**PLCopen**

*for efficiency in automation*

## Technical Paper
## PLCopen Technical Committee

# Function Blocks for motion control:
# Part 2 - Extensions

**PLCopen Official Document, Version 1.0**

DISCLAIMER OF WARRANTIES

THIS DOCUMENT IS PROVIDED ON AN "AS IS" BASIS AND MAY BE SUBJECT TO FUTURE ADDITIONS, MODIFICATIONS, OR CORRECTIONS. PLCOPEN HEREBY DISCLAIMS ALL WARRANTIES OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING ANY WARRANTY OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE, FOR THIS DOCUMENT. IN NO EVENT WILL PLCOPEN BE RESPONSIBLE FOR ANY LOSS OR DAMAGE ARISING OUT OR RESULTING FROM ANY DEFECT, ERROR OR OMISSION IN THIS DOCUMENT OR FROM ANYONE'S USE OF OR RELIANCE ON THIS DOCUMENT.

Date: September 16, 2005

# Function blocks for motion control

This document is a specification as developed by the PLCopen Task Force Motion Control. As such it is an addition to the PLCopen Task Force Motion Control, Technical Document Version 1.0.

The present specification was written thanks to the following members of the PLCopen Task Force Motion Control:

| | |
|---|---|
| Istvan Ulvros | Tetrapak |
| Dieter Hess | 3S |
| Hilmar Panzer | 3S |
| Friedrich Forthuber | B & R Automation |
| Martin Schrott | B & R Automation |
| Bernd Gutknecht | Baumüller |
| Siegfried Schülein | Baumüller |
| Josef Papenfort | Beckhoff |
| Gunter Hertwig | Berger Lahr |
| Roland Schaumburg | Berger Lahr |
| Norbert Sasse | Bosch Rexroth |
| Wolfgang Czech | Bosch Rexroth |
| Alfred Moeltner | Elau |
| Ian Young | Giddings & Lewis |
| Steve Partridge | Giddings & Lewis |
| Wolfgang Brendel | Infoteam |
| Joachim Mayer | ISG |
| Ulrich Eger | ISG |
| Michael Garstenauer | KEBA |
| Thomas Prager | Keba |
| Bob Veltman | Klöckner Tevopharm |
| Fokke Groen | Klöckner Tevopharm |
| Artur Pilz | KUKA |
| Thomas Schmidt | KUKA |
| Candido Ferrio | Omron Yaskawa |
| Frank Durban | Parker Hannifin |
| Christian Ruf | Parker Hannifin |
| Klas Hellmann | Phoenix Contact |
| Paul Brooks | Rockwell Automation |
| Robert Günther | Rockwell Automation |
| Bernhard Lusch | SEW-Eurodrive |
| Hans-Peter Otto | Siemens |
| Willi Gagsteiger | Siemens |
| Eelco van der Wal | PLCopen |

## Change Status List:

| Version number | Date | Change comment |
|---|---|---|
| V 0.1 | September 21, 2000 | Preliminary version – result of the decision on the meeting of July 13 & 14, 2000, to create this extended set |
| V 0.2 | May 9 + 10, 2001 Meeting | MoveContinuous removed form 0.99 and put in here |
| V 0.3 | July 16, 2002 | Added the decisions as made at the meeting of June 11 and 12, 2002, Amsterdam. |
| V 0.4 | September 23 + 24, 2002 | Comments during meeting Sept. 23 and 24, 2002 |
| V 0.5 | December 9 + 10, 2002 | Comments during meeting Dec. 9 and 10, 2002 |
| V 0.6 | March 11 + 12, 2003 | Comments during meeting March 11 and 12, 2003 – not distributed |

| | | |
|---|---|---|
| V 0.7 | May 15, 2003 | Comments during meeting May 14 & 15, 2003 |
| V 0.7a | July 9, 2003 | Added homework in preparation of meeting July |
| V 0.8 | July 18, 2003 | Result of Meeting July |
| V 08A | September 4, 2003 | Split between Part2, 3 and 4. Added State Diagram |
| V 0.9 | September 24, 2003 | Result of the meeting Sept. 22 – 24, and editing by EvdWal |
| V 0.91 | November 21, 2003 | Result of feedback –edited by EvdW |
| V 0.92 | December 18, 2003 | Results of feedback on V 0.91 – edited by EvdW |
| V 0.93 | February 3 / March 22, 2004 | Generation of version for final meeting by EvdW |
| V 0.94 | March 30, 31, 2004 | Results of the meeting at Berger Lahr |
| V 0.95 | April 13, 2004 | Further editing by EvdW |
| V 0.99 | April 16, 2004 | Release for comments – by EvdW |
| V 0.99a | January 20, 2005 | Merge with comments – to be released as V. 1.0. MoveContinuous and MC_Halt added |
| V. 0.99B | Feb. 21, 2005 | In preparation for the meeting. Comments added. Results of meeting. |
| V. 0.99C | April 5, 2005 | Result of meeting April 4 + 5, 2005 at Bosch Rexroth |
| V. 099D | May 19, 2005 | Further editing done by EvdW, incl. State diagram |
| V. 0.99E | May 20, 2005 | Added pictures to torque control |
| V. 0.99F | May 25, 2005 | Behavior of MC_Halt added |
| V. 0.99G | June 7, 2005 | Language checking and corrections |
| V. 0.99H | July 20, 2005 | Final corrections. Final version before 1.0 |
| V. 1.0 | September 16, 2005 | Official Release |

## Table of Contents

## Table of Figures

## Table of Tables

# 1. General Introduction

At the end of 2001, PLCopen released the first release of the specification of an independent library of function blocks for motion control. It included motion functionality for single axes and multiple axes, several administrative tasks, as well as a state diagram. This specification provides the user with a standard command set and structure independent of the underlying architecture.
This structure can be used on many platforms and architectures. In this way one can decide which architecture will be used at a later stage of the development cycle. Advantages for the machine builder are, amongst others, lower costs for supporting the different platforms and the freedom to develop application software in a more independent way, without limiting the productivity of the machine. In addition to those benefits, system maintenance is easier and the education period is shorter. This is a major step forward, and is more and more accepted by users as well as suppliers.

With the release of part 1, it was understood that additional functionality was needed. Part 1 provides the basis for a set of inter-related specifications:

Part 1 - PLCopen Function Blocks for Motion Control
Part 2 - PLCopen Motion Control Extensions
Part 3 - PLCopen Motion Control User Guidelines
Part 4 – PLCopen Motion Control – Interpolation
Part 5 – PLCopen Motion Control - Homing Extensions.

The PLCopen Motion Control Extensions specification, Part 2, as well as the User Guidelines, Part 3, are additions to the PLCopen Function Blocks for Motion Control, and should not be seen as stand alone documents.

The objective of this specification “PLCopen Task Force Motion Control Extensions” is:

*To define a set of extensions to the Part 1 ‘PLCopen Function Blocks for Motion Control’ specification, which will serve the majority of users’ application needs.*

# 2. Overview of the defined extended Function Blocks:

| *Administrative* | | *Motion* | |
|---|---|---|---|
| ***Single Axis*** | ***Multiple Axis*** | ***Single Axis*** | ***Multiple Axis*** |
| MC_TouchProbe | | MC_TorqueControl | MC_GearInPos |
| MC_AbortTrigger | | MC_MoveContinuous | |
| MC_ReadDigitalInput | | MC_Halt | |
| MC_ReadDigitalOutput | | | |
| MC_WriteDigitalOutput | | | |
| MC_SetPosition | | | |
| MC_SetOverride | | | |
| MC_ReadActualVelocity | | | |
| MC_ReadActualTorque | | | |
| MC_DigitalCamSwitch | | | |

Table 1: Overview of the defined Function Blocks

## 2.1. General Remarks to the Function Block Behavior

### *2.1.1. Response Time*

The time taken by the system to respond to a command may vary widely by manufacturer / vendor / supplier and product.
For example in one implementation, writing a variable may result in commanding a request via a serial connection and waiting for a response, while another implementation might not have a serial connection, resulting is a faster response.

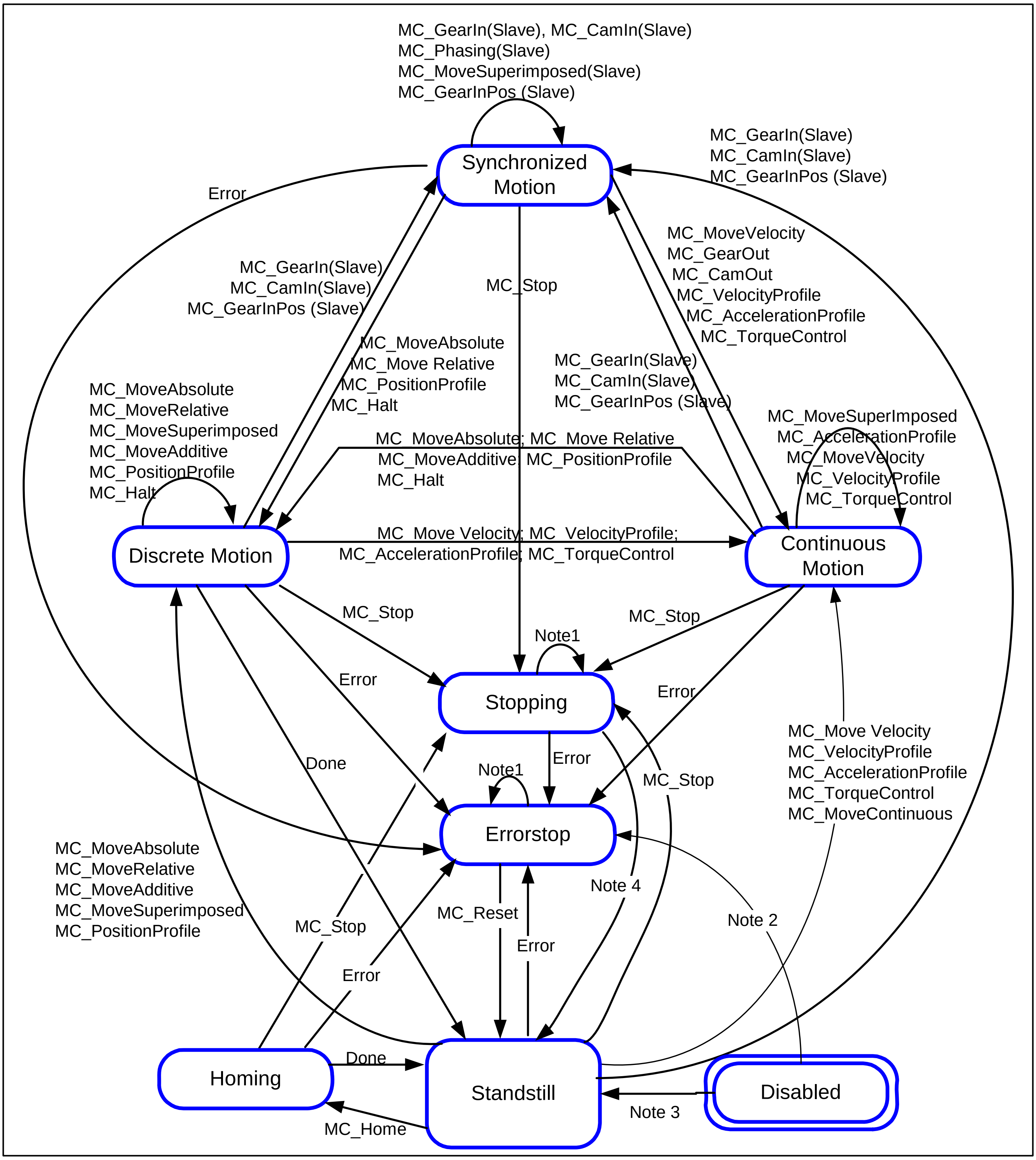
### *2.1.2. Buffered versus Non-buffered modes*

| **Function block** | **Can be specified as a buffered command** | **Can be followed by a buffered command** |
|---|---|---|
| MC_TorqueControl | Yes | Yes |
| MC_MoveContinuous | Yes | Yes |
| MC_Halt | Yes | Yes |
| MC_GearInPos | Yes | Yes |

Table 2: Overview of buffered versus non-buffered modes

## 3. State Diagram

The following state diagram is based on the version as defined in ‘Part 1 – Function Blocks for Motion Control’. This specification adds four Function Blocks: MC_TorqueControl, MC_MoveContinuous, MC_Halt, and MC_GearInPos.
Function Blocks not listed in the state diagram do not affect the State Diagram, meaning that whenever they are called the state does not change. They are: MC_TouchProbe, MC_AbortTrigger, MC_ReadDigitalInput, MC_ReadDigitalOutput, MC_WriteDigitalOutput, MC_SetPosition, MC_SetOverride, MC_ReadActualVelocity, MC_ReadActualTorque, and MC_DigitalCamSwitch.



Note 1: In the ErrorStop and Stopping states all Function Blocks can be called, although they will not be executed, except MC_Reset which will generate a transition to the Standstill state. If an error occurs while the state machine is in the Stopping state a transition to the ErrorStop state is generated.
Note 2: Power.Enable = TRUE and there is an error in the Axis.
Note 3: Power.Enable = TRUE and there is no error in the Axis.
Note 4: MC_Stop.Done AND NOT MC_Stop.Execute.

**Figure 1: State Diagram**

# 4. Defined User Derived Datatypes

The objective of this PLCopen Task Force Motion Control Extensions is defined in chapter 1 General Introduction.
To reach this objective, it is necessary to define additional reference types. These references are a representation of the 'objects' or devices, which are not necessarily a part of the process image.
As a general rule, these new reference datatypes are intended to be used in the same way as the AXIS_REF datatype, meaning that parameters can be read with similar Function Blocks having for instance an INPUT_REF instead of AXIS_REF and using the corresponding I/O parameters.
With the definition of these reference structures (or datatypes), there are Function Blocks defined which give access to the referenced data.

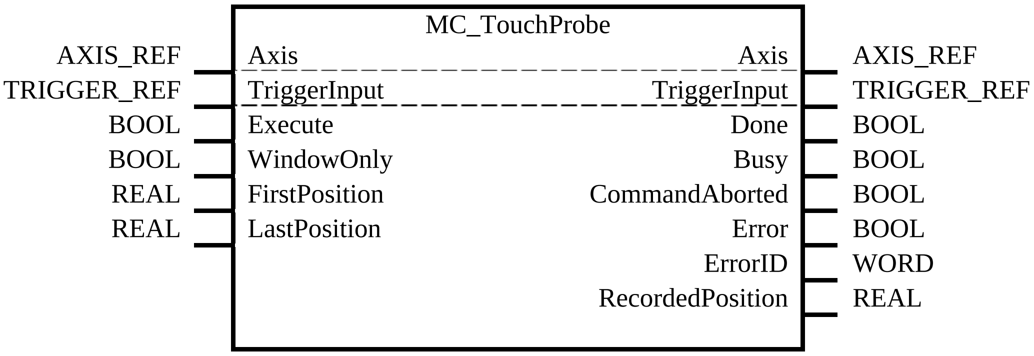
The following reference datatypes are defined within this document:

| **Defined datatype(s)** | **Relevant Function Block(s)** |
|---|---|
| **TRIGGER_REF** is a vendor specific data type that contains information about the trigger input, e.g.:<br>a. Location to the trigger source<br>b. Additional detection pattern information (positive, negative, both, edge, level, pattern recognition, etc) | MC_TouchProbe<br>MC_AbortTrigger |
| **INPUT_REF** is a vendor specific data type that contains a reference to a specific set of inputs, which may be virtual, meaning outside the declaration part. | MC_ReadDigitalInput |
| **OUTPUT_REF** is a vendor specific structure linked to the (physical) outputs | MC_DigitalCamSwitch<br>MC_ReadDigitalOutput<br>MC_WriteDigitalOutput |
| **CAMSWITCH_REF** is a vendor specific reference to the pattern data. | MC_DigitalCamSwitch |
| **TRACK_REF** is vendor specific structure containing information about a track, e.g. the compensations (A track is a set of switches related to one output). | MC_DigitalCamSwitch |

# 5. Function Blocks – Extensions for Motion Control

## 5.1. TouchProbe

| | | | |
|---|---|---|---|
| FB-Name | | **MC_TouchProbe** | |
| The function block is used to record an axis position at a trigger event | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | Identifies the axis for which the position should be recorded at a defined event at the trigger input |
| E | TriggerInput | TRIGGER_REF | Reference to the trigger signal source. Trigger input may be specified by the AXIS_REF. |
| VAR_INPUT | | | |
| B | Execute | BOOL | Starts touch probe recording at rising edge |
| E | WindowOnly | BOOL | If SET, only use the window (defined hereunder) to accept trigger events |
| E | FirstPosition | REAL | Start position from where (positive direction) trigger events are accepted (in technical units [u]). Value included in window. |
| E | LastPosition | REAL | Stop position of the window (in technical units [u]). Value included in window. |
| VAR_OUTPUT | | | |
| B | Done | BOOL | Trigger event recorded |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| E | CommandAborted | BOOL | Command is aborted by another command (MC_AbortTrigger) |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | WORD | Error identification |
| B | RecordedPosition | REAL | Position where trigger event occurred (in technical units [u]) |
| Note:<br>1. Intended for single shot operation, that is the first event after the rising edge at Execute is valid for recording only. Possible following events are ignored<br>2. One Function Block instance should represent exactly one probing command<br>3. In case of multiple instances on the same probe and axis, the elements of TRIGGER_REF should be extended with TouchProbeID - Identification of a unique probing command – this can be linked to MC_AbortTrigger | | | |



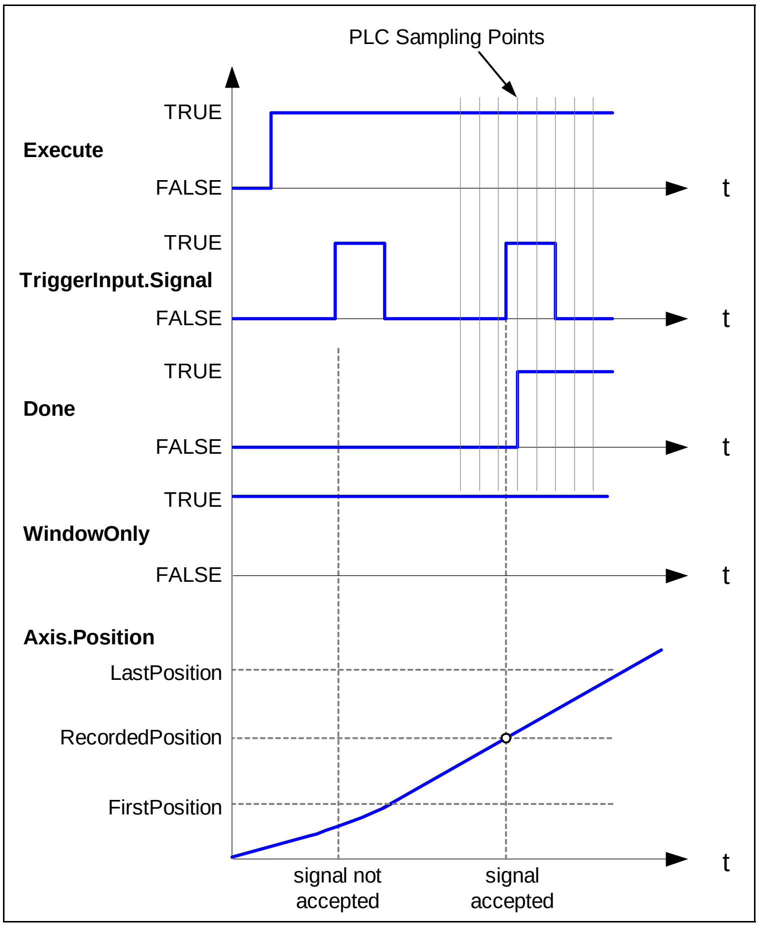


**Figure 2: Timing example MC_TouchProbe**

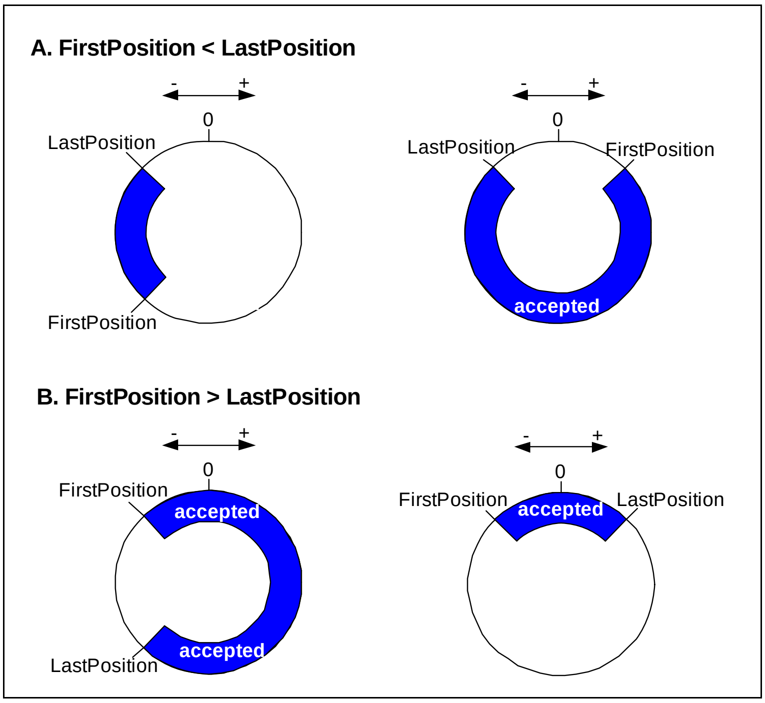
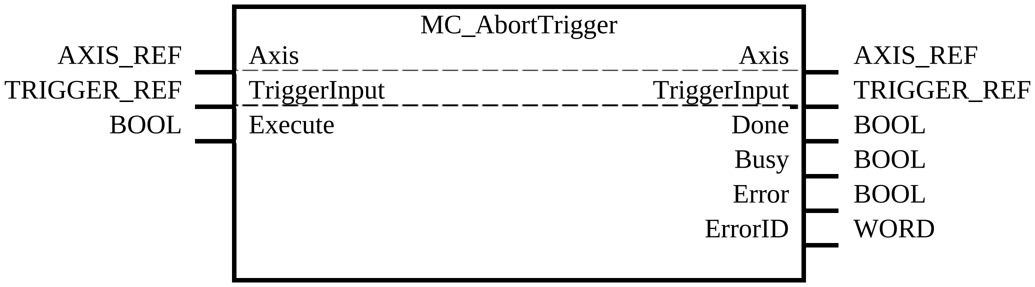


**Figure 3: Examples of windows, where trigger events are accepted (for modulo axes)**

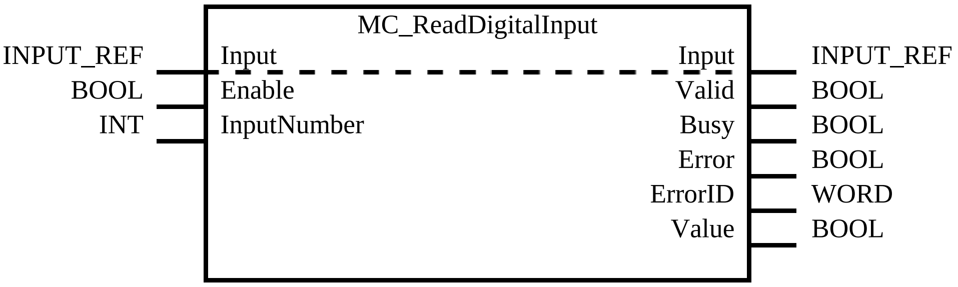
## 5.2. AbortTrigger

| | | | | |
|---|---|---|---|---|
| FB-Name | | | **MC_AbortTrigger** | |
| The function block is used to abort function blocks which are connected to trigger events (e.g. MC_TouchProbe) | | | | |
| VAR_IN_OUT | | | | |
| | B | Axis | AXIS_REF | Identifies the axis to which the trigger functionality is connected. |
| | E | TriggerInput | TRIGGER_REF | Reference to the trigger signal source. TriggerInput may be specified by the AXIS_REF. See Chapter 5.1 TouchProbe |
| VAR_INPUT | | | | |
| | B | Execute | BOOL | Aborts trigger event at rising edge |
| VAR_OUTPUT | | | | |
| | B | Done | BOOL | Trigger functionality aborted |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that an error has occurred within the Function Block |
| | E | ErrorID | WORD | Error identification |
| Notes: - | | | | |



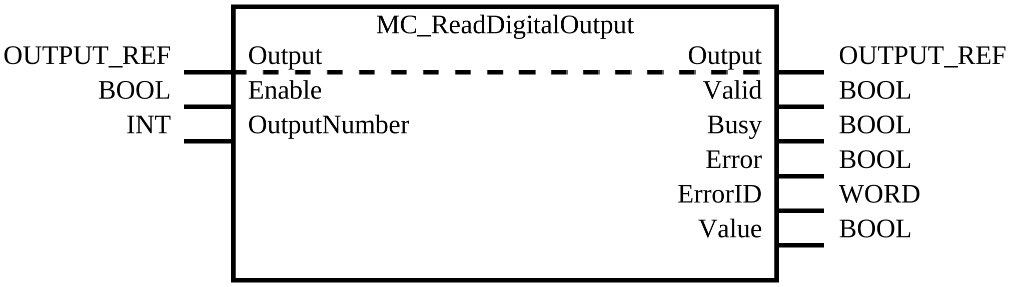
## 5.3. ReadDigitalInput

| FB-Name | | | **MC_ReadDigitalInput** | |
|---|---|---|---|---|
| This function block gives access to the value of the input, referenced by the datatype INPUT_REF. It provides the value of the referenced input (BOOL) | | | | |
| VAR_IN_OUT | | | | |
| | B | Input | INPUT_REF | Reference to the input signal source |
| VAR_INPUT | | | | |
| | B | Enable | BOOL | Get the value of the selected input signal continuously while enabled |
| | E | InputNumber | INT | Selects the input. Can be part of INPUT_REF, if only one single input is referenced. |
| VAR_OUTPUT | | | | |
| | B | Valid | BOOL | Input signal value is valid |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that an error has occurred within the Function Block |
| | E | ErrorID | WORD | Error Identification |
| | B | Value | BOOL | The value of the selected input signal |
| Note: It is not guaranteed that the digital signal will be seen by the FB: a short pulse on the digital input could be over before the next Function Block cycle occurs. | | | | |



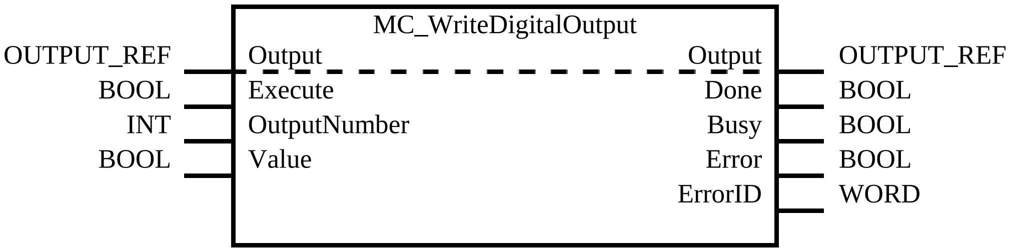
## 5.4. ReadDigitalOutput

| | | | | |
|---|---|---|---|---|
| FB-Name | | | **MC_ReadDigitalOutput** | |
| This Function Block provides access to the value of a digital output, referenced by the datatype OUTPUT_REF. It provides the value of the referenced output (BOOL). | | | | |
| VAR_IN_OUT | | | | |
| | B | Output | OUTPUT_REF | Reference to the signal outputs |
| VAR_INPUT | | | | |
| | B | Enable | BOOL | Get the value of the selected output signal continuously while enabled |
| | E | OutputNumber | INT | Selects the output. Can be part of OUTPUT_REF, if only one single output is referenced. |
| VAR_OUTPUT | | | | |
| | B | Valid | BOOL | Output signal value is valid |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that error has occurred within the Function Block |
| | E | ErrorID | WORD | Error Identification |
| | B | Value | BOOL | The value of the selected output signal |
| Note: It is not guaranteed that the digital signal will be seen by the FB: a short pulse on the digital output could be over before the next Function Block cycle occurs. | | | | |



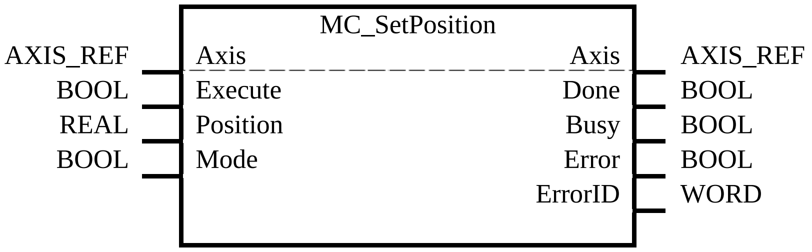
## 5.5. WriteDigitalOutput

| FB-Name | | | **MC_WriteDigitalOutput** | |
|---|---|---|---|---|
| This function block writes a value to the output referenced by the argument "Output" once (with rising edge of Execute). | | | | |
| VAR_IN_OUT | | | | |
| | B | Output | OUTPUT_REF | Reference to the signal output |
| VAR_INPUT | | | | |
| | B | Execute | BOOL | Write the value of the selected output |
| | E | OutputNumber | INT | Selects the output. Can be part of OUTPUT_REF, if only one single input is referenced. |
| | B | Value | BOOL | The value of the selected output |
| VAR_OUTPUT | | | | |
| | B | Done | BOOL | Writing of the output signal value is done |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that an error has occurred within the Function Block |
| | E | ErrorID | WORD | Error Identification |
| Notes: - | | | | |



## 5.6. SetPosition

| | | | | |
|---|---|---|---|---|
| FB-Name | **MC_SetPosition** | | | |
| This Function Block shifts the coordinate system of an axis by manipulating both the set-point position as well as the actual position of an axis with the same value without any movement caused. (Re-calibration with same following error). This can be used for instance for a reference situation. This Function Block can be used during motion without changing the commanded position, which is now positioned in the shifted coordinate system. | | | | |
| VAR_IN_OUT | | | | |
| | B | Axis | AXIS_REF | Identifies the axis to work upon |
| VAR_INPUT | | | | |
| | B | Execute | BOOL | Start setting position in axis |
| | B | Position | REAL | Position unit [u] (Means ‘Distance’ if Mode = RELATIVE) |
| | E | Mode | BOOL | RELATIVE =True, ABSOLUTE = False (Default) |
| VAR_OUTPUT | | | | |
| | B | Done | BOOL | Position has new value |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that an error has occurred within the Function Block |
| | E | ErrorID | WORD | Error identification |
| Note:<br>RELATIVE means that Position is added to the actual position value of the axis at the time of execution. This results in a recalibration by a specified distance. ABSOLUTE means that the actual position value of the axis is set to the value specified in the Position parameter. | | | | |

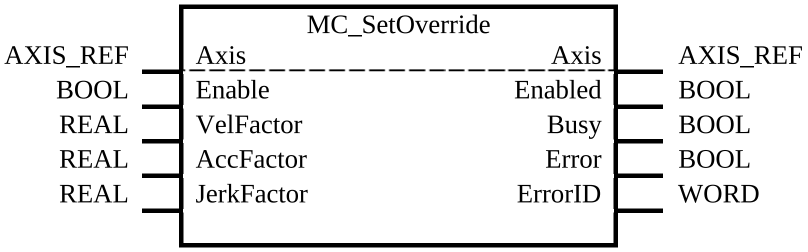


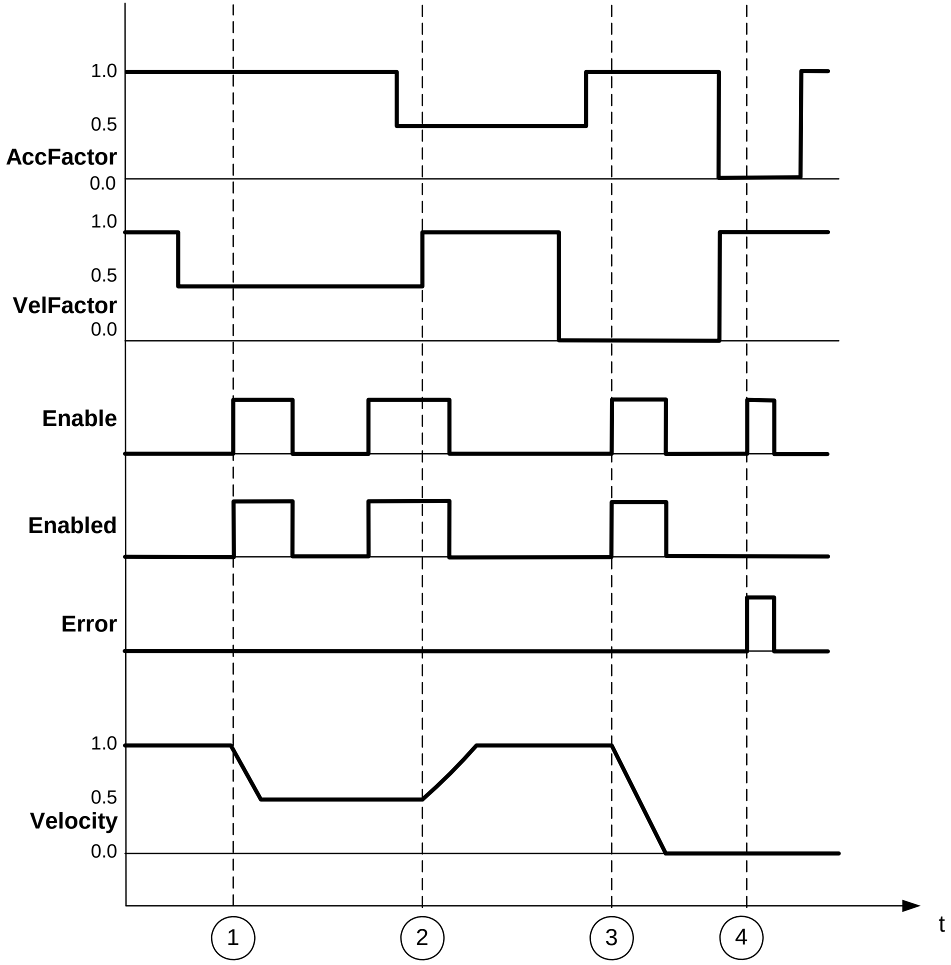
## 5.7. SetOverride

| FB-Name | **MC_SetOverride** | | |
|---|---|---|---|
| This function block sets the values of override for the whole axis, and all functions that are working on that axis. The override parameters act as a factor that is multiplied to the commanded velocity, acceleration, deceleration and jerk of the move function block. | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | Identifies the axis to work upon |
| VAR_INPUT | | | |
| B | Enable | BOOL | If SET, it writes the value of the override factor continuously. If RESET it should keep the last value. |
| B | VelFactor | REAL | New override factor for the velocity |
| E | AccFactor | REAL | New override factor for the acceleration/deceleration |
| E | JerkFactor | REAL | New override factor for the jerk |
| VAR_OUTPUT | | | |
| B | Enabled | BOOL | Signals that the override factor(s) is (are) set successfully |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | WORD | Error identification |

Notes:

1. The Input AccFactor acts on positive and negative acceleration (deceleration).
2. This Function Block sets the factor. The override factor is valid until a new override is set.
3. The default values of the override factor are 1.0.
4. The value of the overrides can be between 0.0 and 1.0. The behavior of values > 1.0 is vendor specific. Values < 0.0 are not allowed. The value 0.0 is not allowed for AccFactor and JerkFactor.
5. The value 0.0 set to the VelFactor stops the axis without bringing it to the state standstill.
6. Override does not act on slave axes. (Axes in the state synchronized motion).
7. The Function Block does not influence the state diagram of the axis.
8. VelFactor can be changed at any time and acts directly on the ongoing motion.
9. If in Discrete motion, reducing the AccFactor and/or JerkFactor can lead to a position overshoot – a possible cause of damage
10. Activating this Function Block on an axis that is under control of MC_PositionProfile, MC_VelocityProfile, or MC_AccelerationProfile, is vendor specific.





1 Axis Velocity changes to 50% with 100% of deceleration

2 Axis Velocity changes back to 100% with 50% acceleration

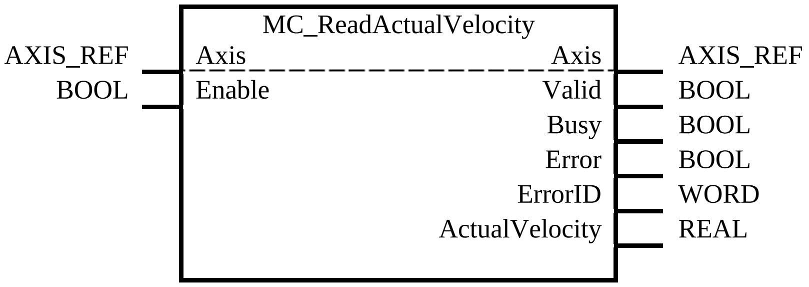
3 Axis Velocity moves to 0% with 100% deceleration

4 No Change, because AccFactor 0.0 is not allowed; Error is set

**Figure 3: Graphical explanation of MC_SetOverride**

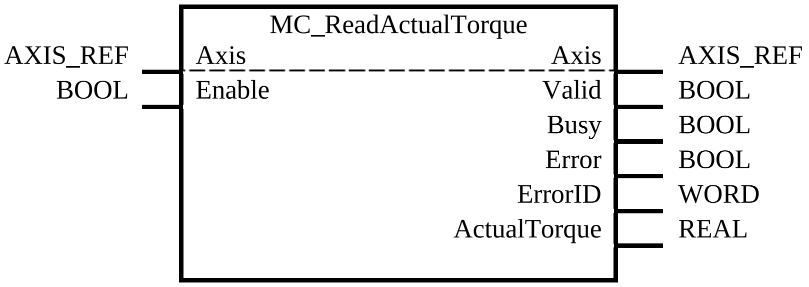
## 5.8. ReadActualVelocity

| | | | | |
|---|---|---|---|---|
| FB-Name | | | **MC_ReadActualVelocity** | |
| This Function Block returns the value of the actual velocity as long as Enable is set. Done is true when the data-output “Velocity” is valid. If Enable is Reset, the data loses its validity, and Done is also reset, no matter if new data is available. | | | | |
| VAR_IN_OUT | | | | |
| | B | Axis | AXIS_REF | Identifies the axis to work upon |
| VAR_INPUT | | | | |
| | B | Enable | BOOL | Get the value of the parameter continuously while enabled |
| VAR_OUTPUT | | | | |
| | B | Valid | BOOL | Valid value is available |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that an error has occurred within the Function Block |
| | E | ErrorID | WORD | Error identification |
| | B | ActualVelocity | REAL | The value of the actual velocity (in axis’ unit [u/s]) |
| Notes: The output ActualVelocity can be a signed value | | | | |



## 5.9. ReadActualTorque

| | | | | |
|---|---|---|---|---|
| FB-Name | | | **MC_ReadActualTorque** | |
| This Function Block returns the value of the actual torque or force as long as Enable is set. Done is true when the data-output “Torque” is valid. If Enable is Reset, the data loses its validity, and Done is also reset, no matter if new data is available. | | | | |
| VAR_IN_OUT | | | | |
| | B | Axis | AXIS_REF | Identifies the axis to work upon |
| VAR_INPUT | | | | |
| | B | Enable | BOOL | Get the value of the parameter continuously while enabled |
| VAR_OUTPUT | | | | |
| | B | Valid | BOOL | Valid value is available |
| | E | Busy | BOOL | Shows that the Function Block is not finished |
| | B | Error | BOOL | Signals that an error has occurred within the Function Block |
| | E | ErrorID | WORD | Error identification |
| | B | ActualTorque | REAL | The value of the actual torque or force (in technical units) |
| Notes: The output ActualTorque can be a signed value | | | | |

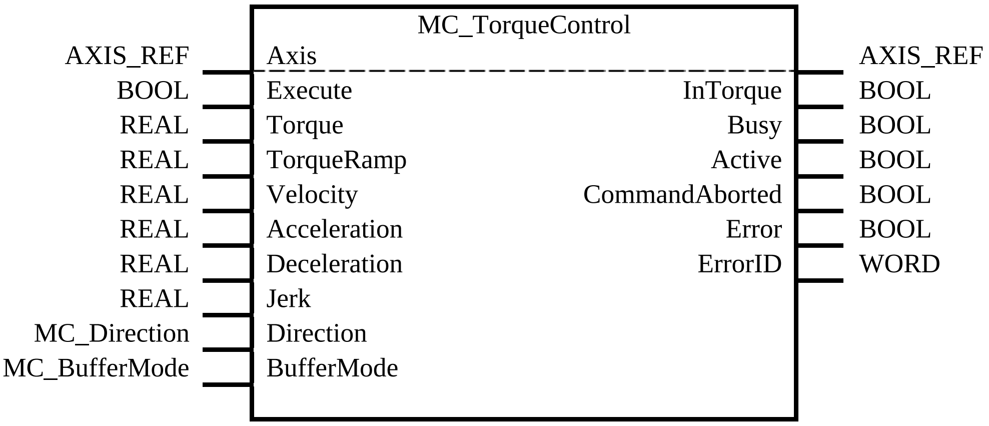


## 5.10. TorqueControl

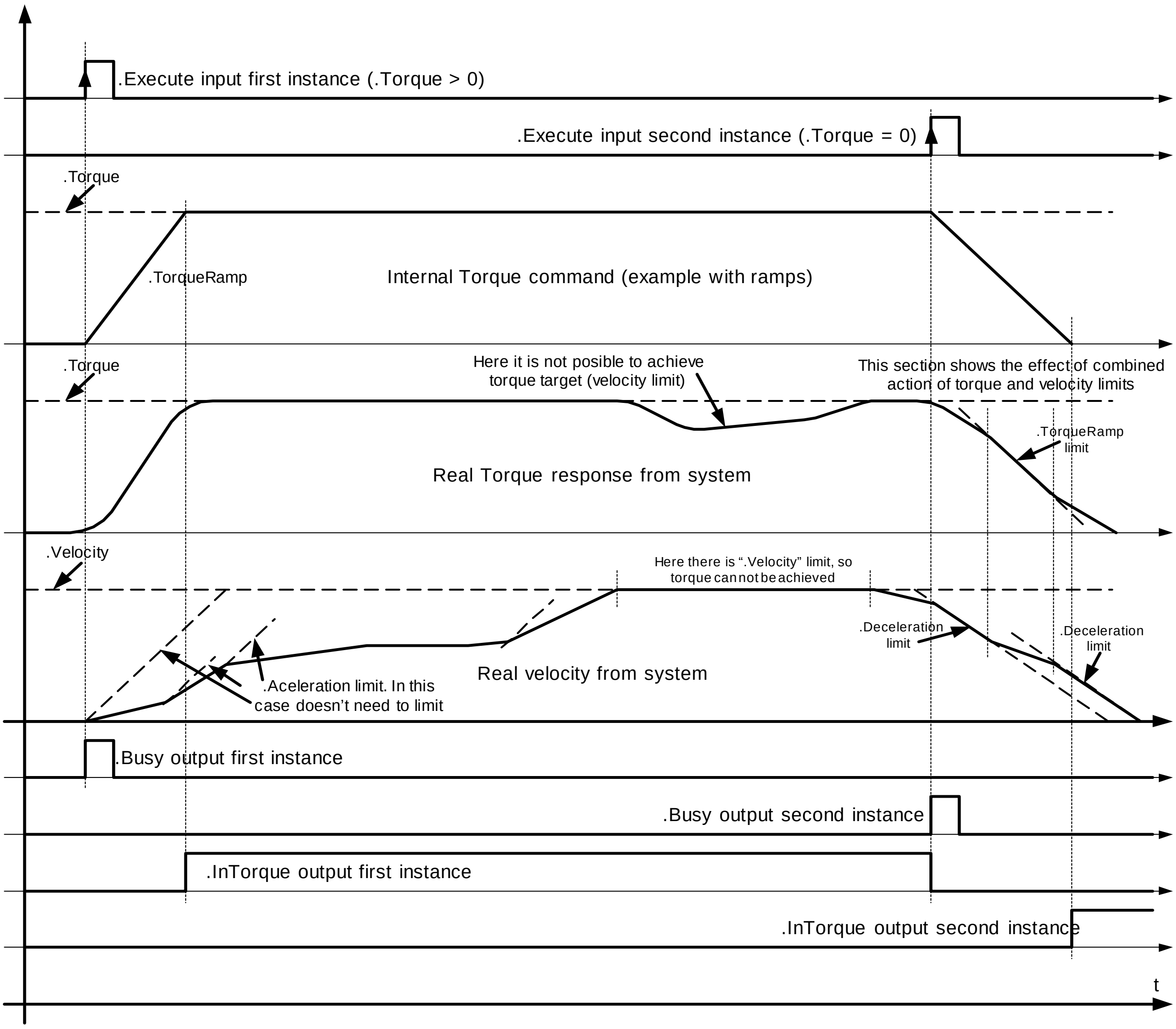
| | | | |
|---|---|---|---|
| FB-Name | **MC_TorqueControl** | | |
| This function block continuously exerts a torque or force of the specified magnitude. This magnitude is approached using a defined ramp (TorqueRamp), and the Function Block sets the InTorque output if the commanded torque level is reached. This function block is applicable for force and torque. When there is no external load, force is applicable. Positive torque is in the positive direction of velocity. | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | Identifies the axis to work upon |
| VAR_INPUT | | | |
| B | Execute | BOOL | Starts on the axis using a rising edge |
| B | Torque | REAL | Value of the torque (Torque or force in t.u.) |
| E | TorqueRamp | REAL | The maximum time derivative of the set value of the torque or force (in t.u. per sec) |
| E | Velocity | REAL | Absolute value of the maximum velocity. |
| E | Acceleration | REAL | Value of the maximum acceleration (acceleration is applicable with same sign of torque and velocity) |
| E | Deceleration | REAL | Value of the maximum deceleration (deceleration is applicable with opposite signs of torque and velocity) |
| E | Jerk | REAL | Value of the maximum jerk |
| E | Direction | MC_Direction | Enum type (1 of 2 values: positive or negative direction.) Note: current direction and shortest way not applicable. |
| E | BufferMode | MC_BufferMode | Defines the behavior of the axis: modes are Aborting, Buffered, Blending |
| VAR_OUTPUT | | | |
| B | InTorque | BOOL | Setpoint value of torque or force is reached for the first time |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| E | Active | BOOL | Indicates that the Function Block has control on the axis |
| E | CommandAborted | BOOL | Command is aborted by another command |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | WORD | Error identification |

Notes:

1. The movement is limited by velocity, acceleration / deceleration, and jerk, or by the value of the torque, depending on the mechanical circumstances.
2. Specific additional tests are outside this FB. For instance, checking on the traveled distance could be done via tracing the actual positions during the action.
3. Velocity is always a positive value. The direction is dependent on the torque and load.
4. The axis ceases to be in torque control mode when any motion control (not administrative) Function Block is accepted on the same axis.



The example below shows the typical behavior of an intermediate “resistive” load (see .Deceleration limit) with some “inertia” (see .TorqueRamp limit).


.Execute input first instance (.Torque > 0)
.Execute input second instance (.Torque = 0)
.Torque
.TorqueRamp
Internal Torque command (example with ramps)
.Torque
Here it is not posible to achieve torque target (velocity limit)
This section shows the effect of combined action of torque and velocity limits
.TorqueRamp limit
Real Torque response from system
.Velocity
Here there is ".Velocity" limit, so torque cannot be achieved
.Deceleration limit
.Deceleration limit
.Aceleration limit. In this case doesn't need to limit
Real velocity from system
.Busy output first instance
.Busy output second instance
.InTorque output first instance
.InTorque output second instance
t

This example could be implemented in a Function Block Diagram like:

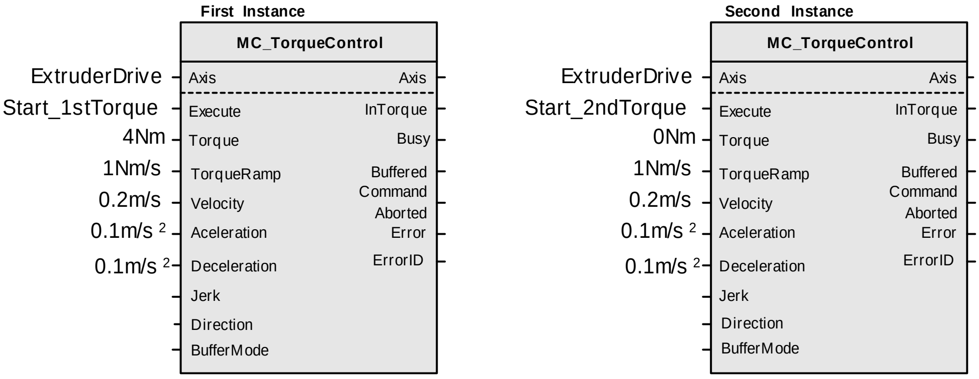


**Figure 4: Example of Torque Control**

With the second example we use opposite signs for Direction & Torque (e.g. Retention or brake control). (In the FB: +Direction –Torque). It is like an unwinding application with torque on the material, and a break in the material.
When the material breaks, as shown in the middle of the picture, this causes a drop in the Real Torque (in absolute terms): the velocity will decrease, limited by the fastest "deceleration" limit specified by the "Deceleration" VAR_INPUT down to zero velocity (with no tension there is a risk of having shock breakings, so we have to limit to the fastest). In this case the torque setpoint might not be achieved.

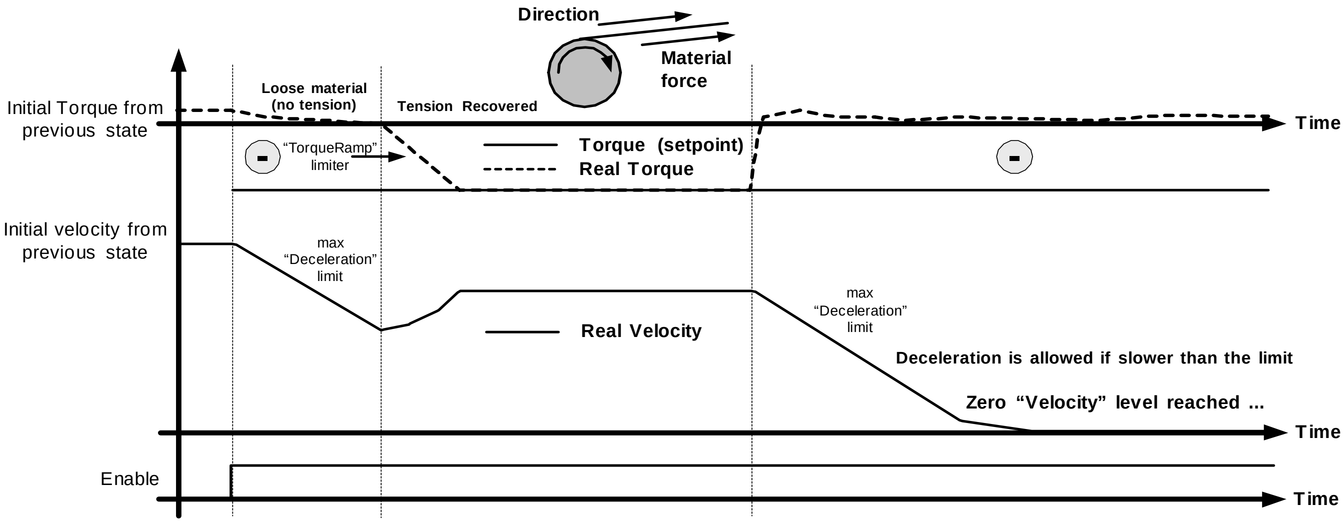
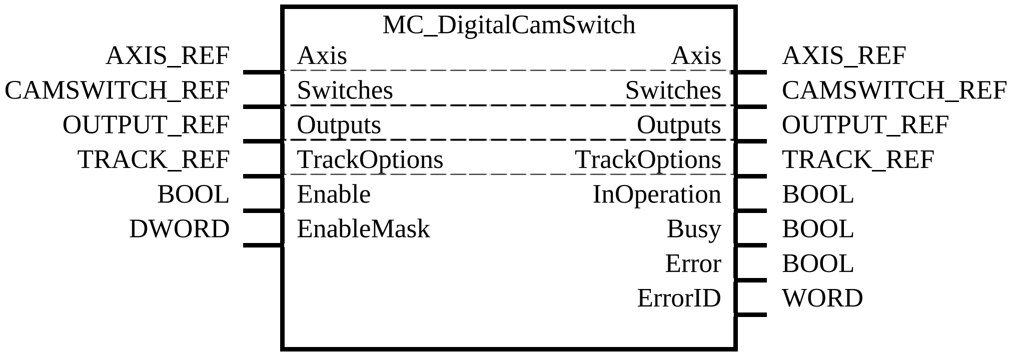


**Figure 5: Second example of Torque Control**

NOTE: In an unwinding application (derived from this brake control) material tension is the target, not motor torque. The instantaneous diameter of the roll should be taken into account to transform the "User tension setpoint". Also additional inertia compensation by modification of the torque setpoint for acceleration / deceleration is common from instantaneous weight data (weight is commonly estimated from diameter). Additionally in unwinding applications, in the case of loose material (same condition as material break), a negative slow velocity reference is usually applied in order to "rewind" the loose material. In this case, this has to be provided by external programming.

## 5.11. DigitalCamSwitch

| FB-Name | **MC_DigitalCamSwitch** | | |
|---|---|---|---|
| This function block is the analogy to switches on a motor shaft: it commands a group of discrete output bits to switch in analogy to a set of mechanical cam controlled switches connected to an axis. Forward and backward movements are allowed. | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | Reference to the axis to which the switches are connected to |
| B | Switches | CAMSWITCH_REF | Reference to the switching actions. |
| E | Outputs | OUTPUT_REF | Reference to the signal outputs, directly related to the referenced tracks. (max. 32 per function block) (First output = first TrackNumber) |
| E | TrackOptions | TRACK_REF | Reference to structure containing track related properties, e.g. the ON and OFF compensations per output/track. |
| VAR_INPUT | | | |
| B | Enable | BOOL | Enables the Switches outputs |
| E | EnableMask | DWORD | 32 bits of BOOL. Enables the different tracks. Least significant data is related to the lowest TrackNumber. With data SET (to ‘1’ resp. TRUE) the related TrackNumber is enabled. |
| VAR_OUTPUT | | | |
| B | InOperation | BOOL | The commanded tracks are enabled |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | WORD | Error Identification |
| Notes:<br>• CAMSWITCH_REF is a vendor specific reference to the pattern data.<br>• OUTPUT_REF is a vendor specific structure linked to the (physical) outputs<br>• TRACK_REF is vendor specific structure containing the track properties, e.g. the compensation per track (A track is a set of switches related to one output). It can contain the reference to the output also.<br>• This functionality is sometimes called PLS – Phase or Position or Programmable Limit Switch | | | |



Basic elements within the structure of CAMSWITCH_REF

| B/E | Parameter | Type | Description |
|---|---|---|---|
| B | TrackNumber | INT | TrackNumber is the reference to the track |
| B | FirstOnPosition [u] | REAL | Lower boundary where the switch is ON |
| B | LastOnPosition [u] | REAL | Upper boundary where the switch is ON |
| E | AxisDirection | INT | Both (=0; Default); Positive (1); Negative (2) |
| E | CamSwitchMode | INT | Position based (=0; Default); Time based (=1) |
| E | Duration | TIME | Coupled to time based CamSwitchMode |

Basic elements within the array structure of TRACK_REF

| B/E | Parameter | Type | Description |
|---|---|---|---|
| E | OnCompensation | TIME | Compensation time with which the switching on is advanced or delayed in time per track. |
| E | OffCompensation | TIME | Time compensation the switching off is delayed per track. |
| E | Hysteresis [u] | REAL | Distance from the switching point (in positive and negative direction) in which the switch is not executed until the axis has left this area, in order to avoid multiple switching around the switching point. |

This definition of a cam has a start and an end position, so the user can define each single cam, which has a **FirstOnPosition** and a **LastOnPosition** (or time). This Function Block is similar to a mechanical cam but has the additional advantages that the outputs can be set for a certain time, and to give it a time compensation and a hysteresis. If (FirstOnPosition > LastOnPosition) it gives an inverse cam switch, which is off only within these positions.

**CamSwitchMode** can be Position, Time or other additional vendor specific types.

**Duration**: Time, the output of a time cam is ON

The time compensation (**OnCompensation** and **OffCompensation**) can be positive or negative. Negative means the output changes before the switching position is reached.

**Hysteresis**: This parameter avoids the phenomenon where the output continually switches if the axis is near the switching point and the actual position is jittering around the switching position. Hysteresis is part of TRACK_REF, which means that a different hysteresis is possible for each track.

Example of CAMSWITCHREF

| **Parameter** | **Type** | **Switch01** | **Switch02** | **Switch03** | **Switch04** | **…** | **SwitchN** |
|---|---|---|---|---|---|---|---|
| TrackNumber | INTEGER | *1* | *1* | *1* | *2* | | |
| FirstOnPosition [u] | REAL | *2000* | *2500* | *4000* | *3000* | | |
| LastOnPosition [u] | REAL | *3000* | *3000* | *1000* | *--* | | |
| AxisDirection | INTEGER | *1=Pos* | *2=Neg* | *0=Both* | *0=Both* | | |
| CamSwitchMode | INTEGER | *0=Position* | *0=Position* | *0=Position* | *1=Time* | | |
| Duration | TIME | -- | -- | -- | *1350* | | |

*Note: Values are Examples*

The example below uses the values from the example for CAMSWITCH_REF above. It uses neither On/OffCompensation, nor hysteresis.
This is the behavior of the outputs, when the axis is moving continuously in the positive direction. The axis is a modulo axis with a modulo length of 5000 u.

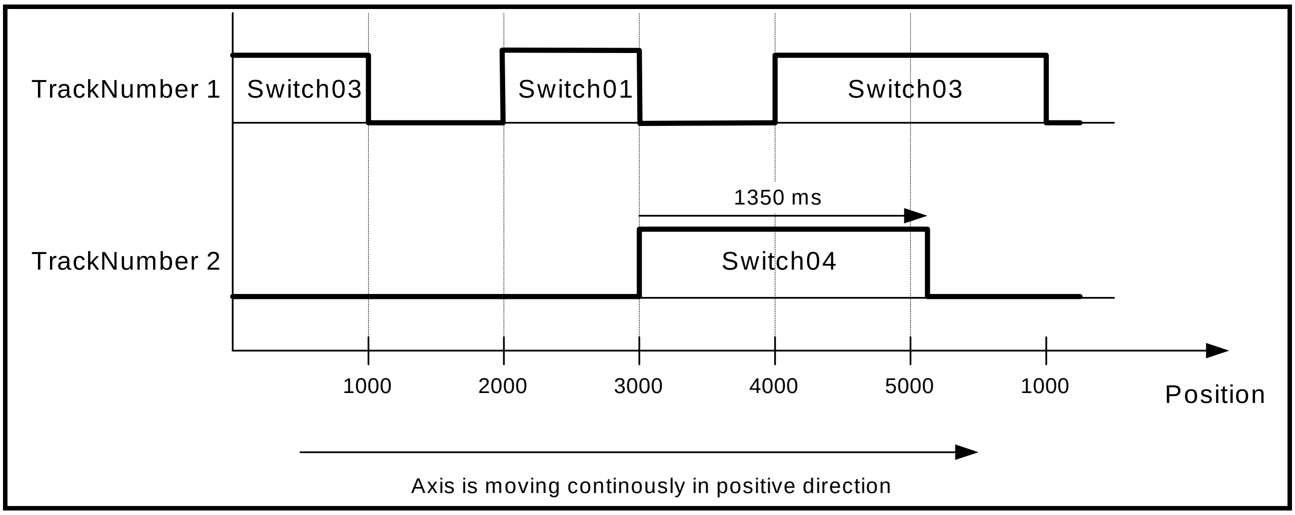


**Figure 6: Example of DigitalCamSwitch**

**Detailed description of Switch01.**
This example additionally uses OnCompensation -125ms and OffCompensation +250ms.

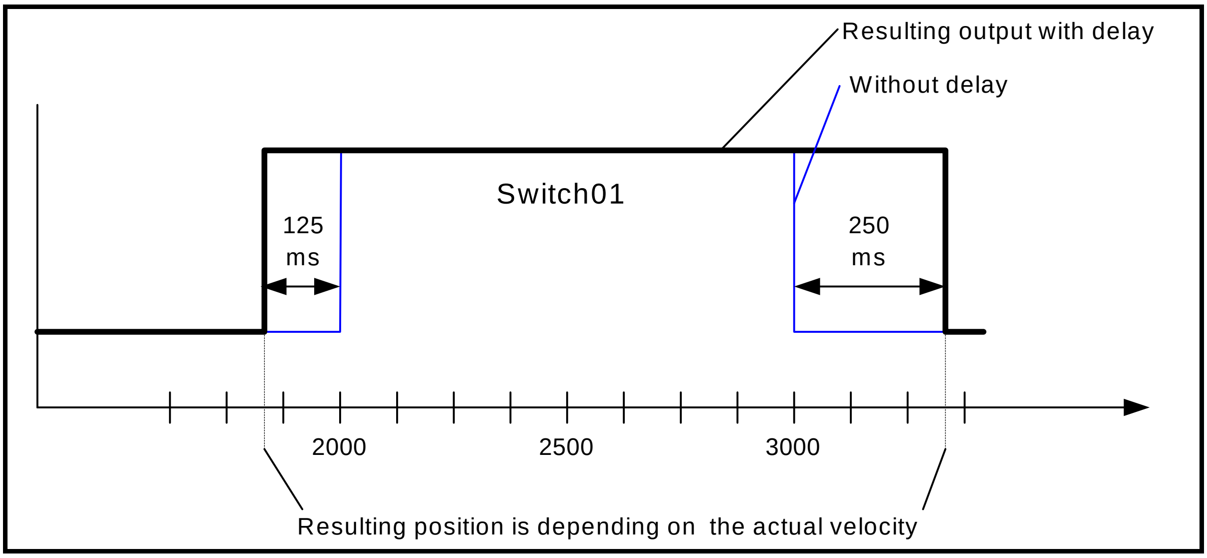

Resulting output with delay
Without delay
Switch01
125
ms
250
ms
2000
2500
3000
Resulting position is depending on the actual velocity

**Figure 7: Detailed description of Switch01.**

Below the behavior of the outputs, when the axis is moving continuously in the negative direction without On/OffCompensation and without Hysteresis.

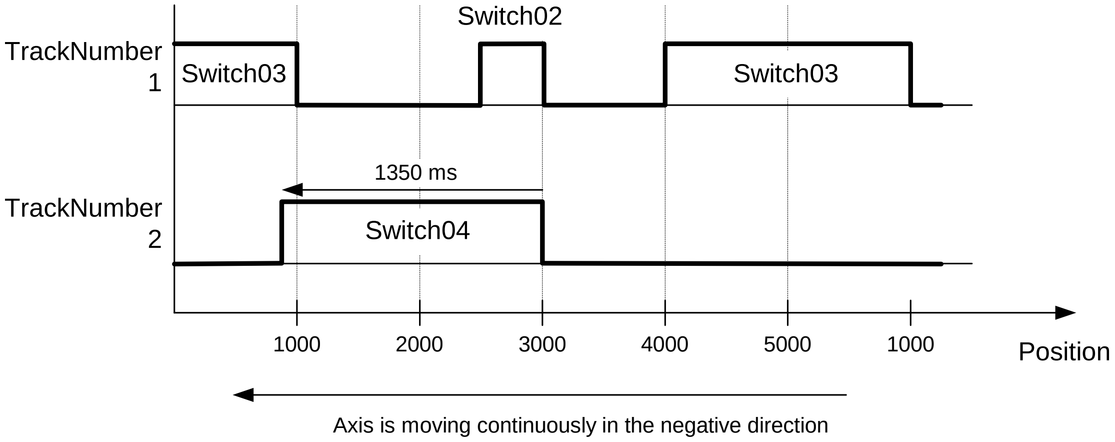


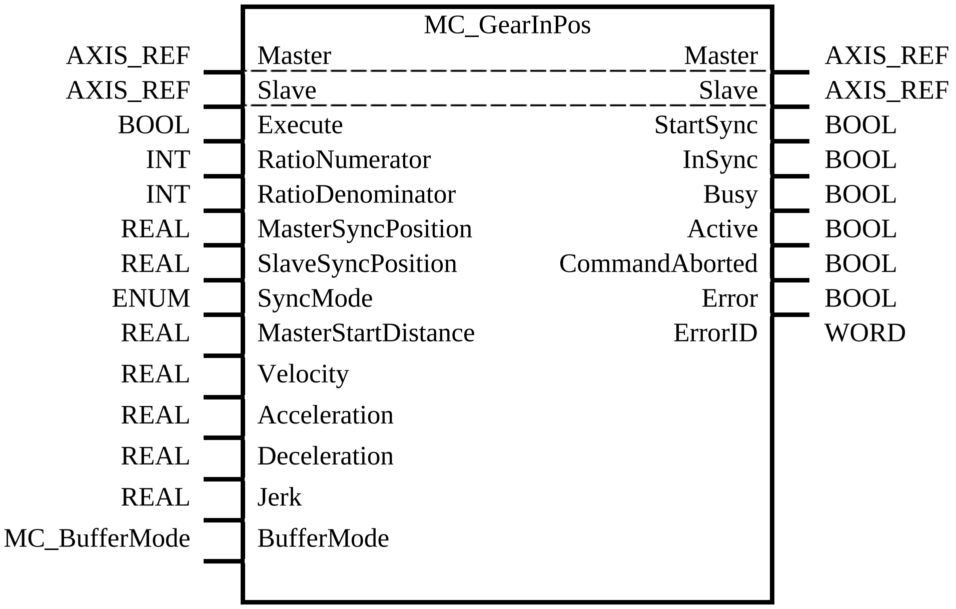
**Figure 8: Example in negative direction**

## 5.12. GearInPos

| | | | |
|---|---|---|---|
| FB-Name | **MC_GearInPos** | | |
| This Function Block commands a gear ratio between the position of the slave and master axes from the synchronization point onwards. | | | |
| VAR_IN_OUT | | | |
| B | Master | AXIS_REF | Reference to master axis |
| B | Slave | AXIS_REF | Reference to slave axis |
| VAR_INPUT | | | |
| B | Execute | BOOL | Start the gearing process at the rising edge |
| B | RatioNumerator | INT | Gear ratio Numerator |
| B | RatioDenominator | INT | Gear ratio Denominator |
| B | MasterSyncPosition | REAL | Master Position at which the axes are running in sync |
| B | SlaveSyncPosition | REAL | Slave Position at which the axes are running in sync |
| E | SyncMode | ENUM | Defines the way to synchronize (like ‘Shortest_Way’; ‘Catch_Up’; ‘Slow_Down’). Vendor specific |
| E | MasterStartDistance | REAL | Master Distance for gear in procedure (when the Slave axis is started to get into synchronization) |
| E | Velocity | REAL | Maximum Velocity during the time difference StartSync and InSync |
| E | Acceleration | REAL | Maximum Acceleration during the time difference StartSync and InSync |
| E | Deceleration | REAL | Maximum Deceleration during the time difference StartSync and InSync |
| E | Jerk | REAL | Maximum Jerk during the time difference StartSync and InSync |
| E | BufferMode | MC_Buffer Mode | Defines the behavior of the axis: modes are Aborting, Buffered, Blending |
| VAR_OUTPUT | | | |
| E | StartSync | BOOL | Commanded gearing starts |
| B | InSync | BOOL | Commanded gearing completed |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| E | Active | BOOL | Indicates that the Function Block has control on the axis |
| B | CommandAborted | BOOL | Command is aborted by another command |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | WORD | Error identification |

Notes:

1. If MasterStartDistance is implemented, any previous motion is continued until master crosses "MasterSyncPosition – MasterStartDistance" in the correct direction (according to the sign of MasterStartDistance). At that point in time the output StartSync is set. When a "Stop" command is executed on the "Slave" axis before the synchronization has happened, it inhibits the synchronization and the function block issues "CommandAborted"
2. If the MasterStartDistance is not specified, the system itself could calculate the set point for StartSync based on the other relevant inputs.



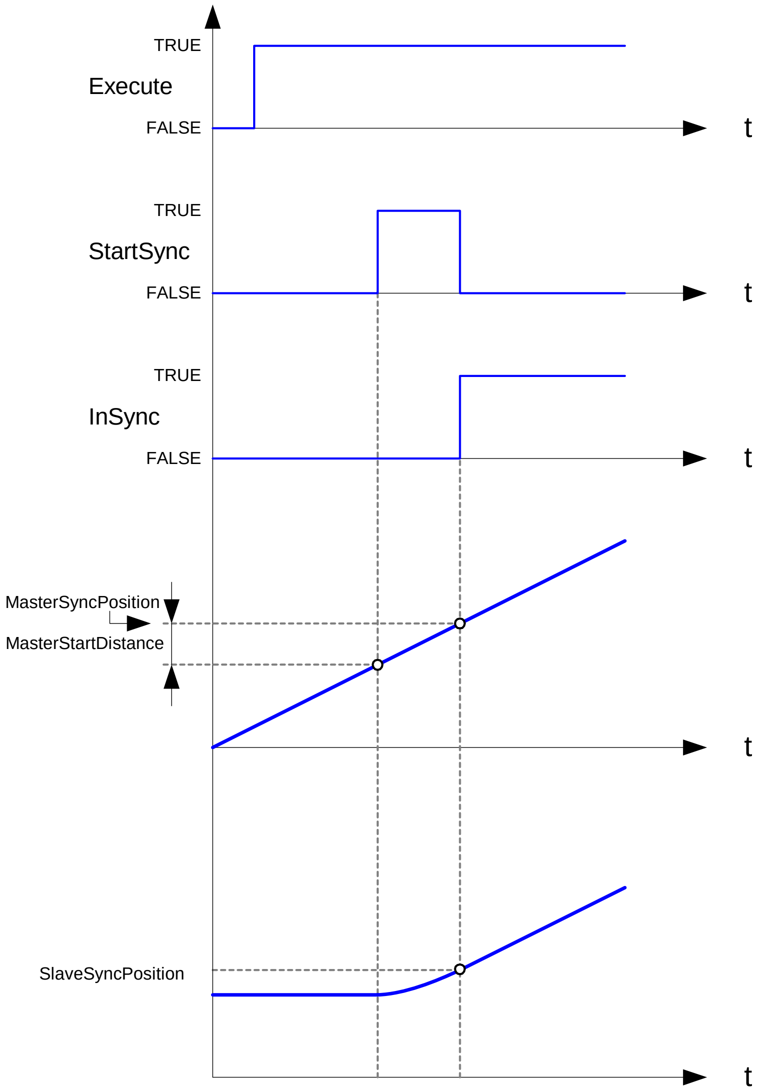


**Figure 9: Timing Diagram of MC_GearInPos**

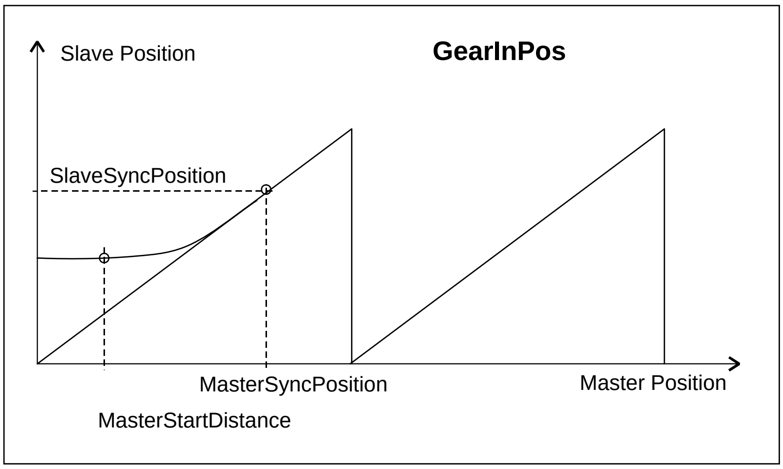


Figure 10.1

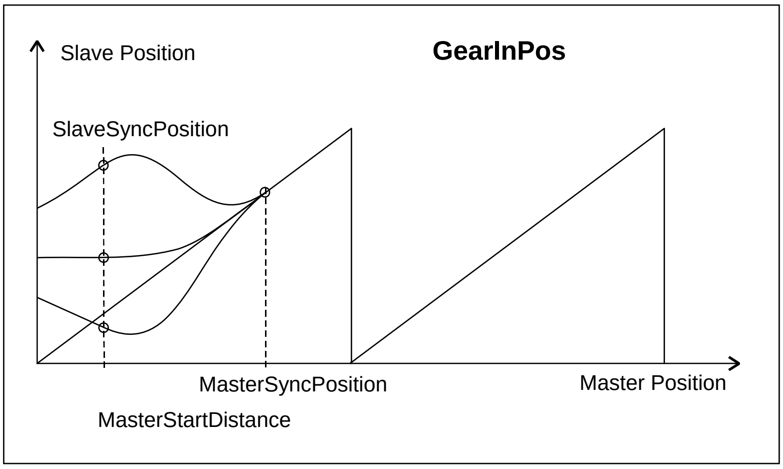


Figure 10.2

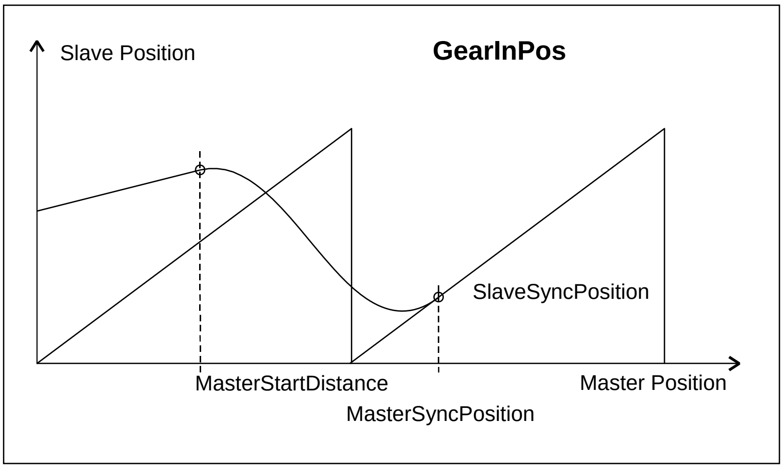
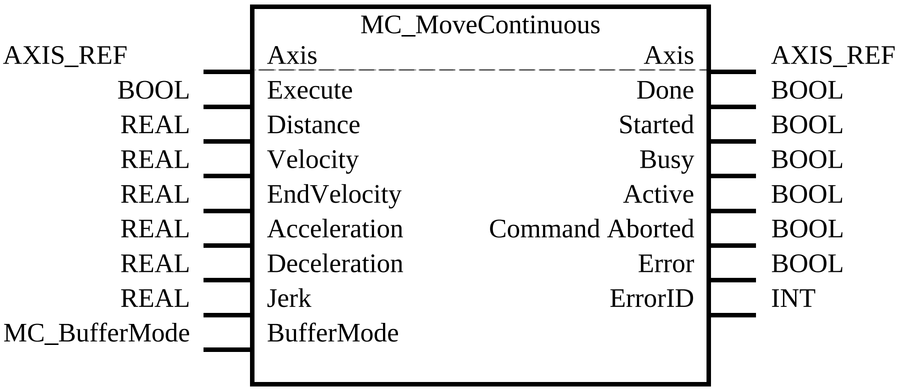


Figure 10.3

**Figure 10: Different examples of MC_GearInPos**

## 5.13. MoveContinuous

| | | | |
|---|---|---|---|
| FB-Name | | MC_MoveContinuous | |
| This function block commands a controlled motion of a specified relative distance ending with the specified velocity. | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | |
| VAR_INPUT | | | |
| B | Execute | BOOL | Start the motion at rising edge |
| B | Distance | REAL | Relative distance for the motion [u] |
| B | Velocity | REAL | Value of the maximum velocity [u/s] |
| B | EndVelocity | REAL | Value of the end velocity [u/s] |
| E | Acceleration | REAL | Value of the acceleration [u/$s^2$] |
| E | Deceleration | REAL | Value of the deceleration [u/$s^2$] |
| E | Jerk | REAL | Value of the Jerk [u/$s^3$] |
| E | BufferMode | MC_Buffer Mode | Defines the behavior of the axis: modes are Aborting, Buffered, Blending |
| VAR_OUTPUT | | | |
| B | InEndVelocity | BOOL | Commanded distance reached and running at requested end velocity |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| E | Active | BOOL | Indicates that the Function Block has control on the axis |
| B | CommandAborted | BOOL | Command aborted |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | INT | Error number |
| Notes:<br>• If the target is reached and no new motion command is put into the buffer, the axis continues to run with the specified ‘EndVelocity’.<br>• This Function Block places the axis into the ”Continuous Motion” state, even during the initial motion by a relative distance, and remains in the “Continuous Motion” state once the EndVelocity has been reached. | | | |



Example of MC_MoveContinuous:

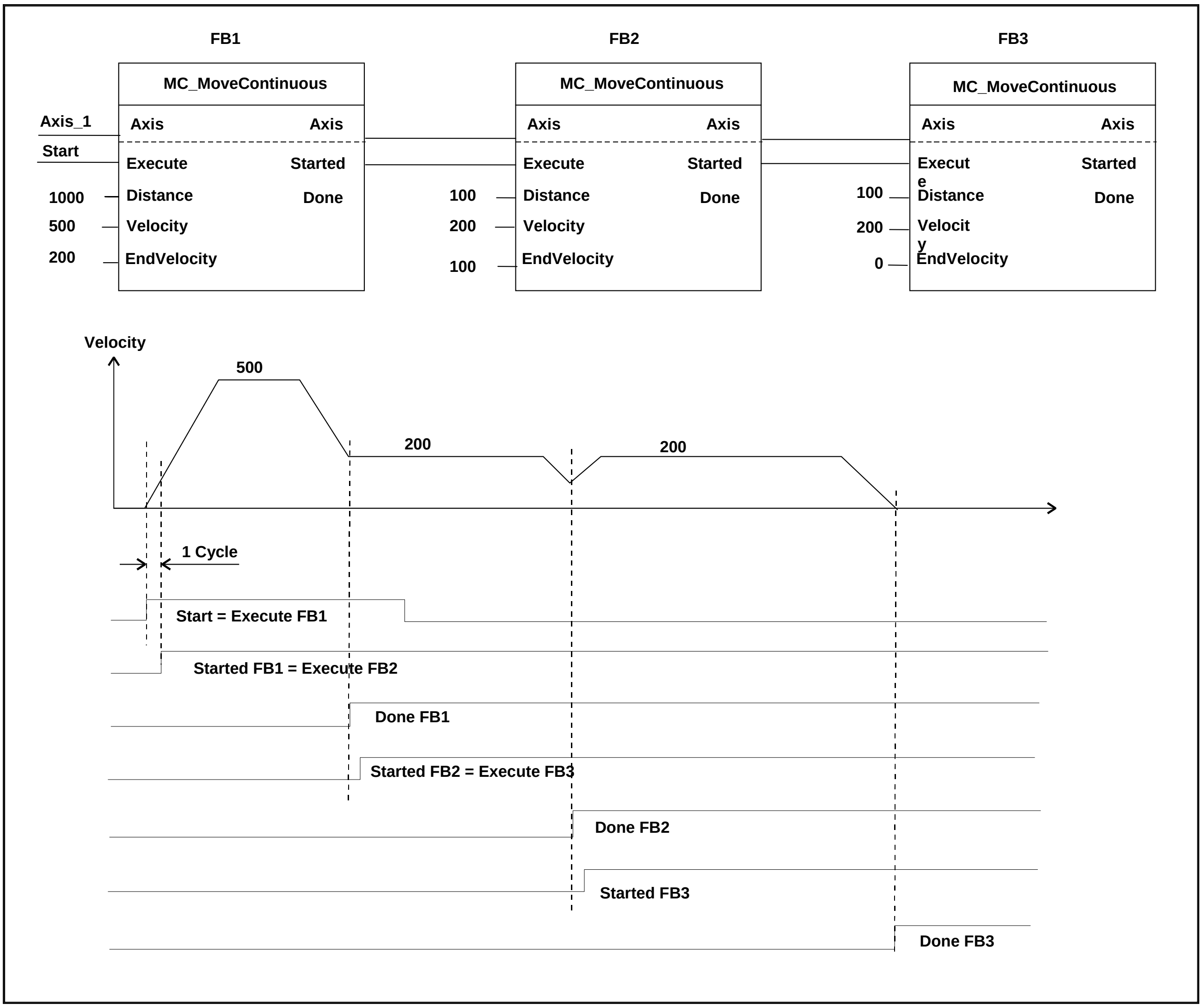
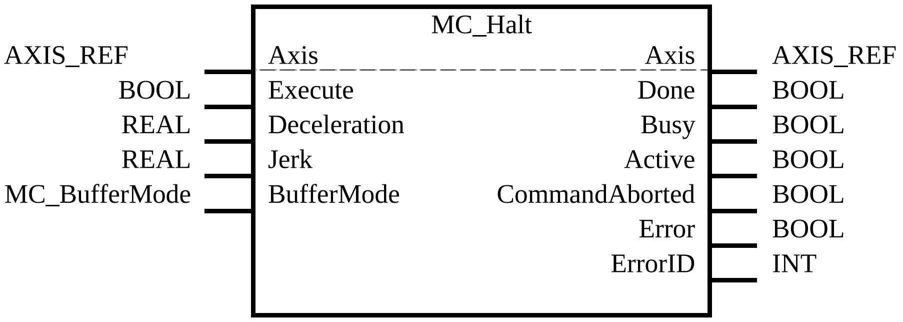


**Figure 11: Example of MC_MoveContinuous**

## 5.14. Halt

| | | | |
|---|---|---|---|
| FB-Name | **MC_Halt** | | |
| This function block commands a controlled motion stop. It aborts any ongoing function block execution. The axis is moved to the state “DiscreteMotion“, until the velocity is zero. With the Done output set, the state is transferred to StandStill. | | | |
| VAR_IN_OUT | | | |
| B | Axis | AXIS_REF | |
| VAR_INPUT | | | |
| B | Execute | BOOL | Start the action at rising edge |
| E | Deceleration | REAL | Value of the deceleration [u/s$^2$] |
| E | Jerk | REAL | Value of the Jerk [u/s$^3$] |
| E | BufferMode | MC_BufferMode | Defines the behavior of the axis: modes are Aborting, Buffered, Blending |
| VAR_OUTPUT | | | |
| B | Done | BOOL | Zero velocity reached |
| E | Busy | BOOL | Shows that the Function Block is not finished |
| E | Active | BOOL | Indicates that the Function Block has control on the axis |
| E | CommandAborted | BOOL | Command is aborted by another command |
| B | Error | BOOL | Signals that an error has occurred within the Function Block |
| E | ErrorID | WORD | Error identification |
| Notes:<br>• MC_Halt is used to stop the axis under normal operation conditions. In non-buffered mode it is possible to set another motion command during deceleration of the axis, which will abort the MC_Halt and will be executed immediately.<br>• If this command is active the next command can be issued. E.g. a driverless vehicle detects an obstacle and needs to stop. MC_Halt is issued. Before the standstill is reached the obstacle is removed and the motion can be continued by setting another motion command, so the vehicle does not stop. | | | |



The example below shows the behavior in combination with a MC_MoveVelocity.
a) A rotating axis is ramped down with Function Block MC_Halt
b) Another motion command overrides the MC_Halt command. MC_Halt allows this, in contrast to MC_Stop. The axis can accelerate again without reaching standstill.

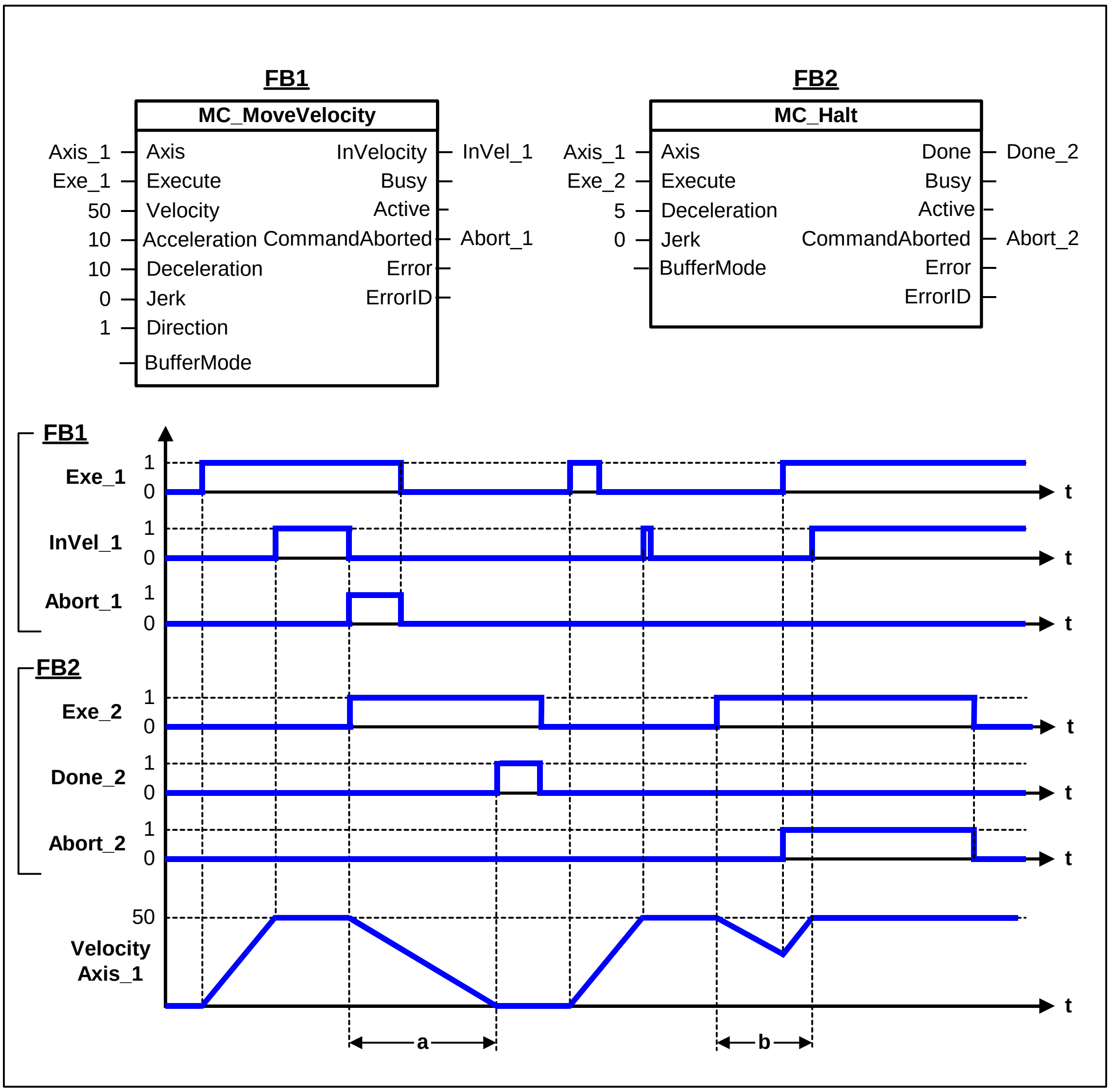


**Figure 12: Example of MC_Halt**

# 6. Appendix A – Compliance Statement

Listed in this Appendix are the requirements for the compliance statement from the supplier of the Motion Control Function Blocks. This part should be seen as integral to Part 1 – Function Blocks for Motion Control.
The compliance statement consists of two main groups: supported datatypes and supported Function Blocks, in combination with the applicable inputs and outputs. The supplier has to fill out the tables for the used datatypes and Function Blocks, according to their product, committing their support to the specification.

By submitting these tables to PLCopen, as well as those from Part 1, and after approval by PLCopen, the list will be published on the PLCopen website, www.plcopen.org , as well as a short form overview, as specified in Appendix A 2 Supported Datatypes and Appendix A 3 Overview of the Function Blocks as below.

In addition to this approval, the supplier is permitted access and usage rights to the PLCopen Motion Control logo, as described in Part 1, chapter Appendix A4 - The PLCopen Motion Control Logo and Its Usage.

## 6.1. Appendix A - Supported Derived Datatypes

Within the specification the following derived datatypes are defined. Define which of these structures are used in this system:

| **Derived datatypes:** | **Where used** | **Supported** | **Which structure** |
|---|---|---|---|
| TRIGGER_REF | MC_TouchProbe<br>MC_AbortTrigger | | |
| INPUT_REF | MC_ReadDigitalInput | | |
| OUTPUT_REF | MC_DigitalCamSwitch<br>MC_ReadDigitalOutput<br>MC_WriteDigitalOutput | | |
| CAMSWITCH_REF | MC_DigitalCamSwitch | | |
| TRACK_REF | MC_DigitalCamSwitch | | |

**Table 3: Supported derived datatypes**

## 6.2. Appendix A - Overview of the Function Blocks

| **Single Axis Function Blocks** | **Supported Yes / No** | **Comments (<= 48 char.)** |
|---|---|---|
| MC_TouchProbe | | |
| MC_AbortTrigger | | |
| MC_ReadDigitalInput | | |
| MC_ReadDigitalOutput | | |
| MC_WriteDigitalOutput | | |
| MC_SetPosition | | |
| MC_SetOverride | | |
| MC_ReadActualVelocity | | |
| MC_ReadActualTorque | | |
| MC_TorqueControl | | |
| MC_DigitalCamSwitch | | |
| MC_GearInPos | | |
| MC_MoveContinuous | | |
| MC_Halt | | |

**Table 4: Short overview of the Function Blocks**

### *6.2.1. TouchProbe*

| If Supported | **MC_TouchProbe** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| E | TriggerInput | | |
| VAR_INPUT | | | |
| B | Execute | | |
| E | WindowOnly | | |
| E | FirstPosition | | |
| E | LastPosition | | |
| VAR_OUTPUT | | | |
| B | Done | | |
| E | Busy | | |
| E | CommandAborted | | |
| B | Error | | |
| E | ErrorID | | |
| B | RecordedPosition | | |

### *6.2.2. AbortTrigger*

| If Supported | **MC_AbortTrigger** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| E | TriggerInput | | |
| VAR_INPUT | | | |
| B | Execute | | |
| VAR_OUTPUT | | | |
| B | Done | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.3. ReadDigitalInput*

| If Supported | **MC_ReadDigitalInput** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Input | | |
| VAR_INPUT | | | |
| B | Enable | | |
| B | InputNumber | | |
| VAR_OUTPUT | | | |
| B | Valid | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |
| B | Value | | |

### *6.2.4. ReadDigitalOutput*

| If Supported | **MC_ReadDigitalOutput** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Output | | |
| VAR_INPUT | | | |
| B | Enable | | |
| B | OutputNumber | | |
| VAR_OUTPUT | | | |
| B | Valid | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |
| B | Value | | |

### *6.2.5. WriteDigitalOutput*

| If Supported | **MC_WriteDigitalOutput** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Output | | |
| VAR_INPUT | | | |
| B | Execute | | |
| B | OutputNumber | | |
| B | Value | | |
| VAR_OUTPUT | | | |
| B | Done | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.6. SetPosition*

| If Supported | **MC_SetPosition** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Execute | | |
| B | Position | | |
| E | Mode | | |
| VAR_OUTPUT | | | |
| B | Done | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.7. SetOverride*

| If Supported | **MC_SetOverride** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Enable | | |
| B | VelFactor | | |
| E | AccFactor | | |
| E | JerkFactor | | |
| VAR_OUTPUT | | | |
| B | Enabled | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.8. ReadActualVelocity*

| If Supported | **MC_ReadActualVelocity** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Enable | | |
| VAR_OUTPUT | | | |
| B | Valid | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |
| B | ActualVelocity | | |

### *6.2.9. ReadActualTorque*

| If Supported | **MC_ReadActualTorque** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Enable | | |
| VAR_OUTPUT | | | |
| B | Valid | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |
| B | ActualTorque | | |

### *6.2.10. TorqueControl*

| If Supported | **MC_TorqueControl** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Execute | | |
| B | Torque | | |
| E | TorqueRamp | | |
| E | Velocity | | |
| E | Acceleration | | |
| E | Deceleration | | |
| E | Jerk | | |
| E | Direction | | |
| E | BufferMode | | |
| VAR_OUTPUT | | | |
| B | InTorque | | |
| E | Busy | | |
| E | Active | | |
| E | CommandAborted | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.11. Digital Cam Switch*

| If Supported | **MC_DigitalCamSwitch** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| B | Switches | | |
| B | Outputs | | |
| E | TrackOptions | | |
| VAR_INPUT | | | |
| B | Enable | | |
| E | EnableMask | | |
| VAR_OUTPUT | | | |
| B | InOperation | | |
| E | Busy | | |
| B | Error | | |
| E | ErrorID | | |

Basic elements within the structure of CAMSWITCH_REF

| **B/E** | **Parameter** | Sup.Y/N | Comments |
|---|---|---|---|
| B | TrackNumber | | |
| B | FirstOnPosition [u] | | |
| B | LastOnPosition [u] | | |
| E | AxisDirection | | |
| E | CamSwitchMode | | |
| E | Duration | | |

Basic elements within the array structure of TRACK_REF

| **B/E** | **Parameter** | Sup.Y/N | Comments |
|---|---|---|---|
| E | OnCompensation | | |
| E | OffCompensation | | |
| E | Hysteresis [u] | | |

### *6.2.12. GearInPos*

| If Supported | **MC_GearInPos** | Sup.Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Master | | |
| B | Slave | | |
| VAR_INPUT | | | |
| B | Execute | | |
| B | RatioNumerator | | |
| B | RatioDenominator | | |
| B | MasterSyncPosition | | |
| B | SlaveSyncPosition | | |
| E | SyncMode | | |
| E | MasterStartDistance | | |
| E | Velocity | | |
| E | Acceleration | | |
| E | Deceleration | | |
| E | Jerk | | |
| E | BufferMode | | |
| VAR_OUTPUT | | | |
| E | StartSync | | |
| B | InSync | | |
| E | Busy | | |
| E | Active | | |
| B | CommandAborted | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.13. MoveContinuous*

| If Supported | **MC_MoveContinuous** | Sup. Y/N | Comments |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Execute | | |
| B | Distance | | |
| B | Velocity | | |
| B | EndVelocity | | |
| E | Acceleration | | |
| E | Deceleration | | |
| E | Jerk | | |
| E | BufferMode | | |
| VAR_OUTPUT | | | |
| B | InEndVelocity | | |
| E | Busy | | |
| E | Active | | |
| B | CommandAborted | | |
| B | Error | | |
| E | ErrorID | | |

### *6.2.14. Halt*

| If Supported | **MC_Halt** | Sup. Y/N | |
|---|---|---|---|
| VAR_IN_OUT | | | |
| B | Axis | | |
| VAR_INPUT | | | |
| B | Execute | | |
| E | Deceleration | | |
| E | Jerk | | |
| E | BufferMode | | |
| VAR_OUTPUT | | | |
| B | Done | | |
| E | Busy | | |
| E | Active | | |
| E | CommandAborted | | |
| B | Error | | |
| E | ErrorID | | |