 bits contained in the status words there is:

- STS_TOP_NEXT_CYCLE: pre-processing trigger bit.
- STS_TOP_CUR_CYCLE: post-processing trigger bit.

These two bits can act as enabling conditions for processing operations written in structured text or ladder language.

Example

This synchronization can be useful for Stop/Run operating modes, or for variation or compensation calculations.

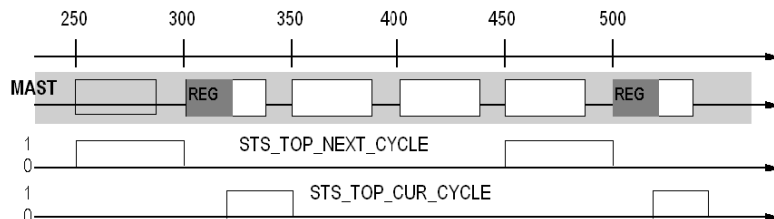
The pre-processing status bit is on 1 during the task cycle before the control processing cycle.

The post-processing status bit is on 1 during the entire task cycle which follows the control handling cycle.

To correctly synchronize the sequential handling with the process control calculations, processing operations must be integrated within the same task.

In manual or tracking mode, the command is generated in all processing cycles. These synchronizing bits are always on 1 while in these modes.

For example, Loop on 200 ms with a MAST task on 50 ms:



Multitasking Application

Optimum Functioning Level

For the optimum and determinist operating level, in a given control channel you are advised to assign to the same task:

- inputs/outputs channels
- sequential pre- and post processing
- sequential processing managing the PLC's operating modes

9.2 Handling process control operations according to PLC operating mode

Handling Process Control Operations According to the PLC Operating Mode

At a Glance

PLC behavior can be altered either by user intervention or by a fault. In this case, the process control channels follow a preset downgraded functioning mode. The following PLC operating modes can alter process control operations.

PLC Power-Up

On PLC power-up, the system searches for a valid application in the user memory space.

If the application is valid, the system changes to a state of configuration and each process control channel is called. The channel context is then set on initial values which can be used on application.

If the application is not valid, the system changes to standby mode to wait for a reconfiguration request.

Processor in RUN Mode

In RUN mode, the processor carries out successively on each cycle:

- the reading of input channels
- the execution of setpoint programs
- the execution of process control loops
- the processing of sequential programming
- the writing of outputs

All process control channels are called in each task cycle:

- Calculation of the process value, (PV), calculation of the Feed Forward (OUT_FF), alarm management, operating modes, setpoint programmers and generation of Manu or Tracking mode commands are all carried out on each cycle.
- Loop commands in Auto mode are generated and setpoints are calculated during the sampling period.

Switch to STOP Mode

The process control channels do not directly initiate the STOP mode for the processor or the task. Switching to STOP mode means that all the current functions in progress stop.

The process control channels are no longer carried out. They refuse all commands (in Auto or Manu, etc.). The calculation results remain in their current state. Physical outputs take on the fallback value set in the configuration.

Inputs are always refreshed. Parameters may therefore be altered. The validity check is carried out on the next start-up.

Cold Restart

A cold restart can be initiated in several ways:

- Changing the cartridge (cold start)
- A reconfiguration (by loading a program, transferring a new application, etc.)
- A new configuration

A cold restart is indicated by the %S0 system bit.

The process control channels monitor their configuration, and initialize their parameters and status as from the first cycle. Algorithm processing is executed from the second cycle.

All commands generated in a sequential operation from the first cycle are accepted, except for orders to track or autotune on the loop controller. The command is refused.

Warm Restart

A warm restart happens on a power cut followed by a power restart.

At the moment of the power cut, parameters are saved. The system and application contexts (e.g. application data, operating modes) are retained.

Any autotuning in progress will be aborted. The process control channels are carried out as from the first cycle.

9.3 Common operating modes for process control loops

Subject of this Section

This Section describes the operating modes which are common to all process control loops: loop control in manual mode, starting autotuning, execution in Tracking mode, etc.

What Is in This Section?

This section contains the following topics:

Topic	Page
Loop Execution in Manual and Automatic Modes	213
Autotuning and Tracking Mode	214
Auto / Manual and Manual / Auto Switching	216
Behavior of Loops During I/O Errors	217

Loop Execution in Manual and Automatic Modes

Control in Manual Mode

Manual mode enables the user to apply a value directly to the loop controller output.

Select this mode from the Unity Pro Debugging screens.

Switching to Manual mode simply requires sending a single command. Once this command is acknowledged, Manual mode is indicated by the STS_AUTO_MANU status bit.

Control of the loop or loop controller command may then be initiated. If the command is a numeric value, it is subject to high and low thresholds, as well as speed limitation. The output is processed each time the task cycles.

Controlling Servo Output in Manual Mode, Without Copying Positions

Manual control is always done on the OUT_MAN variable. This variable ranges from 0 to 100. However, since OUT_MAN is not directly linked to the actual position of the actuator, it's essential to still be able to open and close the actuator, even if OUT_MAN reaches its upper or lower threshold. To do so, it is possible to assign OUT_MAN a value beyond its normal range: OUT_MAN will be limited, yet the change affecting the calculated command will be taken into account by the Servo function.

For example: OUT_MAN = 100.0; actuator is open at 50%. To attain a 70% opening, enter OUT_MAN = 120.0. OUT_MAN then takes on the limited value of 100.00.

Execution in Automatic Mode

In Automatic mode, the command value is calculated by the loop controller based on the setpoint and PV values.

Switching to Automatic mode is done from the Unity Pro screens. You may also switch modes by sending a single command. Once the command is acknowledged, the system uses the STS_AUTO_MANU status bit to confirm the switch to Automatic mode. The output is processed at each cycle of the sampling.

Autotuning and Tracking Mode

Autotuning Execution

Before autotuning, you must first enter step function duration, performance and amplitude of the desired command.

If these parameters have too small or too large values, autotuning will not be executed.

The loop controller can be in automatic or manual mode before autotuning is executed. Sending a command starts the autotuning process. During the autotuning process (2.5 times the step function period), the function controls the loop controller output. This output cannot be modified.

The autotuning function automatically sets loop controller coefficients. The diagnostics word shows any possible faults detected during autotuning. When autotuning is complete, the loop controller assumes its operating mode prior to the autotuning operation.

If the loop controller is in automatic mode, it uses the new parameters. The **Previous adjustment** command sets to the previous parameters.

Tracking Mode Execution

This operating mode is used to force a process control loop's digital outputs. This is often used when closing an open loop to prevent a bump occurring in the command units.

Tracking mode is also used in back-up architectures composed of one active and one passive PLC. Here it makes the passive PLC outputs identical to the active PLC outputs.

Tracking mode uses a parameter (%MF address) and a command (send command):

- Switching to Tracking mode is done by sending a command. The command is refused if the address containing the Tracking value is not entered.
- If the order is sent, the process control loop output value is replaced by the Tracking output value. The internal variables are regularly initialized with the output value.
- If the order is given to abort Tracking mode, the loop controller returns to its previous operating mode without output bump.

Tracking mode has higher priority than automatic, manual and autotuning modes.

When the Tracking mode is activated or a Tracking mode request is launched, it is impossible to execute any command (automatic, manual, and autotuning).

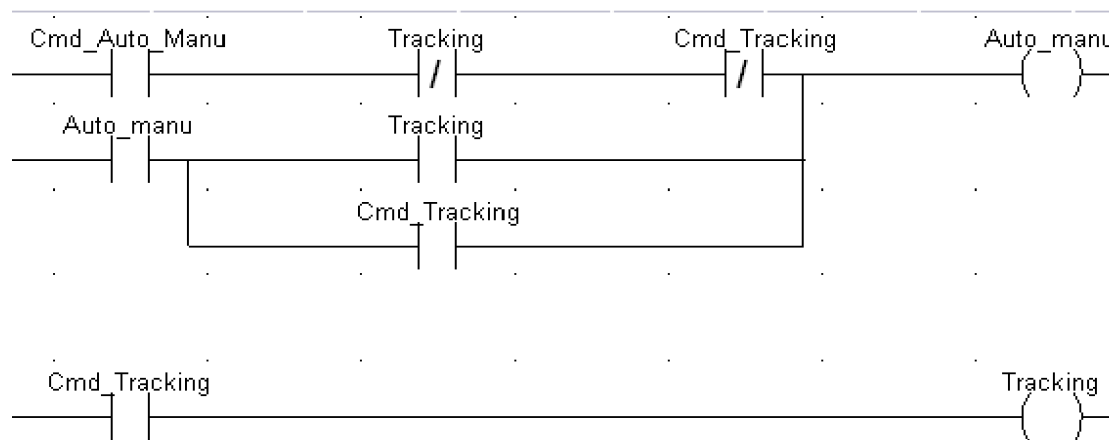
You have to ensure the right bit programming maintaining the commands bits while Tracking mode is On.

Tracking mode is not available for all loop controllers (characteristics none). For example, the ON OFF loop controller does not have this mode.

Tracking mode is global on cascade and auto selector loops.

Example

This example shows, in a Ladder section, how to manage the Auto/Manual mode and the Tracking mode.



Auto / Manual and Manual / Auto Switching

Auto / Manual Switching

The manual command is continuously updated: it tracks the command output. On Auto / Manual switching, the first manual value given is the last value calculated by the loop controller; which prevents bumps.

Manual / Auto Switching

Manual / Auto switching (for any loop controller other than ON OFF) is bumpless on the output command, on condition that the BUMPLESS option is selected on configuration. For the PID, two scenarios can be distinguished:

- **Scenario: PID with integral action ($T_i \neq 0$):**
The PID algorithm in incremental form guarantees there will be no bumps during the transition from Manual to Auto. In this case, the PID algorithm always tracks the applied actual output.
- **Scenario: PID without integral action ($T_i = 0$):**
It is possible to obtain a bumpless Manual/Auto transition if the mode is configured as **bumpless** in the PID function parameters (if the PID has an integral action, this configuration has no effect).
The OUTBIAS manual integral parameter is calculated during the switch so that deviation between the actual output and the output calculated by the PID algorithm in absolute format is accepted.
If **bumpless** mode is not selected, OUTBIAS is not recalculated on switch.

Behavior of Loops During I/O Errors

Description

By design, the process control loops do not accept possible errors that could occur on I/O cards. A loop's operating mode can be made conditional during an I/O error, through the sequential program. Application monitoring, associated module bits and diagnostic words can generate the appropriate command for this loop.

9.4 Operating Modes for Each Process Control Loop

Subject of this Section

This section describes operating modes for each of the process control loops: process loop, single loop, cascade loop and auto selector loop.

What Is in This Section?

This section contains the following topics:

Topic	Page
Process Loop Operating Modes	219
Single Loop Operating Modes (3 Single Loops)	220
Cascade Loop Operating Modes	221
Auto Selector Loop Operating Modes	224

Process Loop Operating Modes

At a Glance

Depending on loop controller type, this loop can have 2, 3 or 4 different operating modes (eg. automatic, manual, autotuning, Tracking modes).

Autotuning can be launched if the loop controller is in automatic or manual mode.

Switching to Tracking mode has priority and will possibly cancel the autotuning in progress.

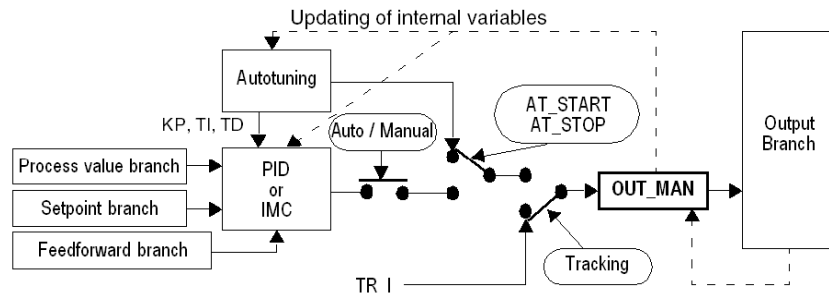
Initial Operating Mode

The initial operating mode of the loop can be configured on cold start. You can specify:

- Whether the setpoint is local or remote, as well as the initial local setpoint value.
- Whether the loop controller starts in automatic or manual mode. The initial manual value, if the loop controller is not an ON OFF loop controller

Process Loop With a PID Loop Controller

The following diagram illustrates a process loop with a PID loop controller:



If the output branch sends dotted lines back to OUT_MAN, the limits have been accepted.

If the loop controller is Split Range or Heat/Cool type, there are 2 output branches. In this instance, autotuning and Servo type output without copy are not available.

If the loop controller is an ON OFF type in 2 or 3 states, there is no Feed forward branch or output branch.

Single Loop Operating Modes (3 Single Loops)

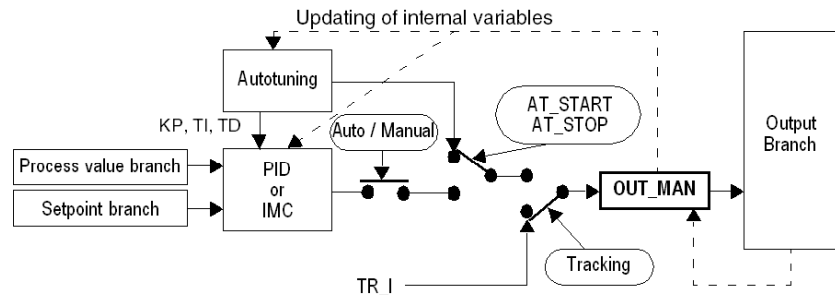
At a Glance

The 3 single loops are independent and are represented by a diagram equivalent to that of the process loop, with the exception of:

- The Feed forward branch does not exist.
- A Split Range or Heat/Cool loop controller cannot be configured.
- The Process Value and Setpoint branches are simplified.

Single Loop With a PID Loop Controller

The following diagram illustrates a single loop with a PID loop controller.



Only one autotuning operation can be launched at a time on the 3 loops from the process control channel. If a second autotuning operation is requested, it is refused.

Cascade Loop Operating Modes

At a Glance

All switching is done without any bump on the loop controller outputs. Managing the slave loop operating mode management is identical to managing the process loop operating mode: this loop behaves as if it were alone.

There are, however, mechanisms specific to the master loop:

- Manual mode and Auto/Manu switching are identical to those of the process loop.
- If the loop controller runs in the Automatic mode (by default), two different situations may be observed:
 - The slave loop controller runs in Automatic mode and uses the Remote setpoint. The cascade is closed in this case, which means the master loop truly runs in Automatic mode.
 - The slave loop controller runs in Automatic mode and uses the local setpoint. Auto-tuning is running. The loop controller runs in Tracking mode. In this case, the master loop controller runs in Tracking mode.

Closing the Cascade

The point of this procedure is to avoid any bump when cascade closes. You may encounter several situations:

- If the slave loop controller uses the local setpoint, the master loop controller follows the slave's local setpoint.
- However, should the slave loop controller have an integral action, the master loop controller follows the slave loop process value.
- Finally, should the slave loop controller not have an integral action, the slave loop controller must be either a P or a PD. The master loop output is calculated and optimized to prevent any bump when cascade close (depending on the controller output and its parameters).

Cold Start

When system is cold-started, the master loop controller always starts in Automatic mode. However, it is possible to re-configure the initial operating mode for the slave loop controller and setpoint type for each loop (Remote/Local).

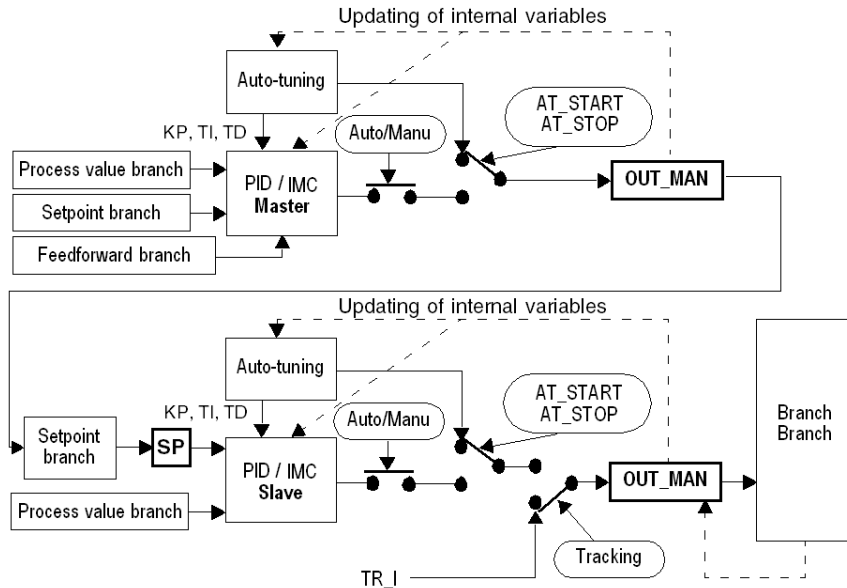
Preventing Further Changes to Outputs

The Master loop characteristically presents one additional functionality: when the master is in Automatic mode and the slave output begins to saturate, the master loop "freezes" (prevents) any changes to the master output, in the direction which saturates the slave. This function limits saturation of the master's integral action. Therefore, it is enabled only if the master loop controller has an integral action.

For instance, when the slave loop controller runs in Automatic mode, it is configured in reverse action and its output reaches the upper limit. In order to dissociate the output from its limit by acting on the setpoint, you must lower said setpoint. The master output is now slanted in a rising, growing direction.

Cascade Loop with a PID Loop Controller

The following schema shows a cascade loop with a PID loop controller.



A cascade loop generally consists of 2 process loops with some restrictions and additional functions.

The OUT_MAN master loop output is the remote setpoint for the slave loop setpoint branch. The OUT_MAN output is therefore expressed in the slave loop scale. It is subject to the slave loop Setpoint branch level limit.

The slave loop controller may be of one of the two following types: Split Range or Hot/Cold.

The Auto/Manual operating mode and the value for the master loop manual command are not accessible from the Unity Pro process control screens. They may however be accessed via the user program.

Restrictions

Restrictions are as follows:

- No totaling on the master loop process value branch.
- No Feed forward branch on slave loop.
- No ON/OFF loop controller on any loop.
- The slave loop Setpoint branch is a single branch, with no scaling.
- Only one model loop controller may be configured on the master or slave loop.

Auto Selector Loop Operating Modes

At a Glance

This loop causes 2 loop controllers to work on the same output. Each loop controller produces an action and a (minimum or maximum) comparator selects the action to be applied. The auto selector loop consists of a main loop made from a process loop and a secondary loop made from a single loop. The two loops share one single output branch.

For example, the auto selector loop is used to implement a restricted process control. The main loop is used to monitor the main value and the secondary loop is used to prevent an auxiliary value from exceeding a limit (or restriction), specified by this loop's setpoint.

Inhibiting One of the Loops

One of the following commands can be used to inhibit a loop to have either a process loop, or a single loop: direct output 1 or direct output 2.

The 2 loop controllers have the same sampling period.

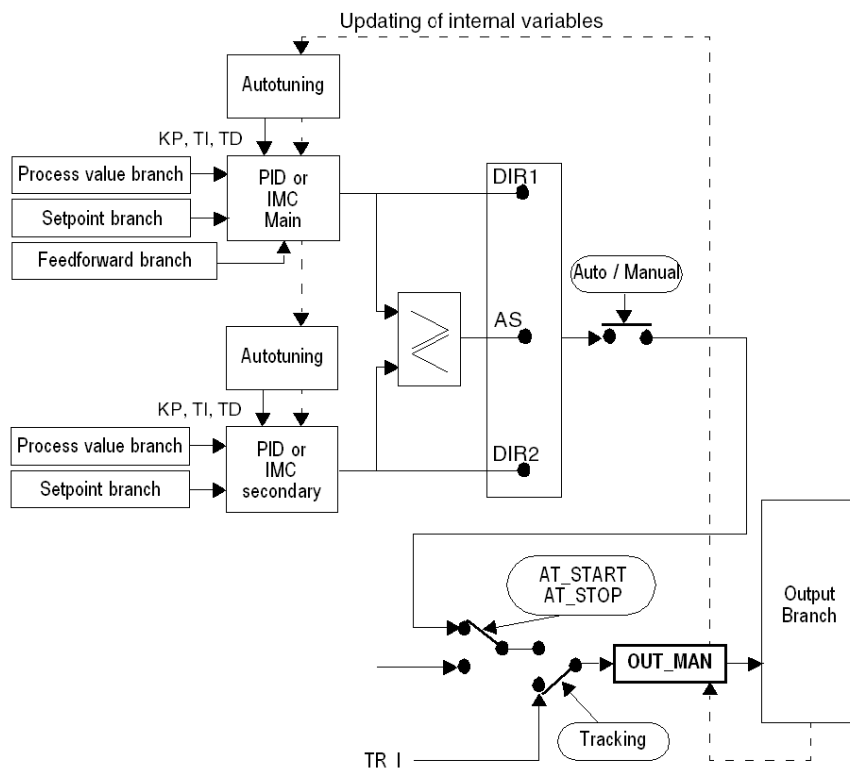
Auto Selector Loop Configuration

An auto selector loop can be configured in 2 ways:

- Case 1: one single Auto/Manual at output branch level after selector. The OUT_MAN manual command value is thus directly applied on the loop output.
- Case 2: an Auto/Manual on the output of each loop controller. Each loop controller's operating mode is then independent. Each loop controller's output value can be manually set upstream of the selector.

Case 1: One Single Auto/Manual at Output Branch Level

The following schema shows an auto selector loop with only one Auto/Manual at output branch level.



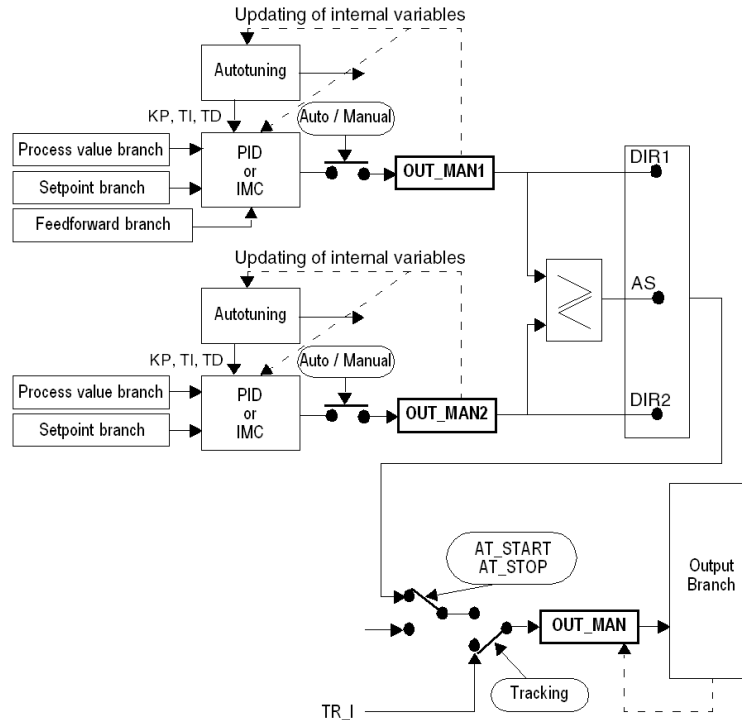
In this case, the 2 loop controllers are always in automatic mode and track the applied OUT_MAN command. When the loop is in automatic mode, the output of these loop controllers is accepted. If this is not the case, the output is not accepted.

As the loop controllers are trackers of the actual output, there is no risk of bump on switching if the loop controller's integral action is used.

The initial loop operating mode and initial setpoint type (Remote/Local) for each loop controller can be configured.

Case 2: An Auto/Manual on the Output of Each Loop Controller

The following schema shows an auto selector loop with one Auto/Manual on each loop controller's output.



In this case, in manual mode, you do not act directly on the actuator command, but at the level of each loop controller's output (OUT_MAN1 and OUT_MAN2).

As long as at least one loop controller is in automatic mode, output selection is made during the loop sampling period. If both loop controllers are in manual mode, selection is made on each task cycle.

Both loop controllers permanently track the OUT_MAN actual output. In automatic mode, if they are using an integral action, they take the previous OUT_MAN output value into account. This signifies that on Manual/Auto switching, the loop controller will not use its last manual value, but the last OUT_MAN actual output value.

The initial operating mode and initial setpoint type (Remote/Local) for each loop controller can be configured.

Starting autotuning forces the auto selector to the direct position of the autotuned loop. At the end of autotuning, the auto selector must be reset to the required position, if it is different from the one imposed.

Process Control Language Objects and IODDT

10

In this Chapter

This Chapter describes the various language objects and IODDTs associated with process control inputs/outputs and channel parameters.

What Is in This Chapter?

This chapter contains the following sections:

Section	Topic	Page
10.1	Process Control Language and IODDT Objects	228
10.2	Language Objects Associated with the Process Control Channels	229
10.3	Language Objects Associated with the Process Loop	238
10.4	Language Objects Associated with the 3 Single Loops	253
10.5	Language Objects Associated with the Single Loop Cascade	274
10.6	Language Objects Associated with the Self-selective Loop	294
10.7	Language Objects Associated with the Setpoint Programmer	315

10.1 Process Control Language and IODDT Objects

Language Objects Associated with the Process Control Channels

Overview

Process control channels may be associated with five IODDTs, depending of the process control type :

- `T_PROC_FLOOP` specific to the process loop,
- `T_PROC_3SING_LOOP` specific to 3 single-loop processes,
- `T_PROC_CASC_LOOP` specific to cascade-type loops,
- `T_PROC_SPP` specific to setpoint programmers,
- `T_PROC_CONST_LOOP` specific to auto-selector loops.

Each IODDT includes a series of language objects used to drive and monitor the operation of control loops. These objects are automatically exchanged at each cycle of the task assigned to the control loops.

NOTE: IODDT variables can be created in two different ways:

- Using the **I/O objects** (see *Unity Pro, Operating Modes*) tab,
- Data Editor (see *Unity Pro, Operating Modes*).

10.2 Language Objects Associated with the Process Control Channels

Subject of this Section

This section describes the language objects associated with the process control channels.

What Is in This Section?

This section contains the following topics:

Topic	Page
Introduction	230
Control Loops Command Word	232
Setpoint Programmer Command Word CMD_ORDER (%MWr.m.c.7)	234
Summary of Selection and Command Words	235
The "Command Order" Double Word	236

Introduction

Introduction

All variables associated with a process control channel (for example: Kp, T_FILT, ...) are in default read and/or write.

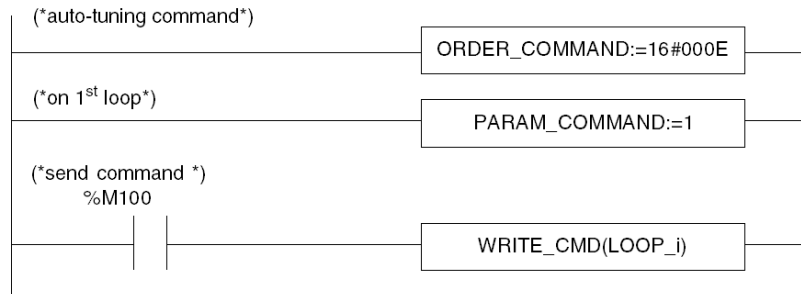
A channel (%CH r.m.c) must be associated to an IODDT (*see page 228*)-type variable. The purpose of such a variable is to simplify explicit reads and writes. The variable allows the following operations :

- Reading module and channel status words.
- Writing parameters.
- Saving parameters.
- Sending commands.

Sending Commands

Explicit instructions are applied to the language object of the *i* channel with the IODDT-type variable LOOP_i.

Example 1: sending an auto-tune instruction to the first loop of a loop controller with 3 single loops. The word being sent contains the explicit command affecting the control loop. The IODDT is of the T_PROC_3SING_LOOP type.



There is no limit to the number of commands in a PLC cycle. Acknowledgment of the instruction and updating of the states of the process control channel concerned come into effect on the task's next cycle.

NOTE: Instructions associated with the operating mode of the loop controller (Auto, Manu, Tracking, Auto-tuning) may not be sent at the same time as part of the same cycle (only the last instruction carried out in the cycle is recognized). In contrast, additional instructions (such as Remote, freeze Totalizer function, etc.) may indeed be sent as part of the same cycle.

Example 2: Manu mode transmission to the slave loop of a cascade-type channel. where LOOP_i is an IODDT-type variable. The IODDT in question may be of the T_PROC_CASC_LOOP type for a cascade-type control loop.

```
! (*slave loop selection*)
  Loop_i.PARAM_COMMAND: =2;
! (*switching to manual mode*)
  Loop_i.ORDER_COMMAND: =16#0023;
! (*send command*)
  IF %M100 THEN WRITE_CMD (LOOP_i);
  END_IF;
```

Values

Command parameter or selection word (%MDr.m.c.j) values.

- Setpoint Programmer
CMD_PARAM (%MDr.m.c.8) = j profile j j={1, ..., 6}
- Cascade loop
PARAM_COMMAND (%MDr.m.c.12) = 1 master loop
PARAM_COMMAND (%MDr.m.c.12) = 2 slave loop
- Auto-selector loop
PARAM_COMMAND (%MDr.m.c.12) = 1 main loop
PARAM_COMMAND (%MDr.m.c.12) = 2 secondary loop
PARAM_COMMAND (%MDr.m.c.12) = 4 global loop

As the process loop includes only one loop, the %MDr.m.c.12 command parameter has no significance.

- 3 single-loop loop controller
PARAM_COMMAND (%MDr.m.c.14) = j profile j j={1, 2, 3}

Control Loops Command Word

Value of the Command Word

ORDER_COMMAND (%MWr.m.c.11) is the command word for process, cascade and auto-selector loops. ORDER_COMMAND (%MWr.m.c.13) is the command word for the 3 single-loops loop controller.

Value	Meaning
16#0001	Switching to Simulation or Non-simulation Mode for the Process Value Input
16#0002	Switching to Remote or Local mode
16#0003	Switching to Manual or Automatic mode
16#0004	Freezing Totaling function
16#0005	Unfreezing Totaling function
16#0006	Resetting Totaling function
16#0007	Selecting Remote1 setpoint
16#0008	Selecting Remote2 setpoint
16#0009	Not used
16#000A	Not used
16#000B	Switching to Simulation or Non-simulation for Feed Forward input
16#000C	Switching to Tracking mode
16#000D	Switching to Non-tracking mode
16#000E	Starts Auto-tuning
16#000F	Stops Auto-tuning
16#0010	Restoring previous settings
16#0011	Using copy (may not be used after a Split Range or Hot/Cold PID)
16#0012	Not using copy (may not be used after a Split Range or Hot/Cold PID)
16#0013	Acknowledging auto-tuning diagnostics
16#0014	Activating RAISE
16#0015	Deactivating RAISE
16#0016	Activating LOWER
16#0017	Deactivating LOWER
16#0018	Not used
16#0019	Resetting Servo1
16#0020	Resetting Servo2
16#0021	Selecting setpoint in Local mode
16#0022	Selecting setpoint in Remote mode

Value	Meaning
16#0023	Switching to Manual
16#0024	Switching to Automatic
16#0025	Selector position in auto-selector
16#0026	Selector position in main loop
16#0027	Selector position in secondary loop

Setpoint Programmer Command Word CMD_ORDER (%MWr.m.c.7)

Command Word Value

The setpoint programmer command word is CMD_ORDER (%MWr.m.c.7).

Value	Meaning
16#0001	Resets the setpoint programmer (affects the current profile)
16#0002	Triggers the selected profile
16#0003	Stops the selected profile
16#0004	Prevents further changes to ("Freezes") the profile
16#0005	Releases ("Unfreezes") the current profile
16#0006	Jumps to the following segment
16#0007	Jumps to the previous segment
16#0008	Inhibits the guaranteed dwell time function
16#0009	Enables the guaranteed dwell time function
16#000A	Freezes / Unfreezes further changes to profile
16#000B	Freezes / Unfreezes the guaranteed dwell time function

Summary of Selection and Command Words

Channel Type

This table summarizes the selection and command words for each type of loop.

Loop Type	Command Word	Command Parameter or Selection Word
Process Loop	%MWr.m.c.11	None
Cascade Loop	%MWr.m.c.11	%MDr.m.c.12 = j (j = 1: master, 2: slave)
Auto-Selector Loop	%MWr.m.c.11	%MDr.m.c.12 = j (j = 1: main, 2: secondary, 4: global loop)
3 single loops	%MWr.m.c.13	%MDr.m.c.14 = j (j = 1, 2 or 3, depending on loop number)
Setpoint Programmer	%MWr.m.c.7	%MDr.m.c.8 = j (j = 1, 2, ... 6 according to profile number)

The "Command Order" Double Word

At a Glance

The **Command Order** double word, defined in the configuration of each loop, enables the transmission of one or more operating mode change commands.

The required %MD double word is entered at the same time as the **loop parameters** of the configuration screen.

The first sixteen bits (from 0 to 15) are identical to those of the periodic data table status word. This single word is copied into the first part of the **Command Order** double word.

NOTE: If the option **Reset %MWi on cold restart** is checked on the processor configuration screen, the X28 bit of the Command Order double word shall be reset to zero during the first cycle following the cold restart. It is therefore essential that the X28 bit be set to 1 by the project in the processing section of the cold restart (project operating mode management).

Description

The following table describes each bit of the Command Order double word.

Double word bit	Associated simple word bit	Description	Indication	Command (1)
%MDi.0	%MWi.0	0 : Manu, 1: Auto	X	X
%MDi.1	%MWi.1	Tracking	X	X
%MDi.2	%MWi.2	Autotuning	X	X
%MDi.3	%MWi.3	0 : Remote, 1: Local	X	X
%MDi.4	%MWi.4	Output 1 from ON/OFF or SERVO	X	X (2)
%MDi.5	%MWi.5	Output 2 from ON/OFF3 or SERVO	X	X(2)
%MDi.6	%MWi.6	Output 1 from SERVO2	X	-
%MDi.7	%MWi.7	Output 2 from SERVO2	X	-
%MDi.8	%MWi.8	SP1 or SP2 selection	X	X
%MDi.9	%MWi.9	Auto-selector in auto-selection mode	X	X
%MDi.10	%MWi.10	Auto-selector in direct main loop mode	X	X
%MDi.11	%MWi.11	Auto-selector in direct secondary loop mode	X	X
%MDi.12	%MWi.12	PID1 or PID2 output selected	X	
%MDi.13	%MWi.13	Reserved	-	-
%MDi.14	%MWi.14	Reserved	-	-
%MDi.15	%MWi.15	Reserved	-	-

Double word bit	Associated simple word bit	Description	Indication	Command (1)
%MDi.1	%MWi.1	0 : copy not used, 1 : copy used	X	X
%MDi.17	%MWi+1.1	0 : releasing totaling, 1: freezing totaling	X	X
%MDi.18	%MWi+1.2	Re-initializing totaling	-	X(3)
%MDi.19	%MWi+1.3	Return to previous adjustments	-	X(3)
%MDi.20	%MWi+1.4	Acknowledging auto-tuning diagnostics	-	X(3)
%MDi.21	%MWi+1.5	Re-initializing SERVO1	-	X(3)
%MDi.22	%MWi+1.6	Re-initializing SERVO2	-	X(3)
%MDi.23	%MWi+1.7	Save parameters	-	X(3)
%MDi.24	%MWi+1.8	Reserved	-	-
%MDi.25	%MWi+1.9	Reserved	-	-
%MDi.26	%MWi+1.10	Reserved	-	-
%MDi.27	%MWi+1.11	Reserved	-	-
%MDi.28	%MWi+1.12	0 : Writing of command word prohibited, 1 : writing of command word authorized	X	X
%MDi.29	%MWi+1.13	Reserved	-	-
%MDi.30	%MWi+1.14	Reserved	-	-
%MDi.31	%MWi+1.15	Reserved	-	-
Legend				
X : Yes - : No (1) : the command is acknowledged only if the X28 bit is at 1 (2) : There is no command associated to the function; it is included merely as information. (3) : the bit is automatically reset to zero				

10.3 Language Objects Associated with the Process Loop

Subject of this Section

This section describes the different language objects associated with the process loop.

What Is in This Section?

This section contains the following topics:

Topic	Page
Detailed Description of Default Language Objects for the T_PROC_PLOOP-Type IODDT	239
Detailed Description of Diagnostic Language Objects for the T_PROC_PLOOP-Type IODDT	243
Detailed Description of Process Control Language Objects for the T_PROC_PLOOP-type IODDT	245
Configuration Language Objects	249

Detailed Description of Default Language Objects for the T_PROC_PLOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_PLOOP-type IODDT. These **default** language objects are associated to the process loop. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_PLOOP-type IODDT_VAR1.

Comments

- Generally speaking, the meaning of any bit is provided for State 1 of said bit. In some specific instances, we provide a meaning for each state of the bit.
- Not all bits are utilized.

%lr.m.c.ERR Error Bit

The following table explains the meaning of the CH_ERROR (%lr.m.c.ERR) error bit.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Process control channel error bit.	%lr.m.c.ERR

Exchange Execution Indicators: EXCH_STS

The following table explains the various meanings of EXCH_STS (%MWr.m.c.0) channel exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Channel status words are in the process of being read.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	The exchange of Command Parameters is in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	The exchange of Adjustment Parameters is in progress.	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	The module reconfiguration is in progress.	%MWr.m.c.0.15

Exchange Report: EXCH_RPT

The following table explains the various meanings of EXCH_RPT (%MW.r.m.c.1) reporting bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error while reading the channel status words.	%MW.r.m.c.1.0
CMD_ERR	BOOL	R	Error while exchanging command parameters.	%MW.r.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters.	%MW.r.m.c.1.2
RECONF_ERR	BOOL	R	Error while reconfiguring the channel. (1 = failure)	%MW.r.m.c.1.15

Standard Channel Errors, CH_FLT

The following table explains the various meanings of the CH_FLT (%MW.r.m.c.2) status word bits.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT	BOOL	R	Fatal internal error.	%MW.r.m.c.2.4
CONF_FLT	BOOL	R	Configuration error.	%MW.r.m.c.2.5
MISSING_ADDR	BOOL	R	Address of IMC register missing.	%MW.r.m.c.2.6
WARN	BOOL	R	Error Sum.	%MW.r.m.c.2.7
STS_ERR_CALC_CORR	BOOL	R	Loop controller branch calculation error.	%MW.r.m.c.2.8
STS_ERR_FLOT_CORR	BOOL	R	Loop controller branch floating point error.	%MW.r.m.c.2.9
STS_ERR_CALC_PV	BOOL	R	PV branch calculation error.	%MW.r.m.c.2.10
STS_ERR_FLOT_PV	BOOL	R	PV branch floating point error.	%MW.r.m.c.2.11
STS_ERR_CALC_OUT	BOOL	R	OUT branch calculation error.	%MW.r.m.c.2.12
STS_ERR_FLOT_OUT	BOOL	R	OUT branch floating point error.	%MW.r.m.c.2.13

Channel States, CH_STATUS2

The following table explains the various meanings of the CH_STATUS2 (%MW.r.m.c.3) status word bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR_SCALE_PV	BOOL	R	PV branch incorrect scale.	%MW.r.m.c.3.0
STS_ERR_TH_SPLRG	BOOL	R	Split Range function thresholds incorrect.	%MW.r.m.c.3.1
STS_ERR_SCALE_OUT1	BOOL	R	OUT1 branch scale incorrect.	%MW.r.m.c.3.2
STS_ERR_SCALE_OUT2	BOOL	R	OUT2 branch scale incorrect.	%MW.r.m.c.3.3
STS_ERR_COPY_POS	BOOL	R	Servo copy address missing.	%MW.r.m.c.3.4

STATUS1 Status Word

The following table explains the various meanings of the STATUS1 (%MWr.m.c.4) status word bits. Word grouping the different Process value/Setpoint status bits

Standard symbol	Type	Access	Meaning	Address
STS_HOLD_TOT	BOOL	R	Freeze of Totaling function.	%MWr.m.c.4.0
STS_PV_SIM	BOOL	R	Simulated process value.	%MWr.m.c.4.1
STS_PV_H_LIM	BOOL	R	Upper limit on process value.	%MWr.m.c.4.2
STS_PV_L_LIM	BOOL	R	Lower limit on process value.	%MWr.m.c.4.3
STS_SP_H_LIM	BOOL	R	Upper limit on setpoint.	%MWr.m.c.4.4
STS_SP_L_LIM	BOOL	R	Lower limit on setpoint.	%MWr.m.c.4.5
STS_L_R	BOOL	R	Remote Setpoint (0) Local Setpoint (1).	%MWr.m.c.4.6
STS_R1_R2	BOOL	R	Remote2 Setpoint (1) Remote1 Setpoint (0).	%MWr.m.c.4.7
STS_ALARMS	BOOL	R	OR logic of process value alarms.	%MWr.m.c.4.8
STS_HH	BOOL	R	Very high alarm.	%MWr.m.c.4.9
STS_H	BOOL	R	High alarm.	%MWr.m.c.4.10
STS_L	BOOL	R	Low Alarm.	%MWr.m.c.4.11
STS_LL	BOOL	R	Very low alarm.	%MWr.m.c.4.12
STS_DEVH	BOOL	R	High alarm for Process value/Setpoint deviation (>0).	%MWr.m.c.4.13
STS_DEVL	BOOL	R	Lower alarm for Process value/Setpoint deviation (<0).	%MWr.m.c.4.14
STS_THLD_DONE	BOOL	R	Totaling threshold reached.	%MWr.m.c.4.15

STATUS2 Status Word

The following table explains the various meanings of the STATUS2 (%MWr.m.c.5) status word bits. Word grouping various Loop Controller/Setpoint status bits.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING	BOOL	R	Auto-tuning in progress.	%MWr.m.c.5.0
STS_TR_S	BOOL	R	Tracking in progress.	%MWr.m.c.5.1
STS_M_A	BOOL	R	Status of PID Operation Mode.	%MWr.m.c.5.3
STS_RAISE1	BOOL	R	Open command.	%MWr.m.c.5.4
STS_LOWER1	BOOL	R	Close command.	%MWr.m.c.5.5
STS_RAISE2	BOOL	R	Branch output 2 open command.	%MWr.m.c.5.6
STS_LOWER2	BOOL	R	Branch output 2 close command.	%MWr.m.c.5.7
STS_OUT_L_LIM	BOOL	R	Loop Controller lower limit reached.	%MWr.m.c.5.8
STS_OUT_H_LIM	BOOL	R	Loop Controller upper limit reached.	%MWr.m.c.5.9
STS_TOP_NEXT_CYCLE	BOOL	R	Sampling pulse on next cycle.	%MWr.m.c.5.10
STS_TOP_CUR_CYCLE	BOOL	R	Sampling pulse in current cycle.	%MWr.m.c.5.11
STS_FF_SIM	BOOL	R	Status of simulation of the FF process value.	%MWr.m.c.5.12

STATUS3 Status Word

The following table explains the various meanings of the STATUS3 (%MWr.m.c.6) status word bits. Word grouping various Servo status bits.

Standard symbol	Type	Access	Meaning	Address
POT_VAL1	BOOL	R	Servo Operation, with copy,	%MWr.m.c.6.0
POT_VAL2	BOOL	R	Servo with copy operation (Reserved).	%MWr.m.c.6.1
RAISE_STOP1	BOOL	R	Open limit reached on servo drive (Reserved).	%MWr.m.c.6.2
LOWER_STOP1	BOOL	R	Close limit reached on servo drive (Reserved).	%MWr.m.c.6.3
RAISE_STOP2	BOOL	R	Open limit reached on servo drive (Reserved).	%MWr.m.c.6.4
LOWER_STOP2	BOOL	R	Close limit reached on servo drive (Reserved).	%MWr.m.c.6.5

Detailed Description of Diagnostic Language Objects for the T_PROC_PLOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_PLOOP-type IODDT. These **diagnostic** language objects are associated to the process loop. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_PLOOP-type IODDT_VAR1.

STATUS4 Status Word

The following table explains the various meanings of the STATUS4 (%MWr.m.c.7) status word bits. Word grouping together the fine diagnostics of different errors (process value, setpoint, FF).

Standard symbol	Type	Access	Meaning	Address
SP_MIN_WARN	BOOL	R	Error while checking SP_MIN et SP_MAX parameters.	%MWr.m.c.7.0
XI_WARN	BOOL	R	Xi parameter check error.	%MWr.m.c.7.1
Yi_WARN	BOOL	R	Yi parameter check error.	%MWr.m.c.7.2
OVER_TOT_WARN	BOOL	R	Totaling overflow error.	%MWr.m.c.7.6
INP_INFR1_WARN	BOOL	R	Error while checking parameters INP_INFR1 and INP_SUPR1.	%MWr.m.c.7.8
INP_INFR2_WARN	BOOL	R	INP_INFR2 and INP_SUPR2 parameters check error.	%MWr.m.c.7.9
RATIO_WARN	BOOL	R	RATIO_MIN and RATIO_MAX parameter check error.	%MWr.m.c.7.10
SP_CALC_WARN	BOOL	R	Setpoint calculation error.	%MWr.m.c.7.11
SP_FLOAT_WARN	BOOL	R	Setpoint floating point error.	%MWr.m.c.7.12
FF_CALC_WARN	BOOL	R	Feed Forward calculation error.	%MWr.m.c.7.13
FF_FLOAT_WARN	BOOL	R	Feed Forward floating point error.	%MWr.m.c.7.14
OUT_FF_WARN	BOOL	R	OUTFF_INF and OUTFF_SUP parameters check error,	%MWr.m.c.7.15

STATUS5 Status Word

The following table explains the various meanings of the STATUS5 (%MWr.m.c.8) status word bits. Word grouping auto-tuning diagnostics.

Standard symbol	Type	Access	Meaning	Address
AT_FAILED	BOOL	R	Auto-tuning failed.	%MWr.m.c.8.0
AT_ABORTED	BOOL	R	Auto-tuning diagnostics interrupted.	%MWr.m.c.8.1
AT_ERR_PARAM	BOOL	R	Auto-tuning diagnostics parameter error.	%MWr.m.c.8.2
AT_PWF_OR_EFB_FAIL	BOOL	R	Auto-tuning diagnostics system error or power failure.	%MWr.m.c.8.3
AT_ERR_SATUR	BOOL	R	Auto-tuning diagnostics process value saturation.	%MWr.m.c.8.4
AT_DV_TOO_SMALL	BOOL	R	Auto-tuning diagnostics insufficient process value deviation.	%MWr.m.c.8.1
AT_TSAMP_HIGH	BOOL	R	Auto-tuning Diagnostics sampling pulse period too long.	%MWr.m.c.8.6
AT_INCONSIST_RESP	BOOL	R	Auto-tuning diagnostics inconsistent response.	%MWr.m.c.8.7
AT_NOT_STAB_INIT	BOOL	R	Auto-tuning diagnostics: initially unstable process value.	%MWr.m.c.8.8
AT_TMAX_TOO_SMALL	BOOL	R	Auto-tuning diagnostics: step duration too brief.	%MWr.m.c.8.9
AT_NOISE_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: process value noise too loud.	%MWr.m.c.8.10
AT_TMAX_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: step duration too long.	%MWr.m.c.8.11
AT_OVERSHOOT	BOOL	R	Auto-tuning diagnostics: overshoot exceeding 10%.	%MWr.m.c.8.12
AT_UNDERSHOOT	BOOL	R	Auto-tuning Diagnostics: non-minimal step too long.	%MWr.m.c.8.13
AT_UNSYMETRICAL_PT	BOOL	R	Auto-tuning diagnostics: procedure too asymmetrical.	%MWr.m.c.8.14
AT_INTEGRATING_PT	BOOL	R	Auto-tuning diagnostics: integrating process.	%MWr.m.c.8.15

Command Word

The following table explains the various meanings of the ORDER_COMMAND (%MWr.m.c.11) command word and of the PARAM_COMMAND (%MDr.m.c.12) command parameters command word.

Standard symbol	Type	Access	Meaning	Address
ORDER_COMMAND	INT	R/W	Command Word (<i>see page 232</i>).	%MWr.m.c.11
PARAM_COMMAND	DINT	R/W	Command Parameter.	%MDr.m.c.12

Detailed Description of Process Control Language Objects for the T_PROC_PLOOP-type IODDT

At a Glance

This section describes the language objects for the T_PROC_PLOOP-type IODDT. These **process control** language objects are associated to the process loop.

Sample Variable Declaration: T_PROC_PLOOP-type IODDT_VAR1.

Objects

The following table lists all process control language objects associated to the process loop.

Standard symbol	Type	Access	Meaning	Default Value	Address
PV_SIM	REAL	R/W	Simulated Process Value	N/A	%MWr.m.c.14
FF_SIM	REAL	R/W	Simulated Feed Forward Input	N/A	%MWr.m.c.15
T_ECH	REAL	R/W	Sampling period.	0.3	%MFr.m.c.16
OUT1	REAL	R	Value of output 1 for Hot/Cold or Split Range.	N/A	%MFr.m.c.18
OUT2	REAL	R	Value of output 2 for Hot/Cold or Split Range.	N/A	%MFr.m.c.20
OUTD	REAL	R	Command Variation Value	N/A	%MFr.m.c.22
OUTFF	REAL	R	Value of Feed Forward action on physical scale.	N/A	%MFr.m.c.24
OUT_MAN	REAL	R/W	Command Value	N/A	%MFr.m.c.26
DEV	REAL	R	Setpoint Process Value Deviation	N/A	%MFr.m.c.28
PV	REAL	R	Process value on physical scale	N/A	%MFr.m.c.30
SP	REAL	R	Setpoint value on physical scale	N/A	%MFr.m.c.32
PV_INF	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.34
PV_SUP	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.36
KP	REAL	R/W	PID proportional coefficient / IMC static gain.	1.0	%MFr.m.c.38
TI	REAL	R/W	PID integral time / IMC time constant.	0.0	%MFr.m.c.40
TD	REAL	R/W	PID derivative time / IMC delay.	0.0	%MFr.m.c.42
OUTBIAS	REAL	R/W	Bias on PID loop controller output / IMC BO/BF ratio.	0.0	%MFr.m.c.44
INT_BAND	REAL	R/W	Integral band.	0.0	%MFr.m.c.46
DBAND	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.48
KD	REAL	R/W	Derivative Filtering	10.0	%MFr.m.c.50
OUTRATE	REAL	R/W	Limit of output 1 variable speed	0.0	%MFr.m.c.52
OUTRATE2	REAL	R/W	Limit of output 2 variable speed	0.0	%MFr.m.c.54

Standard symbol	Type	Access	Meaning	Default Value	Address
OUT1_INF	REAL	R/W	Lower limit for output 1	0.0	%MFr.m.c.56
OUT1_SUP	REAL	R/W	Upper limit for output 1	100.0	%MFr.m.c.58
SP_MIN	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.60
SP_MAX	REAL	R/W	Upper limit for setpoint	100.0	%MFr.m.c.62
OUT2_INF	REAL	R/W	Lower limit for output 2	0.0	%MFr.m.c.64
OUT2_SUP	REAL	R/W	Upper limit for output 2	100.0	%MFr.m.c.66
OUT1_TH1	REAL	R/W	Threshold 1 for Hot/Cold or Split Range output 1	0.0	%MFr.m.c.68
OUT1_TH2	REAL	R/W	Threshold 2 for Hot/Cold or Split Range output 1	50.0	%MFr.m.c.70
OUT2_TH1	REAL	R/W	Threshold 1 for Hot/Cold or Split Range output 2	50.0	%MFr.m.c.72
OUT2_TH2	REAL	R/W	Threshold 2 for Hot/Cold or Split Range output 2	100.0	%MFr.m.c.74
PV_LL	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.76
PV_L	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.78
PV_H	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.80
PV_HH	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.82
RATIO	REAL	R/W	Ratio Value	1.0	%MFr.m.c.84
RATIO_MIN	REAL	R/W	Minimum Ratio Value	0.0	%MFr.m.c.86
RATIO_MAX	REAL	R/W	Maximum Ratio Value	100.0	%MFr.m.c.88
RATIO_BIAS	REAL	R/W	Ratio Bias Value	0.0	%MFr.m.c.90
ONOFF_L	REAL	R/W	Low threshold for ON/OFF Loop Controller	-5.0	%MFr.m.c.92
ONOFF_H	REAL	R/W	High threshold for ON/OFF Loop Controller	5.0	%MFr.m.c.94
HYST	REAL	R/W	Hysteresis of 3-state ON/OFF loop controller	0.0	%MFr.m.c.96
DEV_L	REAL	R/W	Lower threshold of deviation	5.0	%MFr.m.c.98
DEV_H	REAL	R/W	Upper threshold of deviation	5.0	%MFr.m.c.100
T_FILTER	REAL	R/W	Process value filtering time	0.0	%MFr.m.c.102
K_FILTER	REAL	R/W	Multiplication coefficient on process value filtering	1.0	%MFr.m.c.104
FILT_OUT	REAL	R	Filtering output value	N/A	%MFr.m.c.106
SQRT_OUT	REAL	R	Output value of square root	N/A	%MFr.m.c.108
E2_IN	REAL	R/W	Abscissa of first pulse in Segment S2	1428.0	%MFr.m.c.110
E3_IN	REAL	R/W	Abscissa of first pulse in Segment S3	2857.0	%MFr.m.c.112
E4_IN	REAL	R/W	Abscissa of first pulse in Segment S4	4285.0	%MFr.m.c.114
E5_IN	REAL	R/W	Abscissa of first pulse in Segment S5	5714.0	%MFr.m.c.116
E6_IN	REAL	R/W	Abscissa of first pulse in Segment S6	7143.0	%MFr.m.c.118
E7_IN	REAL	R/W	Abscissa of first pulse in Segment S7	8571.0	%MFr.m.c.120

Standard symbol	Type	Access	Meaning	Default Value	Address
E2_OUT	REAL	R/W	Ordinate of first pulse in Segment S2	14.28	%MFr.m.c.122
E3_OUT	REAL	R/W	Ordinate of first pulse in Segment S3	28.57	%MFr.m.c.124
E4_OUT	REAL	R/W	Ordinate of first pulse in Segment S4	42.85	%MFr.m.c.126
E5_OUT	REAL	R/W	Ordinate of first pulse in Segment S5	57.14	%MFr.m.c.128
E6_OUT	REAL	R/W	Ordinate of first pulse in Segment S6	71.43	%MFr.m.c.130
E7_OUT	REAL	R/W	Ordinate of first pulse in Segment S7	85.71	%MFr.m.c.132
THLD	REAL	R/W	Totaling limit	1E+8	%MFr.m.c.134
R_RATE	REAL	R/W	Setpoint speed increase limit	0.0	%MFr.m.c.136
D_RATE	REAL	R/W	Setpoint speed decrease limit	0.0	%MFr.m.c.138
SPEED_LIM_OUT	REAL	R	Setpoint speed limiter output value	N/A	%MFr.m.c.140
INP_INFR1	REAL	R/W	Lower scale for R1 setpoint	0.0	%MFr.m.c.142
INP_SUPR1	REAL	R/W	Higher scale for R1 setpoint	100.0	%MFr.m.c.144
INP_INFR2	REAL	R/W	Lower scale for R2 setpoint	0.0	%MFr.m.c.146
INP_SUPR2	REAL	R/W	Higher scale for R2 setpoint	100.0	%MFr.m.c.148
T1_FF	REAL	R/W	Feed Forward process value filtering time	0.0	%MFr.m.c.150
T2_FF	REAL	R/W	Feed Forward process value filtering time	0.0	%MFr.m.c.152
OUT_FF_INF	REAL	R/W	Lower limit of Feed Forward action	0.0	%MFr.m.c.154
OUT_FF_SUP	REAL	R/W	Upper limit of Feed Forward action	100.0	%MFr.m.c.156
T_MOTOR1	REAL	R/W	Open time for servo-controlled valve	10.0	%MFr.m.c.158
T_MINI1	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.160
T_MOTOR2	REAL	R/W	Open time for servo-controlled valve	10.0	%MFr.m.c.162
T_MINI2	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.164
AT_STEP	REAL	R/W	Auto-tuning step function amplitude	10.0	%MFr.m.c.166
AT_TMAX	REAL	R/W	Auto-tuning step function duration	100.0	%MFr.m.c.168
AT_PERF	REAL	R/W	Criterion for auto-tuning stability	0.5	%MFr.m.c.170
KP_PREV	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.172
TI_PREV	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.174
TD_PREV	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.176
KS	REAL	R/W	IMC static gain	1.0	%MFr.m.c.178
OL_TIME	REAL	R/W	Time Constant in OL.	1.0	%MFr.m.c.180
T_DELAY	REAL	R/W	Current pure delay time	0.0	%MFr.m.c.182
CL_PERF	REAL	R/W	OL/CL Ratio	1.0	%MFr.m.c.184
Reserved	REAL				%MFr.m.c.186

Standard symbol	Type	Access	Meaning	Default Value	Address
Reserved	REAL				%MFr.m.c.188
Reserved	REAL				%MFr.m.c.190
Reserved	REAL				%MFr.m.c.192

Configuration Language Objects

At a Glance

The following table lists all language objects associated with the process loop. These objects are not included in the IODDTs. They may however be displayed with the project program.

Description

Standard symbol	Type	Access	Meaning	Default Value	Address
CONFIG_0	INT	R	This word gathers all different process value configuration bits.	No object	%KWr.m.c.0
Filter	BOOL	R	Filtering function of the process value branch	Missing (0)	%KWr.m.c.0.0
Function Generator	BOOL	R	Function Generator for process value branch	Missing (0)	%KWr.m.c.0.1
Totaling	BOOL	R	Totaling Function for the process value branch	Missing (0)	%KWr.m.c.0.2
Square Root	BOOL	R	Square root function of the process value branch	Missing (0)	%KWr.m.c.0.3
Alarms	BOOL	R	Alarm Function of the process value branch	Present	%KWr.m.c.0.4
PV_CLIP	BOOL	R	Lopping or non-lopping of measurement.	Missing (0)	%KWr.m.c.0.8
EXTRAPOL	BOOL	R	Extrapolation of function generator	No (0)	%KWr.m.c.0.9
PV_UNI_BIP	BOOL	R	Type of process value: unipolar/bipolar.	Unipolar (0)	%KWr.m.c.0.10
PV_EXTERN	BOOL	R	Selection of Standard Process Value (0) vs. External Process Value (1).	Missing (0)	%KWr.m.c.0.11
Totaling	BOOL	R	(bit 13 = 0, bit 14 = 0): phys/ms (bit 13 = 1, bit 14 = 0): phys/s	1	%KWr.m.c.0.13
Totaling	BOOL	R	(bit 13 = 1, bit 14 = 0): phys/ms (bit 13 = 1, bit 14 = 1): phys/s	0	%KWr.m.c.0.14
No object	INT	R	Word grouping the various configuration bits of the setpoint	No object	%KWr.m.c.1
SP_SIMPLE	BOOL	R	Type of setpoint selected: simple.	Selected (1)	%KWr.m.c.1.0
SP_Selection	BOOL	R	Type of setpoint: selection.	Not selected (0)	%KWr.m.c.1.1
SPEED_LIMITER	BOOL	R	Speed limiter on setpoint	Not selected (0)	%KWr.m.c.1.2
SP_SPP	BOOL	R	Type of setpoint selected: programmer.	not selected (0)	%KWr.m.c.1.3

Standard symbol	Type	Access	Meaning	Default Value	Address
LR/L	BOOL	R	Speed limiter either on local setpoint, or in Remote/Local mode.	Local remote (0)	%KWr.m.c.1.4
SEL_MIN	BOOL	R	Function selected if there is a selection setpoint	Missing (0)	%KWr.m.c.1.8
SEL_MAX	BOOL	R	Function selected if there is a selection setpoint	Missing (0)	%KWr.m.c.1.9
SEL_SWITCH	BOOL	R	Function selected if there is a selection setpoint	Present on selection	%KWr.m.c.1.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KWr.m.c.1.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	R1 (0)	%KWr.m.c.1.12
SP_RATIO	BOOL	R	Type of setpoint selected: ratio.	Not selected (0)	%KWr.m.c.1.13
SP_LIMITER	BOOL	R	Setpoint limiter (e.g.: Param_SP).	Not present	%KWr.m.c.1.14
SP_FOLW	BOOL	R	Tracking setpoint.	Non tracking setpoint (0)	%KWr.m.c.1.15
CONFIG_2	INT	R	Word grouping various configuration bits for loop controller and FF	No object	%KWr.m.c.2
PID	BOOL	R	PID function of loop controller branch	Present	%KWr.m.c.2.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	Missing (0)	%KWr.m.c.2.1
ONOFF3	BOOL	R	ON/OFF 3-states branch of loop controller	Missing (0)	%KWr.m.c.2.2
SPLRG/HOTCOLD	BOOL	R	OR Hot/Cold and Split Range presence bits	No object	%KWr.m.c.2.3
Split Range	BOOL	R	Loop controller branch Split Range function.	Missing (0)	%KWr.m.c.2.4
Hot/Cold	BOOL	R	Hot/Cold function of loop controller branch	Not selected	%KWr.m.c.2.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.2.6
Feed Forward	BOOL	R	Presence of Feed Forward input	Missing (0)	%KWr.m.c.2.7
BUMP	BOOL	R	Bump Management when changing operating mode	With bumps (1)	%KWr.m.c.2.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.2.9
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.2.10

Standard symbol	Type	Access	Meaning	Default Value	Address
LR/L	BOOL	R	Speed limiter either on local setpoint, or in Remote/Local mode.	Local remote (0)	%KWr.m.c.1.4
SEL_MIN	BOOL	R	Function selected if there is a selection setpoint	Missing (0)	%KWr.m.c.1.8
SEL_MAX	BOOL	R	Function selected if there is a selection setpoint	Missing (0)	%KWr.m.c.1.9
SEL_SWITCH	BOOL	R	Function selected if there is a selection setpoint	Present on selection	%KWr.m.c.1.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KWr.m.c.1.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	R1 (0)	%KWr.m.c.1.12
SP_RATIO	BOOL	R	Type of setpoint selected: ratio.	Not selected (0)	%KWr.m.c.1.13
SP_LIMITER	BOOL	R	Setpoint limiter (e.g.: Param_SP).	Not present	%KWr.m.c.1.14
SP_FOLW	BOOL	R	Tracking setpoint.	Non tracking setpoint (0)	%KWr.m.c.1.15
CONFIG_2	INT	R	Word grouping various configuration bits for loop controller and FF	No object	%KWr.m.c.2
PID	BOOL	R	PID function of loop controller branch	Present	%KWr.m.c.2.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	Missing (0)	%KWr.m.c.2.1
ONOFF3	BOOL	R	ON/OFF 3-states branch of loop controller	Missing (0)	%KWr.m.c.2.2
SPLRG/HOTCOLD	BOOL	R	OR Hot/Cold and Split Range presence bits	No object	%KWr.m.c.2.3
Split Range	BOOL	R	Loop controller branch Split Range function.	Missing (0)	%KWr.m.c.2.4
Hot/Cold	BOOL	R	Hot/Cold function of loop controller branch	Not selected	%KWr.m.c.2.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.2.6
Feed Forward	BOOL	R	Presence of Feed Forward input	Missing (0)	%KWr.m.c.2.7
BUMP	BOOL	R	Bump Management when changing operating mode	With bumps (1)	%KWr.m.c.2.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.2.9
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.2.10

Standard symbol	Type	Access	Meaning	Default Value	Address
REV_DIR	BOOL	R	Type of loop controller action: 1: opposite action, a positive shift (PV-SP) generates a rise of the output value 0: direct action, a positive shift (PV-SP) generates a fall of the output value	PID opposite action (1)	%KWr.m.c.2.11
MANU/AUTO_INIT	BOOL	R	Initial value of loop controller operating mode	Manu (0)	%KWr.m.c.2.12
Lead Lag	BOOL	R	Lead Lag function of Feed Forward branch	Missing (0)	%KWr.m.c.2.13
FF_UNI_BIP	BOOL	R	Type of Feed Forward process value: unipolar/bipolar.	unipolar	%KWr.m.c.2.14
IMC	BOOL	R	IMC function of loop controller branch	Missing (0)	%KWr.m.c.2.15
CONFIG_3	NIT	R	Word grouping the various output configuration bits	No object	%KWr.m.c.3
Servo	BOOL	R	Type of output selected: Servo.	Selected	%KWr.m.c.3.0
Servo2	BOOL	R	Type of output selected: Servo.	Selected	%KWr.m.c.3.1
Analog1	BOOL	R	Type of output selected: Analog.	Selected	%KWr.m.c.3.2
Analog2	BOOL	R	Type of output selected: Analog.	Selected	%KWr.m.c.3.3
PWM1	BOOL	R	Type of output selected: PWM.	Selected	%KWr.m.c.3.4
PWM2	BOOL	R	Type of output selected: PWM.	Selected	%KWr.m.c.3.5
POT_REV1	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.3.8
POT_REV2	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.3.9
POT_VAL1_INIT	BOOL	R	Presence of Servo copy.	No (0)	%KWr.m.c.3.10
POT_VAL2_INIT	BOOL	R	Presence of (Reserved) Servo copy	Yes (1)	%KWr.m.c.3.11
ANALOG1_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar.	Unipolar	%KWr.m.c.3.12
ANALOG2_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar.	Unipolar (0)	%KWr.m.c.3.13
Loop name	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.4
Loop Unit	INT	R	Loop Unit.		%KWr.m.c.8
Trigger mode	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.11.0

10.4 Language Objects Associated with the 3 Single Loops

Subject of this Section

This section describes the language objects of the 3 single loops.

What Is in This Section?

This section contains the following topics:

Topic	Page
Detailed Description of Default Language Objects for the T_PROC_3SING_LOOP-Type IODDT	254
Detailed Description of Diagnostic Language Objects for the T_PROC_3SING_LOOP-Type IODDT	263
Detailed Description of Process Control Language Objects for the T_PROC_3SING_LOOP-Type IODDT	265
Configuration Language Objects	270

Detailed Description of Default Language Objects for the T_PROC_3SING_LOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_3SING_LOOP-type IODDT. These **default** language objects are associated to 3-single-loop controllers. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_3SING_LOOP-type IODDT_VAR1.

%lr.m.c.ERR Error Bit

The following table explains the meaning of the CH_ERROR (%lr.m.c.ERR) error bit.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Process control channel error bit.	%lr.m.c.ERR

Comments

- Generally speaking, the meaning of any bit is provided for State 1 of said bit. In some specific instances, we provide a meaning for each state of the bit.
- Not all bits are utilized.

Exchange Execution Indicators: EXCH_STS

The following table explains the various meanings of EXCH_STS (%MWr.m.c.0) channel exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Channel status words are in the process of being read.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	The exchange of Command Parameters is in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	The exchange of Adjustment Parameters is in progress.	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	The module reconfiguration is in progress.	%MWr.m.c.0.15

Exchange Report: EXCH_RPT

The following table explains the various meanings of EXCH_RPT (%MWr.m.c.1) reporting bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error while reading the channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error while exchanging command parameters.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error while reconfiguring the channel. (1 = failure)	%MWr.m.c.1.15

Standard Errors for Loop 1, CH_FLT_B1

The following table explains the various meanings of the CH_FLT_B1 (%MWr.m.c.2) status word bits for single-loop 1.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT_B1	BOOL	R	Fatal internal error on Loop 1.	%MWr.m.c.2.4
CONF_FLT_B1	BOOL	R	Configuration error on Loop 1.	%MWr.m.c.2.5
MISSING_ADDR_B1	BOOL	R	IMC register address missing on Loop 1, or Servo copy address missing,	%MWr.m.c.2.6
WARN	BOOL	R	Error Sum.	%MWr.m.c.2.7
STS_ERR_CALC_COR_B1	BOOL	R	Loop controller branch calculation error on Loop 1.	%MWr.m.c.2.8
STS_ERR_FLOT_COR_B1	BOOL	R	Loop controller branch floating point error on Loop 1.	%MWr.m.c.2.9
STS_ERR_CALC_PV_B1	BOOL	R	PV branch calculation error on Loop 1.	%MWr.m.c.2.10
STS_ERR_FLOT_PV_B1	BOOL	R	PV branch floating point error on Loop 1.	%MWr.m.c.2.11
STS_ERR_CALC_OUT_B1	BOOL	R	OUT branch calculation error on Loop 1.	%MWr.m.c.2.12
STS_ERR_FLOT_OUT_B1	BOOL	R	OUT branch floating point error on Loop 1.	%MWr.m.c.2.13

Standard Errors for Loop 2, CH_FLT_B2

The following table explains the various meanings of the CH_FLT_B2 (%MWr.m.c.3) status word bits for single-loop 2.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT_B2	BOOL	R	Fatal internal error on Loop 1.	%MWr.m.c.3.4
CONF_FLT_B2	BOOL	R	Configuration error on Loop 1.	%MWr.m.c.3.5
MISSING_ADDR_B2	BOOL	R	IMC register address missing on Loop 1, or Servo copy address missing,	%MWr.m.c.3.6
STS_ERR_CALC_COR_B2	BOOL	R	Loop controller branch calculation error on Loop 2.	%MWr.m.c.3.8
STS_ERR_FLOT_COR_B2	BOOL	R	Loop controller branch floating point error on Loop 2.	%MWr.m.c.3.9
STS_ERR_CALC_PV_B2	BOOL	R	PV branch calculation error on Loop 2.	%MWr.m.c.3.10
STS_ERR_FLOT_PV_B2	BOOL	R	PV branch floating point error on Loop 2.	%MWr.m.c.3.11
STS_ERR_CALC_OUT_B2	BOOL	R	OUT branch calculation error on Loop 2.	%MWr.m.c.3.12
STS_ERR_FLOT_OUT_B2	BOOL	R	OUT branch floating point error on Loop 2.	%MWr.m.c.3.13

Standard Errors for Loop 3, CH_FLT_B3

The following table explains the various meanings of the CH_FLT_B3 (%MWr.m.c.4) status word bits for single-loop 3.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT_B3	BOOL	R	Fatal internal error on Loop 3.	%MWr.m.c.4.4
CONF_FLT_B3	BOOL	R	Configuration error on Loop 3.	%MWr.m.c.4.5
MISSING_ADDR_B3	BOOL	R	IMC register address missing on Loop 3, or Servo copy address missing,	%MWr.m.c.4.6
STS_ERR_CALC_COR_B3	BOOL	R	Loop controller branch calculation error on Loop 3.	%MWr.m.c.4.8
STS_ERR_FLOT_COR_B3	BOOL	R	Loop controller branch floating point error on Loop 3.	%MWr.m.c.4.9
STS_ERR_CALC_PV_B3	BOOL	R	PV branch calculation error on Loop 3.	%MWr.m.c.4.10
STS_ERR_FLOT_PV_B3	BOOL	R	PV branch floating point error on Loop 3.	%MWr.m.c.4.11
STS_ERR_CALC_OUT_B3	BOOL	R	OUT branch calculation error on Loop 3.	%MWr.m.c.4.12
STS_ERR_FLOT_OUT_B3	BOOL	R	OUT branch floating point error on Loop 3.	%MWr.m.c.4.13

STATUS1 Status Word for Loop 1

The following table explains the various meanings of the STATUS1_B1 (%MW.r.m.c.5) status word for Loop 1. The table lists all different bits for the Process Value/Setpoint status.

Standard symbol	Type	Access	Meaning	Address
STS_HOLD_TOT_B1	BOOL	R	Totaling function state.	%MW.r.m.c.5.0
STS_PV_SIM_B1	BOOL	R	Process value simulation state.	%MW.r.m.c.5.1
STS_PV_H_LIM_B1	BOOL	R	Upper limit on process value branch (PV_SUP).	%MW.r.m.c.5.2
STS_PV_L_LIM_B1	BOOL	R	Lower limit on process value branch (PV_INF).	%MW.r.m.c.5.3
STS_SP_H_LIM_B1	BOOL	R	Upper limit on setpoint branch.	%MW.r.m.c.5.4
STS_SP_L_LIM_B1	BOOL	R	Lower limit on setpoint branch.	%MW.r.m.c.5.5
STS_L_R_B1	BOOL	R	State of the selected Remote/Local setpoint.	%MW.r.m.c.5.6
STS_TR_S_B1	BOOL	R	Tracking status bit.	%MW.r.m.c.5.7
STS_ALARMS_B1	BOOL	R	Sum of process value alarms.	%MW.r.m.c.5.8
STS_HH_B1	BOOL	R	Very high alarm.	%MW.r.m.c.5.9
STS_H_B1	BOOL	R	High alarm.	%MW.r.m.c.5.10
STS_L_B1	BOOL	R	Low Alarm.	%MW.r.m.c.5.11
STS_LL_B1	BOOL	R	Very low alarm.	%MW.r.m.c.5.12
STS_DEVH_B1	BOOL	R	Upper threshold for Process value/Setpoint deviation (>0).	%MW.r.m.c.5.13
STS_DEVL_B1	BOOL	R	Lower threshold for Process value/Setpoint deviation (<0).	%MW.r.m.c.5.14
STS_THLD_DONE_B1	BOOL	R	Totaling threshold reached.	%MW.r.m.c.5.15

STATUS2 Status Word for Loop 1

The following table explains the various meanings of the STATUS2_B1 (%MWr.m.c.6) status word bits for Loop 1. The table lists all different bits for the Process Controller/Output status.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_B1	BOOL	R	Auto-tuning in progress (common to 3 loops)	%MWr.m.c.6.0
STS_M_A_B1	BOOL	R	Status of PID Operation Mode.	%MWr.m.c.6.1
STS_RAISE1_B1	BOOL	R	Open command.	%MWr.m.c.6.2
STS_LOWER1_B1	BOOL	R	Close command.	%MWr.m.c.6.3
STS_OUT_L_LIM_B1	BOOL	R	The calculated output of the PID is greater than or equal to OUT_SUP.	%MWr.m.c.6.4
STS_OUT_H_LIM_B1	BOOL	R	The calculated output of the PID is smaller than or equal to OUT_INF.	%MWr.m.c.6.5
POT_VAL_B1	BOOL	R	Servo Operation, with copy,	%MWr.m.c.6.6
RAISE_STOP_B1	BOOL	R	Open limit reached on Servo drive (Reserved).	%MWr.m.c.6.7
LOWER_STOP_B1	BOOL	R	Close limit reached on Servo drive (Reserved).	%MWr.m.c.6.8
STS_TOP_NEXT_CYC_B1	BOOL	R	Sampling pulse on next cycle.	%MWr.m.c.6.9
STS_TOP_CUR_CYC_B1	BOOL	R	Sampling pulse in current cycle.	%MWr.m.c.6.10
OVER_TOT_WARN_B1	BOOL	R	Totaling overflow error (T_MOTOR1_WARN suppressed).	%MWr.m.c.6.11
INP_INF_WARN_B1	BOOL	R	Error while checking Loop 1 parameters INP_INF and INP_SUP.	%MWr.m.c.6.12
SP_MIN_WARN_B1	BOOL	R	Error while checking Loop 1 parameters SP_MIN et SP_MAX.	%MWr.m.c.6.13
SP_CALC_WARN_B1	BOOL	R	Setpoint calculation error.	%MWr.m.c.6.14
SP_FLOAT_WARN_B1	BOOL	R	Setpoint floating point error.	%MWr.m.c.6.15

STATUS1 Status Word for Loop 2

The following table explains the various meanings of the STATUS1_B2 (%MWr.m.c.7) status word bits for Loop 2. The table lists all different bits for the Process Value/Setpoint status.

Standard symbol	Type	Access	Meaning	Address
STS_HOLD_TOT_B2	BOOL	R	Totaling function state.	%MWr.m.c.7.0
PV_SIM_B2	BOOL	R	Process value simulation state.	%MWr.m.c.7.1
STS_PV_H_LIM_B2	BOOL	R	Upper limit on process value branch (PV_SUP).	%MWr.m.c.7.2
STS_PV_L_LIM_B2	BOOL	R	Lower limit on process value branch (PV_INF).	%MWr.m.c.7.3
STS_SP_H_LIM_B2	BOOL	R	Upper limit on setpoint branch.	%MWr.m.c.7.4
STS_SP_L_LIM_B2	BOOL	R	Lower limit on setpoint branch.	%MWr.m.c.7.5
STS_L_R_B2	BOOL	R	State of the selected Remote/Local setpoint.	%MWr.m.c.7.6
STS_TR_S_B2	BOOL	R	Tracking status bit.	%MWr.m.c.7.7
STS_ALARMS_B2	BOOL	R	Sum of process value alarms.	%MWr.m.c.7.8
STS_HH_B2	BOOL	R	Very high alarm.	%MWr.m.c.7.9
STS_H_B2	BOOL	R	High alarm.	%MWr.m.c.7.10
STS_L_B2	BOOL	R	Low Alarm.	%MWr.m.c.7.11
STS_LL_B2	BOOL	R	Very low alarm.	%MWr.m.c.7.12
STS_DEVH_B2	BOOL	R	Upper threshold for Process value/Setpoint deviation (>0).	%MWr.m.c.7.13
STS_DEVL_B2	BOOL	R	Lower threshold for Process value/Setpoint deviation (<0).	%MWr.m.c.7.14
STS_THLD_DONE_B2	BOOL	R	Totaling threshold reached.	%MWr.m.c.7.15

STATUS2 Status Word for Loop 2

The following table explains the various meanings of the STATUS2_B2 (%MWr.m.c.8) status word bits for Loop 2. The table lists all different bits for the Process Controller/Output status.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_B2	BOOL	R	Auto-tuning in progress (common to 3 loops)	%MWr.m.c.8.0
STS_M_A_B2	BOOL	R	Status of PID Operation Mode.	%MWr.m.c.8.1
STS_RAISE1_B2	BOOL	R	Open command.	%MWr.m.c.8.2
STS_LOWER1_B2	BOOL	R	Close command.	%MWr.m.c.8.3
STS_OUT_L_LIM_B2	BOOL	R	The calculated output of the PID is greater than or equal to OUT_SUP.	%MWr.m.c.8.4
STS_OUT_H_LIM_B2	BOOL	R	The calculated output of the PID is smaller than or equal to OUT_INF.	%MWr.m.c.8.1
POT_VAL_B2	BOOL	R	Servo Operation, with copy,	%MWr.m.c.8.6
RAISE_STOP_B2	BOOL	R	Open limit reached on Servo drive (Reserved).	%MWr.m.c.8.7
LOWER_STOP_B2	BOOL	R	Close limit reached on Servo drive (Reserved).	%MWr.m.c.8.8
STS_TOP_NEXT_CYC_B2	BOOL	R	Sampling pulse on next cycle.	%MWr.m.c.8.9
STS_TOP_CUR_CYC_B2	BOOL	R	Sampling pulse in current cycle.	%MWr.m.c.8.10
OVER_TOT_WARN_B2	BOOL	R	Totaling overflow error (T_MOTOR1_WARN suppressed).	%MWr.m.c.8.11
INP_INF_WARN_B2	BOOL	R	Error while checking Loop 1 parameters INP_INF and INP_SUP.	%MWr.m.c.8.12
SP_MIN_WARN_B2	BOOL	R	Error while checking Loop 1 parameters SP_MIN et SP_MAX.	%MWr.m.c.8.13
SP_CALC_WARN_B2	BOOL	R	Setpoint calculation error.	%MWr.m.c.8.14
SP_FLOAT_WARN_B2	BOOL	R	Setpoint floating point error.	%MWr.m.c.8.15

STATUS1 Status Word for Loop 3

The following table explains the various meanings of the STATUS1_B3 (%MWr.m.c.9) for Loop 3. The table lists all different bits for the Process Value/Setpoint status.

Standard symbol	Type	Access	Meaning	Address
STS_HOLD_TOT_B3	BOOL	R	Totaling function state.	%MWr.m.c.9.0
STS_PV_SIM_B3	BOOL	R	Process value simulation state.	%MWr.m.c.9.1
STS_PV_H_LIM_B3	BOOL	R	Upper limit on process value branch (PV_SUP).	%MWr.m.c.9.2
STS_PV_L_LIM_B3	BOOL	R	Lower limit on process value branch (PV_INF).	%MWr.m.c.9.3
STS_SP_H_LIM_B3	BOOL	R	Upper limit on setpoint branch.	%MWr.m.c.9.4
STS_SP_L_LIM_B3	BOOL	R	Lower limit on setpoint branch.	%MWr.m.c.9.5
STS_L_R_B3	BOOL	R	Setpoint state.	%MWr.m.c.9.6
STS_TR_S_B3	BOOL	R	Tracking status bit.	%MWr.m.c.9.7
STS_ALARMS_B3	BOOL	R	Sum of process value alarms.	%MWr.m.c.9.8
STS_HH_B3	BOOL	R	Very high alarm.	%MWr.m.c.9.9
STS_H_B3	BOOL	R	High alarm.	%MWr.m.c.9.10
STS_L_B3	BOOL	R	Low Alarm.	%MWr.m.c.9.11
STS_LL_B3	BOOL	R	Very low alarm.	%MWr.m.c.9.12
STS_DEVH_B3	BOOL	R	Upper threshold for Process value/Setpoint deviation (>0).	%MWr.m.c.9.13
STS_DEVL_B3	BOOL	R	Lower threshold for Process value/Setpoint deviation (<0).	%MWr.m.c.9.14
STS_THLD_DONE_B3	BOOL	R	Totaling threshold reached.	%MWr.m.c.9.15

STATUS2 Status Word for Loop 3

The following table explains the various meanings of the STATUS2_B3 (%MWr.m.c.10) status word bits for Loop 3. The table lists all different bits for the Process Controller/Output status.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_B3	BOOL	R	Auto-tuning in progress (common to 3 loops)	%MWr.m.c.10.0
STS_M_A_B3	BOOL	R	Status of PID Operation Mode.	%MWr.m.c.10.1
STS_RAISE1_B3	BOOL	R	Open command.	%MWr.m.c.10.2
STS_LOWER1_B3	BOOL	R	Close command.	%MWr.m.c.10.3
STS_OUT_L_LIM_B3	BOOL	R	The calculated output of the PID is greater than or equal to OUT_SUP.	%MWr.m.c.10.4
STS_OUT_H_LIM_B3	BOOL	R	The calculated output of the PID is smaller than or equal to OUT_INF.	%MWr.m.c.10.5
POT_VAL_B3	BOOL	R	Servo Operation, with copy,	%MWr.m.c.10.6
RAISE_STOP_B3	BOOL	R	Open limit reached on servo drive (Reserved).	%MWr.m.c.10.7
LOWER_STOP_B3	BOOL	R	Close limit reached on servo drive (Reserved).	%MWr.m.c.10.8
STS_TOP_NEXT_CYC_B3	BOOL	R	Sampling pulse on next cycle.	%MWr.m.c.10.9
STS_TOP_CUR_CYC_B3	BOOL	R	Sampling pulse in current cycle.	%MWr.m.c.10.10
OVER_TOT_WARN_B3	BOOL	R	Totaling overflow error (T_MOTOR1_WARN suppressed).	%MWr.m.c.10.11
INP_INF_WARN_B3	BOOL	R	Error while checking Loop 3 parameters INP_INF and INP_SUP.	%MWr.m.c.10.12
SP_MIN_WARN_B3	BOOL	R	Error while checking Loop 3 parameters SP_MIN et SP_SUP.	%MWr.m.c.10.13
SP_CALC_WARN_B3	BOOL	R	Setpoint calculation error.	%MWr.m.c.10.14
SP_FLOAT_WARN_B3	BOOL	R	Setpoint floating point error.	%MWr.m.c.10.15

Detailed Description of Diagnostic Language Objects for the T_PROC_3SING_LOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_3SING_LOOP-type IODDT. These **diagnostic** language objects are associated to 3-single-loop controllers. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_3SING_LOOP-type IODDT_VAR1.

STATUS4 Status Word

The following table explains the various meanings of the STATUS4 (%MWr.m.c.11) status word bits. Word grouping fine diagnostics of different errors.

Standard symbol	Type	Access	Meaning	Address
AT_FAILED	BOOL	R	Auto-tuning failed.	%MWr.m.c.11.0
AT_ABORTED	BOOL	R	Auto-tuning diagnostics interrupted.	%MWr.m.c.11.1
AT_ERR_PARAM	BOOL	R	Auto-tuning diagnostics parameter error.	%MWr.m.c.11.2
AT_PWF_OR_EFB_FAIL	BOOL	R	Auto-tuning diagnostics system error or power failure.	%MWr.m.c.11.3
AT_ERR_SATUR	BOOL	R	Auto-tuning diagnostics process value saturation.	%MWr.m.c.11.4
AT_DV_TOO_SMALL	BOOL	R	Auto-tuning diagnostics insufficient process value deviation.	%MWr.m.c.11.5
AT_TSAMP_HIGH	BOOL	R	Auto-tuning Diagnostics sampling pulse period too long.	%MWr.m.c.11.6
AT_INCONSIST_RESP	BOOL	R	Auto-tuning diagnostics inconsistent response.	%MWr.m.c.11.7
AT_NOT_STAB_INIT	BOOL	R	Auto-tuning diagnostics: initially unstable process value.	%MWr.m.c.11.8
AT_TMAX_TOO_SMALL	BOOL	R	Auto-tuning diagnostics: step duration too brief.	%MWr.m.c.11.9
AT_NOISE_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: process value noise too loud.	%MWr.m.c.11.10
AT_TMAX_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: step duration too long.	%MWr.m.c.11.11
AT_OVERSHOOT	BOOL	R	Auto-tuning diagnostics: overshoot exceeding 10%.	%MWr.m.c.11.12
AT_UNDERSHOOT	BOOL	R	Auto-tuning Diagnostics: non-minimal step too long.	%MWr.m.c.11.13

Standard symbol	Type	Access	Meaning	Address
AT_UNSYMETRICAL_PT	BOOL	R	Auto-tuning diagnostics: procedure too asymmetrical.	%MWr.m.c.11.14
AT_INTEGRATING_PT	BOOL	R	Auto-tuning diagnostics: integrating process.	%MWr.m.c.11.15

Command Word

The following table explains the various meanings of the `ORDER_COMMAND` (%MWr.m.c.13) command word and of the `PARAM_COMMAND` (%MDr.m.c.1114) command parameters word.

Standard symbol	Type	Access	Meaning	Address
ORDER_COMMAND	INT	R/W	Command Word (<i>see page 232</i>).	%MWr.m.c.13
PARAM_COMMAND	DINT	R/W	Command Parameter.	%MDr.m.c.14

Detailed Description of Process Control Language Objects for the T_PROC_3SING_LOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_3SING_LOOP-type IODDT. These **process control** language objects are associated to 3-single-loop controllers.

Sample Variable Declaration: T_PROC_3SING_LOOP-type IODDT_VAR1.

Objects

The following table describes the process control language objects associated to 3-single-loop controllers.

Standard symbol	Type	Access	Meaning	Default Value	Address
AT_STEP	REAL	R/W	Auto-tuning step function amplitude	10.0	%MFr.m.c.16
AT_TMAX	REAL	R/W	Auto-tuning step function duration	100.0	%MFr.m.c.18
AT_PERF	REAL	R/W	Criterion for auto-tuning stability.	0.5	%MFr.m.c.20
T_ECH_B1	REAL	R/W	Sampling period.	0.3	%MFr.m.c.22
OUT_MAN_B1	REAL	R/W	Command Value	N/A	%MFr.m.c.24
DEV_B1	REAL	R	Setpoint Process Value Deviation	N/A	%MFr.m.c.26
PV_B1	REAL	R	Process value on physical scale	N/A	%MFr.m.c.28
SP_B1	REAL	R	Setpoint value on physical scale	N/A	%MFr.m.c.30
PV_INF_B1	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.32
PV_SUP_B1	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.34
KP_B1	REAL	R/W	PID proportional coefficient / IMC static gain.	1.0	%MFr.m.c.36
TI_B1	REAL	R/W	PID integral time / IMC time constant.	0.0	%MFr.m.c.38
TD_B1	REAL	R/W	PID derivative time / IMC delay.	0.0	%MFr.m.c.40
OUTBIAS_B1	REAL	R/W	Bias on PID loop controller output / IMC BO/BF ratio.	0.0	%MFr.m.c.42
INT_BAND_B1	REAL	R/W	Integral band.	0.0	%MFr.m.c.44
DBAND_B1	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.46
OUTRATE_B1	REAL	R/W	Limit of output variation speed.	0.0	%MFr.m.c.48
OUT_INF_B1	REAL	R/W	Lower limit for Output 1.	0.0	%MFr.m.c.50
OUT_SUP_B1	REAL	R/W	Upper limit for Output 1.	100.0	%MFr.m.c.52
SP_INF_B1	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.54
SP_SUP_B1	REAL	R/W	Upper limit for setpoint	100.	%MFr.m.c.56

Standard symbol	Type	Access	Meaning	Default Value	Address
PV_LL_B1	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.58
PV_L_B1	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.60
PV_H_B1	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.62
PV_HH_B1	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.64
ONOFF_L_B1	REAL	R/W	Lower threshold for ON/OFF Loop Controller.	-5.0	%MFr.m.c.66
ONOFF_H_B1	REAL	R/W	Upper threshold for ON/OFF Loop Controller.	5.0	%MFr.m.c.68
HYST_B1	REAL	R/W	Hysteresis of 3-state ON/OFF loop controller.	0.0	%MFr.m.c.70
DEV_L_B1	REAL	R/W	Lower threshold of deviation	-5.0	%MFr.m.c.72
DEV_H_B1	REAL	R/W	Upper threshold of deviation	5.0	%MFr.m.c.74
THLD_B1	REAL	R/W	Totaling limit.	1E+8	%MFr.m.c.76
R_RATE_B1	REAL	R/W	Speed limiter ascent speed value.	0.0	%MFr.m.c.78
D_RATE_B1	REAL	R/W	Speed limiter descent speed value.	0.0	%MFr.m.c.80
SPEED_LIM_OUT_B1	REAL	R	Speed limiter output value.	N/A	%MFr.m.c.82
INP_MINR_B1	REAL	R/W	Remote setpoint loop 1 low scale.	0.0	%MFr.m.c.84
INP_MAXR_B1	REAL	R/W	Remote setpoint loop 1 high scale.	100.0	%MFr.m.c.86
T_MOTOR_B1	REAL	R/W	Open time for servo-controlled valve.	0.0	%MFr.m.c.88
T_MINI_B1	REAL	R/W	Minimum open time for servo-controlled valve.	0.0	%MFr.m.c.90
KP_PREV_B1	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.92
TI_PREV_B1	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.94
TD_PREV_B1	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.96
T_ECH_B2	REAL	R/W	Sampling period.	0.3	%MFr.m.c.98
OUT_MAN_B2	REAL	R/W	Command Value	N/A	%MFr.m.c.100
DEV_B2	REAL	R	Setpoint Process Value Deviation	N/A	%MFr.m.c.102
PV_B2	REAL	R	Process value on physical scale	N/A	%MFr.m.c.104
SP_B2	REAL	R	Setpoint value on physical scale	N/A	%MFr.m.c.106
PV_INF_B2	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.108
PV_SUP_B2	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.110
KP_B2	REAL	R/W	Proportional Coefficient	1.0	%MFr.m.c.112
TI_B2	REAL	R/W	Integral Time	0.0	%MFr.m.c.114

Standard symbol	Type	Access	Meaning	Default Value	Address
TD_B2	REAL	R/W	Derivative Time	0.0	%MFr.m.c.116
OUTBIAS_B2	REAL	R/W	Bias on PID loop controller output	0.0	%MFr.m.c.118
INT_BAND_B2	REAL	R/W	Integral band	0.0	%MFr.m.c.120
DBAND_B2	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.122
OUTRATE_B2	REAL	R/W	Limit of output variation speed.	0.0	%MFr.m.c.124
OUT_INF_B2	REAL	R/W	Lower limit for Output 1.	0.0	%MFr.m.c.126
OUT_SUP_B2	REAL	R/W	Upper limit for Output 1.	100.0	%MFr.m.c.128
SP_INF_B2	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.130
SP_SUP_B2	REAL	R/W	Upper limit for setpoint	100.0	%MFr.m.c.132
PV_LL_B2	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.134
PV_L_B2	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.136
PV_H_B2	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.138
PV_HH_B2	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.140
ONOFF_L_B2	REAL	R/W	Lower threshold for ON/OFF Loop Controller.	-5.0	%MFr.m.c.142
ONOFF_H_B2	REAL	R/W	Upper threshold for ON/OFF Loop Controller.	5.0	%MFr.m.c.144
HYST_B2	REAL	R/W	Hysteresis of 3-state ON/OFF loop controller.	0.0	%MFr.m.c.146
DEV_L_B2	REAL	R/W	Lower threshold of deviation	-5.0	%MFr.m.c.148
DEV_H_B2	REAL	R/W	Upper threshold of deviation	5.0	%MFr.m.c.150
THLD_B2	REAL	R/W	Totaling limit.	1E+8	%MFr.m.c.152
R_RATE_B2	REAL	R/W	Speed limiter ascent speed value.	0.0	%MFr.m.c.154
D_RATE_B2	REAL	R/W	Speed limiter descent speed value.	0.0	%MFr.m.c.156
SPEED_LIM_OUT_B2	REAL	R	Speed limiter output value.	N/A	%MFr.m.c.158
INP_MINR_B2	REAL	R/W	Remote setpoint loop 2 low scale.	0.0	%MFr.m.c.160
INP_MAXR_B2	REAL	R/W	Remote setpoint loop 2 high scale.	100.0	%MFr.m.c.162
T_MOTOR_B2	REAL	R/W	Open time for servo-controlled valve.	0.0	%MFr.m.c.164
T_MINI_B2	REAL	R/W	Minimum open time for servo-controlled valve.	0.0	%MFr.m.c.166
KP_PREV_B2	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.168
TI_PREV_B2	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.170
TD_PREV_B2	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.172

Standard symbol	Type	Access	Meaning	Default Value	Address
T_ECH_B3	REAL	R/W	Sampling period.	0.3	%MFr.m.c.174
OUT_MAN_B3	REAL	R/W	Command Value	N/A	%MFr.m.c.176
DEV_B3	REAL	R	Setpoint Process Value Deviation	N/A	%MFr.m.c.178
PV_B3	REAL	R	Process value on physical scale	N/A	%MFr.m.c.180
SP_B3	REAL	R/W	Setpoint value on physical scale	N/A	%MFr.m.c.182
PV_INF_B3	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.184
PV_SUP_B3	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.186
KP_B3	REAL	R/W	Proportional Coefficient	1.0	%MFr.m.c.188
TI_B3	REAL	R/W	Integral Time	0.0	%MFr.m.c.190
TD_B3	REAL	R/W	Derivative Time	0.0	%MFr.m.c.192
OUTBIAS_B3	REAL	R/W	Bias on PID loop controller output	0.0	%MFr.m.c.194
INT_BAND_B3	REAL	R/W	Integral band	0.0	%MFr.m.c.196
DBAND_B3	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.198
OUTRATE_B3	REAL	R/W	Limit of output variation speed.	0.0	%MFr.m.c.200
OUT_INF_B3	REAL	R/W	Lower limit for output 1	0.0	%MFr.m.c.202
OUT_SUP_B3	REAL	R/W	Upper limit for output 1	100.0	%MFr.m.c.204
SP_INF_B3	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.206
SP_SUP_B3	REAL	R/W	Upper limit for setpoint	100.0	%MFr.m.c.208
PV_LL_B3	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.210
PV_L_B3	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.212
PV_H_B3	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.214
PV_HH_B3	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.216
ONOFF_L_B3	REAL	R/W	Lower threshold for ON/OFF Loop Controller.	-5.0	%MFr.m.c.218
ONOFF_H_B3	REAL	R/W	Upper threshold for ON/OFF Loop Controller.	5.0	%MFr.m.c.220
HYST_B3	REAL	R/W	Hysteresis of 3-state ON/OFF loop controller.	0.0	%MFr.m.c.222
DEV_L_B3	REAL	R/W	Lower threshold of deviation	-5.0	%MFr.m.c.224
DEV_H_B3	REAL	R/W	Upper threshold of deviation	5.0	%MFr.m.c.226
THLD_B3	REAL	R/W	Totaling limit.	1E+8	%MFr.m.c.228
R_RATE_B3	REAL	R/W	Speed limiter ascent speed value.	0.0	%MFr.m.c.230
D_RATE_B3	REAL	R/W	Speed limiter descent speed value.	0.0	%MFr.m.c.232
SPEED_LIM_OUT_B3	REAL	R	Speed limiter output value.	N/A	%MFr.m.c.234

Standard symbol	Type	Access	Meaning	Default Value	Address
INP_MINR_B3	REAL	R/W	Remote setpoint loop 3 low scale.	0.0	%MFr.m.c.236
INP_MAXR_B3	REAL	R/W	Remote setpoint loop 3 high scale.	100.0	%MFr.m.c.238
T_MOTOR_B3	REAL	R/W	Open time for servo-controlled valve.	0.0	%MFr.m.c.240
T_MINI_B3	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.242
KP_PREV_B3	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.244
TI_PREV_B3	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.246
TD_PREV_B3	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.248
PV_SIM_B1	INT	R/W	Loop 1 Simulated Process Value.	N/A	%MWFr.m.c.250
PV_SIM_B2	INT	R/W	Loop 2 Simulated Process Value.	N/A	%MWFr.m.c.251
PV_SIM_B3	INT	R/W	Loop 3 Simulated Process Value.	N/A	%MWFr.m.c.252

Configuration Language Objects

Description

The following table describes the process control language objects associated to 3-single-loop controllers.

Standard symbol	Type	Access	Meaning	Default Value	Address
CONFIG_0_B1	INT	R	Word grouping the various configuration bits of the process value	No object	%KWr.m.c.0
Filter	BOOL	R	Filtering function of the process value branch	Not configurable (0)	%KWr.m.c.0.0
Function Generator	BOOL	R	Function Generator for process value branch	Not configurable (0)	%KWr.m.c.0.1
Totaling	BOOL	R	Totaling Function for the process value branch	Missing (0)	%KWr.m.c.0.2
Square Root	BOOL	R	Root function of the process value branch	Square Root	%KWr.m.c.0.3
Alarms	BOOL	R	Alarm Function of the process value branch	Present	%KWr.m.c.0.4
PV_CLIP	BOOL	R	Lopping or non-lopping of process value	Missing (0)	%KWr.m.c.0.8
EXTRAPOL	BOOL	R	Extrapolation of function generator	Not configurable (0)	%KWr.m.c.0.9
PV_UNI_BIP	BOOL	R	Type of process value: unipolar/bipolar	Unipolar (0)	%KWr.m.c.0.10
PV_EXTERN	BOOL	R	Selection of Standard Process Value (0) vs. External Process Value (1)	Missing (0)	%KWr.m.c.0.11
VALID_C1	BOOL	R	Loop used (1) / not used (0)	Enabled (1)	%KWr.m.c.0.12
Totaling: Process value unit	BOOL	R	(bit 13 = 0, bit 14 = 0): phys/ms (bit 13 = 1, bit 14 = 0): phys/s	1	%KWr.m.c.0.13
Totaling: Process value unit	BOOL	R	(bit 13 = 1, bit 14 = 0): phys/mn (bit 13 = 1, bit 14 = 1): phys/ms	0	%KWr.m.c.0.14
CONFIG_1_B1	INT	R	Word grouping the various configuration bits of the setpoint	No object	%KWr.m.c.1
SP_SIMPLE	BOOL	R	Type of setpoint selected: simple.	Selected (1)	%KWr.m.c.1.0
SP_Selection	BOOL	R	Type of setpoint selected: selection.	Not configurable (0)	%KWr.m.c.1.1
SPEED_LIMITER	BOOL	R	Speed limiter on setpoint	Not selected (0)	%KWr.m.c.1.2

Standard symbol	Type	Access	Meaning	Default Value	Address
SP_SPP	BOOL	R	Type of setpoint selected: Programmer.	Not selected (0)	%KWr.m.c.1.3
LR/L	BOOL	R	Speed limiter either on local setpoint or in Remote/Local mode	Local remote (0)	%KWr.m.c.1.4
SEL_MIN	BOOL	R	Function selected if there is a selection setpoint	Not configurable (0)	%KWr.m.c.1.8
SEL_MAX	BOOL	R	Function selected if there is a selection setpoint	Not configurable (0)	%KWr.m.c.1.9
SEL_SWITCH	BOOL	R	Function selected if there is a selection setpoint	Not configurable (0)	%KWr.m.c.1.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KWr.m.c.1.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	Not configurable	%KWr.m.c.1.12
SP_RATIO	BOOL	R	Type of setpoint selected: Ratio.	Not configurable (0)	%KWr.m.c.1.13
SP_LIMITER	BOOL	R	Setpoint Limiter (e.g. PARAM_SP)	not selected (0)	%KWr.m.c.1.14
SP_FOLW	BOOL	R	Tracking Setpoint (0)	No tracking	%KWr.m.c.1.15
CONFIG_2_B1	INT	R	Word grouping various configuration bits for loop controller and FF	No object	%KWr.m.c.2
PID	BOOL	R	PID function of loop controller branch	Present	%KWr.m.c.2.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	Missing (0)	%KWr.m.c.2.1
ONOFF3	BOOL	R	ON/OFF 3-states branch of loop controller	Missing (0)	%KWr.m.c.2.2
SPLRG/HOTCOLD	BOOL	R	OR Hot/Cold and Split Range presence bits	Not configurable (0)	%KWr.m.c.2.3
Split/Range	BOOL	R	Loop controller branch Split Range function.	Not configurable (0)	%KWr.m.c.2.4
Hot/Cold	BOOL	R	Hot/Cold function of loop controller branch	Not configurable (0)	%KWr.m.c.2.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.2.6
Feed Forward	BOOL	R	Presence of Feed Forward input	Not configurable (0)	%KWr.m.c.2.7
BUMP	BOOL	R	Bump Management when changing operating mode	With bumps (1)	%KWr.m.c.2.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.2.9

Standard symbol	Type	Access	Meaning	Default Value	Address
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.2.10
REV_DIR	BOOL	R	Type of loop controller action: 1: opposite action, a positive shift (PV-SP) generates a rise of the output value 0: direct action, a positive shift (PV-SP) generates a fall of the output value	PID opposite action (1)	%KWr.m.c.2.11
MANU/AUTO_INIT	BOOL	R	Initial value of loop controller operating mode	Manu (0)	%KWr.m.c.2.12
Lead Lag	BOOL	R	Lead Lag function of Feed Forward branch	Not configurable (0)	%KWr.m.c.2.13
FF_UNI_BIP	BOOL	R	Type of Feed Forward process value: unipolar/bipolar	Not configurable (0)	%KWr.m.c.2.14
IMC	BOOL	R	IMC function of loop controller branch	Missing (0)	%KWr.m.c.2.15
CONFIG_3_B1	INT	R	Word grouping the various output configuration bits	No object	%KWr.m.c.3
Servo	BOOL	R	Type of output selected: Servo.	Not selected	%KWr.m.c.3.0
Servo2	BOOL	R	Type of output selected: Servo.	Not configurable (0)	%KWr.m.c.3.1
Analog1	BOOL	R	Type of output selected: Analog.	Selected	%KWr.m.c.3.2
Analog2	BOOL	R	Type of output selected: Analog.	Not configurable (0)	%KWr.m.c.3.3
PWM1	BOOL	R	Type of output selected: PWM.	Not selected	%KWr.m.c.3.4
PWM2	BOOL	R	Type of output selected: PWM.	Not configurable (0)	%KWr.m.c.3.5
POT_REV1	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.3.8
POT_REV2	BOOL	R	Servo copy direction	Not configurable (0)	%KWr.m.c.3.9
POT_VAL1_INIT	BOOL	R	Presence of Servo copy.	No (0)	%KWr.m.c.3.10
POT_VAL2_INIT	BOOL	R	Presence of (Reserved) Servo copy	Not configurable (0)	%KWr.m.c.3.11
ANALOG1_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar	Unipolar	%KWr.m.c.3.12
ANALOG2_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar	Not configurable (0)	%KWr.m.c.3.13
Loop name	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.4

Standard symbol	Type	Access	Meaning	Default Value	Address
Loop Unit	INT	R	Loop Unit.		%KWr.m.c.8
IDEM BOUCLE 1 %KW0	INT	R	Loop 2 process value. Unused functions have their bit set at 0.		%KWr.m.c.11
IDEM BOUCLE 1 %KW1	INT	R	Loop 2 setpoint.		%KWr.m.c.12
IDEM BOUCLE 1 %KW2	INT	R	Loop 2 loop controller and FF.		%KWr.m.c.13
IDEM BOUCLE 1 %KW3	INT	R	Loop 2 output.		%KWr.m.c.14
IDEM BOUCLE 1 %KW4	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.15
IDEM BOUCLE 1 %KW8	INT	R	Loop Unit.		%KWr.m.c.19
IDEM BOUCLE 1 %KW0	INT	R	Loop 3 process value. Unused functions have their bit set at 0.		%KWr.m.c.22
IDEM BOUCLE 1 %KW1	INT	R	Loop 3 setpoint.		%KWr.m.c.23
IDEM BOUCLE 1 %KW2	INT	R	Loop 3 loop controller and FF.		%KWr.m.c.24
IDEM BOUCLE 1 %KW3	INT	R	Loop 3 output.		%KWr.m.c.25
IDEM BOUCLE 1 %KW4	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.26
IDEM BOUCLE 1 %KW8	INT	R	Loop Unit.		%KWr.m.c.30
Trigger mode of loop 1	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.34.0
Trigger mode of loop 2	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.35.0
Trigger mode of loop 3	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.36.0

10.5 Language Objects Associated with the Single Loop Cascade

Subject of this Section

 This section describes the cascade loop language objects.

What Is in This Section?

 This section contains the following topics:

Topic	Page
Detailed Description of Default Language Objects for the T_PROC_CASC_LOOP-Type IODDT	275
Detailed Description of Diagnostic Language Objects for the T_PROC_CASC_LOOP-type IODDT	281
Detailed Description of Process Control Language Objects for the T_PROC_CASC_LOOP-Type IODDT	284
Configuration Language Objects	289

Detailed Description of Default Language Objects for the T_PROC_CASC_LOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_CASC_LOOP-type IODDT. These **default** language objects are associated to the cascade loop. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_CASC_LOOP-type IODDT_VAR1.

Comments

- Generally speaking, the meaning of any bit is provided for State 1 of said bit. In some specific instances, we provide a meaning for each state of the bit.
- Not all bits are utilized.

%lr.m.c.ERR Error Bit

The following table explains the meaning of the CH_ERROR (%lr.m.c.ERR) error bit.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Process control channel error bit.	%lr.m.c.ERR

Exchange Execution Indicators: EXCH_STS

The following table explains the various meanings of EXCH_STS (%MWr.m.c.0) channel exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Reading of channel status words in progress.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	Current parameter exchange in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	The exchange of Adjustment Parameters is in progress.	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	The module reconfiguration is in progress.	%MWr.m.c.0.15

Exchange Report: EXCH_RPT

The following table explains the various meanings of EXCH_RPT (%MW.r.m.c.1) reporting bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Reading error for channel status words.	%MW.r.m.c.1.0
CMD_ERR	BOOL	R	Error during command parameter exchange.	%MW.r.m.c.1.1
ADJ_ERR	BOOLB OOL	R	Error while exchanging adjustment parameters.	%MW.r.m.c.1.2
RECONF_ERR	BOOL	R	Error while reconfiguring the channel. (1 = failure)	%MW.r.m.c.1.15

Standard errors for CH_FLT_B1

The following table explains the various meanings of the CH_FLT_B1 (%MW.r.m.c.2) status word bits.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT	BOOL	R	Fatal internal error.	%MW.r.m.c.2.4
CONF_FLT	BOOL	R	Configuration error.	%MW.r.m.c.2.5
MISSING_ADDR_M	BOOL	R	Missing IMC master loop register address.	%MW.r.m.c.2.6
WARN	BOOL	R	Error Sum.	%MW.r.m.c.2.7
STS_ERR_CALC_CORR_M	BOOL	R	Master loop controller branch calculation error.	%MW.r.m.c.2.8
STS_ERR_FLOT_CORR_M	BOOL	R	Master loop controller branch floating point error.	%MW.r.m.c.2.9
STS_ERR_CALC_PV_M	BOOL	R	Master PV branch calculation error.	%MW.r.m.c.2.10
STS_ERR_FLOT_PV_M	BOOL	R	Master PV branch floating point error.	%MW.r.m.c.2.11
STS_ERR_SCALE_PV_M	BOOL	R	Master PV branch incorrect scale.	%MW.r.m.c.2.12

CH_STATUS2 status word

The following table explains the various meanings of the CH_STATUS2 (%MWr.m.c.3) status word bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR_CALC_OUT	BOOL	R	OUT branch calculation error.	%MWr.m.c.3.0
STS_ERR_FLOT_OUT	BOOL	R	OUT branch floating point error.	%MWr.m.c.3.1
STS_ERR_TH_SPLRG	BOOL	R	Split Range function thresholds incorrect.	%MWr.m.c.3.2
STS_ERR_SCALE_OUT1	BOOL	R	OUT1 branch scale incorrect.	%MWr.m.c.3.3
STS_ERR_SCALE_OUT2	BOOL	R	OUT2 branch scale incorrect.	%MWr.m.c.3.4
STS_ERR_COPY_POS	BOOL	R	Missing position copy address.	%MWr.m.c.3.5
MISSING_ADDR_E	BOOL	R	Missing IMC slave loop register address.	%MWr.m.c.3.6
STS_ERR_CALC_CORR_E	BOOL	R	Slave loop controller branch calculation error.	%MWr.m.c.3.8
STS_ERR_FLOT_CORR_E	BOOL	R	Slave loop controller branch floating point error.	%MWr.m.c.3.9
STS_ERR_CALC_PV_E	BOOL	R	Slave PV branch calculation error.	%MWr.m.c.3.10
STS_ERR_FLOT_PV_E	BOOL	R	Slave PV branch floating point error.	%MWr.m.c.3.11
STS_ERR_SCALE_PV_E	BOOL	R	Slave PV branch incorrect scale.	%MWr.m.c.3.12

STATUS1_M status word

The following table explains the various meanings of the STATUS1_M (%MW.r.m.c.4) status word bits. Word grouping the different Process value/Setpoint status bits

Standard symbol	Type	Access	Meaning	Address
STS_PV_SIM_M	BOOL	R	Process value simulation state.	%MW.r.m.c.4.1
STS_PV_H_LIM_M	BOOL	R	Upper limit on process value branch (PV_SUP).	%MW.r.m.c.4.2
STS_PV_L_LIM_M	BOOL	R	Lower limit on process value branch (PV_INF).	%MW.r.m.c.4.3
STS_SP_H_LIM_M	BOOL	R	Upper limit on setpoint branch (SP_SUP).	%MW.r.m.c.4.4
STS_SP_L_LIM_M	BOOL	R	Lower limit on setpoint branch (SP_INF).	%MW.r.m.c.4.5
STS_L_R_M	BOOL	R	State of the selected Remote/Local setpoint.	%MW.r.m.c.4.6
STS_R1_R2_M	BOOL	R	State of selected setpoint.	%MW.r.m.c.4.7
STS_ALARMS_M	BOOL	R	Sum of process value alarms.	%MW.r.m.c.4.8
STS_HH_M	BOOL	R	Very high alarm.	%MW.r.m.c.4.9
STS_H_M	BOOL	R	High alarm.	%MW.r.m.c.4.10
STS_L_M	BOOL	R	Low Alarm.	%MW.r.m.c.4.11
STS_LL_M	BOOL	R	Very low alarm.	%MW.r.m.c.4.12
STS_DEVH_M	BOOL	R	Upper threshold for Process value/Setpoint deviation (>0).	%MW.r.m.c.4.13
STS_DEVL_M	BOOL	R	Lower threshold for Process value/Setpoint deviation (<0).	%MW.r.m.c.4.14

STATUS2_M status word

The following table explains the various meanings of the STATUS2_M (%MW.r.m.c.5) status word bits. Word grouping various status bits for master controller.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_M	BOOL	R	Auto-tuning in progress.	%MW.r.m.c.5.0
STS_TR_S_M	BOOL	R	PID in tracking mode (open cascade).	%MW.r.m.c.5.1
STS_M_A_M	BOOL	R	Status of PID Operation Mode.	%MW.r.m.c.5.3
STS_OUT_L_LIM_M	BOOL	R	Output lower limit reached.	%MW.r.m.c.5.8
STS_OUT_H_LIM_M	BOOL	R	Output upper limit reached.	%MW.r.m.c.5.9
STS_TOP_NEXT_CYC_M	BOOL	R	Sampling pulse on next cycle.	%MW.r.m.c.5.10
STS_TOP_CUR_CYC_M	BOOL	R	Sampling pulse in current cycle.	%MW.r.m.c.5.11
STS_FF_SIM_M	BOOL	R	Simulation state of Feed Forward process value.	%MW.r.m.c.5.12
STS_OUT_CLAMP_LOW	BOOL	R	Output of slanting master in descending order.	%MW.r.m.c.5.13
STS_OUT_CLAMP_HIGH	BOOL	R	Output of slanting master in ascending order.	%MW.r.m.c.5.14

STATUS1_E status word

The following table explains the various meanings of the STATUS1_E (%MWr.m.c.7) status word bits. Word grouping the different Process value/Setpoint slave status bits.

Standard symbol	Type	Access	Meaning	Address
STS_PV_SIM_E	BOOL	R	Process value simulation state.	%MWr.m.c.7.1
STS_PV_H_LIM_E	BOOL	R	Upper limit on process value branch (PV_SUP).	%MWr.m.c.7.2
STS_PV_L_LIM_E	BOOL	R	Lower limit on process value branch (PV_INF).	%MWr.m.c.7.3
STS_SP_H_LIM_E	BOOL	R	Upper limit on setpoint branch (SP_SUP).	%MWr.m.c.7.4
STS_SP_L_LIM_E	BOOL	R	Lower limit on setpoint branch (SP_INF).	%MWr.m.c.7.5
STS_L_R_E	BOOL	R	State of the selected Remote/Local setpoint.	%MWr.m.c.7.6
STS_ALARMS_E	BOOL	R	Sum of process value alarms.	%MWr.m.c.7.8
STS_HH_E	BOOL	R	Very high alarm.	%MWr.m.c.7.9
STS_H_E	BOOL	R	High alarm.	%MWr.m.c.7.10
STS_L_E	BOOL	R	Low Alarm.	%MWr.m.c.7.11
STS_LL_E	BOOL	R	Very low alarm.	%MWr.m.c.7.12
STS_DEVH_E	BOOL	R	Upper threshold for Process value/Setpoint deviation (>0).	%MWr.m.c.7.13
STS_DEVL_E	BOOL	R	Lower threshold for Process value/Setpoint deviation (<0).	%MWr.m.c.7.14
STS_THLD_DONE_E	BOOL	R	Totaling threshold reached.	%MWr.m.c.7.15

STATUS2_E status word

The following table explains the various meanings of the STATUS2_E (%MWr.m.c.8) status word bits. Word grouping various status bits for slave controller.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_E	BOOL	R	Auto-tuning in progress.	%MWr.m.c.8.0
STS_TR_S_E	BOOL	R	Tracking Mode switch.	%MWr.m.c.8.1
STS_M_A_E	BOOL	R	Status of PID Operation Mode.	%MWr.m.c.8.3
STS_RAISE1_E	BOOL	R	Open command.	%MWr.m.c.8.4
STS_LOWER1_E	BOOL	R	Close command.	%MWr.m.c.8.1
STS_RAISE2_E	BOOL	R	Branch output 2 open command.	%MWr.m.c.8.6
STS_LOWER2_E	BOOL	R	Branch output 2 close command.	%MWr.m.c.8.7
STS_OUT_L_LIM_E	BOOL	R	The calculated output of the PID is greater than or equal to OUT_SUP.	%MWr.m.c.8.8
STS_OUT_H_LIM_E	BOOL	R	The calculated output of the PID is smaller than or equal to OUT_INF.	%MWr.m.c.8.9
STS_TOP_NEXT_CYC_E	BOOL	R	Sampling pulse on next cycle.	%MWr.m.c.8.10
STS_TOP_CUR_CYC_E	BOOL	R	Sampling pulse in current cycle.	%MWr.m.c.8.11

Detailed Description of Diagnostic Language Objects for the T_PROC_CASC_LOOP-type IODDT

At a Glance

This section describes the language objects for the T_PROC_CASC_LOOP-type IODDT. These **diagnostic** language objects are associated to the cascade loop. The section groups word objects whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_CASC_LOOP-type IODDT_VAR1.

STATUS3_M Status Word

The following table explains the various meanings of the STATUS3_M(%MWr.m.c.6) status word bits. Word grouping fine diagnostics of different errors (process value, setpoint, Feed Forward).

Standard symbol	Type	Access	Meaning	Address
Xi_WARN_M	BOOL	R	Xi parameter check error.	%MWr.m.c.6.0
Yi_WARN_M	BOOL	R	Yi parameter check error.	%MWr.m.c.6.1
INP_INFR1_WARN_M	BOOL	R	Error while checking parameters INP_INFR1 and INP_SUPR1.	%MWr.m.c.6.2
INP_INFR2_WARN_M	BOOL	R	INP_INFR2 and INP_SUPR2 parameters check error.	%MWr.m.c.6.3
RATIO_WARN_M	BOOL	R	RATIO_MIN and RATIO_MAX parameter check error.	%MWr.m.c.6.4
FF_CALC_WARN_M	BOOL	R	Feed Forward calculation error.	%MWr.m.c.6.5
FF_FLOAT_WARN_M	BOOL	R	Feed Forward floating point error.	%MWr.m.c.6.6
OUT_FF_WARN_M	BOOL	R	OUTFF_INF and OUTFF_SUP parameters check error.	%MWr.m.c.6.7
SP_MIN_WARN_M	BOOL	R	Error while checking SP_MIN et SP_MAX parameters.	%MWr.m.c.6.9
SP_CALC_WARN_M	BOOL	R	Setpoint calculation error.	%MWr.m.c.6.10
SP_FLOAT_WARN_M	BOOL	R	Setpoint floating point error.	%MWr.m.c.6.11

STATUS3_E Status Word

The following table explains the various meanings of the STATUS3_E (%MWr.m.c.9) status word bits. Word grouping output diagnostics.

Standard symbol	Type	Access	Meaning	Address
POT_VAL1_E	BOOL	R	Servo Operation, with copy,	%MWr.m.c.9.0
POT_VAL2_E	BOOL	R	Servo with copy operation (Reserved).	%MWr.m.c.9.1
RAISE_STOP1_E	BOOL	R	Open limit reached on servo drive.	%MWr.m.c.9.2
LOWER_STOP1_E	BOOL	R	Close limit reached on servo drive	%MWr.m.c.9.3
RAISE_STOP2_E	BOOL	R	Open limit reached on servo drive.	%MWr.m.c.9.4
LOWER_STOP2_E	BOOL	R	Close limit reached on servo drive.	%MWr.m.c.9.5
OVER_TOT_WARN_E	BOOL	R	Output capacity overshoot error on Totaling function.	%MWr.m.c.9.8
SP_MIN_WARN_E	BOOL	R	Error while checking SP_MIN et SP_MAX parameters.	%MWr.m.c.9.9
SP_CALC_WARN_E	BOOL	R	Setpoint calculation error.	%MWr.m.c.9.10
SP_FLOAT_WARN_E	BOOL	R	Setpoint floating point error.	%MWr.m.c.9.11

STATUS4 Status Word

The following table explains the various meanings of the STATUS4 (%MWr.m.c.10) status word bits. Word grouping auto-tuning diagnostics.

Standard symbol	Type	Access	Meaning	Address
AT_FAILED	BOOL	R	Auto-tuning failed.	%MWr.m.c.10.0
AT_ABORTED	BOOL	R	Auto-tuning diagnostics interrupted.	%MWr.m.c.10.1
AT_ERR_PARAM	BOOL	R	Auto-tuning diagnostics parameter error.	%MWr.m.c.10.2
AT_PWF_OR_EFB_FAIL	BOOL	R	Auto-tuning diagnostics system error or power failure.	%MWr.m.c.10.3
AT_ERR_SATUR	BOOL	R	Auto-tuning diagnostics process value saturation.	%MWr.m.c.10.4
AT_DV_TOO_SMALL	BOOL	R	Auto-tuning diagnostics insufficient process value deviation.	%MWr.m.c.10.5
AT_TSAMP_HIGH	BOOL	R	Auto-tuning Diagnostics sampling pulse period too long.	%MWr.m.c.10.6
AT_INCONSIST_RESP	BOOL	R	Auto-tuning diagnostics inconsistent response.	%MWr.m.c.10.7
AT_NOT_STAB_INIT	BOOL	R	Auto-tuning diagnostics: initially unstable process value.	%MWr.m.c.10.8
AT_TMAX_TOO_SMALL	BOOL	R	Auto-tuning diagnostics: step duration too brief.	%MWr.m.c.10.9
AT_NOISE_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: process value noise too loud.	%MWr.m.c.10.10

Standard symbol	Type	Access	Meaning	Address
AT_TMAX_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: step duration too long.	%MWr.m.c.10.11
AT_OVERSHOOT	BOOL	R	Auto-tuning diagnostics: overshoot exceeding 10%.	%MWr.m.c.10.12
AT_UNDERSHOOT	BOOL	R	Auto-tuning Diagnostics: non-minimal step too long.	%MWr.m.c.10.13
AT_UNSYMMETRICAL_PT	BOOL	R	Auto-tuning diagnostics: procedure too asymmetrical.	%MWr.m.c.10.14
AT_INTEGRATING_PT	BOOL	R	Auto-tuning diagnostics: integrating process.	%MWr.m.c.10.15

Command Word

The following table explains the various meanings of the `ORDER_COMMAND` (%MWr.m.c.11) command word and of the `PARAM_COMMAND` (%MDr.m.c.12) command parameters command word. Read operation is performed by `READ_STS` (IODDT_VAR1).

Standard symbol	Type	Access	Meaning	Address
ORDER_COMMAND	INT	R/W	Command Word (<i>see page 232</i>).	%MWr.m.c.11
PARAM_COMMAND	DINT	R/W	Command Parameter.	%MDr.m.c.12

Detailed Description of Process Control Language Objects for the T_PROC_CASC_LOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_CASC_LOOP-type IODDT. These **process control** language objects are associated to the cascade loop.

Sample Variable Declaration: T_PROC_CASC_LOOP-type IODDT_VAR1.

Objects

The following table lists all process control language objects associated to the cascade loop.

Standard symbol	Type	Access	Meaning	Default Value	Address
AT_STEP	REAL	R/W	Auto-tuning step function amplitude	10.0	%MFr.m.c.14
AT_TMAX	REAL	R/W	Auto-tuning step function duration	100.0	%MFr.m.c.16
AT_PERF	REAL	R/W	Criterion for auto-tuning stability	0.5	%MFr.m.c.18
T_ECH_M	REAL	R/W	Master loop sampling period.	0.3	%MFr.m.c.20
OUTFF_M	REAL	R	Value of Feed Forward action on physical scale.	N/A	%MFr.m.c.22
OUT_MAN_M	REAL	R/W	Command Value	N/A	%MFr.m.c.24
DEV_M	REAL	R	Setpoint Process Value Deviation	N/A	%MFr.m.c.26
PV_M	REAL	R	Process value on physical scale	N/A	%MFr.m.c.28
SP_M	REAL	R	Setpoint value on physical scale	N/A	%MFr.m.c.30
PV_INF_M	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.32
PV_SUP_M	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.34
KP_M	REAL	R/W	PID proportional coefficient / IMC static gain.	1.0	%MFr.m.c.36
TI_M	REAL	R/W	PID integral time / IMC time constant.	0.0	%MFr.m.c.38
TD_M	REAL	R/W	PID derivative time / IMC delay.	0.0	%MFr.m.c.40
OUTBIAS_M	REAL	R/W	Bias on PID loop controller output / IMC BO/BF ratio.	0.0	%MFr.m.c.42
INT_BAND_M	REAL	R/W	Integral band.	0.0	%MFr.m.c.44
DBAND_M	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.46
KD_M	REAL	R/W	Derivative Filtering	10.0	%MFr.m.c.48
SP_MIN_M	REAL	R/W	Lower limit for master setpoint.	0.0	%MFr.m.c.50
SP_MAX_M	REAL	R/W	Upper limit for master setpoint.	100.0	%MFr.m.c.52
PV_LL_M	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.54

Standard symbol	Type	Access	Meaning	Default Value	Address
PV_L_M	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.56
PV_H_M	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.58
PV_HH_M	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.60
RATIO_M	REAL	R/W	Ratio Value	1.0	%MFr.m.c.62
RATIO_MIN_M	REAL	R/W	Minimum Ratio Value	0.0	%MFr.m.c.64
RATIO_MAX_M	REAL	R/W	Maximum Ratio Value	100.0	%MFr.m.c.66
RATIO_BIAS_M	REAL	R/W	Ratio Bias Value	0.0	%MFr.m.c.68
DEV_L_M	REAL	R/W	Lower threshold of deviation	0.0	%MFr.m.c.70
DEV_H_M	REAL	R/W	Upper threshold of deviation	0.0	%MFr.m.c.72
T_FILTER_M	REAL	R/W	Process value filtering time	0.0	%MFr.m.c.74
K_FILTER_M	REAL	R/W	Multiplication coefficient of process value filtering.	1.0	%MFr.m.c.76
FILT_OUT_M	REAL	R	Filtering output value.		%MFr.m.c.78
SQRT_OUT_M	REAL	R	Output value of square root		%MFr.m.c.80
E2_IN_M	REAL	R/W	Abscissa of first pulse in Segment S2	1428.0	%MFr.m.c.82
E3_IN_M	REAL	R/W	Abscissa of first pulse in Segment S3	2857.0	%MFr.m.c.84
E4_IN_M	REAL	R/W	Abscissa of first pulse in Segment S4	4285.0	%MFr.m.c.86
E5_IN_M	REAL	R/W	Abscissa of first pulse in Segment S5	5714.0	%MFr.m.c.88
E6_IN_M	REAL	R/W	Abscissa of first pulse in Segment S6	7143.0	%MFr.m.c.90
E7_IN_M	REAL	R/W	Abscissa of first pulse in Segment S7	8571.0	%MFr.m.c.92
E2_OUT_M	REAL	R/W	Ordinate of first pulse in Segment S2	14.28.0	%MFr.m.c.94
E3_OUT_M	REAL	R/W	Ordinate of first pulse in Segment S3	28.57	%MFr.m.c.96
E4_OUT_M	REAL	R/W	Ordinate of first pulse in Segment S4	42.85	%MFr.m.c.98
E5_OUT_M	REAL	R/W	Ordinate of first pulse in Segment S5	57.14	%MFr.m.c.100
E6_OUT_M	REAL	R/W	Ordinate of first pulse in Segment S6	71.43	%MFr.m.c.102
E7_OUT_M	REAL	R/W	Ordinate of first pulse in Segment S7	8571.0	%MFr.m.c.104
R_RATE_M	REAL	R/W	Setpoint speed increase limit	0.0	%MFr.m.c.106
D_RATE_M	REAL	R/W	Setpoint speed decrease limit	0.0	%MFr.m.c.108
SPEED_LIM_OUT_M	REAL	R	Setpoint speed limiter output value		%MFr.m.c.110
INP_INFR1_M	REAL	R/W	Lower scale of the master loop R1 setpoint.	0.0	%MFr.m.c.112
INP_SUPR1_M	REAL	R/W	Upper scale of master loop R1 setpoint.	100.0	%MFr.m.c.114
INP_INFR2_M	REAL	R/W	Lower scale of the master loop R2 setpoint.	0.0	%MFr.m.c.116
INP_SUPR2_M	REAL	R/W	Upper scale of master loop R2 setpoint.	100.0	%MFr.m.c.118
T1_FF_M	REAL	R/W	Feed Forward process value filtering time	0.0	%MFr.m.c.120

Standard symbol	Type	Access	Meaning	Default Value	Address
T2_FF_M	REAL	R/W	Feed Forward process value filtering time	0.0	%MFr.m.c.122
OUT_FF_INF_M	REAL	R/W	Lower limit of Feed Forward action	0.0	%MFr.m.c.124
OUT_FF_SUP_M	REAL	R/W	Upper limit of Feed Forward action	100.0	%MFr.m.c.126
KP_PREV_M	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.128
TI_PREV_M	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.130
TD_PREV_M	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.132
OUT1_E	REAL	R	Value of output command 1.	N/A	%MFr.m.c.134
OUT2_E	REAL	R	Value of output command 2.	N/A	%MFr.m.c.136
T_ECH_E	REAL	R/W	Slave loop sampling period.	0.3	%MFr.m.c.138
OUT_MAN_E	REAL	R/W	Slave loop command value.	N/A	%MFr.m.c.140
DEV_E	REAL	R	Setpoint Process Value Deviation	N/A	%MFr.m.c.142
PV_E	REAL	R	PV value on physical scale.	N/A	%MFr.m.c.144
SP_E	REAL	R	Setpoint value on physical scale.	N/A	%MFr.m.c.146
PV_INF_E	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.148
PV_SUP_E	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.150
KP_E	REAL	R/W	PID proportional coefficient / IMC static gain.	1.0	%MFr.m.c.152
TI_E	REAL	R/W	PID integral time / IMC time constant.	0.0	%MFr.m.c.154
TD_E	REAL	R/W	PID derivative time / IMC delay.	0.0	%MFr.m.c.156
OUTBIAS_E	REAL	R/W	Bias on PID loop controller output / IMC BO/BF ratio.	0.0	%MFr.m.c.158
INT_BAND_E	REAL	R/W	Integral band.	0.0	%MFr.m.c.160
DBAND_E	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.162
KD_E	REAL	R/W	Derivative Filtering	10.0	%MFr.m.c.164
OUTRATE_E	REAL	R/W	Limit of output variation speed.	0.0	%MFr.m.c.166
OUTRATE2_E	REAL	R/W	Limit of output 2 variable speed	0.0	%MFr.m.c.168
OUT1_INF_E	REAL	R/W	Lower limit for output 1	0.0	%MFr.m.c.170
OUT1_SUP_E	REAL	R/W	Upper limit for output 1	100.0	%MFr.m.c.172
SP_MIN_E	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.174
SP_MAX_E	REAL	R/W	Upper limit for setpoint	100.0	%MFr.m.c.176
OUT2_INF_E	REAL	R/W	Lower limit for Output 2.	0.0	%MFr.m.c.178
OUT2_SUP_E	REAL	R/W	Upper limit for Output 2.	100.0	%MFr.m.c.180
OUT1_TH1_E	REAL	R/W	Threshold 1 for Hot/Cold or Split Range output 1	0.0	%MFr.m.c.182

Standard symbol	Type	Access	Meaning	Default Value	Address
OUT1_TH2_E	REAL	R/W	Threshold 2 for Hot/Cold or Split Range output 1	50.0	%MFr.m.c.184
OUT2_TH1_E	REAL	R/W	Threshold 1 for Hot/Cold or Split Range output 2	50.0	%MFr.m.c.186
OUT2_TH2_E	REAL	R/W	Threshold 2 for Hot/Cold or Split Range output 2	100.0	%MFr.m.c.188
PV_LL_E	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.190
PV_L_E	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.192
PV_H_E	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.194
PV_HH_E	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.196
DEV_L_E	REAL	R/W	Lower threshold of deviation	0.0	%MFr.m.c.198
DEV_H_E	REAL	R/W	Upper threshold of deviation	0.0	%MFr.m.c.200
T_FILTER_E	REAL	R/W	Process value filtering time	0.0	%MFr.m.c.202
K_FILTER_E	REAL	R/W	Multiplication coefficient of process value filtering.	1.0	%MFr.m.c.204
FILT_OUT_E	REAL	R	Filtering output value.	N/A	%MFr.m.c.206
SQRT_OUT_E	REAL	R	Output value of square root	N/A	%MFr.m.c.208
THLD_E	REAL	R/W	Totaling limit.	1E+8	%MFr.m.c.210
R_RATE_E	REAL	R/W	Setpoint speed increase limit	0.0	%MFr.m.c.212
D_RATE_E	REAL	R/W	Setpoint speed decrease limit	0.0	%MFr.m.c.214
SPEED_LIM_OUT_E	REAL	R	Setpoint speed limiter output value	N/A	%MFr.m.c.216
T_MOTOR1_E	REAL	R/W	Open time for servo-controlled valve	10.0	%MFr.m.c.218
T_MINI1_E	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.220
T_MOTOR2_E	REAL	R/W	Open time for servo-controlled valve	10.0	%MFr.m.c.222
T_MINI2_E	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.224
KP_PREV_E	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.226
TI_PREV_E	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.228
TD_PREV_E	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.230
KS	REAL	R/W	IMC static gain	1.0	%MFr.m.c.232
OL_TIME	REAL	R/W	Time Constant in OL.	1.0	%MFr.m.c.234
T_DELAY	REAL	R/W	Current pure delay time	0.0	%MFr.m.c.236
CL_PERF	REAL	R/W	OL/CL Ratio	0.1	%MFr.m.c.238

Standard symbol	Type	Access	Meaning	Default Value	Address
PV_SIM_M	REAL	R/W	Simulated Process Value	N/A	%MWr.m.c.248
PV_SIM_E	INT	R/W	Simulated Process Value	N/A	%MWr.m.c.249
FF_SIM_M	INT	R/W	Simulated Feed Forward Input	N/A	%MWr.m.c.250

Configuration Language Objects

Description

The following table describes the process control language objects associated with the cascade loop.

Standard symbol	Type	Access	Meaning	Default Value	Address
CONFIG_0_M	INT	R	Word grouping the various configuration bits of the process value	No object	%KW.r.m.c.0
Filter	BOOL	R	Filtering function of the process value branch	Missing (0)	%KW.r.m.c.0.0
Function Generator	BOOL	R	Function Generator for process value branch	Missing (0)	%KW.r.m.c.0.1
Totaling	BOOL	R	Totaling Function for the process value branch	No object	%KW.r.m.c.0.2
Square Root	BOOL	R	Square root function of the process value branch	Missing (0)	%KW.r.m.c.0.3
Alarms	BOOL	R	Alarm Function of the process value branch	Present	%KW.r.m.c.0.4
PV_CLIP	BOOL	R	Lopping or non-lopping of process value	Missing (0)	%KW.r.m.c.0.8
EXTRAPOL	BOOL	R	Extrapolation of function generator	No (0)	%KW.r.m.c.0.9
PV_UNI_BIP	BOOL	R	Type: unipolar/bipolar of the process value.	Unipolar (0)	%KW.r.m.c.0.10
PV_EXTERNE	BOOL	R	Selection of Standard Process Value (0) vs. External Process Value (1).	Not selected (0)	%KW.r.m.c.0.11
CONFIG_1_M	INT	R	Word grouping the various configuration bits of the setpoint	No object	%KW.r.m.c.1
SP_SIMPLE	BOOL	R	Type of setpoint selected: simple.	Selected (1)	%KW.r.m.c.1.0
SP_Selection	BOOL	R	Type of setpoint: selection.	Not selected (0)	%KW.r.m.c.1.1
SPEED_LIMITER	INT	R	Speed limiter on setpoint	Not selected (0)	%KW.r.m.c.1.2
SP_SPP	BOOL	R	Type of setpoint selected: Programmer.	Not selected (0)	%KW.r.m.c.1.3
LR/L	BOOL	R	Speed limiter either on local setpoint, or in Remote/Local mode.	Local remote (0)	%KW.r.m.c.1.4
SEL_MIN	BOOL	R	Function selected for setpoint type Selection.	Missing (0)	%KW.r.m.c.1.8
SEL_MAX	BOOL	R	Function selected for setpoint type Selection.	Missing (0)	%KW.r.m.c.1.9

Standard symbol	Type	Access	Meaning	Default Value	Address
SEL_SWITCH	BOOL	R	Function selected for setpoint type Selection.	Present on Selection	%KWr.m.c.1.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KWr.m.c.1.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	R1 (0)	%KWr.m.c.1.12
SP_RATIO	BOOL	R	Type of setpoint selected: Ratio.	Not selected (0)	%KWr.m.c.1.13
SP_LIMITER	BOOL	R	Setpoint Limiter (e.g. PARAM_SP).	Not present	%KWr.m.c.1.14
SP_FOLW	BOOL	R	Tracking Setpoint (0).	No setpoint	%KWr.m.c.1.15
CONFIG_2_M	INT	R	Word grouping various configuration bits for loop controller and Feed Forward.	No object	%KWr.m.c.2
PID	BOOL	R	PID function of loop controller branch	Present (always)	%KWr.m.c.2.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	No object	%KWr.m.c.2.1
ONOFF3	INT	R	ON/OFF 3-states branch of loop controller	No object	%KWr.m.c.2.2
SPLRG/HOTCOLD	BOOL	R	OR Hot/Cold and Split Range presence bits	No object	%KWr.m.c.2.3
Split/Range	BOOL	R	Loop controller branch Split Range function.	No object	%KWr.m.c.2.4
Hot/Cold	BOOL	R	Hot/Cold function of loop controller branch	No object	%KWr.m.c.2.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.2.6
Feed Forward	BOOL	R	Presence of Feed Forward input	Missing (0)	%KWr.m.c.2.7
BUMP	BOOL	R	Bump Management when changing operating mode.	With bumps (1)	%KWr.m.c.2.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.2.9
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.2.10
REV_DIR	BOOL	R	Type of loop controller action: • 1: opposite action, a positive shift (PV-SP) generates a rise of the output value • 0: direct action, a positive shift (PV-SP) generates a fall of the output value	PID opposite action (1)	%KWr.m.c.2.11

Standard symbol	Type	Access	Meaning	Default Value	Address
MANU/AUTO_INIT	BOOL	R	Initial value of loop controller operating mode	Auto (1)	%KWr.m.c.2.12
Lead Lag	BOOL	R	Lead Lag function of Feed Forward branch	Missing (0)	%KWr.m.c.2.13
FF_UNI_BIP	BOOL	R	Type of Feed Forward process value: unipolar/bipolar	Unipolar	%KWr.m.c.2.14
IMC	BOOL	R	IMC function of loop controller branch	Missing (0)	%KWr.m.c.2.15
Loop name	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.3
Loop Unit	INT	R	Loop Unit.		%KWr.m.c.7
CONFIG_0_E	BOOL	R	This word gathers all different process value configuration bits.	No object	%KWr.m.c.10
Filter	INT	R	Filtering function of the process value branch	Missing (0)	%KWr.m.c.10.0
Function Generator	BOOL	R	Function Generator for process value branch	No object	%KWr.m.c.10.1
Totaling	BOOL	R	Totaling Function for the process value branch	Missing (0)	%KWr.m.c.10.2
Square Root	BOOL	R	Square root function of the process value branch	Missing (0)	%KWr.m.c.10.3
Alarms	BOOL	R	Alarm Function of the process value branch	Present	%KWr.m.c.10.4
PV_CLIP	BOOL	R	Lopping or non-lopping of measurement.	Missing (0)	%KWr.m.c.10.8
EXTRAPOL	BOOL	R	Extrapolation of function generator	No object	%KWr.m.c.10.9
PV_UNI_BIP	BOOL	R	Type of process value: unipolar/bipolar	Unipolar (0)	%KWr.m.c.10.10
Totaling: Process value unit	BOOL	R	(bit 13 = 0, bit 14 = 0): phys/ms (bit 13 = 1, bit 14 = 0): phys/s	1	%KWr.m.c.10.13
Totaling: Process value unit	BOOL	R	(bit 13 = 1, bit 14 = 0): phys/mn (bit 13 = 1, bit 14 = 1): phys/h	0	%KWr.m.c.10.14
CONFIG_1_E	INT	R	Word grouping the various configuration bits of the setpoint	No object	%KWr.m.c.11
SP_SIMPLE	BOOL	R	Type of setpoint selected: Single.	Selected (1)	%KWr.m.c.11.0
SP_Selection	BOOL	R	Type of setpoint: Selection.	No object	%KWr.m.c.11.1
SPEED_LIMITER	BOOL	R	Speed limiter on setpoint	Not selected (0)	%KWr.m.c.11.2
SP_SPP	BOOL	R	Type of setpoint selected: Programmer.	No object	%KWr.m.c.11.3

Standard symbol	Type	Access	Meaning	Default Value	Address
LR/L	BOOL	R	Speed limiter either on local setpoint, or in Remote/Local mode.	Local remote (0)	%KWr.m.c.11.4
SEL_MIN	BOOL	R	Function selected for setpoint type Selection.	No object	%KWr.m.c.11.8
SEL_MAX	BOOL	R	Function selected for setpoint type Selection.	No object	%KWr.m.c.11.9
SEL_SWITCH	BOOL	R	Function selected for setpoint type Selection.	No object	%KWr.m.c.11.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KWr.m.c.11.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	No object	%KWr.m.c.11.12
SP_RATIO	BOOL	R	Type of setpoint selected: Ratio.	No object	%KWr.m.c.11.13
SP_LIMITER	BOOL	R	Setpoint Limiter (e.g. PARAM_SP).	Not present	%KWr.m.c.11.14
SP_FOLW	BOOL	R	Tracking setpoint.	Non tracking setpoint (0)	%KWr.m.c.11.15
CONFIG_2_E	INT	R	Word grouping various configuration bits for loop controller and Feed Forward.	No object	%KWr.m.c.12
PID	BOOL	R	PID function of loop controller branch	Present (always)	%KWr.m.c.12.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	No object	%KWr.m.c.12.1
ONOFF3	BOOL	R	ON/OFF 3-states branch of loop controller	No object	%KWr.m.c.12.2
SPLRG/HOTCOLD	BOOL	R	Or Hot/Cold and Split Range presence bits.	No object	%KWr.m.c.12.3
Split/Range	BOOL	R	Loop controller branch Split Range function.	Missing (0)	%KWr.m.c.12.4
Hot/Cold	BOOL	R	Hot/Cold function of loop controller branch	Not selected	%KWr.m.c.12.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.12.6
Feed Forward	BOOL	R	Presence of Feed Forward input	No object	%KWr.m.c.12.7
BUMP	BOOL	R	Bump Management when changing operating mode.	With bumps (1)	%KWr.m.c.12.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.12.9
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.12.10

Standard symbol	Type	Access	Meaning	Default Value	Address
REV_DIR	BOOL	R	Type of loop controller action: 1: opposite action, a positive shift (PV-SP) generates a rise of the output value 0: direct action, a positive shift (PV-SP) generates a fall of the output value	PID opposite action (1)	%KWr.m.c.12.11
MANU/AUTO_INIT	BOOL	R	Initial value of loop controller operating mode	Manu (0)	%KWr.m.c.12.12
Lead Lag	BOOL	R	Lead Lag function of Feed Forward branch	No object	%KWr.m.c.12.13
FF_UNI_BIP	BOOL	R	Type of Feed Forward process value: unipolar/bipolar	No object	%KWr.m.c.12.14
IMC	BOOL	R	IMC function of loop controller branch	Missing (0)	%KWr.m.c.12.15
CONFIG_3_E	INT	R	Word grouping the various output configuration bits	No object	%KWr.m.c.13
Servo	BOOL	R	Type of output selected: Servo.	Not selected	%KWr.m.c.13.0
Servo2	BOOL	R	Type of output selected: Servo.	Not selected	%KWr.m.c.13.1
Analog1	BOOL	R	Type of output selected: Analog.	Not selected	%KWr.m.c.13.2
Analog2	BOOL	R	Type of output selected: Analog.	Not selected	%KWr.m.c.13.3
PWM1	BOOL	R	Type of output selected: PWM.	Not selected	%KWr.m.c.13.4
PWM2	BOOL	R	Type of output selected: PWM.	Not selected	%KWr.m.c.13.5
POT_REV1	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.13.8
POT_REV2	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.13.9
POT_VAL1_INIT	BOOL	R	Presence of Servo copy.	No (0)	%KWr.m.c.13.10
POT_VAL2_INIT	BOOL	R	Presence of (Reserved) Servo copy	Yes (1)	%KWr.m.c.13.11
ANALOG1_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar	Unipolar	%KWr.m.c.13.12
ANALOG2_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar	Unipolar (0)	%KWr.m.c.13.13
Loop name	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.14
Loop Unit	INT	R	Loop Unit.		%KWr.m.c.18
Trigger mode of Master loop	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.21.0
Trigger mode of Slave loop	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.22.0

10.6 Language Objects Associated with the Self-selective Loop

Subject of this Section

 This section describes the language objects of the self-selective loops.

What Is in This Section?

 This section contains the following topics:

Topic	Page
Detailed Description of Default Language Objects for the T_PROC_CONST_LOOP-Type IODDT	295
Detailed Description of Diagnostic Language Objects for the T_PROC_CONST_LOOP-Type IODDT	301
Detailed Description of Process Control Language Objects for the T_PROC_CONST_LOOP-Type IODDT	304
Configuration Language Objects	309

Detailed Description of Default Language Objects for the T_PROC_CONST_LOOP-Type IODDT

At a Glance

This section describes the language objects for the **T_PROC_CONST_LOOP-type** IODDT. These **default** language objects are associated to the auto-selector (or self-selective) loop. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: **T_PROC_CONST_LOOP-type** IODDT_VAR1.

Comments

- Generally speaking, the meaning of any bit is provided for State 1 of said bit. In some specific instances, we provide a meaning for each state of the bit.
- Not all bits are utilized.

%lr.m.c.ERR Error Bit

The following table explains the meaning of the **CH_ERROR** (%lr.m.c.ERR) error bit.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Process control channel error bit.	%lr.m.c.ERR

Exchange Execution Indicators: EXCH_STS

The following table explains the various meanings of **EXCH_STS** (%MWr.m.c.0) channel exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Channel status words are in the process of being read.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	The exchange of Command Parameters is in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	The exchange of Adjustment Parameters is in progress.	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	The module reconfiguration is in progress.	%MWr.m.c.0.15

Exchange Report: EXCH_RPT

The following table explains the various meanings of EXCH_RPT (%MWr.m.c.1) reporting bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error while reading the channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error while exchanging command parameters.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error while reconfiguring the channel. (1 = failure)	%MWr.m.c.1.15

Standard Errors for CH_FLT

The following table explains the various meanings of the CH_FLT (%MWr.m.c.2) status word bits.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT	BOOL	R	Fatal internal error.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configuration error.	%MWr.m.c.2.5
MISSING_ADDR_2	BOOL	R	Missing secondary loop IMC register address.	%MWr.m.c.2.6
WARN	BOOL	R	Sum of warnings.	%MWr.m.c.2.7
STS_ERR_CALC_CORR_2	BOOL	R	Loop controller branch calculation error.	%MWr.m.c.2.8
STS_ERR_FLOT_CORR_2	BOOL	R	Loop controller branch floating point error.	%MWr.m.c.2.9
STS_ERR_CALC_PV_2	BOOL	R	PV branch calculation error.	%MWr.m.c.2.10
STS_ERR_FLOT_PV_2	BOOL	R	PV branch floating point error.	%MWr.m.c.2.11
STS_ERR_SCALE_PV_2	BOOL	R	PV1 branch scale incorrect.	%MWr.m.c.2.12

CH_STATUS2 Status Word

The following table explains the various meanings of the CH_STATUS2 (%MWr.m.c.3) status word bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR_CALC_OUT	BOOL	R	OUT branch calculation error.	%MWr.m.c.3.0
STS_ERR_FLOT_OUT	BOOL	R	OUT branch floating point error.	%MWr.m.c.3.1
STS_ERR_TH_SPLRG	BOOL	R	Split Range function thresholds incorrect.	%MWr.m.c.3.2
STS_ERR_CALC_CONT	BOOL	R	Secondary branch calculation error.	%MWr.m.c.3.3
STS_ERR_COPY_POS	BOOL	R	Missing position copy address.	%MWr.m.c.3.4
MISSING_ADDR_1	BOOL	R	Missing main loop IMC register address.	%MWr.m.c.3.6
STS_ERR_CALC_CORR_1	BOOL	R	Loop controller branch calculation error.	%MWr.m.c.3.8

Standard symbol	Type	Access	Meaning	Address
STS_ERR_FLOT_CORR_1	BOOL	R	Loop controller branch floating point error.	%MW.r.m.c.3.9
STS_ERR_CALC_PV_1	BOOL	R	PV branch calculation error.	%MW.r.m.c.3.10
STS_ERR_FLOT_PV_1	BOOL	R	PV branch floating point error.	%MW.r.m.c.3.11
STS_ERR_SCALE_PV_1	BOOL	R	PV branch incorrect scale.	%MW.r.m.c.3.12
STS_ERR_SCALE_OUT_1	BOOL	R	C1 branch incorrect scale.	%MW.r.m.c.3.13
STS_ERR_SCALE_OUT_2	BOOL	R	C2 branch incorrect scale.	%MW.r.m.c.3.14
STS_ERR_SCALE	BOOL	R	OR scale errors.	%MW.r.m.c.3.15

STATUS1_C1 Status Word

The following table explains the various meanings of the STATUS1_C1 (%MW.r.m.c.4) status word bits. Word grouping the various Process value/Setpoint status bits of the main loop.

Standard symbol	Type	Access	Meaning	Address
STS_HOLD_TOT	BOOL	R	Freeze of Totaling function.	%MW.r.m.c.4.0
STS_PV_SIM_C1	BOOL	R	Simulated process value.	%MW.r.m.c.4.1
STS_PV_H_LIM_C1	BOOL	R	Upper limit on process value.	%MW.r.m.c.4.2
STS_PV_L_LIM_C1	BOOL	R	Lower limit on process value.	%MW.r.m.c.4.3
STS_SP_H_LIM_C1	BOOL	R	Upper limit on setpoint.	%MW.r.m.c.4.4
STS_SP_L_LIM_C1	BOOL	R	Lower limit on setpoint.	%MW.r.m.c.4.5
STS_L_R_C1	BOOL	R	Remote Setpoint (1) Local Setpoint (0).	%MW.r.m.c.4.6
STS_R1_R2_C1	BOOL	R	Remote2 Setpoint (1) Remote1 Setpoint (0).	%MW.r.m.c.4.7
STS_ALARMS_C1	BOOL	R	OR logic of process value alarms.	%MW.r.m.c.4.8
STS_HH_C1	BOOL	R	Very high alarm.	%MW.r.m.c.4.9
STS_H_C1	BOOL	R	High alarm.	%MW.r.m.c.4.10
STS_L_C1	BOOL	R	Low Alarm.	%MW.r.m.c.4.11
STS_LL_C1	BOOL	R	Very low alarm.	%MW.r.m.c.4.12
STS_DEVH_C1	BOOL	R	High alarm for Process value/Setpoint deviation (>0).	%MW.r.m.c.4.13
STS_DEVL_C1	BOOL	R	Lower alarm for Process value/Setpoint deviation (<0).	%MW.r.m.c.4.14
STS_THLD_DONE_C1	BOOL	R	Totaling threshold reached.	%MW.r.m.c.4.15

STATUS2_C1 Status Word

The following table explains the various meanings of the STATUS2_C1 (%MW.r.m.c.5) status word bits. Word grouping various status bits for master loop controller.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_C1	BOOL	R	Auto-tuning in progress.	%MW.r.m.c.5.0
STS_M_A_C1	BOOL	R	Status of PID Operation Mode.	%MW.r.m.c.5.1
STS_FF_SIM_C1	BOOL	R	Simulation state of Feed Forward process value.	%MW.r.m.c.5.2
STS_TOP_NEXT_CYCLE	BOOL	R	Sampling pulse on next cycle.	%MW.r.m.c.5.6
STS_TOP_CUR_CYCLE	BOOL	R	Sampling pulse in current cycle.	%MW.r.m.c.5.7
STS_TR_S	BOOL	R	Tracking in progress on global loop.	%MW.r.m.c.5.8
STS_M_A	BOOL	R	Global Manu / Auto.	%MW.r.m.c.5.9
STS_RAISE1	BOOL	R	Open command (global loop).	%MW.r.m.c.5.10
STS_LOWER1	BOOL	R	Close command (global loop).	%MW.r.m.c.5.11
STS_RAISE2	BOOL	R	Output 2 branch open command (global loop).	%MW.r.m.c.5.12
STS_LOWER2	BOOL	R	Output 2 branch close command (global loop).	%MW.r.m.c.5.13
STS_OUT_L_LIM	BOOL	R	Upper limit reached for selected PID output (global loop).	%MW.r.m.c.5.14
STS_OUT_H_LIM	BOOL	R	Lower limit reached for selected PID output (global loop).	%MW.r.m.c.5.15

STATUS1_C2 Status Word

The following table explains the various meanings of the STATUS1_C2 (%MWr.m.c.7) status word bits. Word grouping the various Process value/Setpoint status bits of the secondary loop.

Standard symbol	Type	Access	Meaning	Address
STS_PV_SIM_C2	BOOL	R	Process value simulation status.	%MWr.m.c.7.1
STS_PV_H_LIM_C2	BOOL	R	Upper limit on process value branch (PV_SUP).	%MWr.m.c.7.2
STS_PV_L_LIM_C2	BOOL	R	Lower limit on process value branch (PV_INF).	%MWr.m.c.7.3
STS_SP_H_LIM_C2	BOOL	R	Upper limit on setpoint branch.	%MWr.m.c.7.4
STS_SP_L_LIM_C2	BOOL	R	Lower limit on setpoint branch.	%MWr.m.c.7.5
STS_L_R_C2	BOOL	R	Remote Setpoint (1) Local Setpoint (0).	%MWr.m.c.7.6
STS_ALARMS_C2	BOOL	R	Sum of process value alarms.	%MWr.m.c.7.8
STS_HH_C2	BOOL	R	Very high alarm.	%MWr.m.c.7.9
STS_H_C2	BOOL	R	High alarm.	%MWr.m.c.7.10
STS_L_C2	BOOL	R	Low Alarm.	%MWr.m.c.7.11
STS_LL_C2	BOOL	R	Very low alarm.	%MWr.m.c.7.12
STS_DEVH_C2	BOOL	R	Upper threshold for Process value/Setpoint deviation (>0).	%MWr.m.c.7.13
STS_DEVL_C2	BOOL	R	Lower threshold for Process value/Setpoint deviation (<0).	%MWr.m.c.7.14
STS_THLD_DONE_C2	BOOL	R	Totaling threshold reached.	%MWr.m.c.7.15

STATUS2_C2 Status Word

The following table explains the various meanings of the STATUS2_C2 (%MWr.m.c.8) status word bits. Word grouping the various Loop Controller/Setpoint status bits of the secondary loop.

Standard symbol	Type	Access	Meaning	Address
STS_AT_RUNNING_C2	BOOL	R	Auto-tuning in progress.	%MWr.m.c.8.0
STS_M_A_C2	BOOL	R	Status of PID Operation Mode.	%MWr.m.c.8.1
INP_INFR1_WARN_C2	BOOL	R	Error while checking parameters INP_INFR1 and INP_SUPR1.	%MWr.m.c.8.8
SP_MIN_WARN_C2	BOOL	R	Error while checking SP_MIN et SP_MAX parameters.	%MWr.m.c.8.10
SP_CALC_WARN_C2	BOOL	R	Setpoint calculation error.	%MWr.m.c.8.11
SP_FLOAT_WARN_C2	BOOL	R	Setpoint floating point error.	%MWr.m.c.8.12
OVER_TOT_WARN_C2	BOOL	R	Totaling overflow error.	%MWr.m.c.8.13

STATUS3 Status Word

The following table explains the various meanings of the STATUS3 (%MW.r.m.c.9) status word bits. Word grouping the different output bits.

Standard symbol	Type	Access	Meaning	Address
POT_VAL1	BOOL	R	Servo Operation, with copy (global loop).	%MW.r.m.c.9.0
POT_VAL2	BOOL	R	Servo Operation, with copy (global loop).	%MW.r.m.c.9.1
RAISE_STOP1	BOOL	R	Open limit reached on Servo drive (global loop).	%MW.r.m.c.9.2
LOWER_STOP1	BOOL	R	Close limit reached on Servo drive (global loop).	%MW.r.m.c.9.3
RAISE_STOP2	BOOL	R	Open limit reached on Servo drive (global loop).	%MW.r.m.c.9.4
LOWER_STOP2	BOOL	R	Close limit reached on Servo drive (global loop).	%MW.r.m.c.9.5
STS_AS	BOOL	R	Selector positioned on auto-selector.	%MW.r.m.c.9.8
STS_DIR1	BOOL	R	Selector positioned on PID1 output.	%MW.r.m.c.9.9
STS_DIR2	BOOL	R	Selector positioned on PID2 output.	%MW.r.m.c.9.10
STS_SEL_PID1	BOOL	R	1 : Selected output = PID1 output. 0 : Selected output = PID2 output.	%MW.r.m.c.9.11

Detailed Description of Diagnostic Language Objects for the T_PROC_CONST_LOOP-Type IODDT

At a Glance

This section describes the language objects for the **T_PROC_CONST_LOOP-type** IODDT. These **diagnostic** language objects are associated to the auto-selector loop. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_CONST_LOOP-type IODDT_VAR1.

STATUS3_C1 Status Word

The following table explains the various meanings of the STATUS3_C1 (%MWr.m.c.6) status word bits. Word grouping fine diagnostics of different warnings (process value, setpoint, Feed Forward) main loop.

Standard symbol	Type	Access	Meaning	Address
Xi_WARN_C1	BOOL	R	Xi parameter check error.	%MWr.m.c.6.0
Yi_WARN_C1	BOOL	R	Yi parameter check error.	%MWr.m.c.6.1
RATIO_WARN_C1	BOOL	R	RATIO_MIN and RATIO_MAX parameter check error.	%MWr.m.c.6.2
FF_CALC_WARN_C1	BOOL	R	Feed Forward calculation error.	%MWr.m.c.6.3
FF_FLOAT_WARN_C1	BOOL	R	Feed Forward floating point error.	%MWr.m.c.6.4
OUT_FF_WARN_C1	BOOL	R	OUTFF_INF and OUTFF_SUP parameters check error.	%MWr.m.c.6.5
INP_INFR1_WARN_C1	BOOL	R	Error while checking parameters INP_INFR1 and INP_SUPR1.	%MWr.m.c.6.8
INP_INFR2_WARN_C1	BOOL	R	INP_INFR2 and INP_SUPR2 parameters check error.	%MWr.m.c.6.9
SP_MIN_WARN_C1	BOOL	R	Error while checking SP_MIN et SP_MAX parameters.	%MWr.m.c.6.10
SP_CALC_WARN_C1	BOOL	R	Setpoint calculation error.	%MWr.m.c.6.11
SP_FLOAT_WARN_C1	BOOL	R	Setpoint floating point error.	%MWr.m.c.6.12
OVER_TOT_WARN_C1	BOOL	R	Error on totaling overflow.	%MWr.m.c.6.13

STATUS4 Status Word

The following table explains the various meanings of the STATUS4 (%MWr.m.c.10) status word bits. Word grouping auto-tuning diagnostics.

Standard symbol	Type	Access	Meaning	Address
AT_FAILED	BOOL	R	Auto-tuning failed.	%MWr.m.c.10.0
AT_ABORTED	BOOL	R	Auto-tuning diagnostics interrupted.	%MWr.m.c.10.1
AT_ERR_PARAM	BOOL	R	Auto-tuning diagnostics parameter error.	%MWr.m.c.10.2
AT_PWF_OR_EFB_FAIL	BOOL	R	Auto-tuning diagnostics system error or power failure.	%MWr.m.c.10.3
AT_ERR_SATUR	BOOL	R	Auto-tuning diagnostics process value saturation.	%MWr.m.c.10.4
AT_DV_TOO_SMALL	BOOL	R	Auto-tuning diagnostics insufficient process value deviation.	%MWr.m.c.10.5
AT_TSAMP_HIGH	BOOL	R	Auto-tuning Diagnostics sampling pulse period too long.	%MWr.m.c.10.6
AT_INCONSIST_RESP	BOOL	R	Auto-tuning diagnostics inconsistent response.	%MWr.m.c.10.7
AT_NOT_STAB_INIT	BOOL	R	Auto-tuning diagnostics: initially unstable process value.	%MWr.m.c.10.8
AT_TMAX_TOO_SMALL	BOOL	R	Auto-tuning diagnostics: step duration too brief.	%MWr.m.c.10.9
AT_NOISE_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: process value noise too loud.	%MWr.m.c.10.10
AT_TMAX_TOO_HIGH	BOOL	R	Auto-tuning diagnostics: step duration too long.	%MWr.m.c.10.11
AT_OVERSHOOT	BOOL	R	Auto-tuning diagnostics: overshoot exceeding 10%.	%MWr.m.c.10.12
AT_UNDERSHOOT	BOOL	R	Auto-tuning Diagnostics: non-minimal step too long.	%MWr.m.c.10.13
AT_UNSYMMETRICAL_PT	BOOL	R	Auto-tuning diagnostics: procedure too asymmetrical.	%MWr.m.c.10.14
AT_INTEGRATING_PT	BOOL	R	Auto-tuning diagnostics: integrating process.	%MWr.m.c.10.15

Command Word

The following table explains the various meanings of the `ORDER_COMMAND` (%MWr.m.c.11) command word and of the `PARAM_COMMAND` (%MDr.m.c.12) command parameters command word.

Standard symbol	Type	Access	Meaning	Address
ORDER_COMMAND	INT	R/W	Command Word (<i>see page 232</i>)	%MWr.m.c.11
PARAM_COMMAND	DINT	R/W	Command Parameter	%MDr.m.c.12

Detailed Description of Process Control Language Objects for the T_PROC_CONST_LOOP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_CONST_LOOP-type IODDT. These **process control** language objects are associated with the auto-selector (or self-selecting) loop.

Sample Variable Declaration: T_PROC_CONST_LOOP-type IODDT_VAR1.

Objects

The following table lists all process control language objects associated with the self-selective loop.

Standard symbol	Type	Access	Meaning	Default Value	Address
AT_STEP	REAL	R/W	Auto-tuning step function amplitude	10.0	%MFr.m.c.14
AT_TMAX	REAL	R/W	Auto-tuning step function duration	100.0	%MFr.m.c.16
AT_PERF	REAL	R/W	Criterion for auto-tuning stability.	0.5	%MFr.m.c.18
T_ECH	REAL	R/W	Sampling period (common to both PID's).	0.3	%MFr.m.c.20
OUT1	REAL	R	Hot/Cold or Split Range output 1 command value.	N/A	%MFr.m.c.22
OUT2	REAL	R	Hot/Cold or Split Range output 2 command value.	N/A	%MFr.m.c.24
OUTD	REAL	R	Global loop command variation value.	N/A	%MFr.m.c.26
OUT_MAN	REAL	R/W	Value of global command (loop controller output value selected after processing via OUTFRATE and limitations).	N/A	%MFr.m.c.28
OUT_FF_C1	REAL	R	Value of Feed forward action on main loop physical scale.	N/A	%MFr.m.c.30
OUT_MAN_C1	REAL	R/W	Main loop command value.	N/A	%MFr.m.c.32
DEV_C1	REAL	R	Main loop process value setpoint deviation.	N/A	%MFr.m.c.34
PV_C1	REAL	R	PV value on physical scale.	N/A	%MFr.m.c.36
SP_C1	REAL	R	Setpoint value on physical scale.	N/A	%MFr.m.c.38
PV_INF_C1	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.40
PV_SUP_C1	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.42
KP_C1	REAL	R/W	PID proportional coefficient / IMC static gain.	1.0	%MFr.m.c.44
TI_C1	REAL	R/W	PID integral time / IMC time constant.	0.0	%MFr.m.c.46
TD_C1	REAL	R/W	PID derivative time / IMC delay.	0.0	%MFr.m.c.48

Standard symbol	Type	Access	Meaning	Default Value	Address
OUTBIAS_C1	REAL	R/W	Bias on PID loop controller output / IMC BO/BF ratio.	0.0	%MFr.m.c.50
INT_BAND_C1	REAL	R/W	Integral band.	0.0	%MFr.m.c.52
DBAND_C1	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.54
KD_C1	REAL	R/W	Derivative Filtering	10.0	%MFr.m.c.56
SP_MIN_C1	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.58
SP_MAX_C1	REAL	R/W	Upper limit for setpoint	100.0	%MFr.m.c.60
PV_LL_C1	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.62
PV_L_C1	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.64
PV_H_C1	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.66
PV_HH_C1	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.68
RATIO_C1	REAL	R/W	Ratio Value	1.0	%MFr.m.c.70
RATIO_MIN_C1	REAL	R/W	Minimum Ratio Value	0.0	%MFr.m.c.72
RATIO_MAX_C1	REAL	R/W	Maximum Ratio Value	100.0	%MFr.m.c.74
RATIO_BIAS_C1	REAL	R/W	Ratio Bias Value	0.0	%MFr.m.c.76
DEV_L_C1	REAL	R/W	Lower threshold of deviation	-5.0	%MFr.m.c.78
DEV_H_C1	REAL	R/W	Upper threshold of deviation	5.0	%MFr.m.c.80
T_FILTER_C1	REAL	R/W	Process value filtering time	0.0	%MFr.m.c.82
K_FILTER_C1	REAL	R/W	Multiplication coefficient of process value filtering.	1.0	%MFr.m.c.84
FILT_OUT_C1	REAL	R	Filtering output value.	N/A	%MFr.m.c.86
SQRT_OUT_C1	REAL	R	Output value of square root	N/A	%MFr.m.c.88
E2_IN_C1	REAL	R/W	S2 Segment input value.	1428.0	%MFr.m.c.90
E3_IN_C1	REAL	R/W	S3 Segment input value.	2857.0	%MFr.m.c.92
E4_IN_C1	REAL	R/W	S4 Segment input value.	4285.0	%MFr.m.c.94
E5_IN_C1	REAL	R/W	S5 Segment input value.	5714.0	%MFr.m.c.96
E6_IN_C1	REAL	R/W	S6 Segment input value.	7143.0	%MFr.m.c.98
E7_IN_C1	REAL	R/W	S7 Segment input value.	8571.0	%MFr.m.c.100
E2_OUT_C1	REAL	R/W	S2 Segment output value.	14.28	%MFr.m.c.102
E3_OUT_C1	REAL	R/W	S3 Segment output value.	28.57	%MFr.m.c.104
E4_OUT_C1	REAL	R/W	S4 Segment output value.	42.85	%MFr.m.c.106
E5_OUT_C1	REAL	R/W	S5 Segment output value.	57.14	%MFr.m.c.108
E6_OUT_C1	REAL	R/W	S6 Segment output value.	71.43	%MFr.m.c.110
E7_OUT_C1	REAL	R/W	S7 Segment output value.	85.71	%MFr.m.c.112

Standard symbol	Type	Access	Meaning	Default Value	Address
THLD_C1	REAL	R/W	Totaling limit.	1E+8	%MFr.m.c.114
R_RATE_C1	REAL	R/W	Setpoint speed increase limit	0.0	%MFr.m.c.116
D_RATE_C1	REAL	R/W	Setpoint speed decrease limit	0.0	%MFr.m.c.118
SPEED_LIM_OUT_C1	REAL	R	Setpoint speed limiter output value	N/A	%MFr.m.c.120
INP_INFR1_C1	REAL	R/W	Main loop R1 setpoint low scale.	0.0	%MFr.m.c.122
INP_SUPR1_C1	REAL	R/W	Main loop R1 setpoint high scale.	100.0	%MFr.m.c.124
INP_INFR2_C1	REAL	R/W	Main loop R2 setpoint low scale.	0.0	%MFr.m.c.126
INP_SUPR2_C1	REAL	R/W	Main loop R2 setpoint high scale.	100.0	%MFr.m.c.128
T1_FF_C1	REAL	R/W	Feed Forward process value filtering time	0.0	%MFr.m.c.130
T2_FF_C1	REAL	R/W	Feed Forward process value filtering time	0.0	%MFr.m.c.132
OUT_FF_INF_C1	REAL	R/W	Feed forward process value low limit.	0.0	%MFr.m.c.134
OUT_FF_SUP_C1	REAL	R/W	Feed forward process value high limit.	100.0	%MFr.m.c.136
KP_PREV_C1	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.138
TI_PREV_C1	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.140
TD_PREV_C1	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.142
OUT_MAN_C2	REAL	R/W	Secondary loop command value.	N/A	%MFr.m.c.144
DEV_C2	REAL	R	Setpoint Process Value Deviation.	N/A	%MFr.m.c.146
PV_C2	REAL	R	PV value on physical scale.	N/A	%MFr.m.c.148
SP_C2	REAL	R	Setpoint value on physical scale.	N/A	%MFr.m.c.150
PV_INF_C2	REAL	R/W	Lower limit of process value	0.0	%MFr.m.c.152
PV_SUP_C2	REAL	R/W	Upper limit of process value	100.0	%MFr.m.c.154
KP_C2	REAL	R/W	Proportional Coefficient	1.0	%MFr.m.c.156
TI_C2	REAL	R/W	Integral Time	0.0	%MFr.m.c.158
TD_C2	REAL	R/W	Derivative Time	0.0	%MFr.m.c.160
OUTBIAS_C2	REAL	R/W	Bias on PID loop controller output	0.0	%MFr.m.c.162
INT_BAND_C2	REAL	R/W	Integral band.	0.0	%MFr.m.c.164
DBAND_C2	REAL	R/W	Dead band on deviation.	0.0	%MFr.m.c.166
SP_MIN_C2	REAL	R/W	Lower limit for setpoint	0.0	%MFr.m.c.168
SP_MAX_C2	REAL	R/W	Upper limit for setpoint	100.0	%MFr.m.c.170
PV_LL_C2	REAL	R/W	Lowest threshold for process value	5.0	%MFr.m.c.172
PV_L_C2	REAL	R/W	Lower level for process value	5.0	%MFr.m.c.174
PV_H_C2	REAL	R/W	Upper threshold for process value	95.0	%MFr.m.c.176

Standard symbol	Type	Access	Meaning	Default Value	Address
PV_HH_C2	REAL	R/W	Highest threshold for process value	95.0	%MFr.m.c.178
DEV_L_C2	REAL	R/W	Lower threshold of deviation	-5.0	%MFr.m.c.180
DEV_H_C2	REAL	R/W	Upper threshold of deviation	5.0	%MFr.m.c.182
SQRT_OUT_C2	REAL	R	Output value of square root	N/A	%MFr.m.c.184
THLD_C2	REAL	R/W	Totaling limit.	1E+8	%MFr.m.c.186
R_RATE_C2	REAL	R/W	Setpoint speed increase limit	0.0	%MFr.m.c.188
D_RATE_C2	REAL	R/W	Setpoint speed decrease limit	0.0	%MFr.m.c.190
SPEED_LIM_OUT_C2	REAL	R	Setpoint speed limiter output value	N/A	%MFr.m.c.192
INP_INFR1_C2	REAL	R/W	Secondary loop R1 setpoint low scale.	0.0	%MFr.m.c.194
INP_SUPR1_C2	REAL	R/W	Secondary loop R1 setpoint high scale.	100.0	%MFr.m.c.196
KP_PREV_C2	REAL	R	Value before proportional coefficient auto-tuning	N/A	%MFr.m.c.198
TI_PREV_C2	REAL	R	Value before integral coefficient auto-tuning	N/A	%MFr.m.c.200
TD_PREV_C2	REAL	R	Value before derivative coefficient auto-tuning	N/A	%MFr.m.c.202
OUTRATE	REAL	R/W	Output 1 speed limit.	0.0	%MFr.m.c.204
OUTRATE2	REAL	R/W	Output 2 speed limit.	0.0	%MFr.m.c.206
OUT1_INF	REAL	R/W	Lower limit for Output 1.	0.0	%MFr.m.c.208
OUT1_SUP	REAL	R/W	Upper limit for Output 1.	100.0	%MFr.m.c.210
OUT2_INF	REAL	R/W	Lower limit for Output 2.	0.0	%MFr.m.c.212
OUT2_SUP	REAL	R/W	Upper limit for Output 2.	100.0	%MFr.m.c.214
OUT1_TH1	REAL	R/W	Threshold 1 for Hot/Cold or Split Range output 1	0.0	%MFr.m.c.216
OUT1_TH2	REAL	R/W	Threshold 2 for Hot/Cold or Split Range output 1	50.0	%MFr.m.c.218
OUT2_TH1	REAL	R/W	Threshold 1 for Hot/Cold or Split Range output 2	50.0	%MFr.m.c.220
OUT2_TH2	REAL	R/W	Threshold 2 for Hot/Cold or Split Range output 2	100.0	%MFr.m.c.222
T_MOTOR1	REAL	R/W	Open time for servo-controlled valve	10.0	%MFr.m.c.224
T_MINI1	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.226
T_MOTOR2	REAL	R/W	Open time for servo-controlled valve	10.0	%MFr.m.c.228
T_MINI2	REAL	R/W	Minimum open time for servo-controlled valve	0.0	%MFr.m.c.230
KS	REAL	R/W	IMC static gain	1.0	%MFr.m.c.232

Standard symbol	Type	Access	Meaning	Default Value	Address
OL_TIME	REAL	R/W	Time Constant in OL.	1.0	%MFr.m.c.234
T_DELAY	REAL	R/W	Current pure delay time	0.0	%MFr.m.c.236
CL_PERF	REAL	R/W	OL/CL Ratio	0.1	%MFr.m.c.238
PV_SIM_C1	INT	R/W	Simulated Process Value	N/A	%MWr.m.c.248
PV_SIM_C2	INT	R/W	Simulated Process Value	N/A	%MWr.m.c.249
FF_SIM_C1	INT	R/W	Simulated Feed Forward Input	N/A	%MWr.m.c.250

Configuration Language Objects

Description

The following table describes the process control language objects associated with the auto-selector (or self-selecting) loop.

Standard symbol	Type	Access	Meaning	Default Value	Address
CONFIG_0_C1	INT	R	Word grouping the various configuration bits of the C1 process value.	No object	%KWr.m.c.0
Filter	BOOL	R	Filtering function of the process value branch	Missing (0)	%KWr.m.c.0.0
Function Generator	BOOL	R	Function Generator for process value branch	Missing (0)	%KWr.m.c.0.1
Totaling	BOOL	R	Totaling Function for the process value branch	Missing (0)	%KWr.m.c.0.2
Square Root	BOOL	R	Square root function of the process value branch	Missing (0)	%KWr.m.c.0.3
Alarms	BOOL	R	Alarm Function of the process value branch	Present	%KWr.m.c.0.4
PV_CLIP	BOOL	R	Lopping or non-lopping of process value	Missing (0)	%KWr.m.c.0.8
EXTRAPOL	BOOL	R	Extrapolation of function generator	No (0)	%KWr.m.c.0.9
PV_UNI_BIP	BOOL	R	Type of process value: unipolar/bipolar	Unipolar (0)	%KWr.m.c.0.10
PV_EXTERNE	BOOL	R	Selection of Standard Process Value (0) vs. External Process Value (1).	Missing (0)	%KWr.m.c.0.11
Totalizer process value unit	BOOL	R	(bit 13 = 0, bit 14 = 0): phys/ms (bit 13 = 1, bit 14 = 0): phys/s	1	%KWr.m.c.0.13
Totalizer process value unit	BOOL	R	(bit 13 = 1, bit 14 = 0): phys/mn (bit 13 = 1, bit 14 = 1): phys/h	0	%KWr.m.c.0.14
CONFIG_1_C1	INT	R	Word grouping the various configuration bits of the C1 Setpoint.	No object	%KWr.m.c.1
SP_SIMPLE	BOOL	R	Type of setpoint selected : Single.	Selected (1)	%KWr.m.c.1.0
SP_Selection	BOOL	R	Type of setpoint: Selection.	Not selected (0)	%KWr.m.c.1.1
SPEED_LIMITER	INT	R	Setpoint speed limiter.	Not selected (0)	%KWr.m.c.1.2
SP_SPP	BOOL	R	Type of setpoint selected : Programmer.	Not selected (0)	%KWr.m.c.1.3

Standard symbol	Type	Access	Meaning	Default Value	Address
LR/L	BOOL	R	Speed limiter either on local setpoint, or in Remote/Local mode.	Local Remote (0)	%KWr.m.c.1.4
SEL_MIN	BOOL	R	Function selected in the event of a Selection setpoint.	Missing (0)	%KWr.m.c.1.8
SEL_MAX	BOOL	R	Function selected in the event of a Selection setpoint.	Missing (0)	%KWr.m.c.1.9
SEL_SWITCH	BOOL	R	Function selected in the event of a Selection setpoint.	Present on selection	%KWr.m.c.1.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KWr.m.c.1.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	R1 (0)	%KWr.m.c.1.12
SP_RATIO	BOOL	R	Type of setpoint selected : Ratio.	Not selected (0)	%KWr.m.c.1.13
SP_LIMITER	BOOL	R	Setpoint Limiter (e.g. PARAM_SP).	Not present	%KWr.m.c.1.14
SP_FOLW	BOOL	R	Tracking Setpoint (0)	No tracking	%KWr.m.c.1.15
CONFIG_2_C1	INT	R	Word grouping various configuration bits for loop controller and Feed Forward for C1.	No object	%KWr.m.c.2
PID	BOOL	R	PID function of loop controller branch	Always present	%KWr.m.c.2.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	No object	%KWr.m.c.2.1
ONOFF3	INT	R	ON/OFF 3-states branch of loop controller	No object	%KWr.m.c.2.2
SPLRG/HOTCOLD	BOOL	R	OR Hot/Cold and Split Range presence bits	No object	%KWr.m.c.2.3
Split/Range	BOOL	R	Split Range function for selected branch.	Missing (0)	%KWr.m.c.2.4
Hot/Cold	BOOL	R	Hot/Cold function for selected branch.	Not selected	%KWr.m.c.2.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.2.6
Feed Forward	BOOL	R	Presence of Feed Forward input	Missing (0)	%KWr.m.c.2.7
BUMP	BOOL	R	Bump Management when changing operating mode.	With bumps (1)	%KWr.m.c.2.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.2.9

Standard symbol	Type	Access	Meaning	Default Value	Address
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.2.10
REV_DIR	BOOL	R	Type of loop controller action: 1: opposite action, a positive shift (PV-SP) generates a rise of the output value 0: direct action, a positive shift (PV-SP) generates a fall of the output value	PID opposite action (1)	%KWr.m.c.2.11
MANU/AUTO_INIT	BOOL	R	Initial value of loop controller operating mode	Auto (1)	%KWr.m.c.2.12
Lead Lag	BOOL	R	Lead Lag function of Feed Forward branch	Missing (0)	%KWr.m.c.2.13
FF_UNI_BIP	BOOL	R	Type of Feed Forward process value: unipolar/bipolar	Unipolar	%KWr.m.c.2.14
IMC	BOOL	R	Model loop controller in branch loop controller.	Missing (0)	%KWr.m.c.2.15
Loop name	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.3
Loop Unit	INT	R	Loop Unit.		%KWr.m.c.7
CONFIG_0_C2	BOOL	R	Word grouping the various configuration bits of the C2 process value.	No object	%KWr.m.c.10
Filter	INT	R	Filtering function of the process value branch	No object	%KWr.m.c.10.0
Function Generator	BOOL	R	Function Generator for process value branch	Missing (0)	%KWr.m.c.10.1
Totaling	BOOL	R	Totaling Function for the process value branch	Missing (0)	%KWr.m.c.10.2
Square Root	BOOL	R	Square root function of the process value branch	Missing (0)	%KWr.m.c.10.3
Alarms	BOOL	R	Alarm Function of the process value branch	Present	%KWr.m.c.10.4
PV_CLIP	BOOL	R	Lopping or non-lopping of process value	Missing (0)	%KWr.m.c.10.8
EXTRAPOL	BOOL	R	Extrapolation of function generator	No object	%KWr.m.c.10.9
PV_UNI_BIP	BOOL	R	Type of process value: unipolar/bipolar	Unipolar (0)	%KWr.m.c.10.10
PV_EXTERNE	BOOL	R	Selection of Standard Process Value (0) vs. External Process Value (1).	Missing (0)	%KWr.m.c.10.11

Standard symbol	Type	Access	Meaning	Default Value	Address
Totaling process measuring unit	BOOL	R	(bit 13 = 0, bit 14 = 0): phys/ms (bit 13 = 1, bit 14 = 0): phys/s	1	%KW.r.m.c.10.13
Totaling process measuring unit	BOOL	R	(bit 13 = 1, bit 14 = 0): phys/mn (bit 13 = 1, bit 14 = 1): phys/h	0	%KW.r.m.c.10.14
CONFIG_1_C2	BOOL	R	Word grouping the various configuration bits of the C2 Setpoint.	No object	%KW.r.m.c.11
SP_SIMPLE	BOOL	R	Type of setpoint selected : Single.	Selected (1)	%KW.r.m.c.11.0
SP_Selection	BOOL	R	Type of setpoint selected : Selection.	No object	%KW.r.m.c.11.1
SPEED_LIMITER	BOOL	R	Setpoint speed limiter.	Missing (0)	%KW.r.m.c.11.2
SP_SPP	BOOL	R	Type of setpoint selected : Programmer.	No object	%KW.r.m.c.11.3
LR/L	BOOL	R	Speed limiter either on local setpoint, or in Remote/Local mode.	Local Remote (0)	%KW.r.m.c.11.4
SEL_MIN	BOOL	R	Function selected for setpoint type Selection.	No object	%KW.r.m.c.11.8
SEL_MAX	BOOL	R	Function selected for setpoint type Selection.	No object	%KW.r.m.c.11.9
SEL_SWITCH	BOOL	R	Function selected for setpoint type Selection.	No object	%KW.r.m.c.11.10
R/L_INIT	BOOL	R	Initial value of selected Remote/Local setpoint	Local (1)	%KW.r.m.c.11.11
R1/R2_INIT	BOOL	R	Initial value of selected setpoint state	No object	%KW.r.m.c.11.12
SP_RATIO	BOOL	R	Type of setpoint selected : Ratio.	No object	%KW.r.m.c.11.13
SP_LIMITER	BOOL	R	Setpoint Limiter (e.g. PARAM_SP).	Not present (0)	%KW.r.m.c.11.14
SP_FOLW	BOOL	R	Tracking Setpoint (0)	No setpoint	%KW.r.m.c.11.15
CONFIG_2_C2	INT	R	Word grouping various configuration bits for loop controller and Feed Forward for C2.	No object	%KW.r.m.c.12
PID	BOOL	R	PID function of loop controller branch	Present (always)	%KW.r.m.c.12.0
ONOFF2	BOOL	R	ON/OFF 2-states branch of loop controller	No object	%KW.r.m.c.12.1
ONOFF3	BOOL	R	ON/OFF 3-states branch of loop controller	No object	%KW.r.m.c.12.2

Standard symbol	Type	Access	Meaning	Default Value	Address
SPLRG/HOTCOLD	BOOL	R	OR Hot/Cold and Split Range presence bits	No object	%KWr.m.c.12.3
Split/Range	BOOL	R	Loop controller branch Split Range function.	No object	%KWr.m.c.12.4
Hot/Cold	BOOL	R	Hot/Cold function of loop controller branch	No object	%KWr.m.c.12.5
ALARMES_DEV	BOOL	R	Alarm function on diversion of loop controller branch	Present	%KWr.m.c.12.6
Feed Forward	BOOL	R	Presence of Feed Forward input	No object	%KWr.m.c.12.7
BUMP	BOOL	R	Bump Management when changing operating mode.	With bumps (1)	%KWr.m.c.12.8
PV_DEV	BOOL	R	Type of derived action	On process value (0)	%KWr.m.c.12.9
MIX_PAR	BOOL	R	Mixed or parallel loop controller type	PID parallel series	%KWr.m.c.12.10
REV_DIR	BOOL	R	Type of loop controller action: • 1: opposite action, a positive shift (PV-SP) generates a rise of the output value • 0: direct action, a positive shift (PV-SP) generates a fall of the output value	PID opposite action (1)	%KWr.m.c.12.11
MANU/AUTO_INIT	BOOL	R	Initial value of loop controller operating mode	Auto (1)	%KWr.m.c.12.12
Lead Lag	BOOL	R	Leadlag function of the Feed forward branch	No object	%KWr.m.c.12.13
FF_UNI_BIP	BOOL	R	Type of Feed Forward process value: unipolar or bipolar	No object	%KWr.m.c.12.14
IMC	BOOL	R	Model loop controller in branch loop controller.	Missing (0)	%KWr.m.c.12.15
CONFIG_3_C2	INT	R	Word grouping the various output configuration bits	No object	%KWr.m.c.13
Servo	BOOL	R	Type of output selected: Servo.	Not selected	%KWr.m.c.13.0
Servo2	BOOL	R	Type of output selected: Servo.	Not selected	%KWr.m.c.13.1
Analog1	BOOL	R	Type of output selected: Analog.	Not selected	%KWr.m.c.13.2
Analog2	BOOL	R	Type of output selected: Analog.	Not selected	%KWr.m.c.13.3
PWM1	BOOL	R	Type of output selected: PWM.	Not selected	%KWr.m.c.13.4
PWM2	BOOL	R	Type of output selected: PWM.	Not selected	%KWr.m.c.13.5
POT_REV1	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.13.8

Standard symbol	Type	Access	Meaning	Default Value	Address
POT_REV2	BOOL	R	Servo copy direction	Direct (0)	%KWr.m.c.13.9
POT_VAL1_INIT	BOOL	R	Presence of Servo copy.	No (0)	%KWr.m.c.13.10
POT_VAL2_INIT	BOOL	R	Presence of Servo copy.	No (0)	%KWr.m.c.13.11
ANALOG1_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar	Unipolar (0)	%KWr.m.c.13.12
ANALOG2_UNI_BIP	BOOL	R	Type of analog output: unipolar/bipolar	Unipolar (0)	%KWr.m.c.13.13
	INT	R	Loop Name.	Loop i with i [0.9]	%KWr.m.c.14
	INT	R	Loop Unit.		%KWr.m.c.18
CONFIG_0_G	INT	R	Word grouping the various configuration bits of the global loop.	No object	%KWr.m.c.21
MANU/AUTO_G_INIT	BOOL	R	Initial operating value of global loop.	Manu (0)	%KWr.m.c.21.0
AM_G_PID	BOOL	R	Managing A/M blocks initially: on A/M block 0 on the global loop, at 1: A/M blocks on each PID.	On the global loop (0)	%KWr.m.c.21.1
MIN_MAX	BOOL	R	Auto-selector's initial behavior.	Min (0)	%KWr.m.c.21.8
AS_INIT	BOOL	R	Output obtained by restriction on initialization = auto-selector output.	Present (1)	%KWr.m.c.21.9
DIR1_INIT	BOOL	R	Output obtained by restriction on initialization = PID #1 output.	Missing (0)	%KWr.m.c.21.10
DIR2_INIT	BOOL	R	Output obtained by restriction on initialization = PID #2 output.	Missing (0)	%KWr.m.c.21.11
Trigger mode of global loop	BOOL	R	Trigger mode: bit 0 = 0: automatic	0	%KWr.m.c.22.0

10.7 Language Objects Associated with the Setpoint Programmer

Subject of this Section

This section describes the language objects associated with the setpoint programmers.

What Is in This Section?

This section contains the following topics:

Topic	Page
Detailed Description of the T_PROC_SPP-type IODDT's Implicit Exchange Language Objects	316
Detailed Description of Status and Default Language Objects for the T_PROC_SPP-Type IODDT	317
Detailed Description of Process Control Language Objects for the T_PROC_SPP-Type IODDT	322
Configuration Language Objects	326

Detailed Description of the T_PROC_SPP-type IODDT's Implicit Exchange Language Objects

At a Glance

The following table lists all of the T_PROC_SPP -type Implicit Exchange Language Objects applicable to the setpoint programmers.

%lr.m.c.ERR Error Bit

The following table explain the meaning of each error bit CH_ERROR (%lr.m.c.ERR) :

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	BOOL	R	Process control channel error bit.	%lr.m.c.ERR

Detailed Description of Status and Default Language Objects for the T_PROC_SPP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_SPP-type IODDT. These **status** and **default** language objects are associated with the setpoint programmer. The section groups the word objects, whose bits have a specific meaning.

Sample Variable Declaration: T_PROC_SPP-type IODDT_VAR1.

Comments

- Generally speaking, the meaning of any bit is provided for State 1 of said bit. In some specific instances, we provide a meaning for each state of the bit.
- Not all bits are utilized.

%Ir.m.c.ERR Error Bit

The following table explains the meaning of the CH_ERROR (%Ir.m.c.ERR) error bit.

Standard symbol	Type	Access	Meaning	Address
CH_ERROR	EBOOL	R	Error bit for analog output.	%Ir.m.c.ERR

Exchange Execution Indicators: EXCH_STS

The following table explains the various meanings of EXCH_STS (%MWr.m.c.0) channel exchange control bits.

Standard symbol	Type	Access	Meaning	Address
STS_IN_PROGR	BOOL	R	Channel status words are in the process of being read.	%MWr.m.c.0.0
CMD_IN_PROGR	BOOL	R	The exchange of Command Parameters is in progress.	%MWr.m.c.0.1
ADJ_IN_PROGR	BOOL	R	The exchange of Adjustment Parameters is in progress.	%MWr.m.c.0.2
RECONF_IN_PROGR	BOOL	R	The module reconfiguration is in progress.	%MWr.m.c.0.15

Exchange Report: EXCH_RPT

The following table explains the various meanings of EXCH_RPT (%MWr.m.c.1) reporting bits.

Standard symbol	Type	Access	Meaning	Address
STS_ERR	BOOL	R	Error while reading the channel status words.	%MWr.m.c.1.0
CMD_ERR	BOOL	R	Error while exchanging command parameters.	%MWr.m.c.1.1
ADJ_ERR	BOOL	R	Error while exchanging adjustment parameters.	%MWr.m.c.1.2
RECONF_ERR	BOOL	R	Error while reconfiguring the channel. (1 = failure).	%MWr.m.c.1.15

Standard Errors for CH_FLT

The following table explains the various meanings of the CH_FLT (%MWr.m.c.2) status word bits.

Standard symbol	Type	Access	Meaning	Address
INTERNAL_FLT	BOOL	R	Fatal internal error.	%MWr.m.c.2.4
CONF_FLT	BOOL	R	Configuration error.	%MWr.m.c.2.5
COM_FLT	BOOL	R	Communication error.	%MWr.m.c.2.6
WARN	BOOL	R	Error Sum.	%MWr.m.c.2.7
STS_ERR_CALC	BOOL	R	Calculation error.	%MWr.m.c.2.8
STS_ERR_FLOT	BOOL	R	Floating point error.	%MWr.m.c.2.9

STATUS2 Status Word

The following table explains the various meanings of the `STATUS2` (%MWr.m.c.3) status word bits. Word grouping various status bits for output monitoring, SPP "freeze" and profile status.

Standard symbol	Type	Access	Meaning	Address
STOR0	BOOL	R	State of control output 0.	%MWr.m.c.3.0
STOR1	BOOL	R	State of control output 1.	%MWr.m.c.3.1
STOR2	BOOL	R	State of control output 2.	%MWr.m.c.3.2
STOR3	BOOL	R	State of control output 3.	%MWr.m.c.3.3
STOR4	BOOL	R	State of control output 4.	%MWr.m.c.3.4
STOR5	BOOL	R	State of control output 5.	%MWr.m.c.3.5
STOR6	BOOL	R	State of control output 6.	%MWr.m.c.3.6
STOR7	BOOL	R	State of control output 7.	%MWr.m.c.3.7
STS_SPP_HOLD	BOOL	R	Setpoint Programmer function freeze.	%MWr.m.c.3.8
STS_INIT	BOOL	R	1 : all profiles are in INIT.	%MWr.m.c.3.9
STS_RUN	BOOL	R	1 : the current profile is in RUN mode.	%MWr.m.c.3.10
STS_STOP	BOOL	R	1 : the current profile is in STOP mode.	%MWr.m.c.3.11
STS_HOLD_PG	BOOL	R	1 : the guaranteed dwell time function is inhibited.	%MWr.m.c.3.12
STS_ERR_SEG	BOOL	R	Parameter error on current segment.	%MWr.m.c.3.15

STATUS3 Status Word

The following table explains the various meanings of the `STATUS3` (%MWr.m.c.4) status word bits. Word grouping various indicator bits for errors on profiles 1 to 4.

Standard symbol	Type	Access	Meaning	Address
WRN1_RMP_SP	BOOL	R	Profile 1 ramp includes 2 identical setpoints.	%MWr.m.c.4.0
WRN1_RMP_0	BOOL	R	Profile 1 ramp has null speed.	%MWr.m.c.4.1
WRN1_PLR_SP	BOOL	R	A profile 1 dwell time has 2 different setpoints.	%MWr.m.c.4.2
WRN1_PLR_THLD	BOOL	R	Configured guaranteed dwell time on profile 1 and THLD null.	%MWr.m.c.4.3
WRN2_RMP_SP	BOOL	R	Profile 2 ramp includes 2 identical setpoints.	%MWr.m.c.4.4
WRN2_RMP_0	BOOL	R	Profile 2 ramp has null speed.	%MWr.m.c.4.5
WRN2_PLR_SP	BOOL	R	A profile 2 dwell time has 2 different setpoints.	%MWr.m.c.4.6
WRN2_PLR_THLD	BOOL	R	Configured guaranteed dwell time on profile 2 and THLD null.	%MWr.m.c.4.7
WRN3_RMP_SP	BOOL	R	Profile 3 ramp includes 2 identical setpoints.	%MWr.m.c.4.8
WRN3_RMP_0	BOOL	R	Profile 3 ramp has null speed.	%MWr.m.c.4.9

Standard symbol	Type	Access	Meaning	Address
WRN3_PLR_SP	BOOL	R	A profile 3 dwell time has 2 different setpoints.	%MWr.m.c.4.10
WRN3_PLR_THLD	BOOL	R	Configured guaranteed dwell time on profile 3 and THLD null.	%MWr.m.c.4.11
WRN4_RMP_SP	BOOL	R	Profile 4 ramp includes 2 identical setpoints.	%MWr.m.c.4.12
WRN4_RMP_0	BOOL	R	Profile 4 ramp has null speed.	%MWr.m.c.4.13
WRN4_PLR_SP	BOOL	R	A profile 4 dwell time has 2 different setpoints.	%MWr.m.c.4.14
WRN4_PLR_THLD	BOOL	R	Configured guaranteed dwell time on profile 4 and THLD null.	%MWr.m.c.4.15

STATUS4 Status Word

The following table explains the various meanings of the STATUS4 (%MWr.m.c.5) status word bits. Word grouping various indicator bits for errors on profiles 5 to 6.

Standard symbol	Type	Access	Meaning	Address
WRN5_RMP_SP	BOOL	R	Profile 5 ramp includes 2 identical setpoints.	%MWr.m.c.5.0
WRN5_RMP_0	BOOL	R	Profile 5 ramp has null speed.	%MWr.m.c.5.1
WRN5_PLR_SP	BOOL	R	A profile 5 dwell time has 2 different setpoints.	%MWr.m.c.5.2
WRN5_PLR_THLD	BOOL	R	Configured guaranteed dwell time on profile 5 and THLD null.	%MWr.m.c.5.3
WRN6_RMP_SP	BOOL	R	Profile 6 ramp includes 2 identical setpoints.	%MWr.m.c.5.4
WRN6_RMP_0	BOOL	R	Profile 6 ramp has null speed.	%MWr.m.c.5.5
WRN6_PLR_SP	BOOL	R	A profile 6 dwell time has 2 different setpoints.	%MWr.m.c.5.6
WRN6_PLR_THLD	BOOL	R	Configured guaranteed dwell time on profile 6 and THLD null.	%MWr.m.c.5.7

Command Word

The following table explains the various meanings of the CMD_ORDER (%MWr.m.c.7) command word and of the CMD_PARAM (%MDr.m.c.8) command parameters.

Standard symbol	Type	Access	Meaning	Address
CMD_ORDER	INT	R/W	Command Word (<i>see page 232</i>).	%MWr.m.c.7
CMD_PARAM	DINT	R/W	Command Parameter.	%MDr.m.c.8

Parameters

The following table explains different parameters.

Standard symbol	Type	Access	Meaning	Address
CUR_PF	INT	R	Current profile number.	%MWr.m.c.10
SEG_OUT	INT	R	Current segment number.	%MWr.m.c.11
CUR_ITER	INT	R	Number of current iteration (default value is 1).	%MWr.m.c.12
NB_RT_PF1	INT	R/W	Number of Profile 1 iteration (default value is 1).	%MWr.m.c.13
NB_RT_PF2	INT	R/W	Number of Profile 2 iteration (default value is 1).	%MWr.m.c.14
NB_RT_PF3	INT	R/W	Number of Profile 3 iteration (default value is 1).	%MWr.m.c.15
NB_RT_PF4	INT	R/W	Number of Profile 4 iteration (default value is 1).	%MWr.m.c.16
NB_RT_PF5	INT	R/W	Number of Profile 5 iteration (default value is 1).	%MWr.m.c.17
NB_RT_PF6	INT	R/W	Number of Profile 6 iteration (default value is 1).	%MWr.m.c.18

Detailed Description of Process Control Language Objects for the T_PROC_SPP-Type IODDT

At a Glance

This section describes the language objects for the T_PROC_SPP-type IODDT. These **process control** language objects are associated to the setpoint programmer.

Sample Variable Declaration: T_PROC_SPP-type IODDT_VAR1.

Objects

The following table lists all process control language objects associated to the setpoint programmer.

Standard symbol	Type	Access	Meaning	Default Value	Address
SP	REAL	R	Calculated setpoint (output) value.	No object	%MFr.m.c.20
TOTAL_TIME	REAL	R	Value of total time elapsed (including freeze times).	No object	%MFr.m.c.22
CUR_TIME	REAL	R	Value of total time elapsed on the current segment (including freeze times).	No object	%MFr.m.c.24
THLD_PF1	REAL	R/W	Value of guaranteed dwell time threshold for profile 1.	0.0	%MFr.m.c.26
THLD_PF2	REAL	R/W	Value of guaranteed dwell time threshold for profile 2.	0.0	%MFr.m.c.28
THLD_PF3	REAL	R/W	Value of guaranteed dwell time threshold for profile 3.	0.0	%MFr.m.c.30
THLD_PF4	REAL	R/W	Value of guaranteed dwell time threshold for profile 4.	0.0	%MFr.m.c.32
THLD_PF5	REAL	R/W	Value of guaranteed dwell time threshold for profile 5.	0.0	%MFr.m.c.34
THLD_PF6	REAL	R/W	Value of guaranteed dwell time threshold for profile 6.	0.0	%MFr.m.c.36
SP0_PF1	REAL	R/W	Value of Profile 1 initial setpoint.	0.0	%MFr.m.c.38
SP0_PF2	REAL	R/W	Value of Profile 2 initial setpoint.	0.0	%MFr.m.c.40
SP0_PF3	REAL	R/W	Value of Profile 3 initial setpoint.	0.0	%MFr.m.c.42
SP0_PF4	REAL	R/W	Value of Profile 4 initial setpoint.	0.0	%MFr.m.c.44
SP0_PF5	REAL	R/W	Value of Profile 5 initial setpoint.	0.0	%MFr.m.c.46
SP0_PF6	REAL	R/W	Value of Profile 6 initial setpoint.	0.0	%MFr.m.c.48
SP1	REAL	R/W	Target setpoint for segment 1.	0.0	%MFr.m.c.50

Standard symbol	Type	Access	Meaning	Default Value	Address
VAL1	REAL	R/W	Time or speed value for segment 1.	0.0	%MFr.m.c.52
SP2	REAL	R/W	Target setpoint for segment 2.	0.0	%MFr.m.c.54
VAL2	REAL	R/W	Time or speed value for segment 2.	0.0	%MFr.m.c.56
SP3	REAL	R/W	Target setpoint for segment 3.	0.0	%MFr.m.c.58
VAL3	REAL	R/W	Time or speed value for segment 3.	0.0	%MFr.m.c.60
SP4	REAL	R/W	Target setpoint for segment 4.	0.0	%MFr.m.c.62
VAL4	REAL	R/W	Time or speed value for segment 4.	0.0	%MFr.m.c.64
SP5	REAL	R/W	Target setpoint for segment 5.	0.0	%MFr.m.c.66
VAL5	REAL	R/W	Time or speed value for segment 5.	0.0	%MFr.m.c.68
SP6	REAL	R/W	Target setpoint for segment 6.	0.0	%MFr.m.c.70
VAL6	REAL	R/W	Time or speed value for segment 6.	0.0	%MFr.m.c.72
SP7	REAL	R/W	Target setpoint for segment 7.	0.0	%MFr.m.c.74
VAL7	REAL	R/W	Value of the time or speed for segment 0.7	0.0	%MFr.m.c.76
SP8	REAL	R/W	Target setpoint for segment 8.	0.0	%MFr.m.c.78
VAL8	REAL	R/W	Time or speed value for segment 8.	0.0	%MFr.m.c.80
SP9	REAL	R/W	Target setpoint for segment 9.	0.0	%MFr.m.c.82
VAL9	REAL	R/W	Time or speed value for segment 9.	0.0	%MFr.m.c.84
SP10	REAL	R/W	Target setpoint for segment 10.	0.0	%MFr.m.c.86
VAL10	REAL	R/W	Time or speed value for segment 10.	0.0	%MFr.m.c.88
SP11	REAL	R/W	Target setpoint for segment 11.	0.0	%MFr.m.c.90
VAL11	REAL	R/W	Time or speed value for segment 11.	0.0	%MFr.m.c.92
SP12	REAL	R/W	Target setpoint for segment 12.	0.0	%MFr.m.c.94
VAL12	REAL	R/W	Time or speed value for segment 12.	0.0	%MFr.m.c.96
SP13	REAL	R/W	Target setpoint for segment 13.	0.0	%MFr.m.c.98
VAL13	REAL	R/W	Time or speed value for segment 13.	0.0	%MFr.m.c.100
SP14	REAL	R/W	Target setpoint for segment 14.	0.0	%MFr.m.c.102
VAL14	REAL	R/W	Time or speed value for segment 14.	0.0	%MFr.m.c.104
SP15	REAL	R/W	Target setpoint for segment 15.	0.0	%MFr.m.c.106
VAL15	REAL	R/W	Time or speed value for segment 15.	0.0	%MFr.m.c.108
SP16	REAL	R/W	Target setpoint for segment 16.	0.0	%MFr.m.c.110
VAL16	REAL	R/W	Time or speed value for segment 16.	0.0	%MFr.m.c.112
SP17	REAL	R/W	Target setpoint for segment 17.	0.0	%MFr.m.c.114
VAL17	REAL	R/W	Time or speed value for segment 17.	0.0	%MFr.m.c.116
SP18	REAL	R/W	Target setpoint for segment 18.	0.0	%MFr.m.c.118

Standard symbol	Type	Access	Meaning	Default Value	Address
VAL18	REAL	R/W	Time or speed value for segment 18.	0.0	%MFr.m.c.120
SP19	REAL	R/W	Target setpoint for segment 19.	0.0	%MFr.m.c.122
VAL19	REAL	R/W	Time or speed value for segment 19.	0.0	%MFr.m.c.124
SP20	REAL	R/W	Target setpoint for segment 20.	0.0	%MFr.m.c.126
VAL20	REAL	R/W	Time or speed value for segment 20.	0.0	%MFr.m.c.128
SP21	REAL	R/W	Target setpoint for segment 21.	0.0	%MFr.m.c.130
VAL21	REAL	R/W	Time or speed value for segment 21.	0.0	%MFr.m.c.132
SP22	REAL	R/W	Target setpoint for segment 22.	0.0	%MFr.m.c.134
VAL22	REAL	R/W	Time or speed value for segment 22.	0.0	%MFr.m.c.136
SP23	REAL	R/W	Target setpoint for segment 23.	0.0	%MFr.m.c.138
VAL23	REAL	R/W	Time or speed value for segment 23.	0.0	%MFr.m.c.140
SP24	REAL	R/W	Target setpoint for segment 24.	0.0	%MFr.m.c.142
VAL24	REAL	R/W	Time or speed value for segment 24.	0.0	%MFr.m.c.144
SP25	REAL	R/W	Target setpoint for segment 25.	0.0	%MFr.m.c.146
VAL25	REAL	R/W	Time or speed value for segment 25.	0.0	%MFr.m.c.148
SP26	REAL	R/W	Target setpoint for segment 26.	0.0	%MFr.m.c.150
VAL26	REAL	R/W	Time or speed value for segment 26.	0.0	%MFr.m.c.152
SP27	REAL	R/W	Target setpoint for segment 27.	0.0	%MFr.m.c.154
VAL27	REAL	R/W	Time or speed value for segment 27.	0.0	%MFr.m.c.156
SP28	REAL	R/W	Target setpoint for segment 28.	0.0	%MFr.m.c.158
VAL28	REAL	R/W	Time or speed value for segment 28.	0.0	%MFr.m.c.160
SP29	REAL	R/W	Target setpoint for segment 29.	0.0	%MFr.m.c.162
VAL29	REAL	R/W	Time or speed value for segment 29.	0.0	%MFr.m.c.164
SP30	REAL	R/W	Target setpoint for segment 30.	0.0	%MFr.m.c.166
VAL30	REAL	R/W	Time or speed value for segment 30.	0.0	%MFr.m.c.168
SP31	REAL	R/W	Target setpoint for segment 31.	0.0	%MFr.m.c.170
VAL31	REAL	R/W	Time or speed value for segment 31.	0.0	%MFr.m.c.172
SP32	REAL	R/W	Target setpoint for segment 32.	0.0	%MFr.m.c.174
VAL32	REAL	R/W	Time or speed value for segment 32.	0.0	%MFr.m.c.176
SP33	REAL	R/W	Target setpoint for segment 33.	0.0	%MFr.m.c.178
VAL33	REAL	R/W	Time or speed value for segment 33.	0.0	%MFr.m.c.180
SP34	REAL	R/W	Target setpoint for segment 34.	0.0	%MFr.m.c.182
VAL34	REAL	R/W	Time or speed value for segment 34.	0.0	%MFr.m.c.184
SP35	REAL	R/W	Target setpoint for segment 35.	0.0	%MFr.m.c.186

Standard symbol	Type	Access	Meaning	Default Value	Address
VAL35	REAL	R/W	Time or speed value for segment 35.	0.0	%MFr.m.c.188
SP36	REAL	R/W	Target setpoint for segment 36.	0.0	%MFr.m.c.190
VAL36	REAL	R/W	Time or speed value for segment 36.	0.0	%MFr.m.c.192
SP37	REAL	R/W	Target setpoint for segment 37.	0.0	%MFr.m.c.194
VAL37	REAL	R/W	Time or speed value for segment 37.	0.0	%MFr.m.c.196
SP38	REAL	R/W	Target setpoint for segment 38.	0.0	%MFr.m.c.198
VAL38	REAL	R/W	Time or speed value for segment 38.	0.0	%MFr.m.c.200
SP39	REAL	R/W	Target setpoint for segment 39.	0.0	%MFr.m.c.202
VAL39	REAL	R/W	Time or speed value for segment 39.	0.0	%MFr.m.c.204
SP40	REAL	R/W	Target setpoint for segment 40.	0.0	%MFr.m.c.206
VAL40	REAL	R/W	Time or speed value for segment 40.	0.0	%MFr.m.c.208
SP41	REAL	R/W	Target setpoint for segment 41.	0.0	%MFr.m.c.210
VAL41	REAL	R/W	Time or speed value for segment 41.	0.0	%MFr.m.c.212
SP42	REAL	R/W	Target setpoint for segment 42.	0.0	%MFr.m.c.214
VAL42	REAL	R/W	Time or speed value for segment 42.	0.0	%MFr.m.c.216
SP43	REAL	R/W	Target setpoint for segment 43.	0.0	%MFr.m.c.218
VAL43	REAL	R/W	Time or speed value for segment 43.	0.0	%MFr.m.c.220
SP44	REAL	R/W	Target setpoint for segment 44.	0.0	%MFr.m.c.222
VAL44	REAL	R/W	Time or speed value for segment 44.	0.0	%MFr.m.c.224
SP45	REAL	R/W	Target setpoint for segment 45.	0.0	%MFr.m.c.226
VAL45	REAL	R/W	Time or speed value for segment 45.	0.0	%MFr.m.c.228
SP46	REAL	R/W	Target setpoint for segment 46.	0.0	%MFr.m.c.230
VAL46	REAL	R/W	Time or speed value for segment 46.	0.0	%MFr.m.c.232
SP47	REAL	R/W	Target setpoint for segment 47.	0.0	%MFr.m.c.234
VAL47	REAL	R/W	Time or speed value for segment 47.	0.0	%MFr.m.c.236
SP48	REAL	R/W	Target setpoint for segment 48.	0.0	%MFr.m.c.238
VAL48	REAL	R/W	Time or speed value for segment 48.	0.0	%MFr.m.c.240
TIME_SEG	REAL	R	Value of time remaining on current segment.	0.0	%MFr.m.c.242

Configuration Language Objects

Description

This table describes the process control language objects associated with the setpoint programmer.

Standard symbol	Type	Access	Meaning	Default Value	Address
CONFIG_1	INT	R	Word grouping various configuration bits for profile 1.		%KWr.m.c.0
Guaranteed dwell time	BOOL	R	Confirmation of guaranteed dwell time function (0: no, 1: yes).	No (0)	%KWr.m.c.0.0
Dwell Type	BOOL	R	Type of maintain on guaranteed dwell time: 2 bits.	0	%KWr.m.c.0.1
Dwell Type	BOOL	R		0	%KWr.m.c.0.2
Start	BOOL	R	Start with bumps (0: SP0) or bumpless (1: PV).	With bumps (0)	%KWr.m.c.0.3
Retry	BOOL	R	Retry of continuous (1) or non-continuous (0) profile.	Non-continuous (0)	%KWr.m.c.0.4
Retry type	BOOL	R	Retry with bumps (0: SPi) or bumpless (1: PV).	With bumps (0)	%KWr.m.c.0.5
CONFIG_2	INT	R	Word grouping various configuration bits for profile 2.		%KWr.m.c.1
Guaranteed dwell time	BOOL	R	Confirmation of guaranteed dwell time function (0: no, 1: yes).	No (0)	%KWr.m.c.1.0
Dwell Type	BOOL	R	Type of maintain on guaranteed dwell time: 2 bits.	0	%KWr.m.c.1.1
Dwell Type	BOOL	R		0	%KWr.m.c.1.2
Start	BOOL	R	Start with bumps (0: SP0) or bumpless (1: PV).	With bumps (0)	%KWr.m.c.1.3
Retry	BOOL	R	Retry of continuous (1) or non-continuous (0) profile.	Non-continuous (0)	%KWr.m.c.1.4
Retry type	INT	R	Retry with bumps (0: SPi) or bumpless (1: PV).	With bumps (0)	%KWr.m.c.1.5
CONFIG_3	INT	R	Word grouping various configuration bits for profile 3.		%KWr.m.c.2
Guaranteed dwell time	BOOL	R	Confirmation of guaranteed dwell time function (0: no, 1: yes).	No (0)	%KWr.m.c.2.0
Dwell Type	BOOL	R	Type of maintain on guaranteed dwell time: 2 bits.	0	%KWr.m.c.2.1
Dwell Type	BOOL	R		0	%KWr.m.c.2.2

Standard symbol	Type	Access	Meaning	Default Value	Address
Start	BOOL	R	Start with bumps (0: SP0) or bumpless (1: PV).	With bumps (0)	%KWrr.m.c.2.3
Retry	BOOL	R	Retry of continuous (1) or non-continuous (0) profile.	Non-continuous (0)	%KWrr.m.c.2.4
Retry type	BOOL	R	Retry with bumps (0: SPi) or bumpless (1: PV).	With bumps (0)	%KWrr.m.c.2.5
CONFIG_4	INT	R	Word grouping various configuration bits for profile 4.		%KWrr.m.c.3
Guaranteed dwell time	BOOL	R	Confirmation of guaranteed dwell time function (0: no, 1: yes).	No (0)	%KWrr.m.c.3.0
Dwell Type	BOOL	R	Type of maintain on guaranteed dwell time: 2 bits.	0	%KWrr.m.c.3.1
Dwell Type	BOOL	R		0	%KWrr.m.c.3.2
Start	BOOL	R	Start with bumps (0: SP0) or bumpless (1: PV).	With bumps (0)	%KWrr.m.c.3.3
Retry	BOOL	R	Retry of continuous (1) or non-continuous (0) profile.	Non-continuous (0)	%KWrr.m.c.3.4
Retry type	BOOL	R	Retry with bumps (0: SPi) or bumpless (1: PV).	With bumps (0)	%KWrr.m.c.3.5
CONFIG_5	INT	R	Word grouping various configuration bits for profile 5.		%KWrr.m.c.4
Guaranteed dwell time	BOOL	R	Confirmation of guaranteed dwell time function (0: no, 1: yes).	No (0)	%KWrr.m.c.4.0
Dwell Type	BOOL	R	Type of maintain on guaranteed dwell time: 2 bits.	0	%KWrr.m.c.4.1
Dwell Type	BOOL	R		0	%KWrr.m.c.4.2
Start	BOOL	R	Start with bumps (0: SP0) or bumpless (1: PV).	With bumps (0)	%KWrr.m.c.4.3
Retry	BOOL	R	Retry of continuous (1) or non-continuous (0) profile.	Non-continuous (0)	%KWrr.m.c.4.4
Retry type	BOOL	R	Retry with bumps (0: SPi) or bumpless (1: PV).	With bumps (0)	%KWrr.m.c.4.5
CONFIG_6	INT	R	Word grouping various configuration bits for profile 6.		%KWrr.m.c.5
Guaranteed dwell time	BOOL	R	Confirmation of guaranteed dwell time function (0: no, 1: yes).	No (0)	%KWrr.m.c.5.0
Dwell Type	BOOL	R	Type of maintain on guaranteed dwell time: 2 bits.	0	%KWrr.m.c.5.1
Dwell Type	BOOL	R		0	%KWrr.m.c.5.2

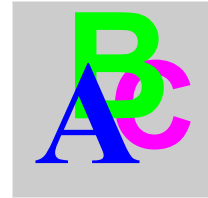
Standard symbol	Type	Access	Meaning	Default Value	Address
Start	BOOL	R	Start with bumps (0: SP0) or bumpless (1: PV).	With bumps (0)	%KWr.m.c.5.3
Retry	BOOL	R	Retry of continuous (1) or non-continuous (0) profile.	Non-continuous (0)	%KWr.m.c.5.4
Retry type	BOOL	R	Retry with bumps (0: SPi) or bumpless (1: PV).	With bumps (0)	%KWr.m.c.5.5
USED_PF1	INT	R	Number of 1 st segment of profile 1.	1	%KWr.m.c.6
USED_PF2	INT	R	Number of 2 st segment of profile 1.	9	%KWr.m.c.7
USED_PF3	INT	R	Number of 3 st segment of profile 1.	17	%KWr.m.c.8
USED_PF4	INT	R	Number of 4 st segment of profile 1.	25	%KWr.m.c.9
USED_PF5	INT	R	Number of 5 st segment of profile 1.	33	%KWr.m.c.10
USED_PF6	INT	R	Number of 6 st segment of profile 1.	41	%KWr.m.c.11
NB_SEG_PF1	INT	R	Number of used segments in profile 1.	8	%KWr.m.c.12
NB_SEG_PF2	INT	R	Number of used segments in profile 2.	8	%KWr.m.c.13
NB_SEG_PF3	INT	R	Number of used segments in profile 3.	8	%KWr.m.c.14
NB_SEG_PF4	INT	R	Number of used segments in profile 4.	8	%KWr.m.c.15
NB_SEG_PF5	INT	R	Number of used segments in profile 5.	8	%KWr.m.c.16
NB_SEG_PF6	INT	R	Number of used segments in profile 6.	8	%KWr.m.c.17
NO_SEG_RT1	INT	R	Number of start segment for profile retry 1.	1	%KWr.m.c.18
NO_SEG_RT2	INT	R	Number of start segment for profile retry 2.	9	%KWr.m.c.19
NO_SEG_RT3	INT	R	Number of start segment for profile retry 3.	17	%KWr.m.c.20
NO_SEG_RT4	INT	R	Number of start segment for profile retry 4.	25	%KWr.m.c.21
NO_SEG_RT5	INT	R	Number of start segment for profile retry 5.	33	%KWr.m.c.22
NO_SEG_RT6	INT	R	Number of start segment for profile retry 6.	41	%KWr.m.c.23

Standard symbol	Type	Access	Meaning	Default Value	Address
CONF_SEG1	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.24
CONF_SEG2	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.25
CONF_SEG3	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.26
CONF_SEG4	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.27
CONF_SEG5	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.28
CONF_SEG6	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.29
CONF_SEG7	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.30
CONF_SEG8	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.31
CONF_SEG9	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.32
CONF_SEG10	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.33
CONF_SEG11	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.34
CONF_SEG12	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.35
CONF_SEG13	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.36
CONF_SEG14	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.37
CONF_SEG15	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.38
CONF_SEG16	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.39
CONF_SEG17	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.40
CONF_SEG18	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.41
CONF_SEG19	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KW r.m.c.42

Standard symbol	Type	Access	Meaning	Default Value	Address
CONF_SEG20	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.43
CONF_SEG21	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.44
CONF_SEG22	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.45
CONF_SEG23	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.46
CONF_SEG24	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.47
CONF_SEG25	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.48
CONF_SEG26	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.49
CONF_SEG27	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.50
CONF_SEG28	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.51
CONF_SEG29	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.52
CONF_SEG30	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 03) of segment.	0	%KWr.m.c.53
CONF_SEG31	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.54
CONF_SEG32	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.55
CONF_SEG33	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.56
CONF_SEG34	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.57
CONF_SEG35	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.58
CONF_SEG36	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.59
CONF_SEG37	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.60
CONF_SEG38	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWr.m.c.61

Standard symbol	Type	Access	Meaning	Default Value	Address
CONF_SEG39	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.62
CONF_SEG40	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.63
CONF_SEG41	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.64
CONF_SEG42	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 03) of segment.	0	%KWrr.m.c.65
CONF_SEG43	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.66
CONF_SEG44	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.67
CONF_SEG45	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.68
CONF_SEG46	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.69
CONF_SEG47	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.70
CONF_SEG48	INT	R	Outputs (bits 8-15) PG (bit 5) Type (bit 4) Unit (bits 0-3) of the segment.	0	%KWrr.m.c.71
SPP_NAME1	INT	R	8 characters over 4 times 2 bytes.		%KWrr.m.c.72
SPP_NAME2	INT	R			%KWrr.m.c.73
SPP_NAME3	INT	R			%KWrr.m.c.74
SPP_NAME4	INT	R			%KWrr.m.c.75

Index



A

adjusting, 175
auto selector loops, 36
auto-tuning, 115
automatic mode, 205

C

cascade loops, 36
channel data structure for process loop modules
 T_PROC_3SING_LOOP, 253
 T_PROC_CASC_LOOP, 274
 T_PROC_CONST_LOOP, 294
 T_PROC_PLOOP, 238
 T_PROC_SPP, 315, 316
configuring, 171

D

debugging, 197

F

feed forward, 79

L

loop controllers, 79

M

manual mode, 205

O

operating modes, 205, 213
 auto selector loops, 218, 224
 cascade loops, 218, 221
 process loops, 218, 219
 single loops, 218, 220
ORDER_COMMAND, 229
outputs, 79

P

PARAM_COMMAND, 229
parameter settings, 229
PID, 115
process control, 33
process loops, 36
process values, 79

S

setpoint programmer, 62
setpoints, 79
single loops, 36

T

T_PROC_3SING_LOOP, 253
T_PROC_CASC_LOOP, 274
T_PROC_CONST_LOOP, 294
T_PROC_PLOOP, 238
T_PROC_SPP, 315

W

WRITE_CMD, 229