

EcoStruxure Machine Expert

EdgeIONTSExpertFunction

Library Guide

EIO000005480.01
04/2026

Legal Information

The information provided in this document contains general descriptions, technical characteristics and/or recommendations related to products/solutions.

This document is not intended as a substitute for a detailed study or operational and site-specific development or schematic plan. It is not to be used for determining suitability or reliability of the products/solutions for specific user applications. It is the duty of any such user to perform or have any professional expert of its choice (integrator, specifier or the like) perform the appropriate and comprehensive risk analysis, evaluation and testing of the products/solutions with respect to the relevant specific application or use thereof.

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

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

Schneider Electric does not grant any right or license for commercial use of the document or its content, except for a non-exclusive and personal license to consult it on an "as is" basis.

Schneider Electric reserves the right to make changes or updates with respect to or in the content of this document or the format thereof, at any time without notice.

To the extent permitted by applicable law, no responsibility or liability is assumed by Schneider Electric and its subsidiaries for any errors or omissions in the informational content of this document, as well as any non-intended use or misuse of the content thereof.

Table of Contents

Safety Information	5
Qualification of Personnel	5
Proper Use	6
Before You Begin	6
Start-up and Test	7
Operation and Adjustments	7
About the Document	9
General Information	15
Presentation of the Library	16
General Information	16
Common Inputs and Outputs	17
General Information	17
Program Organization Units (POU)	18
Encoder Function Blocks	19
General Information	19
<i>Encoder_CAM_NTS</i> : Associates Tracks	19
<i>Encoder_Capture_NTS</i> : Captures the Encoder Position	21
<i>Encoder_Main_NTS</i> : Enables the Encoder Counting	22
<i>Encoder_Preset_NTS</i> : Presets the Encoder	24
<i>Encoder_Reflex_NTS</i> : Sets Threshold Values	25
FastIO_React Function Block	28
<i>FastIO_React_NTS</i> : Generates Output Pulse	28
FreqGen Function Block	30
<i>FreqGen_NTS</i> : Commands a Square Wave Signal	30
HSC Function Blocks	32
<i>HSC_GetParam_NTS</i> : Returns Parameters of HSC	32
<i>HSC_Main_NTS</i> : Control a Main Type Counter for Edge I/O	33
<i>HSC_PWM_NTS</i> : Commands a Pulse Width Modulation Signal	39
<i>HSC_SetParam_NTS</i> : Adjusts Parameters of HSC	41
<i>HSC_Simple_NTS</i> : Controls a Simple Type Counter	43
PTO Administrative Function Blocks	46
<i>MC_AbortTrigger_PTO_NTS</i> : Aborts Function Blocks	46
<i>MC_ReadActualPosition_PTO_NTS</i> : Retrieves Position of the Axis	47
<i>MC_ReadActualVelocity_PTO_NTS</i> : Retrieves Velocity of the Axis	48
<i>MC_ReadAxisError_PTO_NTS</i> : Retrieves Detected Errors	48
<i>MC_ReadBoolParameter_PTO_NTS</i> : Retrieves BOOL Parameters from PTO	49
<i>MC_ReadMotionState_PTO_NTS</i> : Retrieves Motion Status of the Axis	50
<i>MC_ReadParameter_PTO_NTS</i> : Retrieves Parameters from PTO	51
<i>MC_ReadStatus_PTO_NTS</i> : Retrieves the State of the Axis	52
<i>MC_Reset_PTO_NTS</i> : Resets Axis-Related Errors	54
<i>MC_TouchProbe_PTO_NTS</i> : Activates a Trigger Event	55
<i>MC_WriteBoolParameter_PTO_NTS</i> : Writes BOOL Parameters to the PTO	56

<i>MC_WriteParameter_PTO_NTS</i> : Writes Parameters to the PTO	57
PTO Motion Function Blocks	59
Operating Modes	59
Timing Diagram Examples	61
PTO Homing Modes	65
Homing Modes	65
Position Setting	66
Long Reference	67
Long Reference & Index	68
Short Reference Reversal	69
Short Reference No Reversal	70
Short Reference & Index Outside	71
Short Reference & Index Inside	72
PTO Configuration	74
<i>MC_HALT_PTO_NTS</i> : Commands a Controlled Motion Stop until the Velocity equals Zero	75
<i>MC_Home_PTO_NTS</i> : Commands the Axis to Move to a Reference Position	78
<i>MC_MoveAbsolute_PTO_NTS</i> : Commands Motion of a Distance	81
<i>MC_MoveRelative_PTO_NTS</i> : Commands Relative Axis Motion	85
<i>MC_MoveVelocity_PTO_NTS</i> : Controls the Velocity of the Axis	89
<i>MC_Power_PTO_NTS</i> : Manages the Power of the Axis	94
<i>MC_SetPosition_PTO_NTS</i> : Modifies the Reference Position of the Axis	96
<i>MC_Stop_PTO_NTS</i> : Commands a Controlled Motion Stop	97
PWM Function Block	100
PWM Configuration	100
<i>PWM_NTS</i> : Commands a PWM Signal	100
Enumerations and Structures	102
Encoder Enumerations	103
<i>ENCODER_ERROR_ID</i>	103
<i>ENCODER_SUB_FCT_CAM_DIRECTION</i>	104
<i>ENCODER_SUB_FCT_CAM_TYPE</i>	104
Fast_IO Enumeration	105
<i>FAST_IO_ERROR_ID</i>	105
HSC Enumerations	106
<i>HSC_PARAM_TYPE_NTS</i>	106
<i>HSC_ERROR_NTS</i>	107
PTO Enumerations	111
<i>MC_BUFFER_MODE</i>	111
<i>MC_DIRECTION</i>	112
<i>PTO_ERROR_ID</i>	113
<i>PTO_HOMING_MODE</i>	115
<i>PTO_PARAMETER</i>	116
PWM_FREQ_GEN Enumerations	117
<i>FREQ_GEN_ERROR_ID</i>	117
<i>PWM_ERROR_ID</i>	117
Index	119

Safety Information

Important Information

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



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



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

DANGER

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

WARNING

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

CAUTION

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

NOTICE

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

Please Note

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

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

Qualification of Personnel

A qualified person is one who has the following qualifications:

- Skills and knowledge related to the construction and operation of electrical equipment and the installation.
- Knowledge and experience in industrial control programming.
- Received safety-related training to recognize and avoid the hazards involved.

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

Proper Use

- This product is a library to be used together with the control systems and long stator motor segments intended solely for the purposes as described in the present documentation as applied in the industrial sector.
- Always observe the applicable safety-related instructions, the specified conditions, and the technical data.
- Perform a risk evaluation concerning the specific use before using the product. Take protective measures according to the result.
- Since the product is used as a part of an overall system, you must ensure the safety of the personnel by means of the design of this overall system (for example, machine design).
- Any other use is not intended and may be hazardous.

Before You Begin

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

 **WARNING**

UNGUARDED EQUIPMENT

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

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

- This automation equipment and related software is used to control a variety of industrial processes. The type or model of automation equipment suitable for each application will vary depending on factors such as the control function required, degree of protection required, production methods, unusual conditions, government regulations, etc. In some applications, more than one processor may be required, as when backup redundancy is needed.
- Only you, the user, machine builder or system integrator can be aware of all the conditions and factors present during setup, operation, and maintenance of the machine and, therefore, can determine the automation equipment and the related safeties and interlocks which can be properly used. When selecting automation and control equipment and related software for a particular application, you should refer to the applicable local and national standards and regulations. The National Safety Council's Accident Prevention Manual (nationally recognized in the United States of America) also provides much useful information.
- In some applications, such as packaging machinery, additional operator protection such as point-of-operation guarding must be provided. This is necessary if the operator's hands and other parts of the body are free to enter the pinch points or other hazardous areas and serious injury can occur. Software products alone cannot protect an operator from injury. For this reason the software cannot be substituted for or take the place of point-of-operation protection.

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

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

Start-up and Test

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

⚠ WARNING

EQUIPMENT OPERATION HAZARD

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

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

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

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

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

Before energizing equipment:

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

Operation and Adjustments

The following precautions are from the NEMA Standards Publication ICS 7.1-1995:

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

- Regardless of the care exercised in the design and manufacture of equipment or in the selection and ratings of components, there are hazards that can be encountered if such equipment is improperly operated.

- It is sometimes possible to misadjust the equipment and thus produce unsatisfactory or unsafe operation. Always use the manufacturer's instructions as a guide for functional adjustments. Personnel who have access to these adjustments should be familiar with the equipment manufacturer's instructions and the machinery used with the electrical equipment.
- Only those operational adjustments required by the operator should be accessible to the operator. Access to other controls should be restricted to prevent unauthorized changes in operating characteristics.

About the Document

Document Scope

This document describes the functions of the EdgeIONTSExpertFunction library.

For detailed descriptions of the functions, refer to the indicated user guide:

- For High Speed Counter (HSC) functions, refer to the Modicon Edge I/O NTS Counting Modules User Guide.
- For Expert modules functions, refer to the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Validity Note

This document has been updated for the release of EcoStruxure™ Machine Expert V2.6.

Product Related Information

⚠ WARNING

LOSS OF CONTROL

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

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

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

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

⚠ WARNING

IMPROPER USE OF PROGRAM ORGANIZATION UNITS

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

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

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

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

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

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

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION, DATA LOSS, OR FILE CORRUPTION

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

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

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

⚠ WARNING

UNINTENDED EQUIPMENT OPERATION

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

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

General Cybersecurity Information

In recent years, the growing number of networked machines and production plants has seen a corresponding increase in the potential for cyber threats, such as unauthorized access, data breaches, and operational disruptions. You must, therefore, consider all possible cybersecurity measures to help protect assets and systems against such threats.

To help keep your Schneider Electric products secure and protected, it is in your best interest to implement the cybersecurity best practices as described in the [Cybersecurity Best Practices](#) document.

Schneider Electric provides additional information and assistance:

- Subscribe to the Schneider Electric [security newsletter](#).
- Visit the [Cybersecurity Support Portal](#) web page to:
 - Find Security Notifications.
 - Report vulnerabilities and incidents.
- Visit the [Schneider Electric Cybersecurity and Data Protection Posture](#) web page to:
 - Access the cybersecurity posture.
 - Learn more about cybersecurity in the cybersecurity academy.
 - Explore the cybersecurity services from Schneider Electric.

Product Related Cybersecurity Information

Refer to the [Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment User Guide](#).

NOTE: Schneider Electric adheres to industry best practices in the development and implementation of control systems. This includes a "Defense-in-Depth" approach to secure an Industrial Control System. This approach places the controllers behind one or more firewalls to restrict access to authorized personnel and protocols only.

⚠ WARNING

UNAUTHENTICATED ACCESS AND SUBSEQUENT UNAUTHORIZED MACHINE OPERATION

- Evaluate whether your application environments are connected to your critical infrastructure and, if so, take appropriate steps in terms of prevention, based on Defense-in-Depth, before connecting the automation system to any network.
- Limit the number of devices connected to a network to the minimum necessary.
- Isolate your industrial network from other networks inside your company.
- Protect any network against unintended access by using firewalls, VPN, or other, proven security measures such as an Intrusion Prevention System or Intrusion Detection System.
- Monitor activities within your systems.
- Prevent subject devices from direct access or direct link by unauthorized parties or unauthenticated actions.
- Install certificates that are issued by publicly known Trusted Certificate Authorities.
- Keep your systems up to date and rely only on legitimate sources.
- Prepare a recovery plan including backup of your system and process information.

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

For more information on organizational measures and rules covering access to infrastructures, refer to ISO/IEC 27000 series, Common Criteria for Information Technology Security Evaluation, ISO/IEC 15408, IEC 62351, ISA/IEC 62443, NIST Cybersecurity Framework, Information Security Forum - Standard of Good Practice for Information Security and refer to [Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment](#).

Related Documents

Document title	Reference
Cybersecurity Best Practices	CS-Best-Practices-2019-340
Cybersecurity Guidelines for EcoStruxure Machine Expert, Modicon and PacDrive Controllers and Associated Equipment User Guide	EIO0000004242
EcoStruxure Machine Expert, Functions and Libraries User Guide	EIO0000002829 (ENG); EIO0000002830 (FRE); EIO0000002831 (GER); EIO0000002832 (ITA); EIO0000002833 (SPA); EIO0000002834 (CHS);
EcoStruxure Machine Expert, Programming Guide	EIO0000002854 (ENG); EIO0000002855 (FRE); EIO0000002856 (GER); EIO0000002857 (ITA); EIO0000002858 (SPA); EIO0000002859 (CHS);
Modicon Edge I/O NTS, Counting Modules, User Guide	EIO0000005262 (ENG) EIO0000005263 (FRE) EIO0000005264 (GER) EIO0000005265 (SPA) EIO0000005266 (ITA) EIO0000005267 (CHS) EIO0000005268 (POR) EIO0000005269 (TUR)
Modicon Edge I/O NTS Motion Expert Modules, User Guide	EIO0000005254 (ENG)

To find documents online, visit the Schneider Electric download center (www.se.com/ww/en/download/).

Information on Non-Inclusive or Insensitive Terminology

As a responsible, inclusive company, Schneider Electric is constantly updating its communications and products that contain non-inclusive or insensitive terminology. However, despite these efforts, our content may still contain terms that are deemed inappropriate by some customers.

Terminology Derived from Standards

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

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

Among others, these standards include:

Standard	Description
IEC 61131-2:2007	Programmable controllers, part 2: Equipment requirements and tests.
ISO 13849-1:2023	Safety of machinery: Safety related parts of control systems. General principles for design.
EN 61496-1:2020	Safety of machinery: Electro-sensitive protective equipment. Part 1: General requirements and tests.
ISO 12100:2010	Safety of machinery - General principles for design - Risk assessment and risk reduction
EN 60204-1:2006	Safety of machinery - Electrical equipment of machines - Part 1: General requirements
ISO 14119:2013	Safety of machinery - Interlocking devices associated with guards - Principles for design and selection
ISO 13850:2015	Safety of machinery - Emergency stop - Principles for design
IEC 62061:2021	Safety of machinery - Functional safety of safety-related electrical, electronic, and electronic programmable control systems
IEC 61508-1:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: General requirements.
IEC 61508-2:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Requirements for electrical/electronic/programmable electronic safety-related systems.
IEC 61508-3:2010	Functional safety of electrical/electronic/programmable electronic safety-related systems: Software requirements.
IEC 61784-3:2021	Industrial communication networks - Profiles - Part 3: Functional safety fieldbuses - General rules and profile definitions.
2006/42/EC	Machinery Directive
2014/30/EU	Electromagnetic Compatibility Directive
2014/35/EU	Low Voltage Directive

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

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

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

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

General Information

What’s in This Part

Presentation of the Library 16

Common Inputs and Outputs..... 17

Presentation of the Library

What’s in This Chapter

General Information..... 16

General Information

Library Overview

The EdgeIONTSExpertFunction library provides dedicated function blocks to manage Modicon Edge I/O NTS Motion Expert modules.

Characteristics of the Library

The table indicates the characteristics of the library:

Characteristic	Value
Library title	EdgeIONTSExpertFunction
Company	Schneider Electric
Category	Devices > Edge I/O NTS
Component	EdgeIONTSTRepository
Default namespace	EIOEF
Language model attribute	Qualified-access-only (see Functions and Libraries User Guide)
Forward compatible library	Yes (see Functions and Libraries User Guide)

NOTE: For this library, qualified-access-only is set. The POU, data structures, enumerations, and constants must be accessed using the namespace of the library. The default namespace of the library is **EIOEF**.

Controller Platforms

To identify the controllers that are compatible with the library, refer to the List of Controllers Compatible with your Libraries in the Libraries Overview online help.

Common Inputs and Outputs

What's in This Chapter

General Information.....	17
--------------------------	----

General Information

Overview

This chapter provides general information on managing common input and output variables.

Management of Input Variables

When a rising edge is detected at the *Execute* input, the function block starts.

Further modifications of the input variables are not taken into account.

Following the IEC 61131-3 standards, if a variable input to a function block is unavailable, that is, left open or unconnected, then the value from the previous call of the instance of the function block is used. In the first call, the initial, configured value is applied in this case. To help avoid debugging issues, ensure that known values are attributed to the inputs. It is a good practice to use the instance once, and to use the instance in the main task.

Management of Output Variables

The *Done* output is mutually exclusive with the *Busy* and *Error* outputs: only one of them can be TRUE in one function block. If the *Execute* input is TRUE, one of these outputs is TRUE.

At the rising edge of the *Execute* input, the *Busy* output is set. This *Busy* output remains set during the function block execution, and is reset at the rising edge of one of the other outputs (*Done* or *Error*).

The *Done* output is set when the function block execution has been completed successfully.

When a function block execution is terminated due to a detected error, the *Error* output is set and the detected error code is provided with the *ErrorId* output.

The *Done*, *Error*, and *ErrorId* outputs are reset with the falling edge of *Execute*. If the *Execute* input is reset before execution is finished, then the outputs are set for one task cycle at the execution ending.

When an instance of a function block receives another *Execute* before it is finished, the function block does not return feedback, such as *Done*, for the previous action.

Handling a Detected Error

Two outputs can report a detected error during the execution of the function block:

- *Error* = TRUE when an error is detected.
- *ErrorId* returns the error code *HSC_ERROR_NTS*, page 107.

Program Organization Units (POU)

What's in This Part

Encoder Function Blocks	19
FastIO_React Function Block	28
FreqGen Function Block.....	30
HSC Function Blocks.....	32
PTO Administrative Function Blocks	46
PTO Motion Function Blocks	59
PWM Function Block	100

Encoder Function Blocks

What's in This Chapter

General Information.....	19
<i>Encoder_CAM_NTS</i> : Associates Tracks	19
<i>Encoder_Capture_NTS</i> : Captures the Encoder Position.....	21
<i>Encoder_Main_NTS</i> : Enables the Encoder Counting	22
<i>Encoder_Preset_NTS</i> : Presets the Encoder	24
<i>Encoder_Reflex_NTS</i> : Sets Threshold Values	25

General Information

Overview

The encoder function blocks are used to retrieve information from and send commands to the encoder system.

Supported Encoder Modes

The following encoder modes are supported:

Encoder mode	Refer to the chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide
Incremental	Incremental Mode Principle Description
SSI	SSI Mode Principle Description
BiSS-C	BiSS-C Mode Principle Description
SinCos	SinCos Mode Principle Description

Supported Sub-Functions

The following sub-functions are supported:

Sub-Function	Refer to the chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide
CAM	CAM Mode Principle Description
Preset	Preset Sub-Function
Capture	Capture Sub-Function
Reflex output and compare	Reflex Output and Compare Sub-Function

Encoder_CAM_NTS: Associates Tracks

Function Block Description

The *Encoder_CAM_NTS* function block allows you to modify the association between the tracks and the physical outputs of the Modicon Edge I/O NTS Expert I/O Module (for example, NTSECS0121).

For further information, refer to the description of the CAM sub-function in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variables Description

This table describes the input variables:

Input	Data type	Description
<i>Channel</i>	<i>EncoderRef</i>	Reference to the encoder instance.
<i>Enable</i>	BYTE	When the bit associated to the output is TRUE, set the output value according to the track that is to be associated (if <i>SingleTurnPos</i> is valid). - Bit 0 is associated to output 0. ... - Bit 7 is associated to output 7.
<i>Force</i>	BYTE	When TRUE, the bit associated to the output is forced regarding the value of the input <i>ForcedValue</i> . The value of the <i>Enable</i> input is ignored. - Bit 0 is associated to output 0. ... - Bit 7 is associated to output 7.
<i>ForcedValue</i>	BYTE	Value to apply to the output <i>Value</i> when the input <i>Force</i> is set. - Bit 0 is associated to output 0. ... - Bit 7 is associated to output 7.

This table describes the output variables:

Output	Data type	Comment
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>Value</i>	BYTE	Value of the output. - Bit 0 is associated to output 0. ... - Bit 7 is associated to output 7.
<i>TrackStatus</i>	UINT	Each bit represents the status of the corresponding track. - Bit 0 is associated to track 0. ... - Bit 15 is associated to track 15.
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected and the CAM outputs are disabled. The CAM outputs are disabled. You can trigger a rising edge on <i>Enable</i> to reset the detected error.
<i>ErrorId</i>	<i>ENCODER_ERROR_ID</i> , page 103	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.

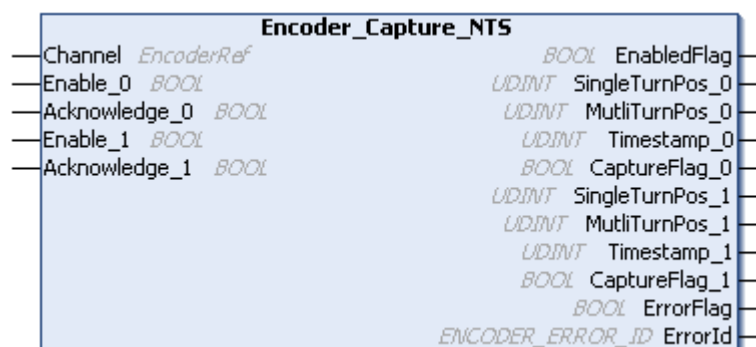
Encoder_Capture_NTS: Captures the Encoder Position

Function Block Description

The *Encoder_Capture_NTS* function block is used to capture the encoder position with two different capture inputs.

For further information, refer to the description of the capture sub-function in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data Type	Description
<i>Channel</i>	EncoderRef	Reference to the encoder instance.
<i>Enable_0</i>	BOOL	When TRUE, the capture input 0 captures the encoder position. The output <i>CaptureFlag_0</i> is set to TRUE and capture 0 is locked if the parameter <i>Autolock</i> of the capture sub-function is TRUE. Default value: FALSE
<i>Acknowledge_0</i>	BOOL	When a rising edge is detected, the output <i>CaptureFlag_0</i> is reset and capture 0 is unlocked if the parameter <i>Autolock</i> of the capture sub-function is TRUE. Default value: FALSE
<i>Enable_1</i>	BOOL	When TRUE, the capture input 1 captures the encoder position. The output <i>CaptureFlag_1</i> is set to TRUE and capture 1 is locked if the parameter <i>Autolock</i> of the capture sub-function is TRUE. Default value: FALSE
<i>Acknowledge_1</i>	BOOL	When a rising edge is detected, the output <i>CaptureFlag_1</i> is reset and capture 1 is unlocked if the parameter <i>Autolock</i> of the capture sub-function is TRUE. Default value: FALSE

This table describes the output variables:

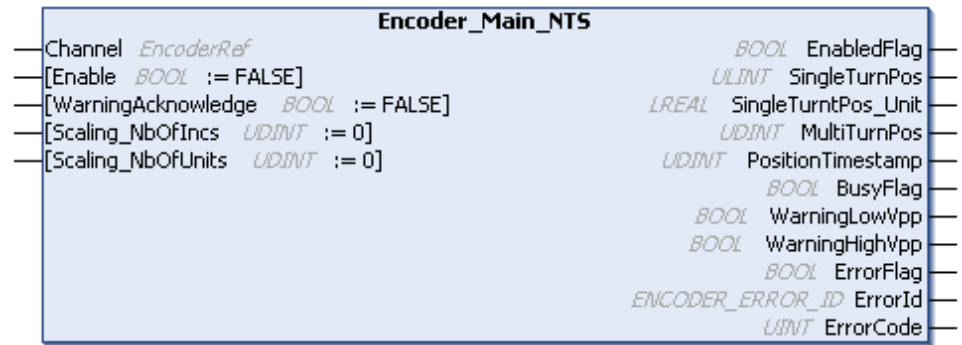
Output	Data Type	Comment
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>SingleTurnPos_0</i>	UDINT	<i>SingleTurnPos</i> value captured at the capture condition 0 event. The value is refreshed whenever a rising edge is detected at <i>CaptureFlag_0</i> .
<i>MultiTurnPos_0</i>	UDINT	<i>MultiTurnPos</i> value captured at the capture condition 0 event. <i>MultiTurnPos</i> values are exclusive to the encoder modes SSI and BiSS-C. The value is refreshed whenever a rising edge is detected at <i>CaptureFlag_0</i> .
<i>Timestamp_0</i>	UDINT	Indicates the time of position 0 capturing.
<i>CaptureFlag_0</i>	BOOL	When TRUE, the captured value at capture input 0 is valid and capture 0 is locked if the parameter <i>Autolock</i> of the capture sub-function is TRUE. A rising edge at input <i>Acknowledge_0</i> resets the output <i>CaptureFlag_0</i> and unlocks capture 0 if the parameter <i>Autolock</i> of the capture sub-function is TRUE.
<i>SingleTurnPos_1</i>	UDINT	<i>SingleTurnPos</i> value captured at the capture condition 1 event. The value is refreshed whenever a rising edge is detected at <i>CaptureFlag_1</i> .
<i>MultiTurnPos_1</i>	UDINT	<i>MultiTurnPos</i> value captured at the capture condition 1 event. <i>MultiTurnPos</i> values are exclusive to the encoder modes SSI and BiSS-C. The value is refreshed whenever a rising edge is detected at <i>CaptureFlag_1</i> .
<i>Timestamp_1</i>	UDINT	Indicates the time of position 1 capturing.
<i>CaptureFlag_1</i>	BOOL	When TRUE, the captured value at capture input 1 is valid and capture 1 is locked if the parameter <i>Autolock</i> of the capture sub-function is TRUE. A rising edge at input <i>Acknowledge_1</i> resets the output <i>CaptureFlag_1</i> and unlocks capture 1 if the parameter <i>Autolock</i> of the capture sub-function is TRUE.
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected. The error needs to be resolved to continue capturing. You can trigger a rising edge on <i>Enable_0</i> or <i>Enable_1</i> to reset the detected error.
<i>ErrorId</i>	<i>ENCODER_ERROR_ID</i> , page 103	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.

Encoder_Main_NTS: Enables the Encoder Counting

Function Block Description

The *Encoder_Main_NTS* function block is used to enable the encoder counting.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Channel</i>	EncoderRef	Reference to the encoder instance.
<i>Enable</i>	BOOL	TRUE enables the function block. When a rising edge is detected, the values of the following scaling parameters are taken into account: <i>Scaling_NbOfIncs</i> <i>Scaling_NbOfUnits</i> If you modify these values, trigger a rising edge on this <i>Enable</i> input to take them into account. Default value: FALSE
<i>WarningAcknowledge</i>	BOOL	This input is exclusive to the SinCos mode. For further information, refer to the <i>SinCos Mode Principle Description</i> in the Modicon Edge I/O NTS Motion Expert Modules User Guide. When a rising edge is detected, the parameters <i>WarningLowVpp</i> and <i>WarningHighVpp</i> are reset. Default value: FALSE
<i>Scaling_NbOfIncs</i>	UDINT	0 indicates that the scaling is disabled. The value in user units of <i>SingleTurnPos_Unit</i> is equal to the value in pulses <i>SingleTurnPos</i>. A value that is greater than 0 indicates that the scaling is enabled. The value in user units is calculated from the value in pulses <i>SingleTurnPos</i>, such as: $\text{SingleTurnPos_Unit} = \text{SingleTurnPos} * (\text{Scaling_NbOfUnits} / \text{Scaling_NbOfIncs})$ Default value: 0
<i>Scaling_NbOfUnits</i>	UDINT	0 indicates that the scaling is disabled. The value in user units of <i>SingleTurnPos_Unit</i> is equal to the value in pulses <i>SingleTurnPos</i>. A value that is greater than 0 indicates that the scaling is enabled. The value in user units is calculated from the value in pulses <i>SingleTurnPos</i>, such as: $\text{SingleTurnPos_Unit} = \text{SingleTurnPos} * (\text{Scaling_NbOfUnits} / \text{Scaling_NbOfIncs})$ Default value: 0

This table describes the output variables:

Output	Data type	Description
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE. Default value: FALSE
<i>SingleTurnPos</i>	ULINT	Indicates a single-turn position of the machine in pulses. Refreshed with each cycle.
<i>SingleTurnPos_Unit</i>	LREAL	0 indicates that the scaling is disabled. The value of <i>SingleTurnPos_Unit</i> is equal to <i>SingleTurnPos</i> . A value that is unequal to 0 indicates that the scaling is enabled. The value of <i>SingleTurnPos_Unit</i> is calculated from $SingleTurnPos * Scaling_NbOfUnits / Scaling_NbOfIncs$ Default value: 0
<i>MultiTurnPos</i>	UDINT	Indicates a multi-turn position of the machine in pulses. Refreshed with each cycle.
<i>PositionTimestamp</i>	UDINT	Indicates the time of position capturing.
<i>BusyFlag</i>	BOOL	When TRUE, indicates that the encoder is being enabled.
<i>WarningLowVpp</i>	BOOL	When TRUE, the lower limit of the SinCos signal (low Vpp) is not taken into account.
<i>WarningHighVpp</i>	BOOL	When TRUE, the upper limit of the SinCos signal (high Vpp) is not taken into account.
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected. You can trigger a rising edge on <i>Enable</i> to reset the detected error. Default value: FALSE
<i>ErrorId</i>	<i>ENCODER_ERROR_ID</i> , page 103	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.
<i>ErrorCode</i>	UINT	Indicates the code of the detected error when <i>ErrorFlag</i> is TRUE and <i>ErrorId</i> is <i>PROTOCOL</i> .

Encoder_Preset_NTS: Presets the Encoder

Function Block Description

The *Encoder_Preset_NTS* function block is used to preset the encoder with a reference position value.

For further information, refer to the description of the Preset sub-function in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Inputs	Data type	Description
<i>Channel</i>	EncoderRef	Reference to the encoder instance.
<i>Enable</i>	BOOL	When TRUE, the encoder preset sub-function is enabled and when a preset event is detected, a preset is triggered and the <i>PresetFlag</i> output is set to TRUE. Default value: FALSE
<i>Force</i>	BOOL	When a rising edge is detected, the preset is forced and the <i>PresetFlag</i> output is set to TRUE. Default value: FALSE
<i>Acknowledge</i>	BOOL	When a rising edge is detected, the <i>PresetFlag</i> output is reset. Default value: FALSE
<i>SingleTurnPos</i>	UDINT	Sets the preset position for single-turn bits. NOTE: It is not verified whether the position is within the valid range.
<i>MultiTurnPos</i>	UDINT	<i>MultiTurnPos</i> values are exclusive to the encoder modes SSI and BiSS-C. Sets the preset position for multi-turn bits. NOTE: It is not verified whether the position is within the valid range.

This table describes the output variables:

Output	Data type	Description
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>PresetFlag</i>	BOOL	TRUE indicates that a preset is performed. Default value: FALSE
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected. You can trigger a rising edge on <i>Enable</i> to reset the detected error. Default value: FALSE
<i>ErrorId</i>	ENCODER_ERROR_ID, page 103	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.

Encoder_Reflex_NTS: Sets Threshold Values

Function Block Description

The *Encoder_Reflex_NTS* function block is used to set outputs on threshold values.

For further information, refer to the description of the Reflex Output and Compare sub-function in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Channel</i>	EncoderRef	Reference to the encoder instance.
<i>Enable</i>	BYTE	Bit field used to enable reflex outputs 0 through 3. When the bit associated to the output is TRUE, set the output used for the reflex value (if <i>SingleTurnPos</i> is valid). - Bit 0 is associated to reflex 0. - Bit 1 is associated to reflex 1. - Bit 2 is associated to reflex 2. - Bit 3 is associated to reflex 3. Bits 4...7 are reserved.
<i>Force</i>	BYTE	Bit field used to force reflex outputs 0 through 3. When TRUE, the associated bit corresponding to the reflex value is forced regarding the value of the output <i>ReflexForcedValue</i> of the reflex output and compare sub-function. The value of the <i>Enable</i> input is ignored. - Bit 0 is associated to reflex 0. - Bit 1 is associated to reflex 1. - Bit 2 is associated to reflex 2. - Bit 3 is associated to reflex 3. Bits 4...7 are reserved.
<i>ForcedValue</i>	BYTE	Bit field of the forced values for reflex outputs 0 through 3. Value to apply when the function block output <i>ReflexForce</i> of the reflex output and compare sub-function is set. - Bit 0 is associated to reflex 0. - Bit 1 is associated to reflex 1. - Bit 2 is associated to reflex 2. - Bit 3 is associated to reflex 3. Bits 4...7 are reserved.
<i>SuspendCompare</i>	BOOL	When TRUE, the reflex output values are latched and maintained independently of the present values of the encoder and the threshold values. If <i>SuspendCompare</i> is TRUE, a reflex output can still be forced to the <i>ForcedValue</i> with the <i>Force</i> input.

This table describes the output variables:

Output	Data type	Description
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>Value</i>	BYTE	Present value of the reflex bit. • Bit 0 is associated to reflex 0. • Bit 1 is associated to reflex 1. • Bit 2 is associated to reflex 2. • Bit 3 is associated to reflex 3. Bits 4...7 are reserved. NOTE: The physical outputs depend on the configured channel. For further information, refer to the description of the Reflex Output and Compare sub-function in the Modicon Edge I/O NTS Motion Expert Modules User Guide.
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected. You can trigger a rising edge on <i>Enable</i> to reset the detected error. Default value: FALSE
<i>ErrorId</i>	<i>ENCODER_ERROR_ID</i> , page 103	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.

FastIO_React Function Block

What’s in This Chapter

FastIO_React_NTS: Generates Output Pulse..... 28

FastIO_React_NTS: Generates Output Pulse

Function Block Description

The *FastIO_React_NTS* function block supports the Output Pulse After Input Event mode and generates an output pulse after an input edge and a delay. For further information, refer to the chapter Output Pulse After Input Event Mode Principle Description in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Channel</i>	FastIoRef	Reference to the fast I/O instance.
<i>Enable</i>	BOOL	When TRUE, the input is monitored and the pulse generation is enabled. Default value: FALSE
<i>EventTime</i>	UDINT	The time (in ns) between detecting the input event and generating the output pulse. Value range: 0...4294967295 Default value: 0
<i>PulseTime</i>	UDINT	The duration (in ns) of the generated output pulse. Value range: 0...4294967295 Default value: 0

This table describes the output variables:

Output	Data type	Description
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>PulseNb</i>	BYTE	The number of pulses that are generated when an input event is detected. Default value: 0
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected. You can trigger a rising edge on <i>Enable</i> to reset the detected error. Default value: FALSE
<i>ErrorId</i>	<i>FAST_IO_ERROR_ID</i> , page 105	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.

FreqGen Function Block

What’s in This Chapter

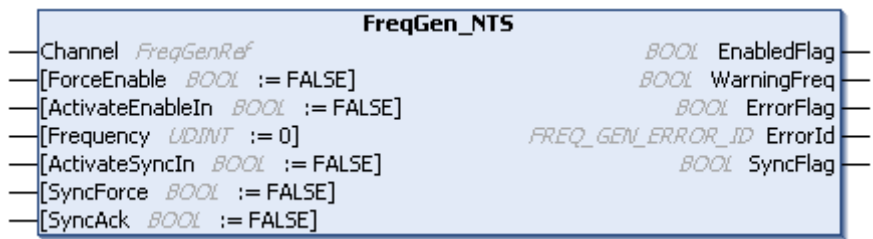
FreqGen_NTS: Commands a Square Wave Signal 30

FreqGen_NTS: Commands a Square Wave Signal

Function Block Description

The *FreqGen_NTS* function block supports the Frequency Generator mode and commands a square wave signal output at the specified frequency. For further information, refer to the chapter Frequency Generator Mode Principle Description in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Channel</i>	FreqGenRef	Reference to the frequency generator instance.
<i>ForceEnable</i>	BOOL	When TRUE, forces the pulse generation. The <i>ActivateEnableIn</i> input is ignored. Default value: FALSE
<i>ActivateEnableIn</i>	BOOL	When TRUE, enables pulse generation provided that the <i>EnConfigured</i> input is TRUE. Default value: FALSE
<i>Frequency</i>	UDINT	Frequency to apply to the pulse generation in steps of 0.1 Hz. Value range: 0...1,000,000 (0...100 kHz) Default value: 0
<i>ActivateSyncIn</i>	BOOL	When TRUE, the synchronization is started when a <i>SyncConfigured</i> input parameter is defined other than 0 and the <i>SyncFlag</i> output is set to TRUE. Default value: FALSE
<i>SyncForce</i>	BOOL	When a rising edge is detected, forces the synchronization and sets the <i>SyncFlag</i> output to TRUE independently of the defined <i>SyncConfigured</i> input parameter. Default value: FALSE
<i>SyncAck</i>	BOOL	When a rising edge is detected, the <i>SyncFlag</i> output is reset. Default value: FALSE

This table describes the output variables:

Output	Data type	Description
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>WarningFreq</i>	BOOL	When TRUE, the value of the input <i>Frequency</i> exceeds the maximum frequency. The applied frequency is the maximum frequency. Default value: FALSE
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected. Pulse generation is stopped. You can trigger a rising edge on <i>ForceEnable</i> to reset the detected error. Default value: FALSE
<i>ErrorId</i>	<i>FREQ_GEN_ERROR_ID</i> , page 117	Indicates the identification number of the detected error when <i>ErrorFlag</i> is TRUE.
<i>SyncFlag</i>	BOOL	When TRUE, synchronization is being performed. Default value: FALSE

HSC Function Blocks

What's in This Chapter

<i>HSC_GetParam_NTS</i> : Returns Parameters of HSC	32
<i>HSC_Main_NTS</i> : Control a Main Type Counter for Edge I/O.....	33
<i>HSC_PWM_NTS</i> : Commands a Pulse Width Modulation Signal	39
<i>HSC_SetParam_NTS</i> : Adjusts Parameters of HSC	41
<i>HSC_Simple_NTS</i> : Controls a Simple Type Counter	43

HSC_GetParam_NTS: Returns Parameters of HSC

Function Block Description

The *HSC_GetParam_NTS* administrative function block returns a parameter value of a High Speed Counter (HSC).

Graphical Representation



I/O Variables Description

This table describes the input variables:

Inputs	Type	Comment
<i>HSC_REF_NTS</i>	HSCRef	Reference of the HSC instance. Must not be changed during function block execution.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>Param</i>	<i>HSC_PARAM_TYPE_NTS</i> , page 106	Parameter to read.

This table describes the output variables:

Outputs	Type	Comment
<i>Done</i>	BOOL	TRUE indicates that <i>ParameterValue</i> is valid. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>HSC_ERROR_NTS</i> , page 107	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>ParameterValue</i>	DINT	Value of the parameter that has been read.

NOTE: For more information about the *Done*, *Busy*, *Error*, and *ErrorId* parameters, refer to General Information, page 17.

HSC_Main_NTS: Control a Main Type Counter for Edge I/O

Function Block Description

The *HSC_Main_NTS* function block controls a **Main** type counter with the following functions:

Function	Refer to the chapter in the Modicon Edge I/O NTS Counting Modules User Guide for further information
Frequency meter	Frequency Meter Function
Ratio meter	Ratio Meter Function
Period meter	Period Meter Function
Single event counting	Single Phase Event Counting Function
Single phase counting	Single Phase Counting Function
Dual phase counting	Dual Phase Counting Function
Reflex	Reflex Output Sub Function

The *HSC_Main_NTS* function block is mandatory when using **Main** counter.

The function block instance name must match the name defined by configuration. Hardware related information managed by this function block is synchronized with the MAST task cycle.

WARNING

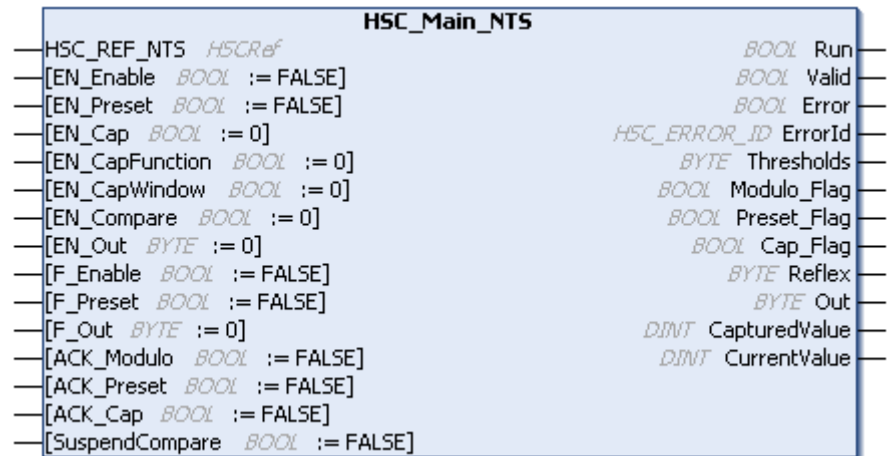
UNINTENDED OUTPUT VALUES

- Only use the function block instance in the MAST task.
- Do not use the same function block instance in a different task.

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

NOTE: Forcing the logical output values of the function block is allowed by EcoStruxure Machine Expert but it will have no impact on physical outputs if the function is active (executing).

Graphical Representation



I/O Variables Description

This table describes the input variables:

Input	Type	Description
<i>HSC_REF_NTS</i>	HSCRef	Reference to the HSC instance. Must not be changed during function block execution.
<i>EN_Enable</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 0. For further information, refer to the Enable Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE authorizes enabling of the counter using the <i>Enable</i> input.
<i>EN_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 1. For further information, refer to the Preset Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE authorizes counter synchronization and start using the input defined in the <i>SynchInputLocation</i> parameter.
<i>EN_Cap</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 2. For further information, refer to the Capture Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE enables the input defined in the <i>CaptureInputLocation</i> parameter (if configured in Single Phase Counting and Dual Phase Counting).
<i>EN_CapFunction</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 3. For further information, refer to the Capture Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. If <i>EN_CapFunction</i> and <i>EN_Cap</i> are TRUE, the capture function can operate. NOTE: <i>EN_Cap</i> is the prerequisite for <i>EN_CapFunction</i> . <i>EN_CapFunction</i> is the prerequisite for <i>EN_CapWindow</i> .
<i>EN_CapWindow</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 4. For further information, refer to the Capture Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. If TRUE, the capture window is enabled. The range of the capture window is configured with <i>CaptureWindowStartPosition</i> and <i>CaptureWindowEndPosition</i> . If FALSE, the capture function can operate without the restriction of capture window.
<i>EN_Compare</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 5. For further information, refer to the Compare Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. If TRUE, the comparator for reflex output (using thresholds 0, 1, 2, 3) is enabled.

Input	Type	Description
<i>EN_Out</i>	BYTE	Corresponds to <i>EnableReflexOutput</i> of the Reflex Output Sub Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. Set bits to 1 to enable the corresponding physical outputs to echo the configured function value (Reflex or Stop) as a result of the comparison function. Only active when outputs are configured in the HSC editor: • Bit 0: Output 0 enabled. • Bit 1: Output 1 enabled. • Bit 2: Output 2 enabled. • Bit 3: Output 3 enabled. • Bits 4...7: Not used. Association of the HSC output <i>Outx</i> with the physical output <i>Qy</i> is done by configuration.
<i>F_Enable</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 7. For further information, refer to the Enable Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE activates the counter and takes into account pulses on the counter input.
<i>F_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 8. For further information, refer to the Preset Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. When a rising edge is detected, the counting function synchronization is authorized and the counters are started.
<i>F_Out</i>	BYTE	Corresponds to <i>ForceReflexOutput</i> of the Reflex Output Sub Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. Set bits to 1 to force the corresponding physical outputs to 1 if associated with HSC by configuration. Takes priority over <i>EN_Out</i>. Only active when outputs are configured in the HSC editor: • Bit 0: Output 0 forced. • Bit 1: Output 1 forced. • Bit 2: Output 2 forced. • Bit 3: Output 3 forced. • Bits 4...7: Not used. Association of the HSC output <i>Outx</i> with the physical output <i>Qy</i> is done by configuration.
<i>ACK_Modulo</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 10. When a rising edge is detected, the <i>Modulo_Flag</i> output is reset.
<i>ACK_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 11. When a rising edge is detected, the <i>Preset_Flag</i> is reset.

Input	Type	Description
<i>ACK_Cap</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 12. When a rising edge is detected, the <i>Cap_Flag</i> is reset.
<i>SuspendCompare</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 6. TRUE disables the comparator for the <i>Reflex</i> output. When <i>SuspendCompare</i> (<i>OperationalCommand</i> bit 6) is set to TRUE, the compare function is suspended. The parameters related to the compare function, <i>ReflexState</i>, <i>ThresholdState</i>, <i>OutputState</i> are maintained at their values. NOTE: If the compare function is suspended while the <i>OutputState</i> is set to TRUE for the duration of a <i>ReflexOutput•PulseWidth</i> value, the <i>OutputState</i> continues the normal operation and is set to FALSE at the end of the configured duration. For further information, refer to the chapter <i>Reflex Output Sub Function</i> in the <i>Modicon Edge I/O NTS Counting Modules User Guide</i>.

This table describes the output variables:

Outputs	Type	Comment
<i>Run</i>	BOOL	Corresponds to <i>OperationalState</i> bit 0. TRUE activates the counter.
<i>Valid</i>	BOOL	Corresponds to <i>OperationalState</i> bit 1. TRUE indicates that valid data is available at the function block output pin.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished. For further information, refer to <i>General Information</i> , page 17.
<i>ErrorId</i>	<i>HSC_ERROR_NTS</i> , page 107	Indicates the identification number of the detected error when <i>Error</i> is TRUE. For further information, refer to <i>General Information</i> , page 17.
<i>Thresholds</i>	BYTE	Corresponds to <i>ThresholdState</i> of the Reflex Output Sub Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. The bits are set to 1 when the counter value <i>CurrentValue</i> is superior or equal to the <i>Thresholds</i> value for the corresponding threshold: • Bit 0: $CurrentValue \geq Threshold0$ • Bit 1: $CurrentValue \geq Threshold1$ • Bit 2: $CurrentValue \geq Threshold2$ • Bit 3: $CurrentValue \geq Threshold3$ • Bits 4...7: Not used
<i>Modulo_Flag</i>	BOOL	Corresponds to <i>OperationalState</i> bit 2. This output indicates if the counter looped back to 0 after reaching the modulo value. • TRUE when the counter loops back to 0. • FALSE when a rising edge is detected at the <i>ACK_Modulo</i> input.
<i>Preset_Flag</i>	BOOL	Corresponds to <i>OperationalState</i> bit 3. Set to TRUE when you synchronize the counter value.
<i>Cap_Flag</i>	BOOL	Corresponds to <i>OperationalState</i> bit 4. TRUE indicates that a value has been latched in the capture register. This flag must be reset before a new capture can occur.
<i>Reflex</i>	BYTE	Corresponds to <i>ReflexState</i> of the Reflex Output Sub Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. State of the reflex function: • Bit 0: Reflex 0 • Bit 1: Reflex 1 • Bit 2: Reflex 2 • Bit 3: Reflex 3 • Bits 4...7: Not used

Outputs	Type	Comment
<i>Out</i>	BYTE	Corresponds to <i>OutputState</i> of the Reflex Output Sub Function in the Modicon Edge I/O NTS, Counting Modules, User Guide). State of the physical outputs: • Bit 0: Output 0 • Bit 1: Output 1 • Bit 2: Output 2 • Bit 3: Output 3 • Bits 4...7: Not used Only active when the outputs are configured in the Counters configuration tab. Association of the HSC output <i>Outx</i> with the physical output <i>Qy</i> is done by the configuration.
<i>CapturedValue</i>	DINT	The value of the captured counter value. -2,147,483,648...2,147,483,647
<i>CurrentValue</i>	DINT	The value of the counter. -2,147,483,648...2,147,483,647

HSC_PWM_NTS: Commands a Pulse Width Modulation Signal

Function Block Description

The *HSC_PWM_NTS* function block controls the PWM (Pulse Width Modulated) output.

The function block instance name must match the name defined by configuration. Hardware related information managed by this function block is synchronized with the MAST task cycle.

For detailed information, refer to the PWM Output Function chapter in the Modicon Edge I/O NTS Counting Modules User Guide.

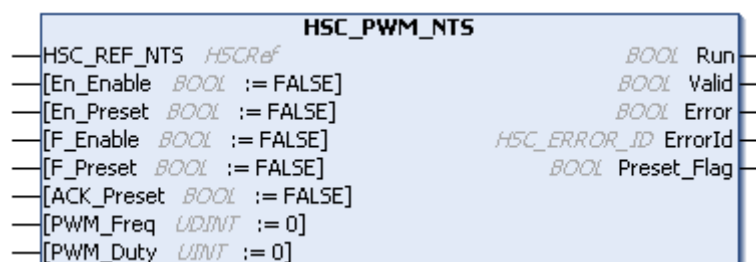
⚠ WARNING

UNINTENDED OUTPUT VALUES

- Only use the function block instance in the MAST task.
- Do not use the same function block instance in a different task.

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

Graphical Representation



I/O Variables Description

This table describes the input variables:

Inputs	Type	Comment
<i>HSC_REF_NTS</i>	HSCRef	Reference to the HSC instance. Must not be changed during function block execution.
<i>En_Enable</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 0. For further information, refer to the Enable Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE authorizes enabling of the Pulse Width Modulation (PWM) using the <i>Enable</i> input.
<i>En_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 1. For further information, refer to the Preset Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE authorizes PWM output synchronization and start using the input defined in the <i>SyncInputLocation</i> parameter.
<i>F_Enable</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 7. For further information, refer to the Enable Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. TRUE activates the <i>PWMOutputLocation</i> output.
<i>F_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 8. For further information, refer to the Preset Function in the Modicon Edge I/O NTS, Counting Modules, User Guide. When a rising edge is detected, the <i>PWMOutputLocation</i> output function synchronization is authorized.
<i>ACK_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 11. When a rising edge is detected, the <i>Preset_Flag</i> is reset.
<i>PWM_Freq</i>	UDINT	Frequency of the <i>PWM</i> output. Value range: 0...2000000 (20 kHz) For example, 123456 means 1234.56 Hz.
<i>PWM_Duty</i>	UINT	Duty cycle of the <i>PWM</i> output. Value range: 0...1000 (100 %) For example, 123 means 12.3 %.

This table describes the output variables:

Outputs	Type	Comment
<i>Run</i>	BOOL	Corresponds to <i>OperationalState</i> bit 0. TRUE indicates that the <i>PWMOutputLocation</i> output is activated.
<i>Valid</i>	BOOL	Corresponds to <i>OperationalState</i> bit 1. TRUE indicates that the <i>PWMOutputLocation</i> output is valid. Frequency and duty values are within the allowed range.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished. For further information, refer to <i>General Information</i> , page 17.
<i>ErrorId</i>	<i>HSC_ERROR_NTS</i> , page 107	Indicates the identification number of the detected error when <i>Error</i> is TRUE. For further information, refer to <i>General Information</i> , page 17.
<i>Preset_Flag</i>	BOOL	Corresponds to <i>OperationalState</i> bit 3. Set to TRUE when you synchronize the <i>PWMOutputLocation</i> output.

HSC_SetParam_NTS: Adjusts Parameters of HSC

Function Block Description

The *HSC_SetParam_NTS* administrative function block modifies the value of a parameter of an HSC.

Graphical Representation



I/O Variables Description

This table describes the input variables:

Inputs	Type	Comment
<i>HSC_REF_NTS</i>	HSCRef	Reference to the HSC function block. Must not be changed during function block execution.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>Param</i>	<i>HSC_PARAM_TYPE_NTS</i> , page 106	Parameter to set.
<i>ParameterValue</i>	DINT	Parameter value to write.

This table describes the output variables:

Outputs	Type	Comment
<i>Done</i>	BOOL	TRUE indicates that the parameter value was written. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>HSC_ERROR_NTS</i> , page 107	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

NOTE: For more information about the *Done*, *Busy*, *Error*, and *ErrorId* parameters, refer to General Information, page 17.

HSC_Simple_NTS: Controls a Simple Type Counter

Function Block Description

The *HSC_Simple_NTS* function block controls a **Simple Counting** counter with the following operating modes:

- One-shot counting
- Modulo-loop counting

The *HSC_Simple_NTS* function block is mandatory when using a **Simple Counting** counter.

For detailed information, refer to the Simple Counting Function chapter in the Modicon Edge I/O NTS Counting Modules User Guide.

The function block instance name must match the name defined by configuration. Hardware related information managed by this function block is synchronized with the MAST task cycle.

⚠ WARNING

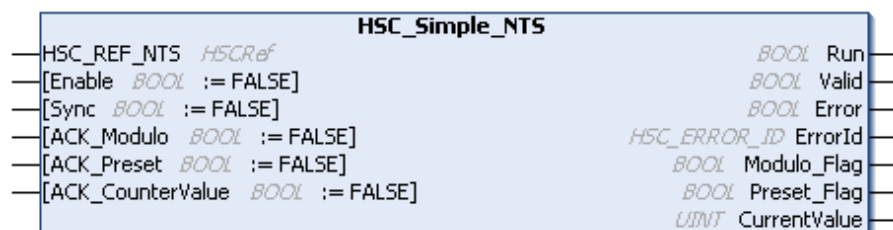
UNINTENDED OUTPUT VALUES

- Only use the function block instance in the MAST task.
- Do not use the same function block instance in a different task.

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

NOTE: Forcing the logical output values of the function block is allowed by EcoStruxure Machine Expert but it will have no impact on physical outputs if the function is active (executing).

Graphical Representation



I/O Variables Description

This table describes the input variables:

Inputs	Type	Comment
<i>HSC_REF_NTS</i>	HSCRef	Reference to the HSC instance.
<i>Enable</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 0. TRUE activates the counter and takes into account pulses on the counter input.
<i>Sync</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 1. One-shot counter: When a rising edge is detected, loads the preset of the counter. Modulo-loop counter: When a rising edge is detected, resets and initializes the counter.
<i>ACK_Modulo</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 2. Modulo-loop mode: When a rising edge is detected, resets the modulo flag <i>Modulo_Flag</i> .
<i>ACK_Preset</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 3. When a rising edge is detected, resets the <i>Preset_Flag</i> .
<i>ACK_CounterValue</i>	BOOL	Corresponds to <i>OperationalCommand</i> bit 4. When a rising edge is detected, resets the present counter value to 0.

This table describes the output variables:

Outputs	Type	Comment
<i>Run</i>	BOOL	Corresponds to <i>OperationalState</i> bit 0. TRUE indicates that the counter is activated. In One-shot mode, switches to 0 when <i>CurrentValue</i> reaches 0. A rising edge on <i>Sync</i> is needed to restart the counter. In Modulo-loop mode, TRUE indicates that the counter is activated.
<i>Valid</i>	BOOL	Corresponds to <i>OperationalState</i> bit 1. TRUE indicates that the output values on the function block are valid.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished. For further information, refer to <i>General Information</i>, page 17.
<i>ErrorId</i>	<i>HSC_ERROR_NTS</i> , page 107	Indicates the identification number of the detected error when <i>Error</i> is TRUE. For further information, refer to <i>General Information</i>, page 17.
<i>Modulo_Flag</i>	BOOL	Corresponds to <i>OperationalState</i> bit 2. Modulo-loop mode: Set to TRUE when the counter exceeds the modulo value.
<i>Preset_Flag</i>	BOOL	Corresponds to <i>OperationalState</i> bit 3. Set to TRUE when you synchronize the counter value. - In one-shot mode: preset the counter value. - In modulo-loop mode: reset the counter value.
<i>CurrentValue</i>	UINT	The value of the counter. Value range: 0...65,535

PTO Administrative Function Blocks

What's in This Chapter

MC_AbortTrigger_PTO_NTS: Aborts Function Blocks

46

MC_ReadActualPosition_PTO_NTS: Retrieves Position of the Axis

47

MC_ReadActualVelocity_PTO_NTS: Retrieves Velocity of the Axis

48

MC_ReadAxisError_PTO_NTS: Retrieves Detected Errors

48

MC_ReadBoolParameter_PTO_NTS: Retrieves BOOL Parameters from PTO

49

MC_ReadMotionState_PTO_NTS: Retrieves Motion Status of the Axis

50

MC_ReadParameter_PTO_NTS: Retrieves Parameters from PTO

51

MC_ReadStatus_PTO_NTS: Retrieves the State of the Axis

52

MC_Reset_PTO_NTS: Resets Axis-Related Errors

54

MC_TouchProbe_PTO_NTS: Activates a Trigger Event

55

MC_WriteBoolParameter_PTO_NTS: Writes BOOL Parameters to the PTO

56

MC_WriteParameter_PTO_NTS: Writes Parameters to the PTO

57

MC_AbortTrigger_PTO_NTS: Aborts Function Blocks

Function Block Description

The *MC_AbortTrigger_PTO_NTS* function block aborts the *MC_TouchProbe_PTO_NTS*, page 55 function block as it executes the *AbortTrigger* administrative command. For further information, refer to the table that describes PTO administrative commands in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
Axis	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
Execute	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.

This table describes the output variables:

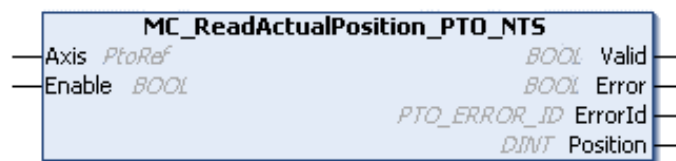
Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the function block execution is finished with no error detected. The execution of the administrative command is done.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

MC_ReadActualPosition_PTO_NTS: Retrieves Position of the Axis

Function Block Description

The *MC_ReadActualPosition_PTO_NTS* function block returns the present position of the axis.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Enable</i>	BOOL	When TRUE, the function block is executed. The position values of the axis are continuously retrieved. When FALSE, function block execution is terminated and the outputs are reset.

This table describes the output variables:

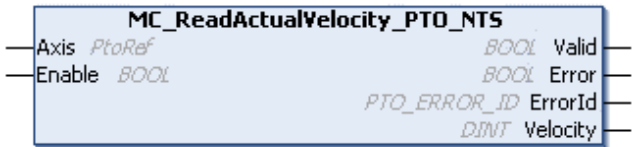
Output	Data type	Description
<i>Valid</i>	BOOL	TRUE indicates that valid data is available at the function block output pin.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>Position</i>	DINT	Present position of the axis (in number of pulses).

MC_ReadActualVelocity_PTO_NTS: Retrieves Velocity of the Axis

Function Block Description

The *MC_ReadActualVelocity_PTO_NTS* function block returns the value of the commanded velocity of the axis.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
Axis	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
Enable	BOOL	When TRUE, the function block is executed. The velocity values of the axis are continuously retrieved. When FALSE, terminates the function block execution and resets its outputs.

This table describes the output variables:

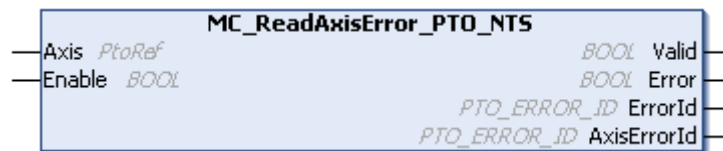
Output	Data type	Description
Valid	BOOL	TRUE indicates that valid data is available at the function block output pin.
Error	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
ErrorId	PTO_ERROR_ID, page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
Velocity	DINT	Present velocity of the axis (in Hz).

MC_ReadAxisError_PTO_NTS: Retrieves Detected Errors

Function Block Description

The *MC_ReadAxisError_PTO_NTS* retrieves the axis control error. If no axis control error is detected, the function block returns *AxisErrorId* = 0.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Enable</i>	BOOL	When TRUE, the function block is executed. The information on detected axis errors are continuously retrieved. When FALSE, terminates the function block execution and resets its outputs.

This table describes the output variables:

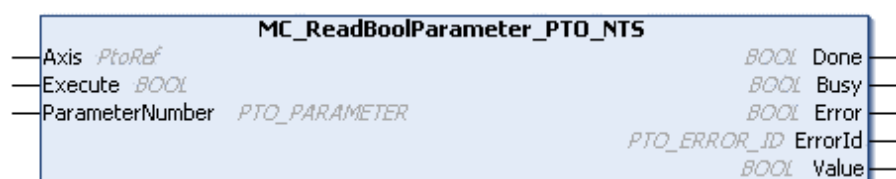
Output	Data type	Description
<i>Valid</i>	BOOL	TRUE indicates that valid data is available at the function block output pin.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>AxisErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected axis error (described in the Index 1000 of <i>PTO_ERROR_ID</i> , page 113).

MC_ReadBoolParameter_PTO_NT5: Retrieves BOOL Parameters from PTO

Function Block Description

The *MC_ReadBoolParameter_PTO_NT5* function block retrieves the value of a Boolean parameter.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>ParameterNumber</i>	<i>PTO_PARAMETER</i> , page 116	Parameter name or parameter number of the parameter requested from <i>PTO_PARAMETER</i> , page 116.

This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that valid data is available at the function block output pin.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>Value</i>	BOOL	Value of the requested Boolean parameter.

MC_ReadMotionState_PTO_NTS: Retrieves Motion Status of the Axis

Function Block Description

The *MC_ReadMotionState_PTO_NTS* returns the motion status of the axis.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Enable</i>	BOOL	When TRUE, information on the motion status of the axis are continuously retrieved When FALSE, terminates the function block execution and resets its outputs.

This table describes the output variables:

Output	Data type	Description
<i>Valid</i>	BOOL	TRUE indicates that valid data is available at the function block output pin.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>ConstantVelocity</i>	BOOL	The velocity is constant. The value 0 is allowed.
<i>Accelerating</i>	BOOL	The velocity value is increasing.
<i>Decelerating</i>	BOOL	The velocity value is decreasing.

MC_ReadParameter_PTO_NTS: Retrieves Parameters from PTO

Function Block Description

The *MC_ReadParameter_PTO_NTS* function block retrieves the value of a vendor-specific parameter, except of a Boolean data type.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>ParameterNumber</i>	<i>PTO_PARAMETER</i> , page 116	ID of the parameter requested from <i>PTO_PARAMETER</i> , page 116.

This table describes the output variables:

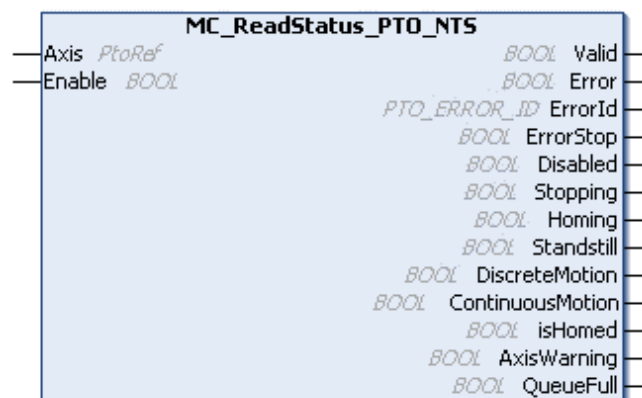
Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that valid data is available at the function block output pin.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>Value</i>	DINT	Value of the requested parameter.

MC_ReadStatus_PTO_NTS: Retrieves the State of the Axis

Function Block Description

The *MC_ReadStatus_PTO_NTS* function block retrieves the state diagram, page 59 status of the axis.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Enable</i>	BOOL	When TRUE, information on the state diagram, page 59 status of the axis are continuously retrieved. When FALSE, terminates the function block execution and resets its outputs.

This table describes the output variables:

Output	Data type	Description
<i>Valid</i>	BOOL	TRUE indicates that valid data is available at the function block output pin.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>ErrorStop</i>	BOOL	The <i>ErrorStop</i> state is assigned the highest priority. It is applicable if an error is detected on the axis or in the controller. The <i>ErrorStop</i> state is maintained as long as the <i>Error</i> is TRUE. No motion command is accepted until a reset is performed using the function block <i>MC_Reset_PTO_NTS</i> , page 54.
<i>Disabled</i>	BOOL	The <i>Disabled</i> state is the initial state of the axis. No motion command is allowed. The axis is not homed.
<i>Stopping</i>	BOOL	The <i>Stopping</i> state is applicable when the axis is controlled by the function block <i>MC_Stop_PTO_NTS</i> , page 97.
<i>Homing</i>	BOOL	The <i>Homing</i> state is applicable when the axis is controlled by the function block <i>MC_Home_PTO_NTS</i> , page 78.
<i>Standstill</i>	BOOL	The <i>Standstill</i> state is the initial state of the axis. No error is detected. No motion commands are executed on the axis.
<i>DiscreteMotion</i>	BOOL	The <i>DiscreteMotion</i> state is applicable when the axis is controlled by one of the function blocks: • <i>MC_Halt_PTO_NTS</i>, page 75 • <i>MC_MoveAbsolute_PTO_NTS</i>, page 81 • <i>MC_MoveRelative_PTO_NTS</i>, page 85
<i>ContinuousMotion</i>	BOOL	The <i>ContinuousMotion</i> state is applicable when the axis is controlled by the function block <i>MC_MoveVelocity_PTO_NTS</i> , page 89.
<i>IsHomed</i>	BOOL	When TRUE, the reference point is valid. Absolute motion is allowed.
<i>AxisWarning</i>	BOOL	When TRUE, an error is detected on the axis (refer to <i>MC_ReadAxisError_PTO_NTS</i> , page 48 for further information).
<i>QueueFull</i>	BOOL	When TRUE, the motion queue is full. No additional buffered motion is allowed.

MC_Reset_PTO_NTS: Resets Axis-Related Errors

Function Block Description

The *MC_Reset_PTO_NTS* function block resets the detected axis-related errors. If conditions are permitting, it allows a transition from the *ErrorStop* state to *Standstill*. It updates the *AxisState*. If the function block is executed when there is no detected error, it does not affect motion commands that are in progress.

The *MC_Reset_PTO_NTS* function block executes the *Reset* administrative command. For further information, refer to the table that describes PTO administrative commands in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.

This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the reset is finished. Function block execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

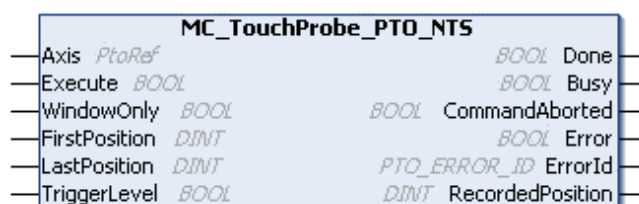
MC_TouchProbe_PTO_NTS: Activates a Trigger Event

Function Block Description

The *MC_TouchProbe_PTO_NTS* function block is used to activate a trigger event on the probe input. This trigger event allows to record the axis position, and/or to start a buffered move.

The *MC_TouchProbe_PTO_NTS* function block executes the *TouchProbe* administrative command. For further information, refer to the table that describes PTO administrative commands in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution.
<i>WindowOnly</i>	BOOL	When TRUE, trigger events are only accepted in the range configured with the inputs <i>FirstPosition</i> and <i>LastPosition</i> .
<i>FirstPosition</i>	DINT	The start position of the range in which trigger events are accepted.
<i>LastPosition</i>	DINT	The end position of the range in which trigger events are accepted.
<i>TriggerLevel</i>	BOOL	When FALSE, the trigger event occurs on a falling edge of the PROBE input. When TRUE, the trigger event occurs on a rising edge of the PROBE input. Default value: FALSE

This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the trigger event on the PROBE input is detected and the TouchProbe function block is finished. When a rising edge is detected, the output <i>RecordedPosition</i> is updated, and the motion command, sent with the <i>seTrigger</i> buffer mode, page 111, is executed.
<i>Busy</i>	BOOL	TRUE indicates that the function block is active and waiting for the trigger event on the PROBE input to be detected.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is aborted by the function block <i>MC_AbortTrigger_PTO_NTS</i> , page 46 or by a detected error. The TouchProbe execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>RecordedPosition</i>	DINT	The position where the trigger event is detected.

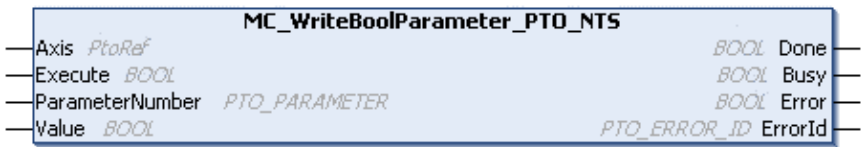
NOTE: Once the function block is enabled by the *Execute* input, the *Busy* output is TRUE and remains TRUE until the first valid event. At that point, the *Busy* output is set to FALSE and the *Done* output is set to TRUE and remains so until another rising edge is detected at the *Execute* input. Subsequent events are ignored until the function block is re-enabled.

MC_WriteBoolParameter_PTO_NTS: Writes BOOL Parameters to the PTO

Function Block Description

The *MC_WriteBoolParameter_PTO_NTS* function block modifies the value of a Boolean parameter.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>ParameterNumber</i>	<i>PTO_PARAMETER</i> , page 116	ID of the parameter requested from <i>PTO_PARAMETER</i> , page 116.
<i>Value</i>	BOOL	Value to be written to the requested Boolean parameter.

This table describes the output variables:

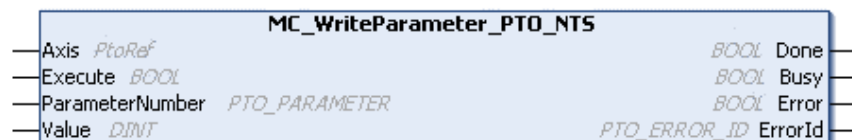
Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the Boolean parameter was written.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

MC_WriteParameter_PTO_NTS: Writes Parameters to the PTO

Function Block Description

The *MC_WriteParameter_PTO_NTS* function block modifies the value of a parameter, except of a Boolean data type.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>ParameterNumber</i>	PTO_PARAMETER, page 116	ID of the parameter requested from PTO_PARAMETER, page 116.
<i>Value</i>	DINT	Value to be written to the requested parameter.

This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the parameter was written.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	PTO_ERROR_ID, page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

PTO Motion Function Blocks

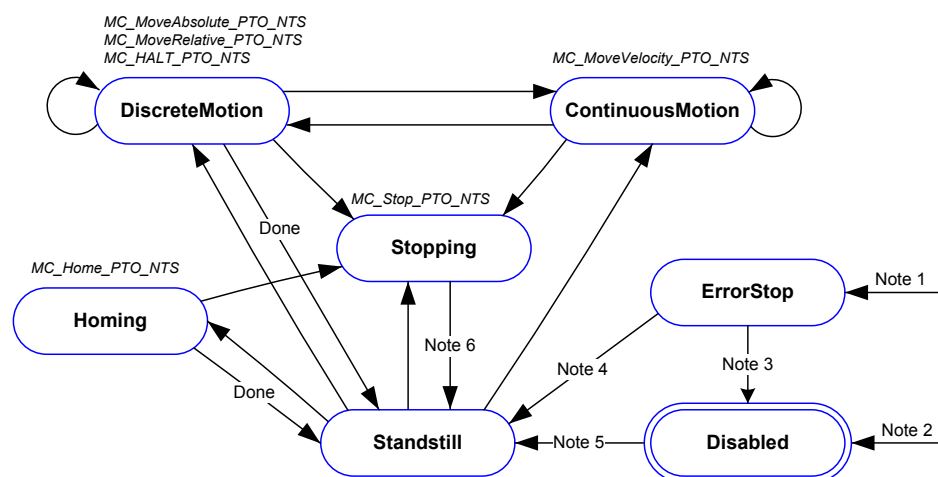
What's in This Chapter

Operating Modes.....	59
Timing Diagram Examples	61
PTO Homing Modes	65
PTO Configuration.....	74
<i>MC_HALT_PTO_NTS</i> : Commands a Controlled Motion Stop until the Velocity equals Zero	75
<i>MC_Home_PTO_NTS</i> : Commands the Axis to Move to a Reference Position	78
<i>MC_MoveAbsolute_PTO_NTS</i> : Commands Motion of a Distance	81
<i>MC_MoveRelative_PTO_NTS</i> : Commands Relative Axis Motion	85
<i>MC_MoveVelocity_PTO_NTS</i> : Controls the Velocity of the Axis	89
<i>MC_Power_PTO_NTS</i> : Manages the Power of the Axis	94
<i>MC_SetPosition_PTO_NTS</i> : Modifies the Reference Position of the Axis.....	96
<i>MC_Stop_PTO_NTS</i> : Commands a Controlled Motion Stop	97

Operating Modes

Motion State Diagram

The axis is in one of the states defined in this diagram:



Note 1 From any state, when an error is detected.

Note 2 From any state except *ErrorStop*, when *MC_Power_PTO_NTS.Status* equals FALSE.

Note 3 *MC_Reset_PTO.Done* equals TRUE and *MC_Power_PTO_NTS.Status* equals FALSE.

Note 4 *MC_Reset_PTO.Done* equals TRUE and *MC_Power_PTO_NTS.Status* equals TRUE.

Note 5 *MC_Power_PTO_NTS.Status* equals TRUE.

Note 6 *MC_Stop_PTO.Done* equals TRUE and *MC_Stop_PTO.Execute* equals FALSE.

The table describes the axis states:

State	Description
Disabled	Initial state of the axis, no motion command is allowed. The axis is not homed.
Standstill	Power is on, there is no error detected, and there are no motion commands active on the axis. A motion command is allowed.
ErrorStop	Highest priority, applicable when an error is detected on the axis or in the controller. Ongoing motion is aborted by an implicit stop command using the Fast Stop Deceleration parameter value defined in the configuration. For further information, refer to the <i>PTO Configuration</i> chapter of the Modicon Edge I/O NTS Motion Expert Modules User Guide. The <i>Error</i> output is set to TRUE on applicable function blocks, and the <i>ErrorId</i> output indicates the identification number of the detected error. No further motion command is accepted until a reset has been performed using <i>MC_Reset_PTO_NTS</i> .
Homing	Applicable when <i>MC_Home_PTO_NTS</i> controls the axis.
DiscreteMotion	Applicable when <i>MC_MoveRelative_PTO_NTS</i> , <i>MC_MoveAbsolute_PTO_NTS</i> , or <i>MC_Halt_PTO_NTS</i> controls the axis.
ContinuousMotion	Applicable when <i>MC_MoveVelocity_PTO_NTS</i> controls the axis.
Stopping	Applicable when <i>MC_Stop_PTO_NTS</i> controls the axis.

NOTE: Function blocks that are not listed in the state diagram do not affect a change of state of the axis.

For further information, refer to the description of the Driven_Operational state in the *Pulse Train Output (PTO) Principle Description* chapter of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

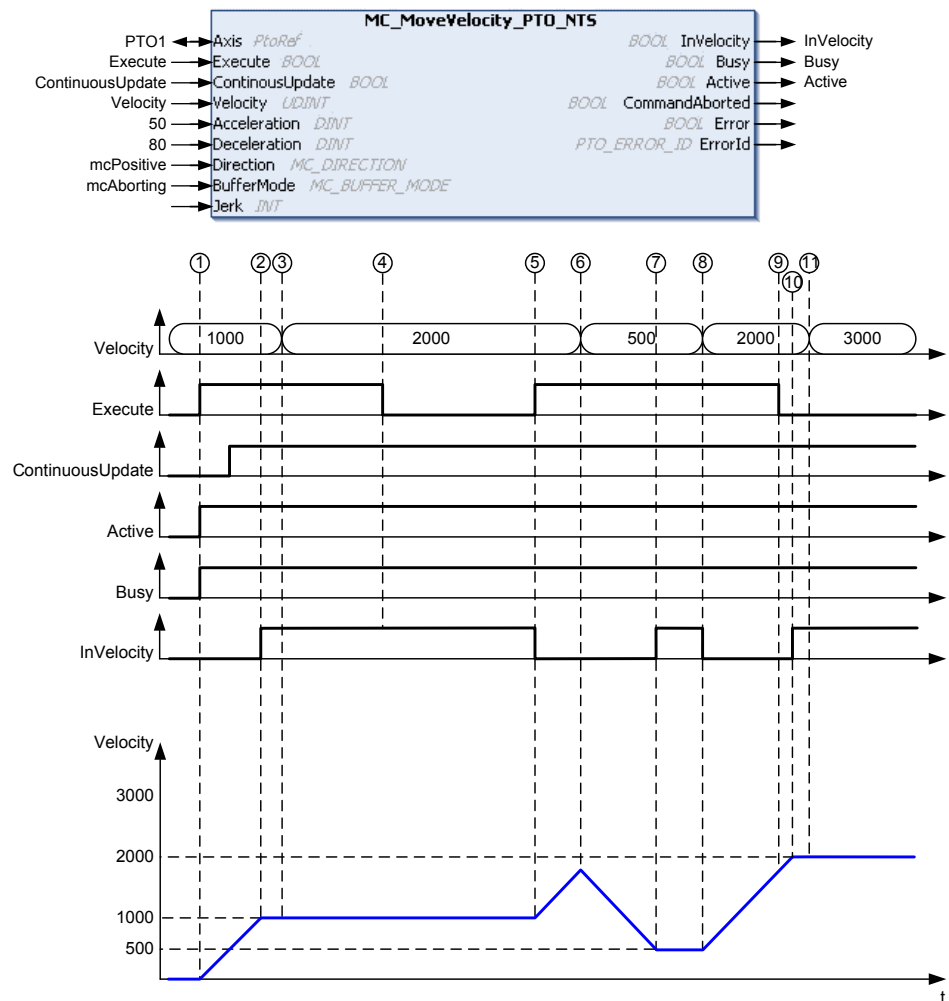
Motion Command Abort Table

A new motion command can abort an ongoing motion command as indicated in the following table:

Motion Command		New					
		<i>Home</i>	<i>Velocity</i>	<i>Relative</i>	<i>Absolute</i>	<i>Halt</i>	<i>Stop</i>
Ongoing	<i>Home</i>	–	–	–	–	–	✓
	<i>Velocity</i>	–	✓	✓	✓	✓	✓
	<i>Relative</i>	–	✓	✓	✓	✓	✓
	<i>Absolute</i>	–	✓	✓	✓	✓	✓
	<i>Halt</i>	–	✓	✓	✓	✓	✓
	<i>Stop</i>	–	–	–	–	–	–
✓ The ongoing motion command is aborted and the new motion command begins execution in accordance with the <i>BufferMode</i> parameter value. For further information, refer to <i>BufferMode Parameter Description</i> in the <i>Pulse Train Output (PTO) Principle Description</i> chapter of the Modicon Edge I/O NTS Motion Expert Modules User Guide. – The new motion command is rejected and results in a detected error.							

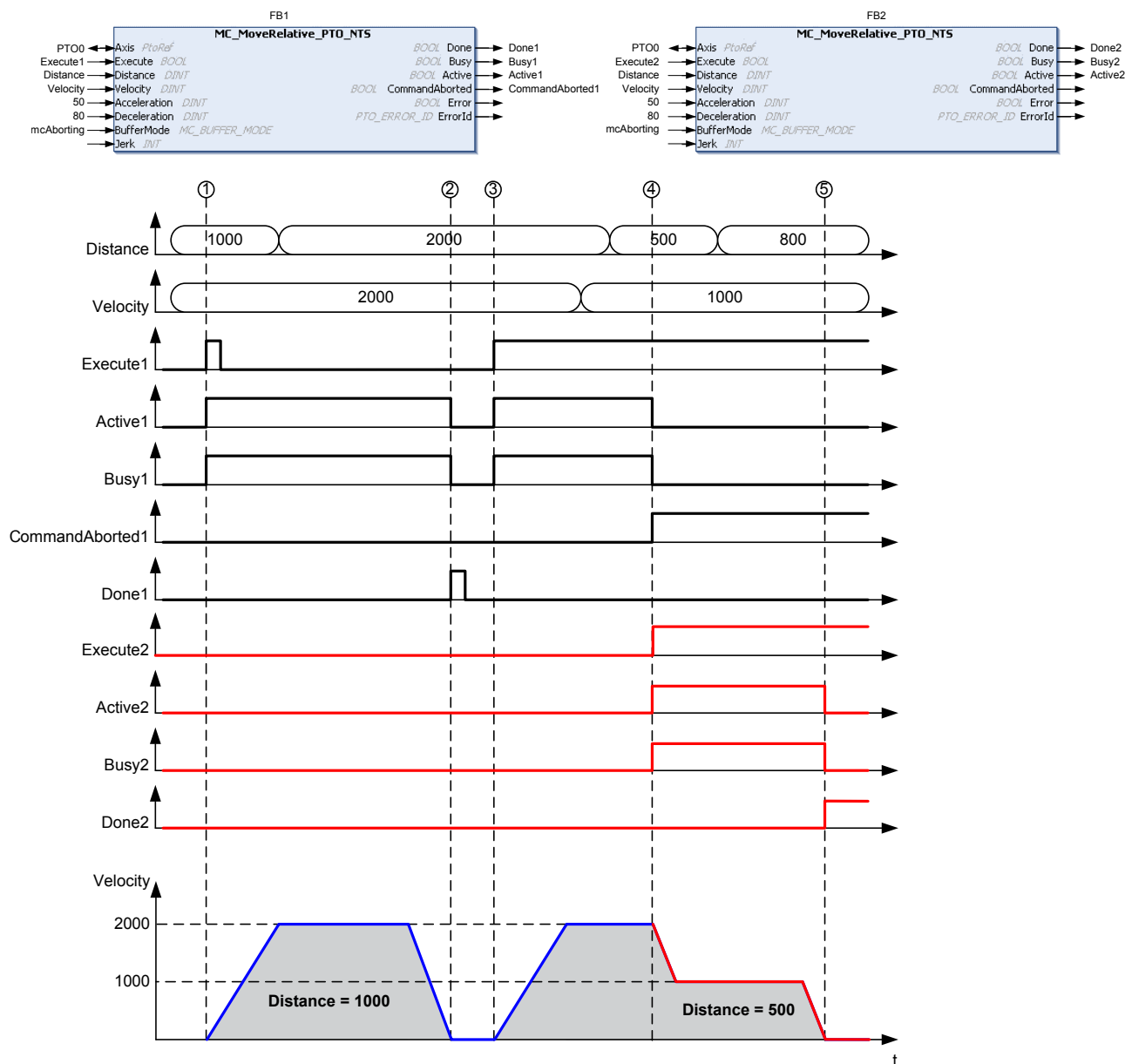
Timing Diagram Examples

Move Velocity to Move Velocity with mcAborting



- 1** *Execute* rising edge: command parameters are latched, movement is started with target velocity 1000.
- 2** Target velocity 1000 is reached.
- 3** *Velocity* parameter is changed to 2000: not applied at this point (no rising edge on *Execute* input, and *ContinuousUpdate* was latched with value 0 at the start of the movement).
- 4** *Execute* falling edge: status bits are maintained (movement is still active).
- 5** *Execute* rising edge: command parameters are latched, movement is started with target velocity 2000 and *ContinuousUpdate* active.
- 6** *Velocity* parameter is changed to 500: applied (*ContinuousUpdate* is TRUE). The previous target velocity 2000 was not reached.
- 7** Target velocity 500 is reached.
- 8** *Velocity* parameter is changed to 2000: applied (*ContinuousUpdate* is TRUE).
- 9** *Execute* falling edge: status bits are maintained (movement is still active).
- 10** Target velocity 2000 is reached, *InVelocity* is set (*Execute* pin is FALSE).
- 11** *Velocity* parameter is changed to 3000: not applied (movement is still active).

Move Relative to Move Relative with mcAborting



1 FB1 *Execute* rising edge: command parameters are latched, movement is started with target *velocity* 2000 and *distance* 1000.

2 Movement ends: distance traveled is 1000.

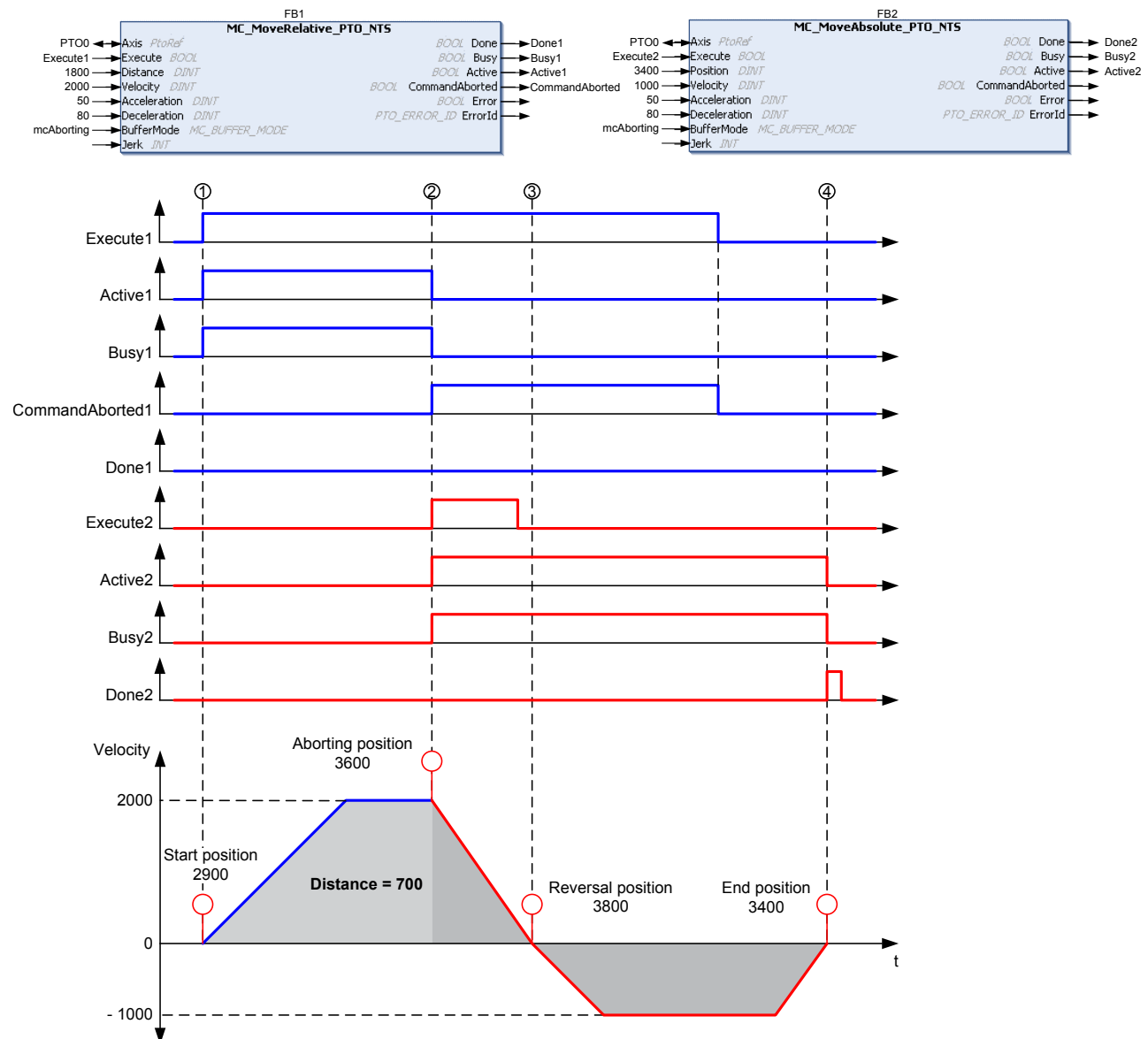
3 FB1 *Execute* rising edge: command parameters are latched, movement is started with target *velocity* 2000 and *distance* 2000.

4 FB2 *Execute* rising edge: command parameters are latched, movement is started with target *velocity* 1000 and *distance* 500.

NOTE: FB1 is aborted.

5 Movement ends.

Move Relative to Move Absolute with mcAborting



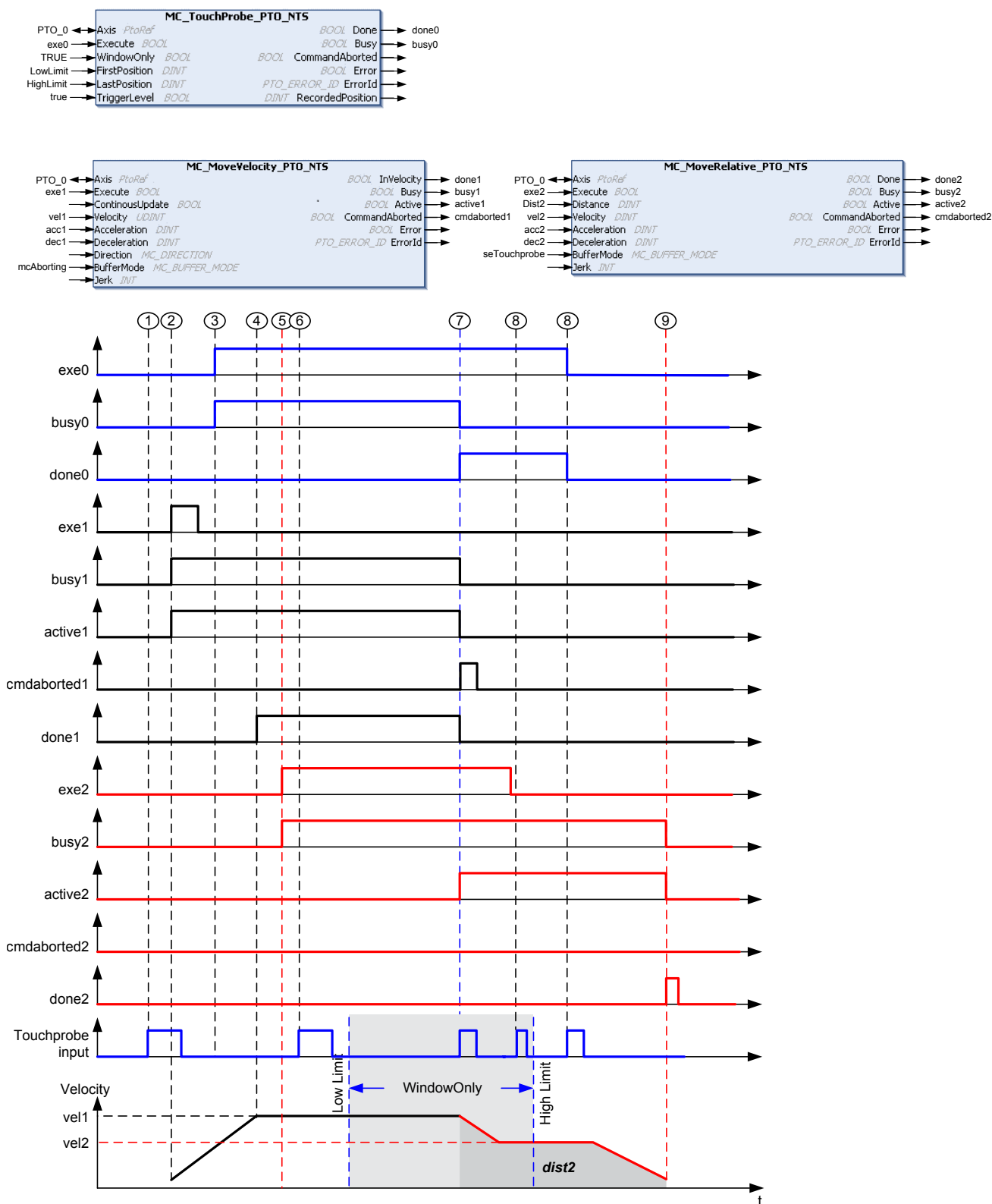
1 FB1 *Execute* rising edge: command parameters are latched, movement is started with target *velocity* 2000 and *distance* 1800.

2 FB2 *Execute* rising edge: command parameters are latched, FB1 is aborted, and movement continues with target *velocity* 1000 and target *position* 3400. Automatic direction management: direction reversal is needed to reach target position, move to stop at *deceleration* of FB2.

3 Velocity 0, direction reversal, movement resumes with target *velocity* 1000 and target *position* 3400.

4 Movement ends: target position 3400 is reached.

Move Velocity to Move Relative with seTrigger



1 **MC_TouchProbe_PTO_NTS** not busy at this point even if the probe input is active, but is not monitored because the *Busy* signal `busy0` is not active.

2 **MC_MoveVelocity_PTO_NTS** *Execute* rising edge: command parameters are latched, movement is started with target *velocity* `vel1`.

3 **MC_TouchProbe_PTO_NTS** *Execute* rising edge: The **MC_TouchProbe_PTO_NTS** function block is active, monitoring for a trigger event in the specified window.

4 `vel1` is reached.

5 *MC_MoveRelative_PTO_NTS Execute* rising edge: command parameters are latched, waiting for probe event to start.

6 Probe event outside of enable window: event is ignored.

7 A valid event is detected. *MC_MoveRelative_PTO_NTS* aborts *MC_MoveVelocity_PTO_NTS*. The *MC_TouchProbe_PTO_NTS* function block is done, the monitoring for a trigger event is stopped.

8 Subsequent events are ignored.

9 Movement ends.

PTO Homing Modes

This chapter describes the PTO homing modes.

Homing Modes

Description

Homing is the method to establish the reference point or origin for absolute movement.

A homing movement is performed using the following homing movement types:

- Position Setting, page 66
- Long Reference, page 67,
- Long Reference & Index, page 68,
- Short Reference Reversal, page 69,
- Short Reference No Reversal, page 70,
- Short Reference & Index Outside, page 71,
- Short Reference & Index Inside, page 72.

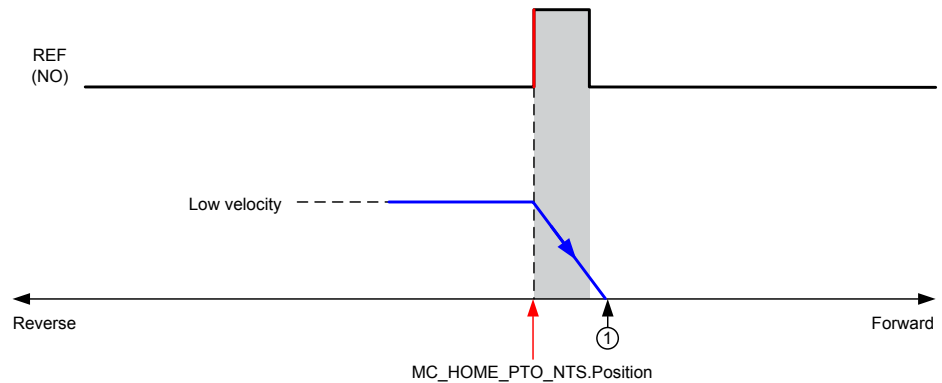
A homing movement must be completed without interruption for the new reference point to be valid. If the reference movement is interrupted, it needs to be started again.

Also refer to *MC_Home_PTO_NTS*, page 78 and *PTO_HOMING_MODE*, page 115.

Home Position

Homing is done with an external switch (REF input), and the homing position is defined on the switch edge. Then the motion is decelerated until stopped.

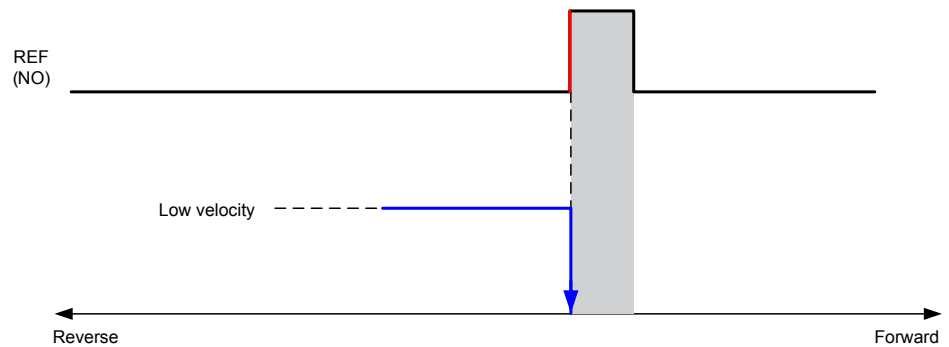
The position of the axis at the end of the motion sequence may therefore differ from the position parameter set on the function block:



REF (NO) Reference point (Normally Open)

1 Position at the end of motion = *MC_HOME_PTO_NTS.Position* + “deceleration to stop” distance.

To simplify the representation of a stop in the homing mode diagrams, the following presentation is made to represent the position of the axis:



REF (NO): REF input (Normally Open)

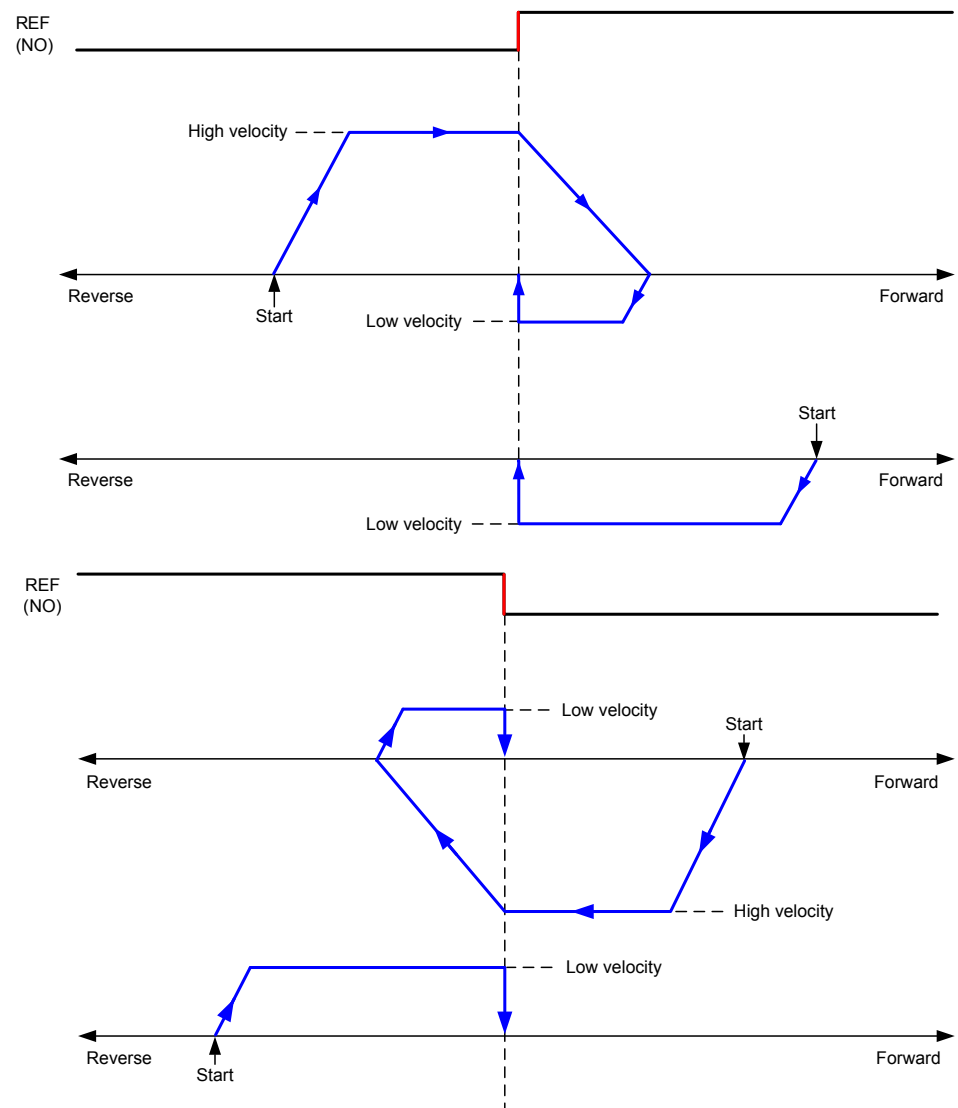
Position Setting

Description

In the case of position setting, the present position is set to the specified position value. No move is performed.

Long Reference

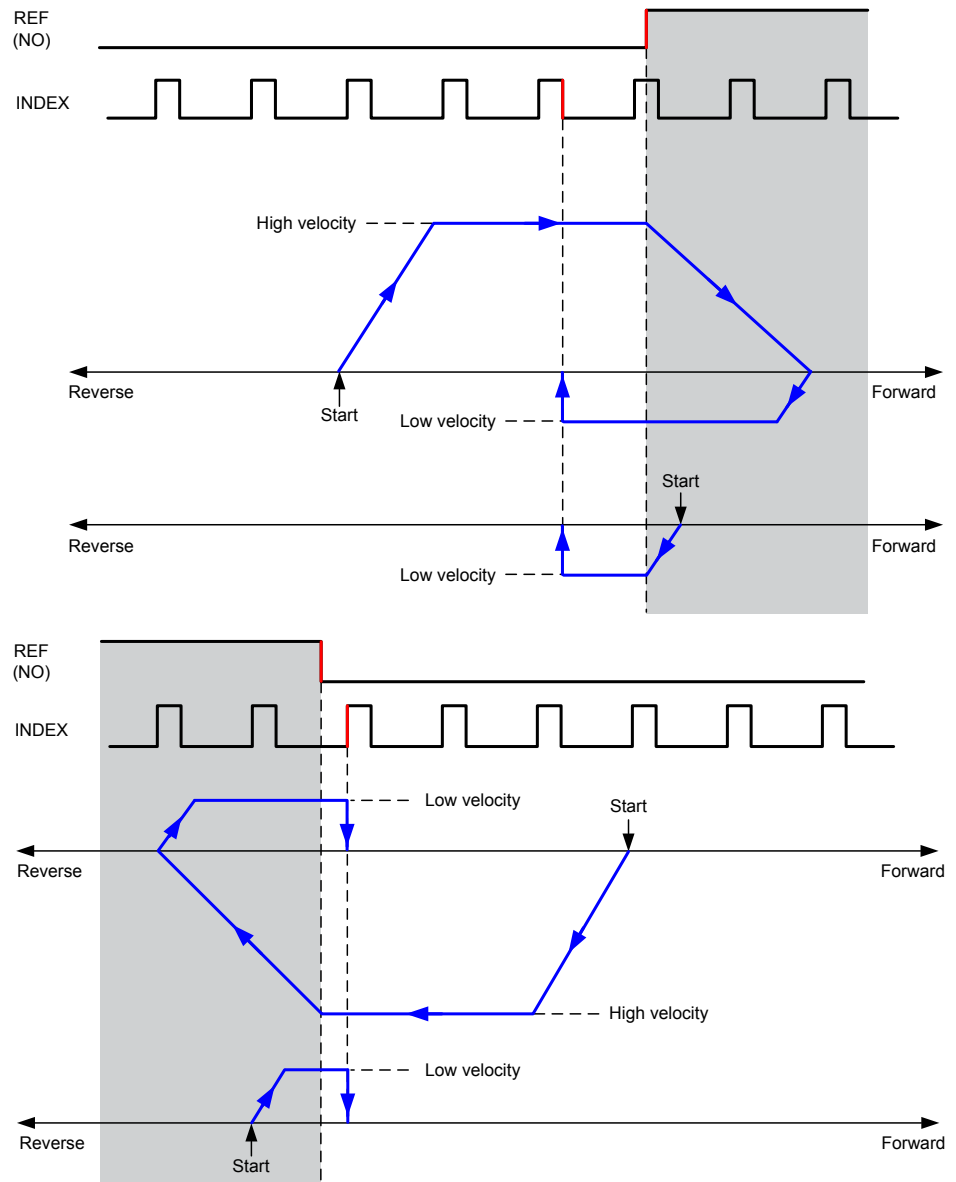
The homing profile depends on the REF input signal:



REF (NO): REF input (Normally Open)

Long Reference & Index

The homing profile depends on the REF input and the INDEX input signal:

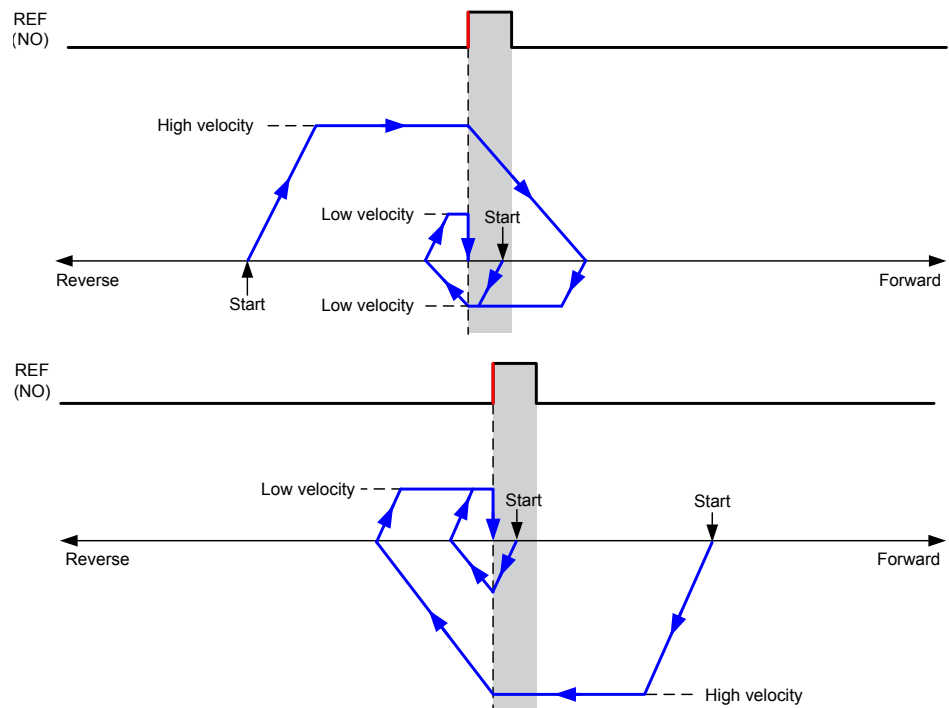


REF (NO): REF input (Normally Open)

INDEX: INDEX input

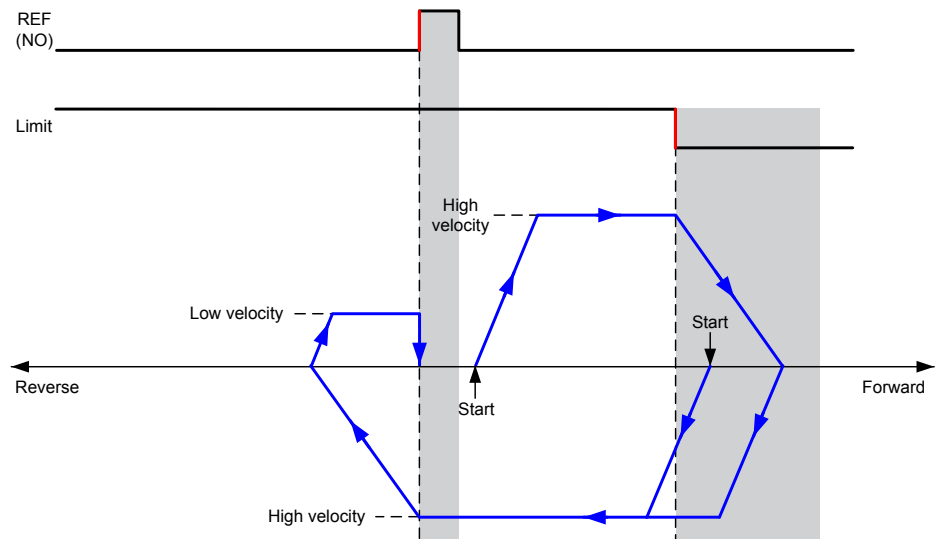
Short Reference Reversal

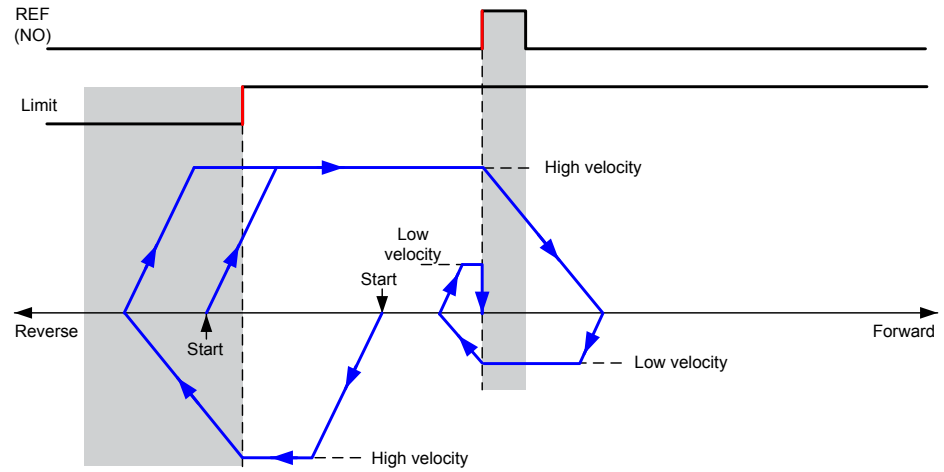
The homing profile depends on the REF input signal:



REF (NO) Reference point (Normally Open)

If limits (hardware or software) are configured, the movement reverses to continue searching for the REF input signal:

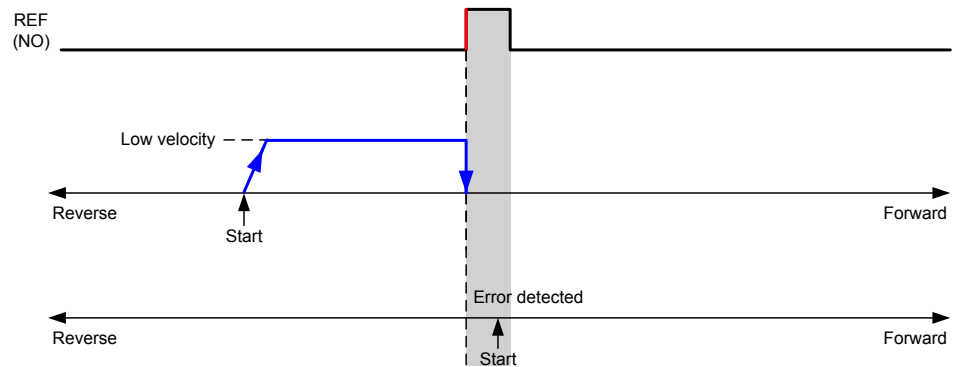




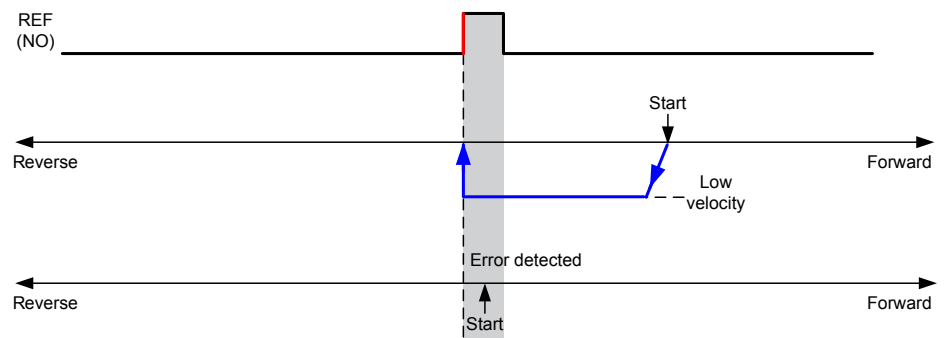
REF (NO): REF input (Normally Open)

Short Reference No Reversal

The homing profile searches the REF input at *LowVelocity* with no reversal:



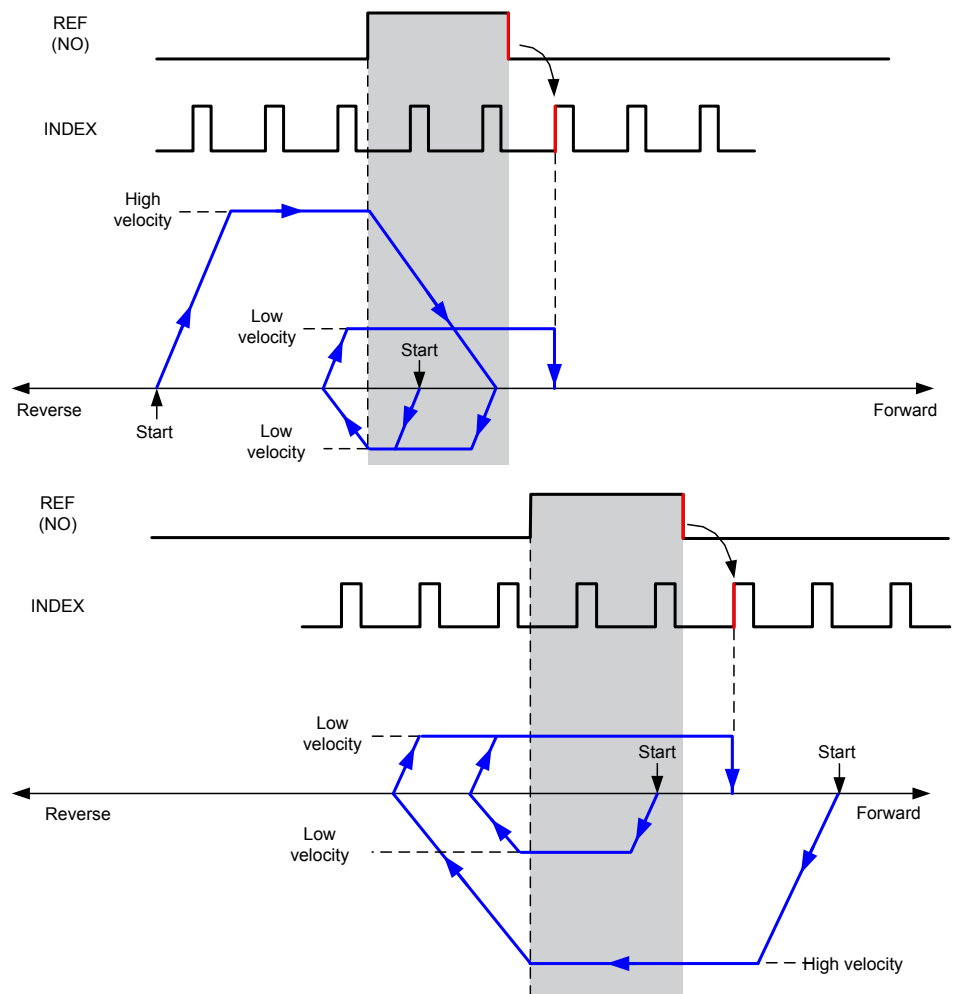
REF (NO): REF input (Normally Open)



REF (NO): REF input (Normally Open)

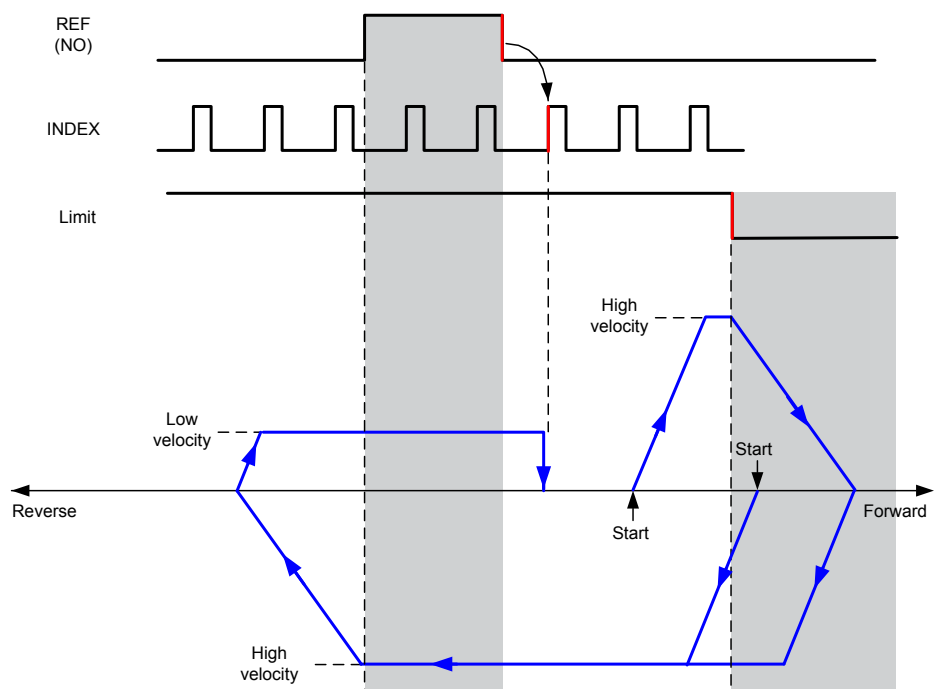
Short Reference & Index Outside

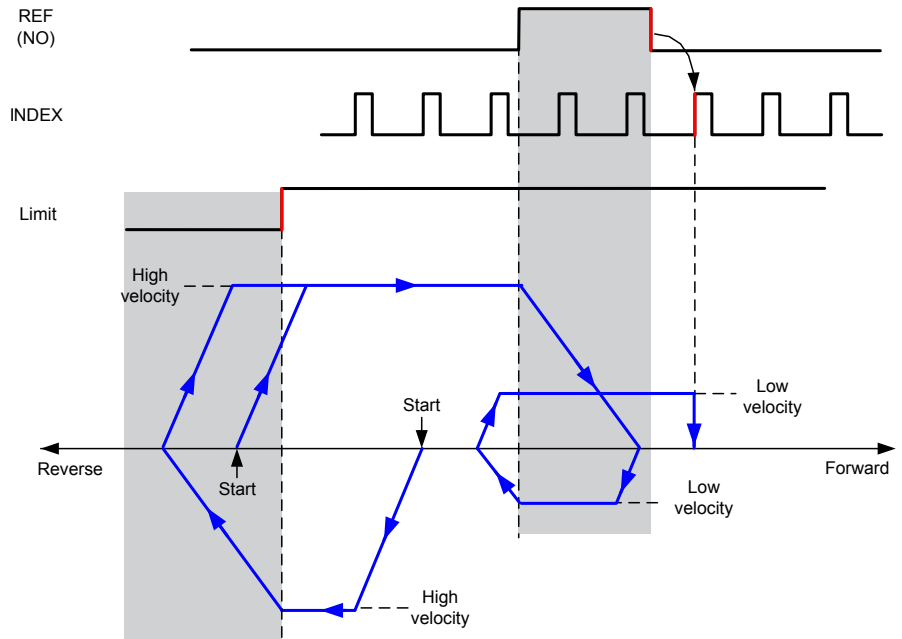
The homing profile depends on the REF input and the INDEX input signal:



REF (NO): REF input (Normally Open)

If limits (hardware or software) are configured, the movement reverses to continue searching for the REF input signal:

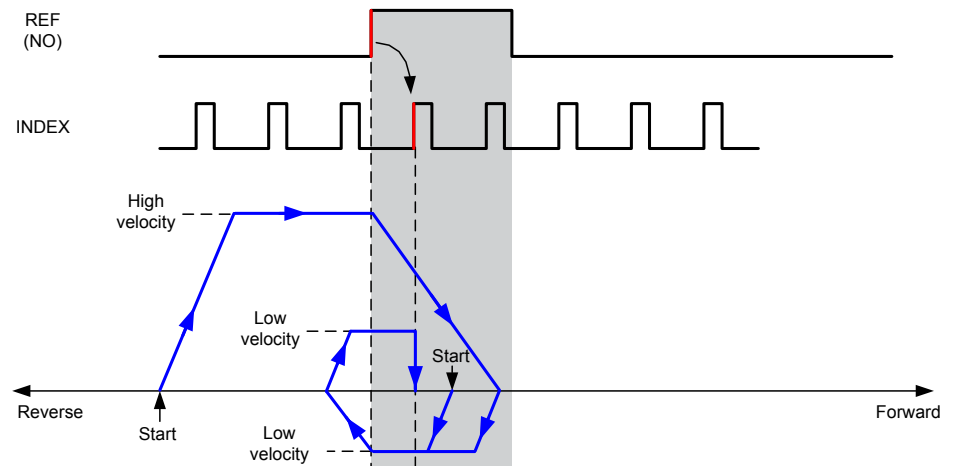


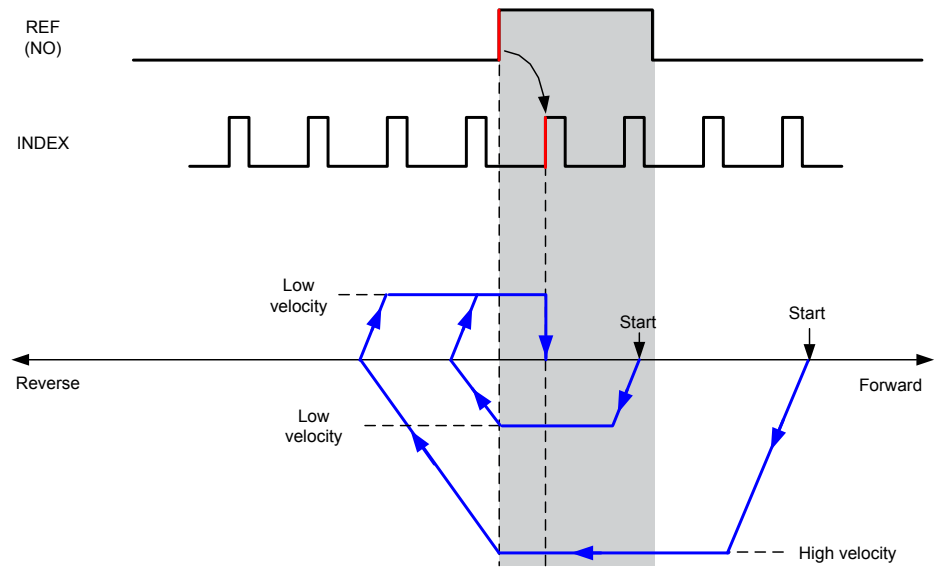


REF (NO): REF input (Normally Open)

Short Reference & Index Inside

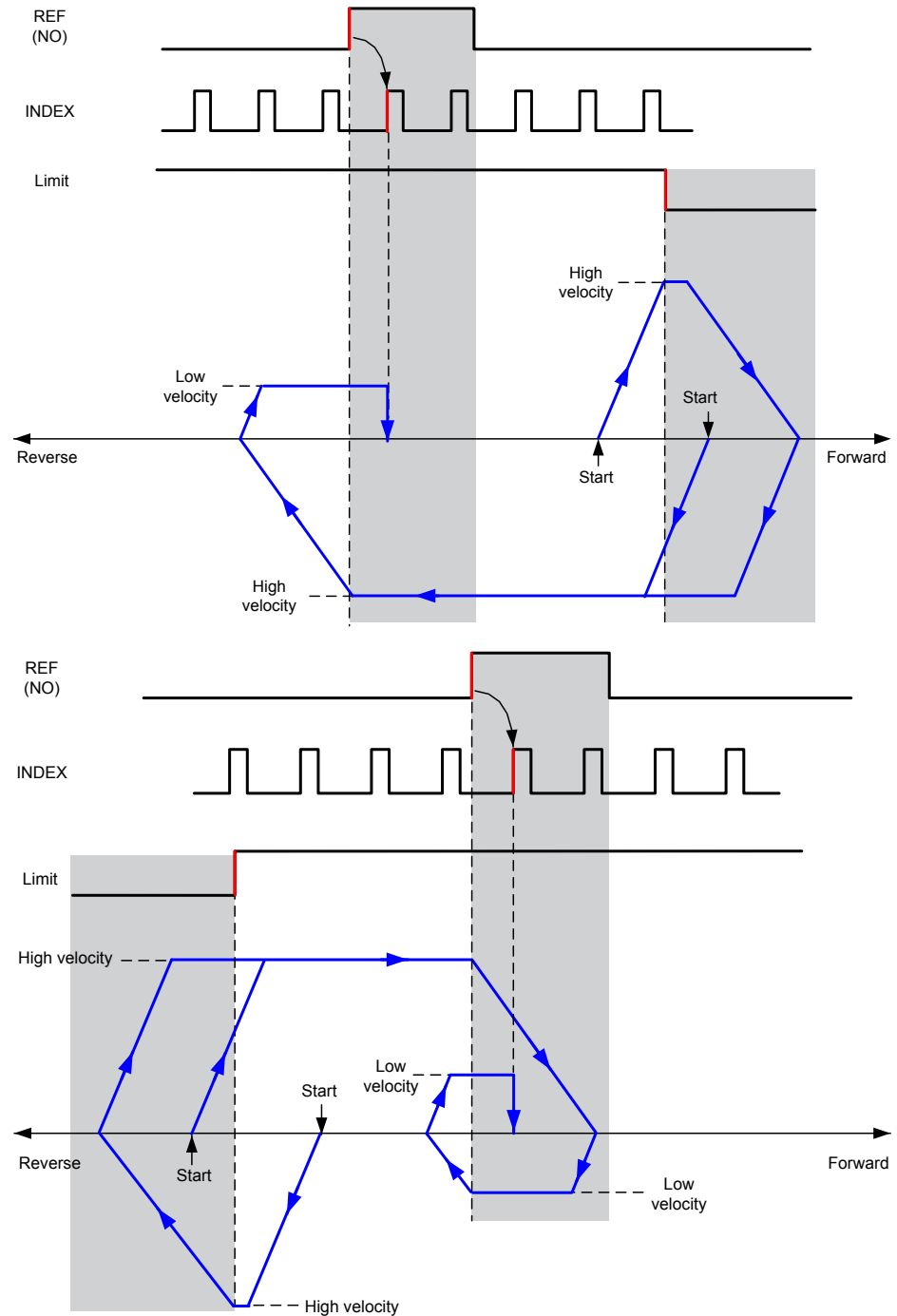
The homing profile depends on the REF input and the INDEX input signal:





REF (NO): REF input (Normally Open)

If limits (hardware or software) are configured, the movement reverses to continue searching for the REF input signal:



REF (NO): REF input (Normally Open)

PTO Configuration

For information on PTO configuration in the **Edge I/O NTS Editor** of the Logic Builder, refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

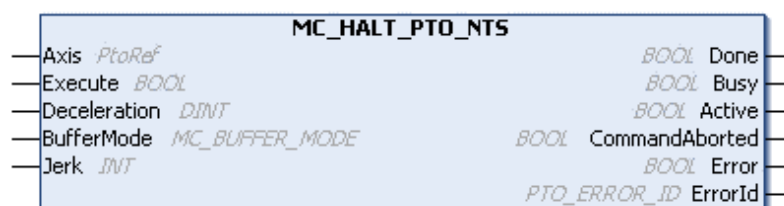
MC_HALT_PTO_NTS: Commands a Controlled Motion Stop until the Velocity equals Zero

Function Block Description

The *MC_HALT_PTO_NTS* function block commands a controlled motion stop and the axis is set to the state *DiscreteMotion* until the velocity is zero. With the *Done* output set to TRUE, the state is set to *Standstill*.

The *MC_HALT_PTO_NTS* function block executes the *Halt* motion command. For further information, refer to the description of the Halt motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block sends the motion command. When a falling edge is detected, the outputs of the function block are reset. When <i>Execute</i> is TRUE, the outputs of the function block are updated. When <i>Execute</i> is FALSE, the outputs of the function block are not updated. NOTE: Setting <i>Execute</i> to FALSE does not cancel the motion command sent on a rising edge.
<i>Deceleration</i>	DINT	Deceleration in Hz/ms or in ms (according to the configuration). Value range (Hz/ms): 1...Maximum Deceleration Value range (ms): Maximum Deceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>BufferMode</i>	MC_BUFFER_MODE, page 111	When TRUE, transition is performed from an ongoing motion. Refer to MC_BUFFER_MODE, page 111.
<i>Jerk</i>	INT	Sets the percentage used to create an S-curve profile. Value range: 0...65,535* Default value: 0 *If the provided value is above 100, the <i>Jerk</i> is set to 100 and an advisory is issued. If the <i>Jerk</i> parameter is set to 100%, then the acceleration and deceleration values are double the value defined in the acceleration and deceleration parameters. For example, if <i>Jerk</i> is set to 100%, then the acceleration and deceleration values equal to the value defined in the <i>Acceleration</i> and <i>Deceleration</i> parameters multiplied by 2. The duration for the acceleration and deceleration is maintained regardless of the <i>Jerk</i> parameter value. To maintain this duration, the acceleration or deceleration provided to the motion command is adjusted accordingly. NOTE: If the new calculated acceleration or deceleration exceeds the Maximum Acceleration or Maximum Deceleration parameter value, an advisory is issued and <i>Jerk</i> is changed to the closest possible <i>Jerk</i> value. For further information about acceleration and deceleration ramps, refer to the chapter <i>Pulse Train Output (PTO) Principle Description</i> of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

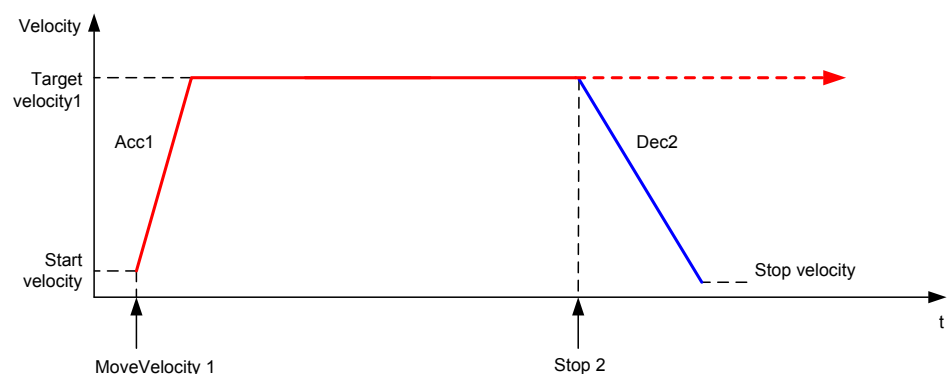
This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the Halt is finished. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Active</i>	BOOL	When TRUE, the function block controls the motion of the axis. Only one function block at a time can control one axis.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is cancelled due to another move command or a detected error. The motion execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

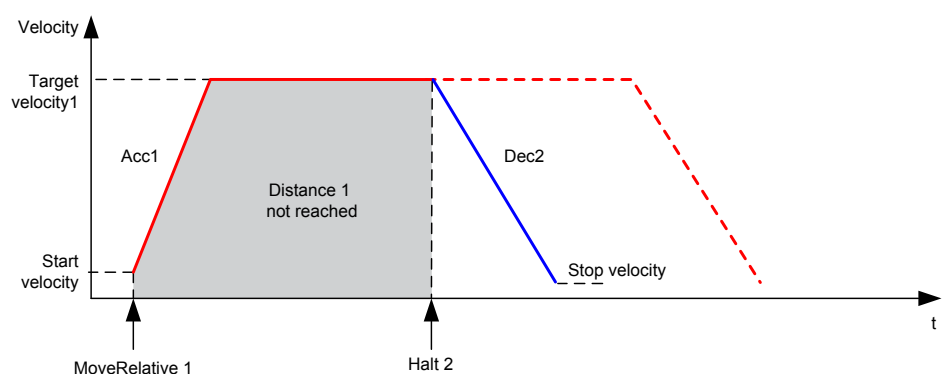
NOTE: The function block completes with velocity zero.

Timing Diagram Example

The following diagram shows the profile of the motion command when axis is in *ContinuousMotion* state. For further information, refer to the *Motion State Diagram*, page 59.



The following diagram shows the profile of the motion command when axis is in *DiscreteMotion* state. For further information, refer to the *Motion State Diagram*, page 59.



NOTE: In this diagram, *BufferMode* of the *Absolute* motion command is set to *mcAborting* (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

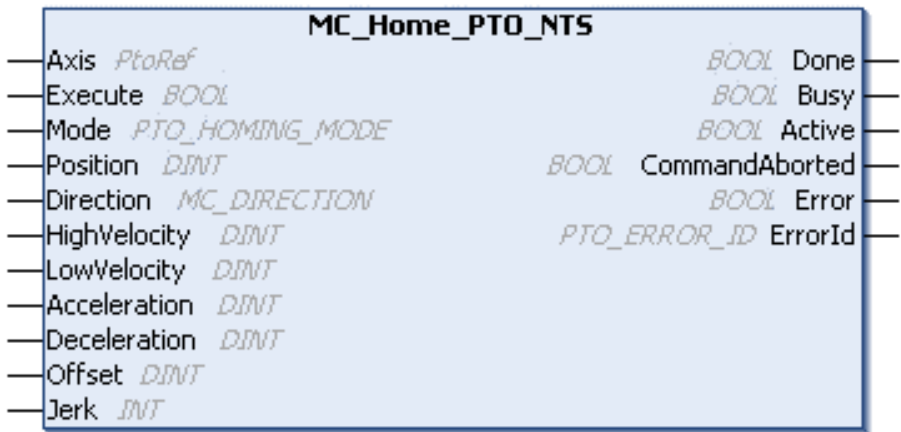
MC_Home_PTO_NTS: Commands the Axis to Move to a Reference Position

Function Block Description

The *MC_Home_PTO_NTS* function block commands the axis to move to the reference position, and the axis is set to the state *Homing*. The details of this sequence depend on the *Homing* mode defined. The **REF Enabled** input parameter and the **INDEX Enabled** input parameter must be defined in the **Edge I/O NTS Editor** of the Logic Builder. For further information, (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

The *MC_Home_PTO_NTS* function block executes the *Homing* motion command. For further information, refer to the description of the Homing motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Inputs	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block sends the motion command. When a falling edge is detected, the outputs of the function block are reset. When <i>Execute</i> is TRUE, the outputs of the function block are updated. When <i>Execute</i> is FALSE, the outputs of the function block are not updated. NOTE: Setting <i>Execute</i> to FALSE does not cancel the motion command sent on a rising edge.
<i>Mode</i>	<i>PTO_HOMING_MODE</i> , page 115	The predefined type of homing mode. Refer to <i>PTO_HOMING_MODE</i>, page 115. Default value: <i>PositionSetting</i>
<i>Position</i>	DINT	Sets the position when the origin event is detected. Default value: 0
<i>Direction</i>	<i>MC_DIRECTION</i> , page 112	The starting direction that is exclusive to Homing. The values <i>mcPositiveDirection</i> and <i>mcNegativeDirection</i> are valid. Default value: <i>mcNegativeDirection</i>
<i>HighVelocity</i>	DINT	The target homing velocity for searching the limit or reference switch. Value range (in Hz): 1...Maximum Velocity Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>LowVelocity</i>	DINT	The target homing velocity for searching the reference switch or index signal. The motion stops when the switching point is detected. Value range (in Hz): 1...<i>HighVelocity</i> Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Acceleration</i>	DINT	Acceleration in Hz/ms or in ms (according to the configuration). Value range (in Hz/ms): 1...Maximum Acceleration Value range (in ms): Maximum Acceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Deceleration</i>	DINT	Deceleration in Hz/ms or in ms (according to the configuration). Value range (Hz/ms): 1...Maximum Deceleration Value range (ms): Maximum Deceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.

Inputs	Data type	Description
<i>Offset</i>	DINT	Sets the number of pulses to move at the end of the Homing motion command due to mechanical reasons. For further information, refer to Homing Modes, page 65. The sign defines the direction of the move and <i>LowVelocity</i> is the velocity at which the move is executed. The parameter is used during a reference movement without index pulse. NOTE: The offset movement is part of the Homing motion command and is executed 500 ms after <i>CurrentVelocity</i> reaches 0. Value range: -2,147,483,648...2,147,483,647 Default value: 0
<i>Jerk</i>	INT	Sets the percentage used to create an S-curve profile. Value range: 0...65,535* Default value: 0 *If the provided value is above 100, the <i>Jerk</i> is set to 100 and an advisory is issued. If the <i>Jerk</i> parameter is set to 100%, then the acceleration and deceleration values are double the value defined in the acceleration and deceleration parameters. For example, if <i>Jerk</i> is set to 100%, then the acceleration and deceleration values equal to the value defined in the <i>Acceleration</i> and <i>Deceleration</i> parameters multiplied by 2. The duration for the acceleration and deceleration is maintained regardless of the <i>Jerk</i> parameter value. To maintain this duration, the acceleration or deceleration provided to the motion command is adjusted accordingly. NOTE: If the new calculated acceleration or deceleration exceeds the Maximum Acceleration or Maximum Deceleration parameter value, an advisory is issued and <i>Jerk</i> is changed to the closest possible <i>Jerk</i> value. For further information about acceleration and deceleration ramps, refer to the chapter <i>Pulse Train Output (PTO) Principle Description</i> of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the homing is finished. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Active</i>	BOOL	When TRUE, the function block controls the motion of the axis. Only one function block at a time can control one axis.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is cancelled due to another move command or a detected error. The execution of the motion is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

Timing Diagram Example

For further information, refer to [Homing Modes](#), page 65.

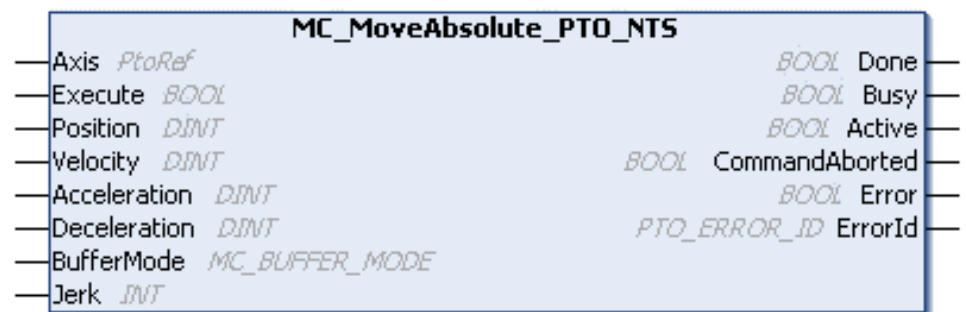
MC_MoveAbsolute_PTO_NTS: Commands Motion of a Distance

Function Block Description

The *MC_MoveAbsolute_PTO_NTS* function block causes the specified axis to move towards a given position at the specified speed, and the axis is set to the state *DiscreteMotion*. To use the *MC_MoveAbsolute_PTO_NTS* function block, *IsHomed* must be TRUE. Otherwise, the function block will terminate in error (*Error* set to 1 and *ErrorId* set to *InvalidAbsolute*).

The *MC_MoveAbsolute_PTO_NTS* function block executes the *Absolute* motion command. For further information, refer to the description of the Absolute motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block sends the motion command. When a falling edge is detected, the outputs of the function block are reset. When <i>Execute</i> is TRUE, the outputs of the function block are updated. When <i>Execute</i> is FALSE, the outputs of the function block are not updated. NOTE: Setting <i>Execute</i> to FALSE does not cancel the motion command sent on a rising edge.
<i>Position</i>	DINT	The target position. Default value: 0
<i>Velocity</i>	DINT	The target velocity in Hz. Value range: 1...<i>MaxVelocityAppl</i>, page 116 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Acceleration</i>	DINT	Acceleration in Hz/ms or in ms (according to the configuration). Value range (in Hz/ms): 1...Maximum Acceleration Value range (in ms): Maximum Acceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Deceleration</i>	DINT	Deceleration in Hz/ms or in ms (according to the configuration). Value range (Hz/ms): 1...Maximum Deceleration Value range (ms): Maximum Deceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.

Input	Data type	Description
<i>BufferMode</i>	<i>MC_BUFFER_MODE</i> , page 111	Transition is performed from an ongoing motion. Refer to <i>MC_BUFFER_MODE</i> , page 111. Default value: <i>mcAborting</i>
<i>Jerk</i>	INT	Sets the percentage used to create an S-curve profile. Value range: 0...65,535* Default value: 0 *If the provided value is above 100, the <i>Jerk</i> is set to 100 and an advisory is issued. If the <i>Jerk</i> parameter is set to 100%, then the acceleration and deceleration values are double the value defined in the acceleration and deceleration parameters. For example, if <i>Jerk</i> is set to 100%, then the acceleration and deceleration values equal to the value defined in the <i>Acceleration</i> and <i>Deceleration</i> parameters multiplied by 2. The duration for the acceleration and deceleration is maintained regardless of the <i>Jerk</i> parameter value. To maintain this duration, the acceleration or deceleration provided to the motion command is adjusted accordingly. NOTE: If the new calculated acceleration or deceleration exceeds the Maximum Acceleration or Maximum Deceleration parameter value, an advisory is issued and <i>Jerk</i> is changed to the closest possible <i>Jerk</i> value. For further information about acceleration and deceleration ramps, refer to the chapter <i>Pulse Train Output (PTO) Principle Description</i> of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

This table describes the output variables:

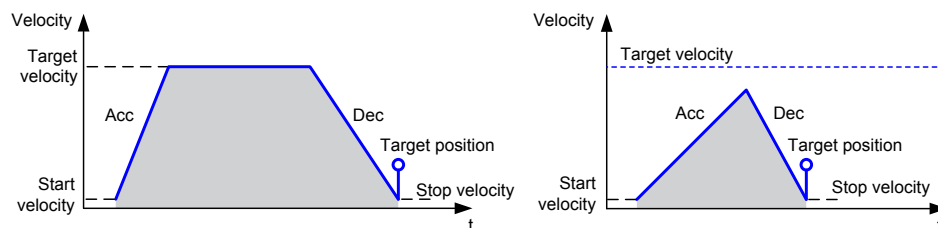
Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the MoveAbsolute process is finished. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Active</i>	BOOL	When TRUE, the function block controls the motion of the axis. Only one function block at a time can control one axis.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is cancelled due to another move command or a detected error. The motion execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

NOTE:

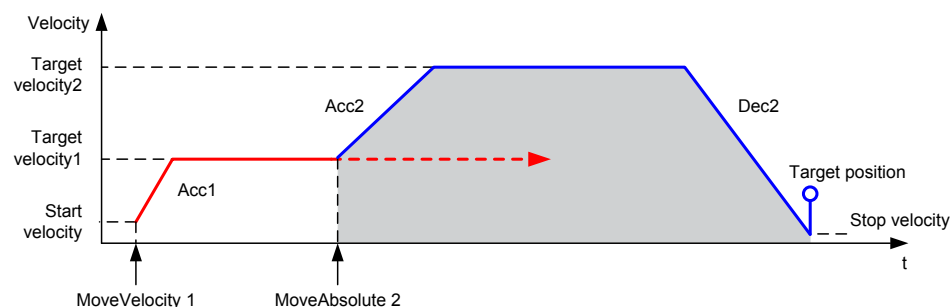
- The function block completes with velocity zero if no further function blocks are pending.
- The motion direction is automatically set, according to the present and targeted positions.
- If the distance is insufficient for the target velocity to be reached, the movement profile is triangular, rather than trapezoidal.
- If the position cannot be reached with the ongoing direction, the direction reversal is automatically managed. If a motion is ongoing, it is first halted with the deceleration of the *MC_MoveAbsolute_PTO_NTS* function block, and then the motion resumes backwards.

Timing Diagram Examples

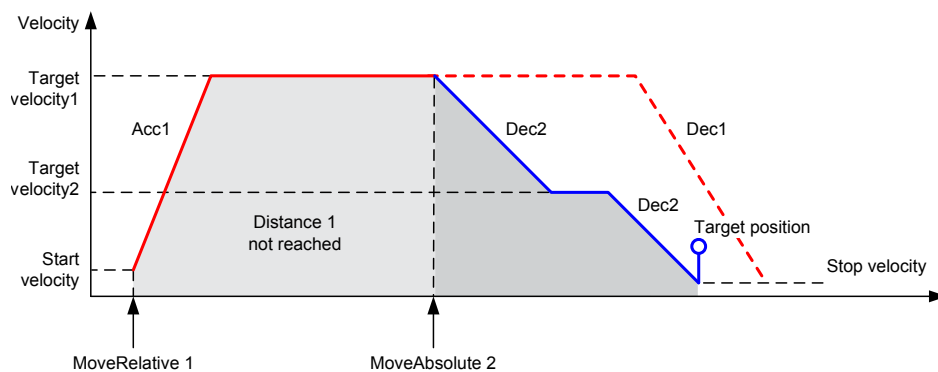
The following diagram shows the profile of the motion command when the axis is in *Standstill* state. For further information, refer to the [Motion State Diagram](#), page 59.



The following diagram shows the profile of the motion command when the axis is in *ContinuousMotion* state. For further information, refer to the [Motion State Diagram](#), page 59.

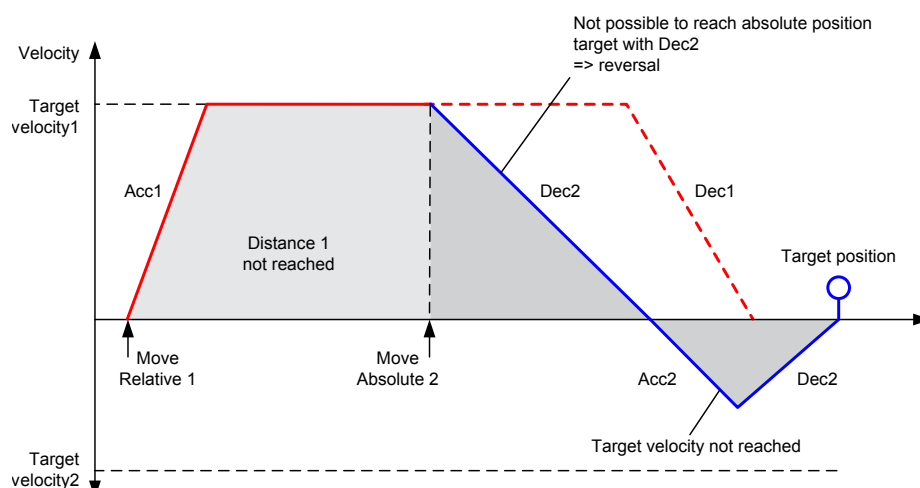


The following diagram shows the profile of the motion command when the axis is in *DiscreteMotion* state. For further information, refer to the [Motion State Diagram](#), page 59.



NOTE: In this diagram, *BufferMode* of the *Absolute* motion command is set to *mcAborting* (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

The following diagram shows the profile of the motion command when the axis is in *DiscreteMotion* state and there is a direction change. For further information, refer to the Motion State Diagram, page 59.



NOTE: In this diagram, *BufferMode* of the *Absolute* motion command is set to *mcAborting* (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

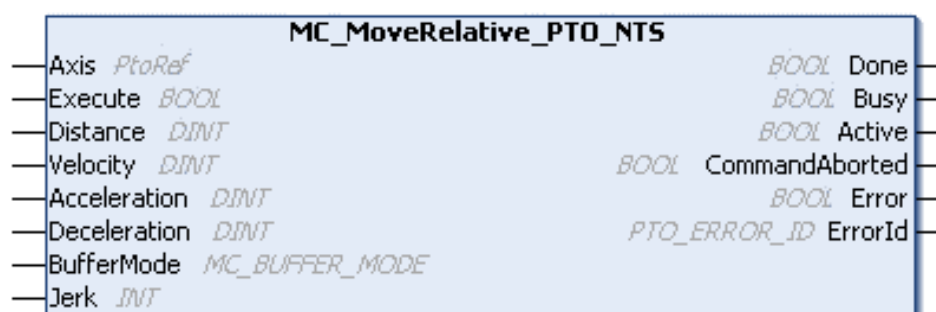
MC_MoveRelative_PTO_NTS: Commands Relative Axis Motion

Function Block Description

The *MC_MoveRelative_PTO_NTS* function block causes the specified axis to move of an incremental distance, and the axis is set to the *DiscreteMotion* state. The target position is referenced from the position at execution time, incremented by a distance.

The *MC_MoveRelative_PTO_NTS* function block executes the *Relative* motion command. For further information, refer to the description of the Relative motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Inputs	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block sends the motion command. When a falling edge is detected, the outputs of the function block are reset. When <i>Execute</i> is TRUE, the outputs of the function block are updated. When <i>Execute</i> is FALSE, the outputs of the function block are not updated. NOTE: Setting <i>Execute</i> to FALSE does not cancel the motion command sent on a rising edge.
<i>Distance</i>	DINT	The distance of the motion in number of pulses. The sign specifies the direction. Default value: 0
<i>Velocity</i>	DINT	The target velocity in Hz. Value range: 1...<i>MaxVelocityAppl</i>, page 116 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Acceleration</i>	DINT	Acceleration in Hz/ms or in ms (according to the configuration). Value range (in Hz/ms): 1...Maximum Acceleration Value range (in ms): Maximum Acceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Deceleration</i>	DINT	Deceleration in Hz/ms or in ms (according to the configuration). Value range (Hz/ms): 1...Maximum Deceleration Value range (ms): Maximum Deceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.

Inputs	Data type	Description
<i>BufferMode</i>	<i>MC_BUFFER_MODE</i> , page 111	Transition is performed from an ongoing motion. Refer to <i>MC_BUFFER_MODE</i> , page 111. Default value: <i>mcAborting</i>
<i>Jerk</i>	INT	Sets the percentage used to create an S-curve profile. Value range: 0...65,535* Default value: 0 *If the provided value is above 100, the <i>Jerk</i> is set to 100 and an advisory is issued. If the <i>Jerk</i> parameter is set to 100%, then the acceleration and deceleration values are double the value defined in the acceleration and deceleration parameters. For example, if <i>Jerk</i> is set to 100%, then the acceleration and deceleration values equal to the value defined in the <i>Acceleration</i> and <i>Deceleration</i> parameters multiplied by 2. The duration for the acceleration and deceleration is maintained regardless of the <i>Jerk</i> parameter value. To maintain this duration, the acceleration or deceleration provided to the motion command is adjusted accordingly. NOTE: If the new calculated acceleration or deceleration exceeds the Maximum Acceleration or Maximum Deceleration parameter value, an advisory is issued and <i>Jerk</i> is changed to the closest possible <i>Jerk</i> value. For further information about acceleration and deceleration ramps, refer to the chapter <i>Pulse Train Output (PTO) Principle Description</i> of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

This table describes the output variables:

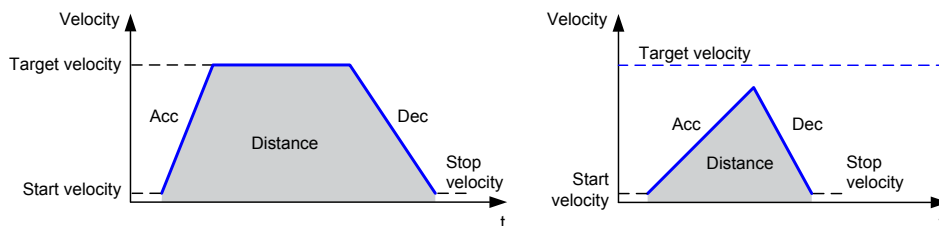
Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the MoveRelative process is finished. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Active</i>	BOOL	When TRUE, the function block controls the motion of the axis. Only one function block at a time can control one axis.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is cancelled due to another move command or a detected error. The motion execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

NOTE:

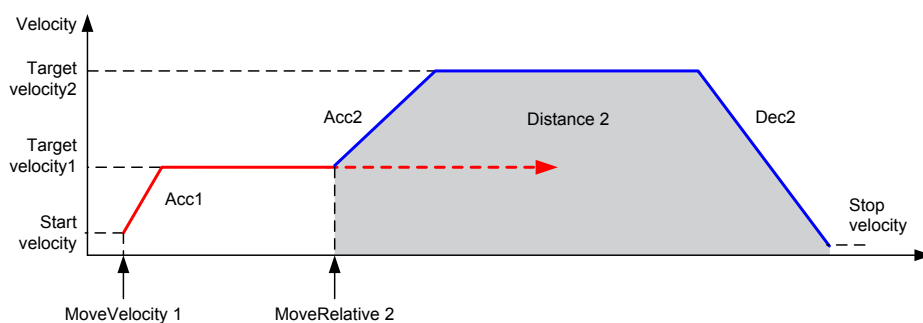
- The function block completes with velocity zero if no further function blocks are pending.
- If the distance is insufficient for the target velocity to be reached, the movement profile is triangular, rather than trapezoidal.
- If a motion is ongoing, and the commanded distance is exceeded due to the motion parameters, the direction reversal is automatically managed: the motion is first halted with the deceleration of the *MC_MoveRelative_PTO_NTS* function block, and then the motion resumes backwards.

Timing Diagram Examples

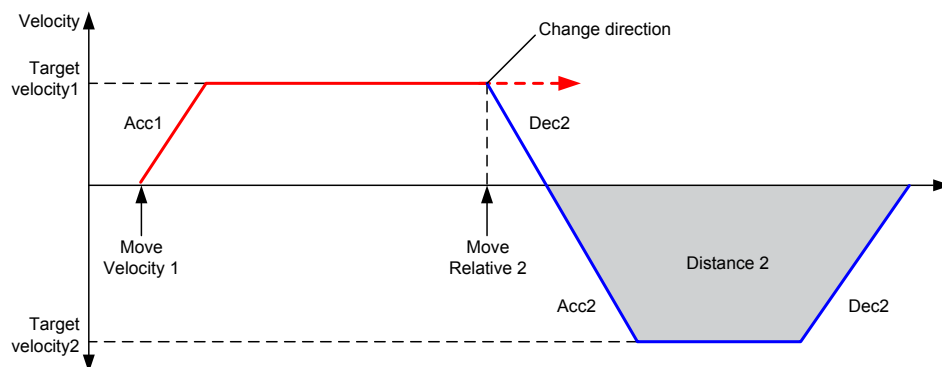
The following diagram shows the profile of the motion command when the axis is in *Standstill* state. For further information, refer to the [Motion State Diagram](#), page 59.



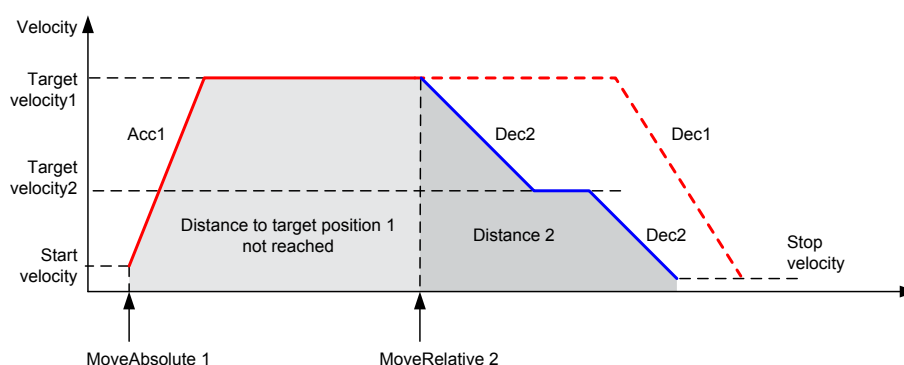
The following diagram shows the profile of the motion command when the axis is in *ContinuousMotion* state. For further information, refer to the [Motion State Diagram](#), page 59.



The following diagram shows the profile of the motion command when the axis is in *ContinuousMotion* state and there is a direction change. For further information, refer to the [Motion State Diagram](#), page 59.

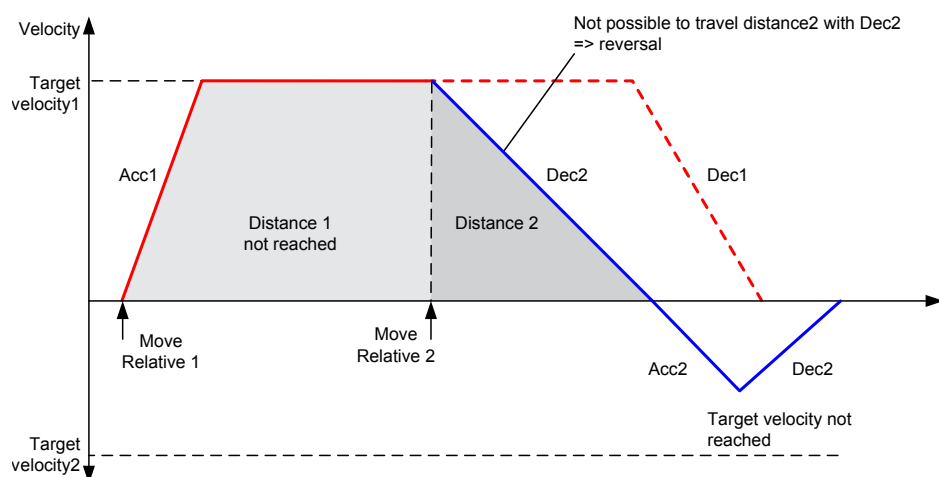


The following diagram shows the profile of the motion command when the axis is in *DiscreteMotion* state. For further information, refer to the *Motion State Diagram*, page 59.



NOTE: In this diagram, *BufferMode* of the *Relative* motion command is set to *mcAborting* (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

The following diagram shows the profile of the motion command when the axis is in *DiscreteMotion* state and there is a direction change. For further information, refer to the *Motion State Diagram*, page 59.



NOTE: In this diagram, *BufferMode* of the *Relative* motion command is set to *mcAborting* (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

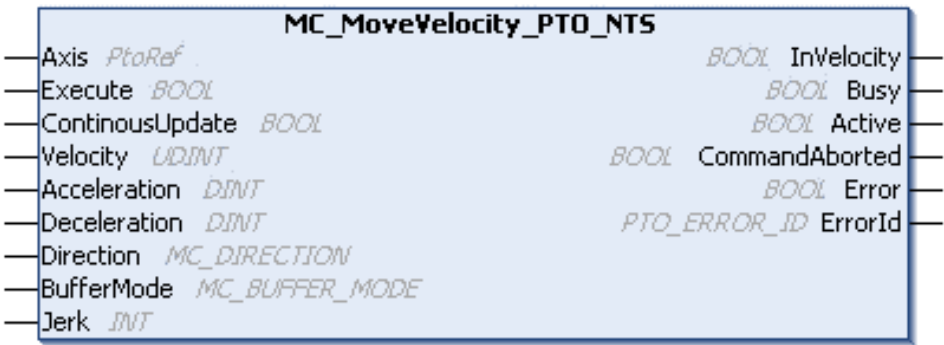
MC_MoveVelocity_PTO_NTS: Controls the Velocity of the Axis

Function Block Description

The *MC_MoveVelocity_PTO_NTS* function block causes the specified axis to move at the specified speed, and the axis is set to the state *ContinuousMotion*. This continuous movement is maintained until either a software limit is reached, a cancelling move is triggered, or a transition to the *ErrorStop* state is detected.

The *MC_MoveVelocity_PTO_NTS* function block executes the *Velocity* motion command. For further information, refer to the description of the *Velocity* motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block sends the motion command. When a falling edge is detected, the outputs of the function block are reset. When <i>Execute</i> is TRUE, the outputs of the function block are updated. When <i>Execute</i> is FALSE, the outputs of the function block are not updated. NOTE: Setting <i>Execute</i> to FALSE does not cancel the motion command sent on a rising edge.
<i>ContinuousUpdate</i>	BOOL	When TRUE, the function block applies the values of the input variables <i>Velocity</i>, <i>Acceleration</i>, <i>Deceleration</i>, <i>Jerk</i>, and <i>Direction</i> to the ongoing command. When <i>Execute</i> is TRUE, the parameters of the ongoing velocity command can be modified. When <i>Execute</i> is FALSE, the parameters of the ongoing velocity command cannot be modified.
<i>Velocity</i>	DINT	The target velocity in Hz. Value range: 1...<i>MaxVelocityAppl</i>, page 116 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Acceleration</i>	DINT	Acceleration in Hz/ms or in ms (according to the configuration). Value range (in Hz/ms): 1...Maximum Acceleration Value range (in ms): Maximum Acceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Deceleration</i>	DINT	Deceleration in Hz/ms or in ms (according to the configuration). Value range (Hz/ms): 1...Maximum Deceleration Value range (ms): Maximum Deceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Direction</i>	<i>MC_DIRECTION</i> , page 112	The direction of the motion. Refer to <i>MC_DIRECTION</i>, page 112. Default value: <i>mcPositiveDirection</i>

Input	Data type	Description
<i>BufferMode</i>	<i>MC_BUFFER_MODE</i> , page 111	Transition is performed from an ongoing motion. Refer to <i>MC_BUFFER_MODE</i> , page 111. Default value: <i>mcAborting</i>
<i>Jerk</i>	INT	Sets the percentage used to create an S-curve profile. Value range: 0...65,535* Default value: 0 *If the provided value is above 100, the <i>Jerk</i> is set to 100 and an advisory is issued. If the <i>Jerk</i> parameter is set to 100%, then the acceleration and deceleration values are double the value defined in the acceleration and deceleration parameters. For example, if <i>Jerk</i> is set to 100%, then the acceleration and deceleration values equal to the value defined in the <i>Acceleration</i> and <i>Deceleration</i> parameters multiplied by 2. The duration for the acceleration and deceleration is maintained regardless of the <i>Jerk</i> parameter value. To maintain this duration, the acceleration or deceleration provided to the motion command is adjusted accordingly. NOTE: If the new calculated acceleration or deceleration exceeds the Maximum Acceleration or Maximum Deceleration parameter value, an advisory is issued and <i>Jerk</i> is changed to the closest possible <i>Jerk</i> value. For further information about acceleration and deceleration ramps, refer to the chapter <i>Pulse Train Output (PTO) Principle Description</i> of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

This table describes the output variables:

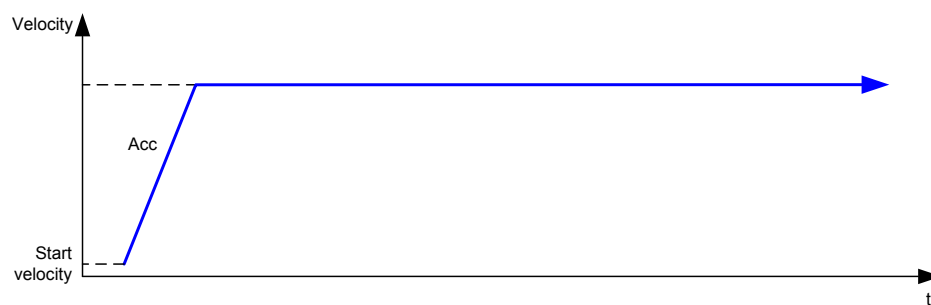
Output	Data type	Description
<i>InVelocity</i>	BOOL	When TRUE, the target velocity is reached. The motion is ongoing and the motion execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>Active</i>	BOOL	When TRUE, the function block controls the motion of the axis. Only one function block at a time can control one axis.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is cancelled due to another move command or a detected error. The motion execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

NOTE:

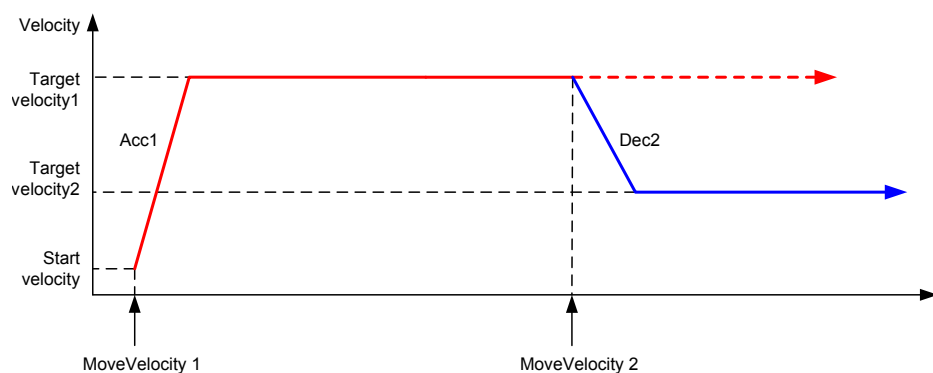
- To stop the motion, the function block has to be interrupted by another function block issuing a new command.
- If a motion is ongoing, and the direction is reversed, first the motion is halted with the deceleration of the *MC_MoveVelocity_PTO_NTS* function block, and then the motion resumes backwards.

Timing Diagram Examples

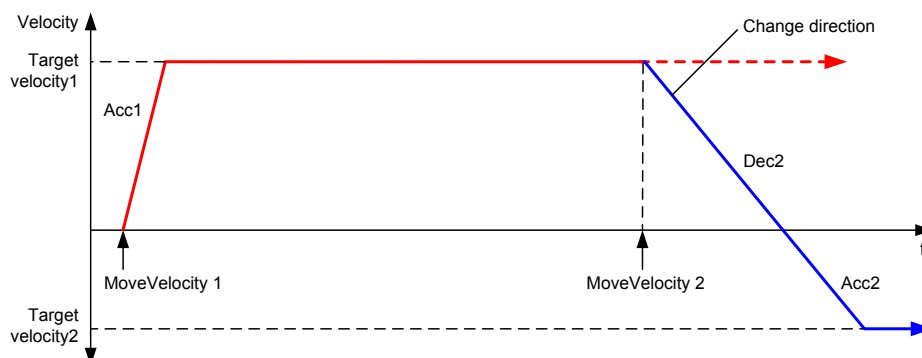
The following diagram shows the profile of the motion command when the axis is in *Standstill* state:



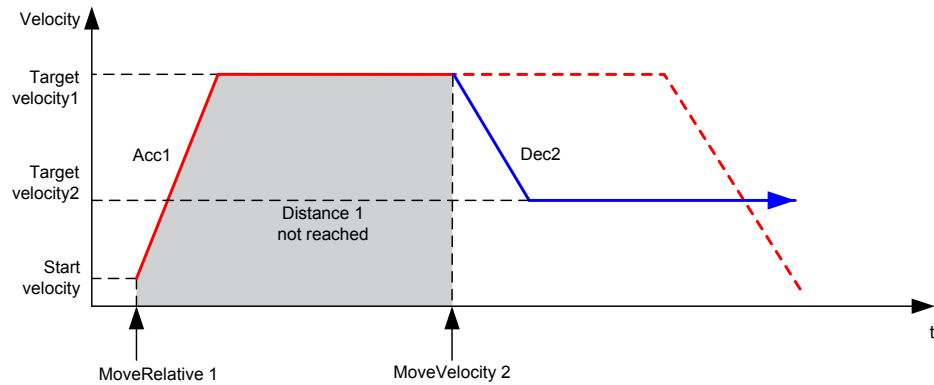
The following diagram shows the profile of the motion command when the axis is in *ContinuousMotion* state:



The following diagram shows the profile of the motion command when the axis is in *ContinuousMotion* state and there is a direction change:



The following diagram shows the profile of the motion command when the axis is in *DiscreteMotion* state:



NOTE: In this diagram, *BufferMode* of the *Velocity* motion command is set to *mcAborting* (refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide).

MC_Power_PTO_NTS: Manages the Power of the Axis

Function Block Description

The *MC_Power_PTO_NTS* function block is mandatory for execution of the other PTO function blocks. It allows enabling power and control to the axis, switching the axis state from *Disabled* to *Standstill*.

This function block needs to be the first PTO function block that is called.

No motion function block is allowed to affect the axis until the *MC_Power_PTO_NTS.Status* bit is TRUE.

Disabling power (*MC_Power_PTO_NTS.Enable* equals to FALSE) switches the axis:

- from *Standstill* to *Disabled* state.
- from an ongoing motion to *ErrorStop*, and then to *Disabled* when the detected error is reset.

If the **DriveReady Enabled** input parameter is reset, the axis state switches to *ErrorStop*.

Graphical Representation



I/O Variable Description

This table describes the input variables:
For further information, refer to the *PTO Configuration* chapter of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

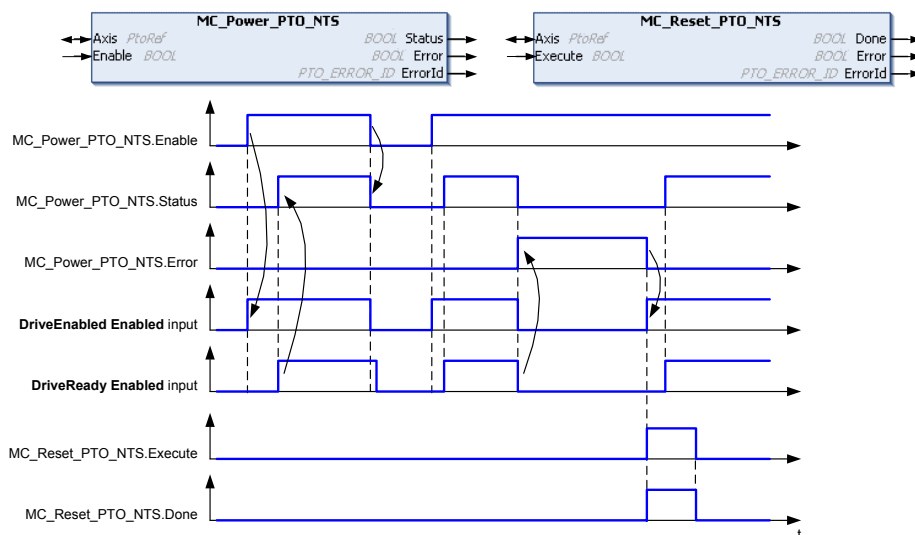
Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Enable</i>	BOOL	When TRUE and the DriveReady Enabled input parameter is TRUE, and no error is detected, power is supplied to the axis. The axis state is switched from <i>Disabled</i> to <i>Standstill</i> . When FALSE, terminates the function block execution and resets the outputs.

This table describes the output variables:

Output	Data type	Description
<i>Status</i>	BOOL	When TRUE, power is enabled, motion commands can be executed.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

Timing Diagram Example

The diagram illustrates the function block operation:



The **DriveEnabled Enabled** input parameter and the **DriveReady Enabled** input parameter are configured in the **Edge I/O NTS Editor** of the Logic Builder. For further information, refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

MC_SetPosition_PTO_NTS: Modifies the Reference Position of the Axis

Function Block Description

The *MC_SetPosition_PTO_NTS* function block modifies the coordinates of the position of the axis without physical movement. This function block can only be used while the axis is in *Standstill* state. It can be used as a simple homing mode. Once executed, absolute positioning is allowed.

The *MC_SetPosition_PTO_NTS* function block executes the *Set Position* motion command. For further information, refer to the description of the Set Position motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block starts execution. When a falling edge is detected, the function block stops execution and the outputs are reset.
<i>Position</i>	DINT	New value of the position of the Axis. Default value: 0

This table describes the output variables:

Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the SetPosition process is finished. Function block execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	PTO_ERROR_ID, page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

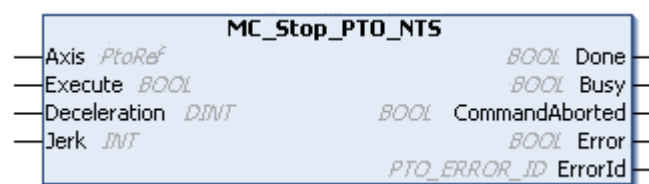
MC_Stop_PTO_NTS: Commands a Controlled Motion Stop

Function Block Description

The *MC_Stop_PTO_NTS* function block commands a controlled motion stop and the axis is set to the state *Stopping*. It aborts ongoing motion execution and the motion queue is cleared. While the axis is in state *Stopping*, no other function block can perform motion on the axis.

The *MC_Stop_PTO_NTS* function block executes the *Stop* motion command. For further information, refer to the description of the *Stop* motion command in the chapter *Pulse Train Output (PTO) Principle Description* of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variable Description

This table describes the input variables:

Input	Data type	Description
<i>Axis</i>	PtoRef	Reference to the name of the axis (instance) for which the function block is to be executed. In the Devices tree , the name is declared in the controller configuration.
<i>Execute</i>	BOOL	When a rising edge is detected, the function block sends the motion command. When a falling edge is detected, the outputs of the function block are reset. When <i>Execute</i> is TRUE, the outputs of the function block are updated. When <i>Execute</i> is FALSE, the outputs of the function block are not updated. NOTE: Setting <i>Execute</i> to FALSE does not cancel the motion command sent on a rising edge.
<i>Deceleration</i>	DINT	Deceleration in Hz/ms or in ms (according to the configuration). Value range (Hz/ms): 1...Maximum Deceleration Value range (ms): Maximum Deceleration...400,000 Default value: 0 NOTE: The default value of ZERO will cause the function block to detect an error, thereby requiring you to set an appropriate value for the parameter.
<i>Jerk</i>	INT	Sets the percentage used to create an S-curve profile. Value range: 0...65,535* Default value: 0 *If the provided value is above 100, the <i>Jerk</i> is set to 100 and an advisory is issued. If the <i>Jerk</i> parameter is set to 100%, then the acceleration and deceleration values are double the value defined in the acceleration and deceleration parameters. For example, if <i>Jerk</i> is set to 100%, then the acceleration and deceleration values equal to the value defined in the <i>Acceleration</i> and <i>Deceleration</i> parameters multiplied by 2. The duration for the acceleration and deceleration is maintained regardless of the <i>Jerk</i> parameter value. To maintain this duration, the acceleration or deceleration provided to the motion command is adjusted accordingly. NOTE: If the new calculated acceleration or deceleration exceeds the Maximum Acceleration or Maximum Deceleration parameter value, an advisory is issued and <i>Jerk</i> is changed to the closest possible <i>Jerk</i> value. For further information about acceleration and deceleration ramps, refer to the chapter <i>Pulse Train Output (PTO) Principle Description</i> of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

This table describes the output variables:

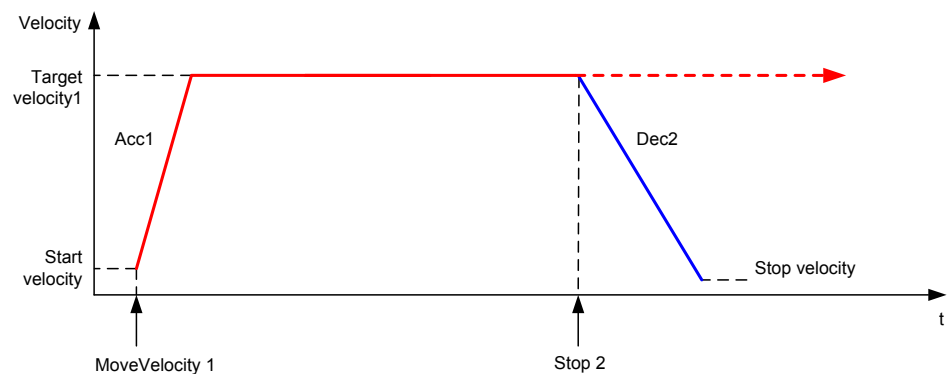
Output	Data type	Description
<i>Done</i>	BOOL	TRUE indicates that the stop is finished. Function block execution is finished.
<i>Busy</i>	BOOL	TRUE indicates that the function block is busy processing data.
<i>CommandAborted</i>	BOOL	When TRUE, the function block execution is cancelled due to another move command or a detected error. The motion execution is finished.
<i>Error</i>	BOOL	TRUE indicates that an error is detected. Function block execution is finished.
<i>ErrorId</i>	<i>PTO_ERROR_ID</i> , page 113	Indicates the identification number of the detected error when <i>Error</i> is TRUE.

NOTE:

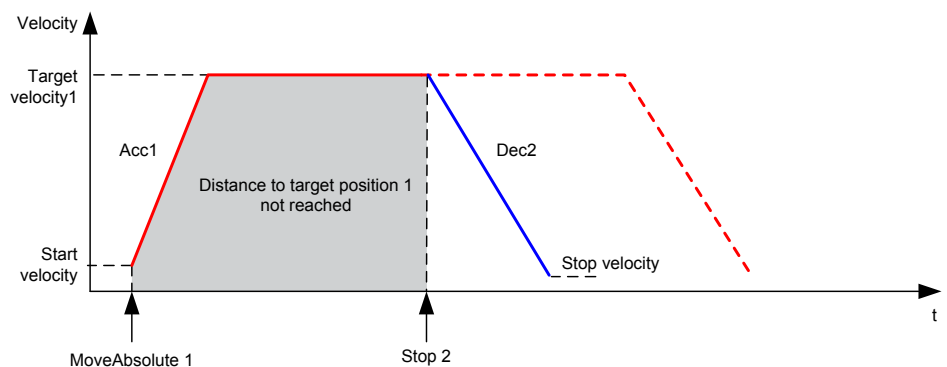
- Calling this function block in state *Standstill* changes the state to *Stopping*, and back to *Standstill* when *Execute* is FALSE.
- The state *Stopping* is maintained as long as the input *Execute* is TRUE or the motion command is ongoing.
- The *Done* output is set to TRUE when the stop ramp is finished.
- If *Deceleration* is equal to 0, the fast stop deceleration is used.
- The function block completes with velocity zero.

Timing Diagram Examples

The following diagram shows the profile of the motion command when the axis is in *ContinuousMotion* state:



The following diagram shows the profile of the motion command when the axis is in *DiscreteMotion* state:



PWM Function Block

What's in This Chapter

PWM Configuration 100

PWM_NTS: Commands a PWM Signal..... 100

PWM Configuration

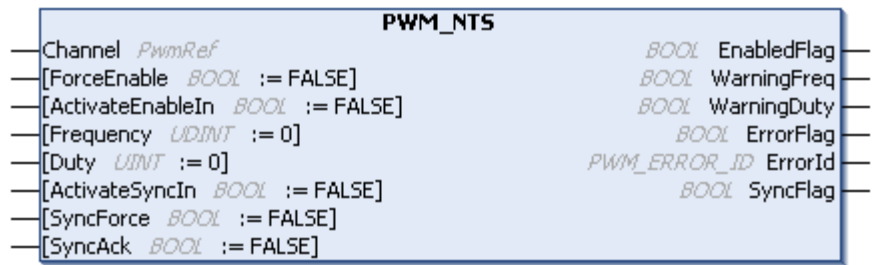
For information on PWM configuration in the **Edge I/O NTS Editor** of the Logic Builder, refer to the *PWM Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

PWM_NTS: Commands a PWM Signal

Function Block Description

The *PWM_NTS* function block commands a Pulse Width Modulation (PWM) signal output at the specified frequency and duty cycle. For further information, refer to the chapter Pulse Width Modulation Mode Principle Description in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Graphical Representation



I/O Variables Description

This table describes the input variables:

Input	Data type	Description
<i>Channel</i>	PwmRef	Reference to the PWM instance.
<i>ForceEnable</i>	BOOL	When TRUE, forces the pulse generation. The <i>ActivateEnableIn</i> input is ignored. Default value: FALSE
<i>ActivateEnableIn</i>	BOOL	When TRUE, enables pulse generation provided that the <i>EnConfigured</i> input that is configured is TRUE. Default value: FALSE
<i>Frequency</i>	UDINT	Frequency to apply to the pulse generation in steps of 0.1 Hz. Value range: 0...200,000 (0...20 kHz) Default value: 0
<i>Duty</i>	UINT	Duty cycle to apply to the pulse generation in steps of 0.1%. Value range: 0...1000 (0...100%) Default value: 0
<i>ActivateSyncIn</i>	BOOL	When TRUE, the synchronization is started when a <i>SyncConfigured</i> input parameter is defined other than 0 and the <i>SyncFlag</i> output is set to TRUE. Default value: FALSE
<i>SyncForce</i>	BOOL	When a rising edge is detected, forces the synchronization and sets the <i>SyncFlag</i> output to TRUE independently of the defined <i>SyncConfigured</i> input parameter. Default value: FALSE
<i>SyncAck</i>	BOOL	When a rising edge is detected, the <i>SyncFlag</i> output is reset. Default value: FALSE

This table describes the output variables:

Output	Data type	Description
<i>EnabledFlag</i>	BOOL	TRUE indicates that the output values on the function block are valid. If the function block is disabled, the output is set to FALSE.
<i>WarningFreq</i>	BOOL	When TRUE, the value of the input <i>Frequency</i> exceeds the maximum frequency. The applied frequency is the maximum frequency. Default value: FALSE
<i>WarningDuty</i>	BOOL	When TRUE, the value of the input <i>Duty</i> exceeds the maximum duty. The applied duty is the maximum duty. Default value: FALSE
<i>ErrorFlag</i>	BOOL	TRUE indicates that an error is detected.
<i>ErrorId</i>	PWM_ERROR_ID, page 117	Indicates the identification number of the detected error when <i>Error</i> is TRUE.
<i>SyncFlag</i>	BOOL	When TRUE, synchronization has been performed. Default value: FALSE

Enumerations and Structures

What’s in This Part

Encoder Enumerations	103
Fast_IO Enumeration	105
HSC Enumerations.....	106
PTO Enumerations.....	111
PWM_FREQ_GEN Enumerations	117

Encoder Enumerations

What's in This Chapter

<i>ENCODER_ERROR_ID</i>	103
<i>ENCODER_SUB_FCT_CAM_DIRECTION</i>	104
<i>ENCODER_SUB_FCT_CAM_TYPE</i>	104

ENCODER_ERROR_ID

Description

The *ENCODER_ERROR_ID* enumeration describes the types of errors that can be detected on an encoder function block.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>NO_ERROR</i>	0	No error is detected.
<i>INTERNAL_FIELD_POWER_SUPPLY</i>	1	Internal field power supply is not detected.
<i>POWER</i>	2	The power monitoring of the encoder does not detect power return of the encoder.
<i>FAULT</i>	3	Power cannot be supplied to the encoder.
<i>SHORTCUT</i>	4	A short-circuit is detected in the channel.
<i>FALLBACK</i>	5	An implicit communication issue is detected. The module is in fallback state. The expert function is stopped.
<i>QUADRATURE</i>	10	The signals from the incremental encoder are incorrect.
<i>FRAME_FORMAT</i>	20	Loss of communication between module and encoder.
<i>FRAME_PARITY</i>	21	Loss of communication between module and encoder.
<i>FRAME_START</i>	22	Loss of communication between module and encoder.
<i>FRAME_STUCK_AT_1</i>	23	Loss of communication between module and encoder.
<i>TIMEOUT</i>	30	No reply received from the encoder to a communication request.
<i>PROTOCOL</i>	31	Internal error detected in the encoder.
<i>CRC</i>	32	Communication between module and encoder is established but the content is incorrect.
<i>CONF_NOT_SUPPORTED</i>	33	The encoder used is not supported by the module.
<i>SINCOS_SIGNAL</i>	40	The signals received from the encoder are incorrect.
<i>SINCOS_SYNCHRO</i>	41	A synchronization issue between the number of period and the angle in the period has been detected. The position cannot be calculated.
<i>NOT_CONFIGURED</i>	50	The encoder is not configured. The encoder reference is invalid.
<i>ENC_NOT_ENABLED</i>	80	The encoder is not enabled. Use the <i>Encoder_Main_NTS</i> function block to enable the encoder.
<i>COMM_ERR</i>	100	The module is unavailable or the Network Interface Module (NIM) is unavailable or the GCS (Group Cyclic Status) value is incorrect.

ENCODER_SUB_FCT_CAM_DIRECTION

Description

The *ENCODER_SUB_FCT_CAM_DIRECTION* enumeration defines the three modes of direction that are available for CAMs. For further information, refer to the *CAM Mode Principle Description* in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>BOTH</i>	0	The CAM is active in forward and reverse directions.
<i>FORWARD</i>	1	The CAM is active in forward direction.
<i>REVERSE</i>	2	The CAM is active in reverse direction.

ENCODER_SUB_FCT_CAM_TYPE

Description

The *ENCODER_SUB_FCT_CAM_TYPE* enumeration defines the three types of CAMs that are available. For further information, refer to the *CAM Mode Principle Description* in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>POSITION</i>	0	A Position type CAM is available.
<i>MONOSTABLE</i>	1	A Monostable type CAM is available.
<i>BRAKING</i>	2	A Braking type CAM is available.

Fast_IO Enumeration

What's in This Chapter

FAST_IO_ERROR_ID 105

FAST_IO_ERROR_ID

Description

The *FAST_IO_ERROR_ID* enumeration describes the types of errors that can be detected on the *FastIO_React_NTS* function block, page 28.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>NO_ERROR</i>	0	No error is detected.
<i>INTERNAL_FIELD_POWER_SUPPLY</i>	1	Internal field power supply is not detected.
<i>SHORTCUT</i>	2	A short-circuit is detected in the channel.
<i>FALLBACK</i>	3	An implicit communication issue is detected. The module is in fallback state. The expert function is stopped.
<i>COMM_ERR</i>	100	The module is unavailable or the NIM is unavailable or the GCS (Group Cyclic Status) value is incorrect.

HSC Enumerations

What's in This Chapter

<i>HSC_PARAM_TYPE_NTS</i>	106
<i>HSC_ERROR_NTS</i>	107

HSC_PARAM_TYPE_NTS

Description

The *HSC_PARAM_TYPE_NTS* enumeration is dedicated to parameters that perform a Get or Set operation on HSC function blocks.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>HSC_PRESET</i>	0x01	To get or set the Preset value for the counting functions Simple Counting , Single Phase Counting and Dual Phase Counting . For further information, refer to the <i>Modicon Edge I/O NTS Counting Modules User Guide</i> .
<i>HSC_TIMEBASE</i>	0x02	To get or set the Time Base value for the counting functions Frequency Meter , Ratio Meter and Single Phase Event Counting . For further information, refer to the <i>Modicon Edge I/O NTS Counting Modules User Guide</i> .
<i>HSC_CALIBRATION_FACTOR</i>	0x03	To get or set the Calibration Factor value for the counting functions Frequency Meter and Ratio Meter .
<i>HSC_ABSOLUTE_LIMIT</i>	0x04	To get or set the Absolute Limit value for the counting function Ratio Meter .
<i>HSC_RESOLUTION</i>	0x05	To get or set the Resolution value for the counting function Period Meter .
<i>HSC_HYSTERESIS</i>	0x06	To get or set the Hysteresis value for the counting function dual phase in free-large mode.
<i>HSC_CAPTURE_WINDOW_START_POSITION</i>	0x07	To get or set the capture window start position CAP Window Start value for the counting functions Single Phase Event Counting and Dual Phase Counting .
<i>HSC_CAPTURE_WINDOW_END_POSITION</i>	0x08	To get or set the capture window end position CAP Window End value for the counting functions Single Phase Event Counting and Dual Phase Counting .
<i>HSC_THRESHOLD0</i>	0x09	To get or set the Threshold0 value for the counting subfunction reflex output.
<i>HSC_THRESHOLD1</i>	0x0A	To get or set the Threshold1 value for the counting subfunction reflex output.
<i>HSC_THRESHOLD2</i>	0x0B	To get or set the Threshold2 value for the counting subfunction reflex output.
<i>HSC_THRESHOLD3</i>	0x0C	To get or set the Threshold3 value for the counting subfunction reflex output.

HSC_ERROR_NTS

Description

The *HSC_ERROR_NTS* enumeration contains the possible values that indicate the errors detected by the function block.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>HSC_NO_ERROR</i>	0x00	No error detected.
<i>HSC_UNKNOWN_REF</i>	0x01	The HSC reference is incorrect or not configured.
<i>HSC_UNKNOWN_PARAMETER</i>	0x02	The parameter reference is incorrect. For valid parameters, refer to <i>HSC_PARAM_TYPE_NTS</i> , page 106.

Name	Value	Description
<i>HSC_INVALID_PARAMETER</i>	0x03	The value of the parameter is incorrect. The following conditions must be met: • HSC_PRESET ◦ Simple Counting: $0 \leq \text{preset} \leq 65,535$ ◦ Single Phase Counting: $0 \leq \text{preset} \leq 2,147,483,647$ ◦ Dual Phase Counting: – modulo loop: $0 \leq \text{preset} \leq 2,147,483,647$ – free large: $-2,147,483,648 \leq \text{preset} \leq 2,147,483,647$ • HSC_TIMEBASE ◦ Frequency Meter – 0: 10 ms – 1: 100 ms – 2: 1000 ms – 3: 60 s ◦ Ratio Meter – 0: 10 ms – 1: 100 ms – 2: 1000 ms ◦ Single Phase Event Counting – 0: 1 ms – 1: 10 ms – 2: 100 ms – 3: 1000 ms – 4: 10 s – 5: 60 s • HSC_CALIBRATION_FACTOR: $-100 \leq \text{calibration factor} \leq 100$ • HSC_ABSOLUTE_LIMIT: $10 \leq \text{absolute limit} \leq 60,000,000$ • HSC_RESOLUTION ◦ Period Meter: – 0: 0.1 μs – 1: 1 μs – 2: 10 μs – 3: 100 μs – 4: 1000 μs • HSC_HYSTERESIS: $-128 < \text{Hysteresis} < 127$ • HSC_CAPTURE_WINDOW_START_POSITION ◦ The value of <i>HSC_CAPTURE_WINDOW_START_POSITION</i> must be less than the value of <i>HSC_CAPTURE_WINDOW_END_POSITION</i>. ◦ The value of <i>HSC_CAPTURE_WINDOW_START_POSITION</i> must be less than or equal to the Preset value. • HSC_CAPTURE_WINDOW_END_POSITION ◦ The value of <i>HSC_CAPTURE_WINDOW_END_POSITION</i> must be larger than the value of <i>HSC_CAPTURE_WINDOW_START_POSITION</i>. ◦ The value of <i>HSC_CAPTURE_WINDOW_END_POSITION</i> must be less than or equal to the Preset value. • HSC_THRESHOLD_X: $\text{Threshold } 0 < \text{Threshold } 1 < \text{Threshold } 2 < \text{Threshold } 3$
<i>HSC_COM_ERROR</i>	0x04	A communication error is detected with the HSC module.
<i>HSC_SHORTWIRE_GROUP0_DETECTED</i>	0x05	An output short-circuit group 0 is detected. (Q0...Q3).

Name	Value	Description
<i>HSC_SHORTWIRE_GROUP1_DETECTED</i>	0x06	An output short-circuit group 1 is detected. (Q4...Q7).
<i>HSC_PDM_ERROR</i>	0x07	A 24 V power supply error is detected.

PTO Enumerations

What's in This Chapter

<i>MC_BUFFER_MODE</i>	111
<i>MC_DIRECTION</i>	112
<i>PTO_ERROR_ID</i>	113
<i>PTO_HOMING_MODE</i>	115
<i>PTO_PARAMETER</i>	116

MC_BUFFER_MODE

Description

The *MC_BUFFER_MODE* enumeration provides the buffer modes. For a general description of the buffer mode, refer to the section Buffer Mode Description in the Pulse Train Output Interface > Operating Mode chapter of the Modicon Edge I/O NTS Motion Expert Modules User Guide.

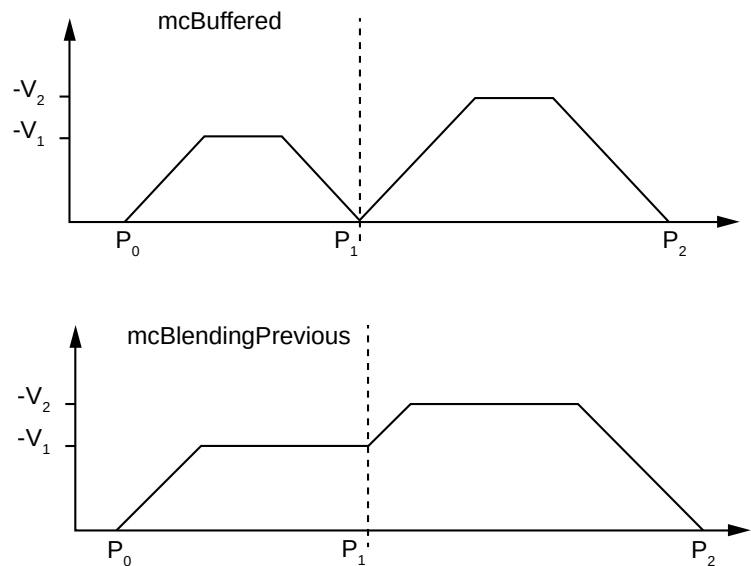
Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>mcAborting</i>	0	The function block is enabled immediately. Ongoing motion is aborted and the motion queue is cleared. Default mode
<i>mcBuffered</i>	1	The function block is enabled after the ongoing motion has been completed (the <i>Done</i> or <i>InVelocity</i> bit is set to TRUE). Blending is not performed.
<i>mcBlendingPrevious</i>	3	The velocity is blended with the velocity of the first function block (blending with the velocity of function block 1 at the end position of function block 1).
<i>seTrigger</i>	10	When the function block is executed, the motion command is buffered. When the touchprobe event conditions are met, the motion command is executed aborting any ongoing motion command.
<i>seBufferedDelay</i>	11	This buffer mode operates like the <i>mcBuffered</i> but the execution of the function block starts after the Delay parameter value duration. For further information, refer to the <i>PTO Configuration</i> chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Examples

The examples below show a movement executed by two motion commands. The axis moves from the position P_0 to P_1 and then P_2 . The second command is passed while the axis is executing the first command but before the stopping ramp is reached. For each motion profile below, P_1 is the reference point for the blending calculation. The buffer mode determines whether velocity V_1 or V_2 is reached at position P_1 .



MC_DIRECTION

Description

The *MC_DIRECTION* enumeration provides the motion direction.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>mcNegativeDirection</i>	-1	CCW (Counter Clock Wise), backward, reverse, negative motion
<i>mcPositiveDirection</i>	1	CW (Clock Wise), forward, positive motion
<i>mcCurrentDirection</i>	2	Motion is performed in the direction that was last used.

PTO_ERROR_ID

Description

The *PTO_ERROR_ID* enumeration describes the types of PTO errors that can be detected:

- Detected axis errors are in the range from 1000...1999
- Detected motion state errors are in the range from 2000...2999
- Detected range errors are in the range from 3000...3999

Enumeration Elements

The enumeration data type ENUM contains the following values. For parameters that are configured in the **Edge I/O NTS Editor** of the Logic Builder, refer to the *PTO Configuration* chapter in the Modicon Edge I/O NTS Motion Expert Modules User Guide.

Name	Value	Description
<i>NoError</i>	0	No error is detected.
<i>Communication Error</i>	100	The module is unavailable or the NIM is unavailable or the GCS (Group Cyclic Status) value is incorrect.
<i>InternalError</i>	1000	Module internal error detected.
<i>DisabledAxis</i>	1001	The motion command could not be started or was aborted because the axis is not ready.
<i>HwPositionLimitP</i>	1002	The hardware positive position limit is reached.
<i>HwPositionLimitN</i>	1003	The hardware negative position limit is reached.
<i>SwPositionLimitP</i>	1004	The software positive position limit is reached.
<i>SwPositionLimitN</i>	1005	The software negative position limit is reached.
<i>ApplicationStopped</i>	1006	Application execution has been stopped.
<i>OutputProtection</i>	1007	Short-circuit output protection is active on the PTO channels.
<i>INTERNAL_FIELD_POWER_SUPPLY</i>	1008	Internal field power supply is not detected.
<i>Fallback</i>	1009	Outputs are in fallback state.
<i>WarningVelocityValue</i>	1100	Commanded <i>Velocity</i> parameter is out of range.
<i>WarningAccelerationValue</i>	1101	Commanded <i>Acceleration</i> parameter is out of range.
<i>WarningDecelerationValue</i>	1102	Commanded <i>Deceleration</i> parameter is out of range.
<i>WarningJerkValue</i>	1104	Commanded <i>JerkRatio</i> parameter is limited by the configured maximum acceleration or deceleration. In this case, the <i>JerkRatio</i> is recalculated to respect these maximums.
<i>ErrorStopActive</i>	2000	The motion command could not be started or was aborted because the axis is in <i>ErrorStop</i> state.
<i>StoppingActive</i>	2001	The motion command could not be started or was aborted because the axis is controlled by the function block <i>MC_Stop_PTO_NTS</i> , page 97 (either the axis is stopping or the <i>MC_Stop_PTO_NTS.Execute</i> input is TRUE).
<i>InvalidTransition</i>	2002	Transition is not allowed. Refer to the <i>Motion State Diagram</i> , page 59.
<i>SetPositionInvalid</i>	2003	To execute the <i>MC_SetPosition_PTO_NTS</i> , page 96, the axis must be in <i>Standstill</i> state.
<i>HomingError</i>	2004	<i>MC_Home_PTO_NTS</i> , page 78 cannot be executed.
<i>InvalidProbeConf</i>	2005	The PROBE Enabled input parameter must be configured.
<i>InvalidHomingConf</i>	2006	The homing input parameters (REF Enabled , INDEX Enabled) must be configured for this homing mode.
<i>AbsoluteInvalid</i>	2007	An absolute motion cannot be executed while the axis is not successfully referenced to an origin position. A homing sequence needs to be executed first (<i>MC_Home_PTO_NTS</i> , page 78 or <i>MC_SetPosition_PTO_NTS</i> , page 96).
<i>MotionQueueFull</i>	2008	The motion command could not be buffered because the buffer queue is full.
<i>InvalidAxis</i>	3000	The function block is not applicable for the specified axis.

Name	Value	Description
<i>InvalidPositionValue</i>	3001	The value of the <i>Position</i> parameter is out of range, or the <i>Distance</i> parameter provides a position that is out of range.
<i>InvalidVelocityValue</i>	3002	The value of the <i>Velocity</i> parameter is out of range. The value must be greater than 0.
<i>InvalidAccelerationValue</i>	3003	The value of the <i>Acceleration</i> parameter is out of range.
<i>InvalidDecelerationValue</i>	3004	The value of the <i>Deceleration</i> parameter is out of range.
<i>InvalidBufferModeValue</i>	3005	The <i>buffer mode</i> , page 111 does not correspond to a valid enumeration value.
<i>InvalidDirectionValue</i>	3006	The direction does not correspond to a valid enumeration value, or the direction is invalid because the software position limit is exceeded.
<i>InvalidHomeMode</i>	3007	The homing <i>Mode</i> parameter, page 78 value is incorrect.
<i>InvalidParameter</i>	3008	The parameter number does not exist for the specified axis.
<i>InvalidParameterValue</i>	3009	The parameter value is out of range.
<i>ReadOnlyParameter</i>	3010	The parameter is read-only.

PTO_HOMING_MODE

Description

The *PTO_HOMING_MODE* enumeration describes the homing movement types. For further information, refer to *Homing Modes*, page 65.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>PositionSetting</i>	0	Homing movement type position setting, page 66.
<i>LongReference</i>	1	Homing movement type long reference, page 67.
<i>LongReferenceAndIndex</i>	10	Homing movement type long reference & index, page 68.
<i>ShortReference_Reversal</i>	20	Homing movement type short reference reversal, page 69.
<i>ShortReference_NoReversal</i>	21	Homing movement type short reference no reversal, page 70.
<i>ShortReferenceAndIndex_Outside</i>	30	Homing movement type short reference & index outside, page 71.
<i>ShortReferenceAndIndex_Inside</i>	31	Homing movement type short reference & index inside, page 72.

PTO_PARAMETER

Description

The *PTO_PARAMETER* enumeration describes the types of PTO parameters.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Parameter name	Data type	Parameter number	R/W	Description
<i>CommandedPosition</i>	DINT	1	R	Provides the distance traveled during the previous motion command.
<i>SWPosLimit</i>	DINT	2	R/W	The value of the SW Limit Positive Enabled parameter.
<i>SWNegLimit</i>	DINT	3	R/W	The value of the SW Positive Limit parameter.
<i>EnableSWLimitPos</i>	BOOL	4	R/W	The value of the SW Limit Negative Enabled parameter.
<i>EnableSWLimitNeg</i>	BOOL	5	R/W	The value of the SW Negative Limit parameter.
<i>EnableDirPos</i>	BOOL	6	R/W	The value of the Direction Positive Enabled parameter.
<i>EnableDirNeg</i>	BOOL	7	R/W	The value of the Direction Negative Enabled parameter.
<i>MaxVelocityAppl</i>	DINT	9	R/W	The value of the Maximum Velocity parameter.
<i>CommandedVelocity</i>	DINT	11	R	Provides the target velocity value of the previous motion command.
<i>MaxAccelerationAppl</i>	DINT	13	R/W	The value of the Maximum Acceleration parameter.
<i>MaxDecelerationAppl</i>	DINT	15	R/W	The value of the Maximum Deceleration parameter.
<i>Delay</i>	DINT	1000	R/W	The value of the Delay parameter.

PWM_FREQ_GEN Enumerations

What's in This Chapter

<i>FREQ_GEN_ERROR_ID</i>	117
<i>PWM_ERROR_ID</i>	117

FREQ_GEN_ERROR_ID

Description

The *FREQ_GEN_ERROR_ID* enumeration describes the types of errors that can be detected on the *FreqGen_NTS* function block, page 30.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>NO_ERROR</i>	0	No error is detected.
<i>INTERNAL_FIELD_POWER_SUPPLY</i>	1	Internal field power supply is not detected.
<i>SHORTCUT</i>	2	A short-circuit is detected in the channel.
<i>FALLBACK</i>	3	An implicit communication issue is detected. The module is in fallback state. The expert function is stopped.
<i>COMM_ERR</i>	100	The module is unavailable or the NIM is unavailable or the GCS (Group Cyclic Status) value is incorrect.

PWM_ERROR_ID

Description

The *PWM_ERROR_ID* enumeration describes the types of errors that can be detected in Pulse Width Modulation (PWM) mode.

Enumeration Elements

The enumeration data type ENUM contains the following values:

Name	Value	Description
<i>NO_ERROR</i>	0	No error is detected.
<i>INTERNAL_FIELD_POWER_SUPPLY</i>	1	Internal field power supply is not detected.
<i>SHORTCUT</i>	2	A short-circuit is detected in the channel.
<i>FALLBACK</i>	3	An implicit communication issue is detected. The module is in fallback state. The expert function is stopped.
<i>COMM_ERR</i>	100	The module is unavailable or the NIM is unavailable or the GCS (Group Cyclic Status) value is incorrect.

Index

A

<i>AbsoluteInvalid</i>	114
<i>ApplicationStopped</i>	114

B

<i>BOTH</i>	104
<i>BRAKING</i>	104
<i>Busy</i>	
management of status variables	17

C

<i>COMM_ERR</i>	103, 105, 117
<i>CommandedPosition</i>	116
<i>CommandedVelocity</i>	116
<i>Communication Error</i>	114
<i>CONF_NOT_SUPPORTED</i>	103
<i>CRC</i>	103

D

<i>Delay</i>	116
<i>DisabledAxis</i>	114
<i>Done</i>	
management of status variables	17

E

Edge I/O NTS Editor	74, 100
<i>EnableDirNeg</i>	116
<i>EnableDirPos</i>	116
<i>EnableSWLimitNeg</i>	116
<i>EnableSWLimitPos</i>	116
<i>ENC_NOT_ENABLED</i>	103
<i>Encoder_CAM_NTS</i>	19
<i>Encoder_Capture_NTS</i>	21
<i>ENCODER_ERROR_ID</i>	103
<i>Encoder_Main_NTS</i>	22
<i>Encoder_Preset_NTS</i>	24
<i>Encoder_Reflex_NTS</i>	25
<i>ENCODER_SUB_FCT_CAM_DIRECTION</i>	104
<i>ENCODER_SUB_FCT_CAM_TYPE</i>	104
<i>Error</i>	
handling a detected error	17
management of status variables	17
<i>ErrorId</i>	
handling a detected error	17
management of status variables	17
<i>ErrorStopActive</i>	114
<i>Execute</i>	
management of status variables	17

F

<i>Fallback</i>	114
<i>FALLBACK</i>	103, 105, 117
<i>FAST_IO_ERROR_ID</i>	105
<i>FastIO_React_NTS</i>	28
<i>FAULT</i>	103
<i>FORWARD</i>	104
<i>FRAME_FORMAT</i>	103

<i>FRAME_PARITY</i>	103
<i>FRAME_START</i>	103
<i>FRAME_STUCK_AT_1</i>	103
<i>FREQ_GEN_ERROR_ID</i>	117
<i>FreqGen_NTS</i>	30

H

handling a detected error	
<i>Error</i>	17
<i>ErrorId</i>	17
<i>HomingError</i>	114
<i>HSC_ABSOLUTE_LIMIT</i>	106
<i>HSC_CALIBRATION_FACTOR</i>	106
<i>HSC_CAPTURE_WINDOW_END_POSITION</i>	106
<i>HSC_CAPTURE_WINDOW_START_POSITION</i>	106
<i>HSC_COM_ERROR</i>	109
<i>HSC_GetParam_NTS</i>	32
<i>HSC_HYSTERESIS</i>	106
<i>HSC_INVALID_PARAMETER</i>	109
<i>HSC_Main_NTS</i>	33
<i>HSC_NO_ERROR</i>	108
<i>HSC_PDM_ERROR</i>	110
<i>HSC_PRESET</i>	106
<i>HSC_PWM_NTS</i>	39
commanding a pulse width modulation signal	39
<i>HSC_RESOLUTION</i>	106
<i>HSC_SetParam_NTS</i>	41
<i>HSC_SHORTWIRE_GROUP0_DETECTED</i>	109
<i>HSC_SHORTWIRE_GROUP1_DETECTED</i>	110
<i>HSC_Simple_NTS</i>	43
<i>HSC_THRESHOLD0</i>	106
<i>HSC_THRESHOLD1</i>	106
<i>HSC_THRESHOLD2</i>	106
<i>HSC_THRESHOLD3</i>	106
<i>HSC_TIMEBASE</i>	106
<i>HSC_UNKNOWN_PARAMETER</i>	108
<i>HSC_UNKNOWN_REF</i>	108
<i>HwPositionLimitN</i>	114
<i>HwPositionLimitP</i>	114

I

<i>INTERNAL_FIELD_POWER_SUPPLY</i>	103, 105, 114, 117
<i>InternalError</i>	114
<i>InvalidAccelerationValue</i>	115
<i>InvalidAxis</i>	114
<i>InvalidBufferModeValue</i>	115
<i>InvalidDecelerationValue</i>	115
<i>InvalidDirectionValue</i>	115
<i>InvalidHomeMode</i>	115
<i>InvalidHomingConf</i>	114
<i>InvalidParameter</i>	115
<i>InvalidParameterValue</i>	115
<i>InvalidPositionValue</i>	115
<i>InvalidProbeConf</i>	114
<i>InvalidTransition</i>	114
<i>InvalidVelocityValue</i>	115

L

<i>Logic Builder</i>	74, 100
<i>LongReference</i>	115
<i>LongReferenceAndIndex</i>	115

M		<i>PWM_ERROR_ID</i>	117
management of status variables		<i>PWM_NTS</i>	100
<i>Busy</i>	17	Q	
<i>Done</i>	17	<i>QUADRATURE</i>	103
<i>Error</i>	17	R	
<i>ErrorId</i>	17	<i>ReadOnlyParameter</i>	115
<i>Execute</i>	17	<i>REVERSE</i>	104
<i>MaxAccelerationAppl</i>	116	S	
<i>MaxDecelerationAppl</i>	116	<i>seBufferedDelay</i>	111
<i>MaxVelocityAppl</i>	116	<i>setPositionInvalid</i>	114
<i>MC_AbortTrigger_PTO_NTS</i>	46	<i>seTrigger</i>	111
<i>MC_BUFFER_MODE</i>	111	<i>SHORTCUT</i>	103, 105, 117
<i>MC_DIRECTION</i>	112	<i>ShortReference_NoReversal</i>	115
<i>MC_HALT_PTO_NTS</i>	75	<i>ShortReference_Reversal</i>	115
<i>MC_Home_PTO_NTS</i>	78	<i>ShortReferenceAndIndex_Inside</i>	115
<i>MC_MoveAbsolute_PTO_NTS</i>	81	<i>ShortReferenceAndIndex_Outside</i>	115
<i>MC_MoveRelative_PTO_NTS</i>	85	<i>SINCOS_SIGNAL</i>	103
<i>MC_MoveVelocity_PTO_NTS</i>	89	<i>SINCOS_SYNCHRO</i>	103
<i>MC_Power_PTO_NTS</i>	94	<i>StoppingActive</i>	114
<i>MC_ReadActualPosition_PTO_NTS</i>	47	<i>SWNegLimit</i>	116
<i>MC_ReadActualVelocity_PTO_NTS</i>	48	<i>SwPositionLimitN</i>	114
<i>MC_ReadAxisError_PTO_NTS</i>	48	<i>SwPositionLimitP</i>	114
<i>MC_ReadBoolParameter_PTO_NTS</i>	49	<i>SWPosLimit</i>	116
<i>MC_ReadMotionState_PTO_NTS</i>	50	T	
<i>MC_ReadParameter_PTO_NTS</i>	51	<i>TIMEOUT</i>	103
<i>MC_ReadStatus_PTO_NTS</i>	52	W	
<i>MC_Reset_PTO_NTS</i>	54	<i>WarningAccelerationValue</i>	114
<i>MC_SetPosition_PTO_NTS</i>	96	<i>WarningDecelerationValue</i>	114
<i>MC_Stop_PTO_NTS</i>	97	<i>WarningJerkValue</i>	114
<i>MC_TouchProbe_PTO_NTS</i>	55	<i>WarningVelocityValue</i>	114
<i>MC_WriteBoolParameter_PTO_NTS</i>	56		
<i>MC_WriteParameter_PTO_NTS</i>	57		
<i>mcAborting</i>	111		
<i>mcBlendingPrevious</i>	111		
<i>mcBuffered</i>	111		
<i>mcCurrentDirection</i>	112		
<i>mcNegativeDirection</i>	112		
<i>mcPositiveDirection</i>	112		
<i>MONOSTABLE</i>	104		
<i>MotionQueueFull</i>	114		
N			
<i>NoError</i>	114		
<i>NO_ERROR</i>	103, 105, 117		
<i>NOT_CONFIGURED</i>	103		
O			
<i>OutputProtection</i>	114		
P			
<i>POSITION</i>	104		
<i>PositionSetting</i>	115		
<i>POWER</i>	103		
<i>PROTOCOL</i>	103		
<i>PTO</i>			
configuration	74		
<i>PTO_ERROR_ID</i>	113		
<i>PTO_HOMING_MODE</i>	115		
<i>PTO_PARAMETER</i>	116		
pulse width modulation			
<i>HSC_PWM_NTS</i>	39		
PWM			
configuration	100		
<i>HSC_PWM_NTS</i>	39		

Schneider Electric
35 rue Joseph Monier
92500 Rueil Malmaison
France

www.se.com

As standards, specifications, and design change from time to time, please ask for confirmation of the information given in this publication.

© 2026 – Schneider Electric. All rights reserved.

EIO000005480.01