MotionWorks IEC

PLCopenPlus Function Blocks for Motion Control - 2013-04-12

Table Of Contents

Overview 1

Introduction 1

The State Diagram 3

Error Handling 6

Function block interface 8

Data Types 29

Data Types 29

Data Type: AXIS_REF 30

Data Type: CONTINUOUS_REF 32

Data Type: INPUT_REF 34

Data Type: OUTPUT_REF 36

Data Type: PATTERN_REF 38

Data Type: PrmStruct 39

Data Type: RTC_STRUCT 40

Data Type: TRIGGER_REF 41

Data Type: Y_DISENGAGE_DATA 43

Data Type: Y_ENGAGE_DATA 44

Data Type: Y_MS_CAM_STRUCT 45

Enumerated Types..47

Function Block List..49

Function Block List 49

Function Blocks for Motion Control 61

MC_AbortTrigger 61

MC_FinishHoming 65

MC_GearIn 68

MC_GearInPos..74

MC_GearOut 83

MC_MoveAbsolute..86

MC_MoveRelative..92

MC_MoveSuperImposed 97

MC_MoveVelocity..109

MC_Power...114

MC_ReadActualPosition...121

MC_ReadActualTorque...123

MC_ReadActualVelocity 126

MC_ReadAxisError 128

MC_ReadBoolParameter..132

MC_ReadParameter..135

MC_ReadStatus 138

MC_Reset 142

MC_SetPosition..146

MC_StepLimitSwitch...150

MC_StepRefPulse..155

MC_Stop..160

MC_TorqueControl 165

MC_TouchProbe..176

MC_WriteBoolParameter 185

MC_WriteParameter 188

Y_CamFileSelect 191

Y_CamIn 197

Y_CamOut 210

Y_CamScale...215

Y_CamShift..220

Y_CamStructSelect 227

Y_ClearAlarms...231

Y_DirectControl 233

Y_HoldPosition 239

Y_ProbeContinuous 242

Y_ReadAlarm 248

Y_ReadCamTable 250

Y_ReadDriveParameter 254

Y_ReadMultipleParameters 258

Y_ReadStringParameter 261

Y_RebootController 264

Y_ReleaseCamTable 266

Y_ResetAbsoluteEncoder 268

Y_ResetMechatrolink 273

Y_SetRTC 276

Y_SlaveOffset 278

Y_VerifyParameters 285

Y_WriteCamTable 290

Y_WriteDriveParameter 293

Y_WriteParameters 297

Controller AlarmID List 301

Function Block ErrorID List 333

Axis Parameter List 341
High Speed Output 349
Timing Diagram 350
Camming 353
Camming Introduction 353
CamState 353
Cam Masters 355
Master Cycle 355
Camming Function Blocks 356
Creating a Cam Table 357
Externally Created Cam Data 358
Transferring Cam Files to an MPiec Controller 359
User File Storage 362
Configuring FileName Input for Y_CamFileSelect 363
Internally Created Cam Data 365
Cam Table Types 367
On-The-Fly Adjustments 368
Camming Block Diagram 369
Engage / Disengage Window 370

Cam Transitions Matrix 371

Motion Details 375

Acceleration/Deceleration Limits 375

Position Limits 377

Velocity Limits 379

Moving Average Filter (S-Curve) 380

Determining When Motion is Complete 383

External Encoder Block Diagram 385

Commanded Position Output 386

Command Filtering (MPiec with MECHATROLINK interface) 387

Command Filtering (MP2600iec - Dual Port RAM interface) 388

Overview

This document contains instructions for using the PLCopenPlus and YMotion firmware library function blocks integrated in the MPiec Series Controllers.

Introduction

This manual is adopted from the PLCopen for motion control specification at www.plcopen.org, and includes additional information for functionality with Yaskawa and other components.

Each function block is listed in alphabetical order, and is also linked to the feature or function from the software environment. A comprehensive list of axis parameters and error codes is at the back of the manual. A subset of specific errors that each function block may generate is included under each function block description.

The other main concepts covered in this manual are the Motion State Diagram, and documentation concerning the Data Types supplied with the PLCopenPlus Firmware Library.

The Firmware Library is the set of all PLCopen function blocks, plus Yaskawa specific functions. The firmware library is called PLCopen Plus, and is automatically loaded when a new project is created.

Model

The PLCopenPlus Function Block (FB) library is designed for the purpose of controlling axes via the language elements consistent with those defined in the IEC 61131-3 standard. It provides a set of command oriented function blocks that have a reference to the axis, e.g. the abstract data type ‘Axis’, which offers flexibility, ease of use and reusability.

The State Diagram

The state diagram shown defines the behavior of the axis at a high level when motion control function blocks are "simultaneously" activated. This combination of motion profiles is useful in building a more complicated profile or to treat exceptions within a program.

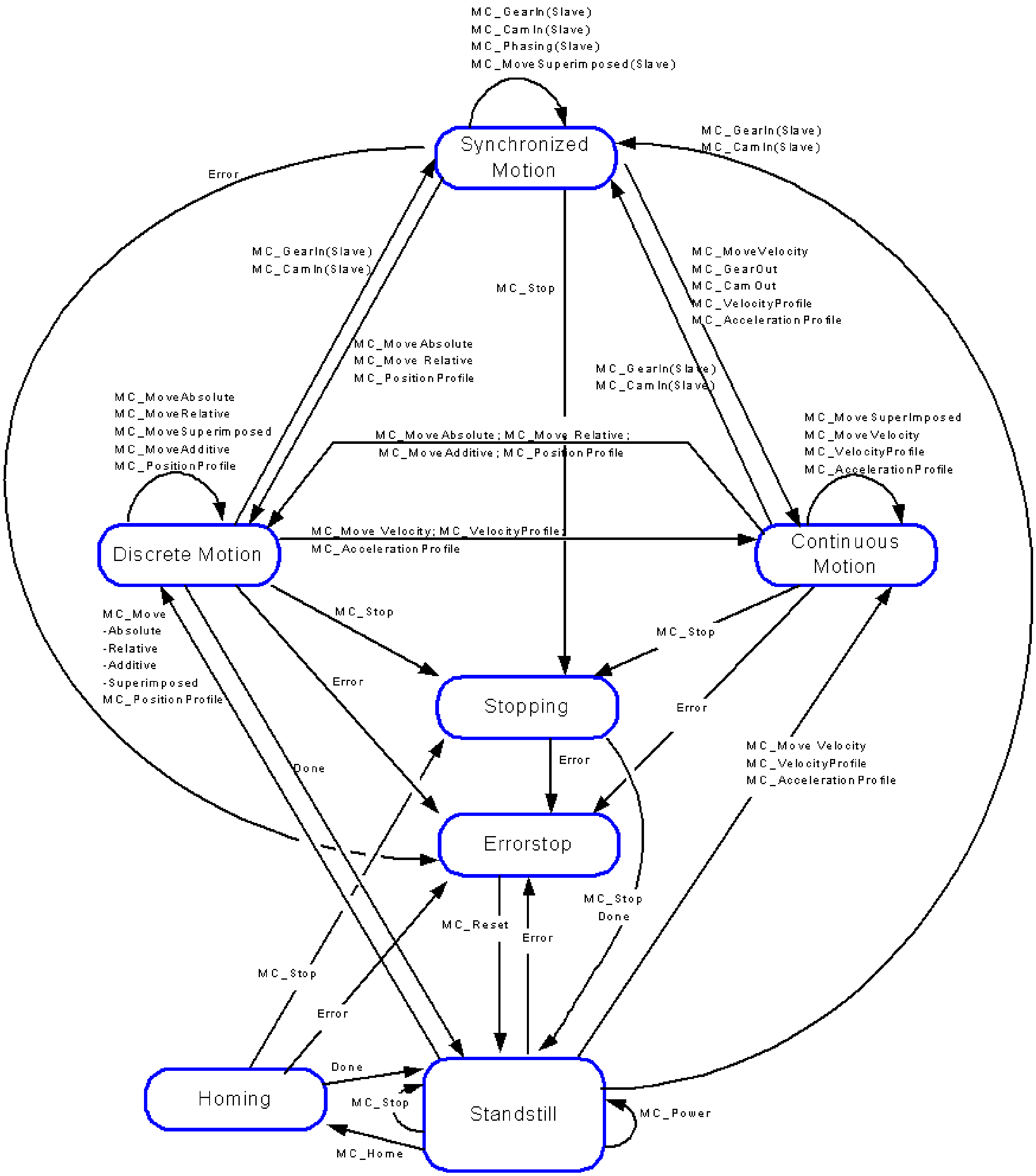
The basic rule is that motion commands are always taken sequentially. These commands act on the axis' state diagram. The axis is always in one of the following defined states:

- Standstill (no movement)
- Homing (movement to reference position)
- Discrete Motion (movement towards target position)
- Continuous Motion (jogging)
- Synchronized Motion (synchronized movement of master and slave)
- Stopping (axis is stopped)
- ErrorStop (axis error occurred)

Any motion command is a transition that changes the state of the axis and, as a consequence, modifies the way the current motion is computed. A normal procedure would start in Standstill. In this state, the power can be switched on per axis (via the Power command). Also, one can access the Homing state (via the issue of the Home command per axis), which after normal completion returns to Standstill. From here, one can transfer an axis to either Discrete Motion or Continuous Motion. Via the Stopping state, one can return to Standstill. ErrorStop is a state to which the axis transfers in case of an error. Via a Reset command, one can return to Standstill, from which the machine can be moved to an operational state again. Please note that the states define the functionality of the Function Blocks.

The diagram is focused on the states of a single axis. The multiple axis function blocks such as MC_CamIn and MC_GearIn change the state whereas these axis can have specific states.

Connecting a slave axis to a master axis has no influence on the master axis.

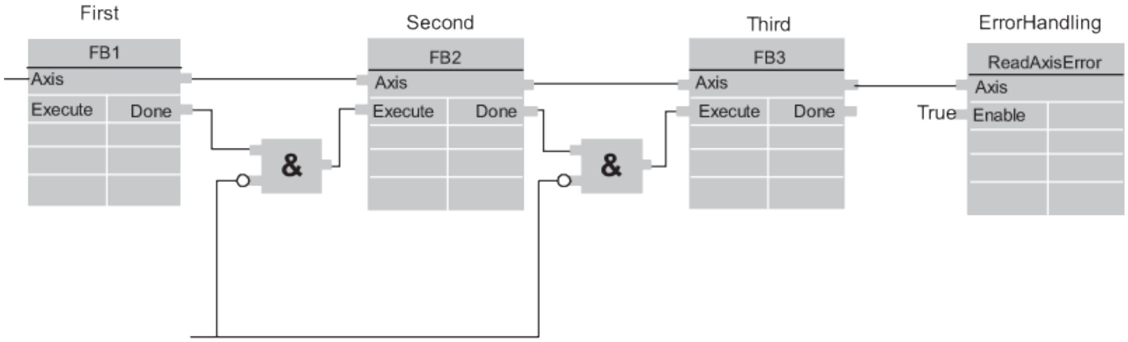

MC_GearIn(Slave)
MC_CamIn(Slave)
MC_Phasing(Slave)
MC_MoveSuperimposed(Slave)
Synchronized Motion
MC_GearIn(Slave)
MC_CamIn(Slave)
Error
MC_GearIn(Slave)
MC_CamIn(Slave)
MC_Stop
MC_MoveVelocity
MC_GearOut
MC_CamOut
MC_VelocityProfile
MC_AccelerationProfile
MC_MoveAbsolute
MC_Move Relative
MC_PositionProfile
MC_GearIn(Slave)
MC_CamIn(Slave)
MC_MoveAbsolute
MC_MoveRelative
MC_MoveSuperimposed
MC_MoveAdditive
MC_PositionProfile
MC_MoveAbsolute; MC_Move Relative;
MC_MoveAdditive; MC_PositionProfile
MC_MoveSuperImposed
MC_MoveVelocity
MC_VelocityProfile
MC_AccelerationProfile
Discrete Motion
MC_Move Velocity; MC_VelocityProfile;
MC_AccelerationProfile
Continuous Motion
MC_Move
-Absolute
-Relative
-Additive
-Superimposed
MC_PositionProfile
MC_Stop
MC_Stop
Error
Stopping
Error
Done
Error
MC_Move Velocity
MC_VelocityProfile
MC_AccelerationProfile
Errorstop
MC_Stop
Done
MC_Reset
Error
MC_Stop
Error
Done
Homing
MC_Stop
Standstill
MC_Power
MC_Home

Error Handling

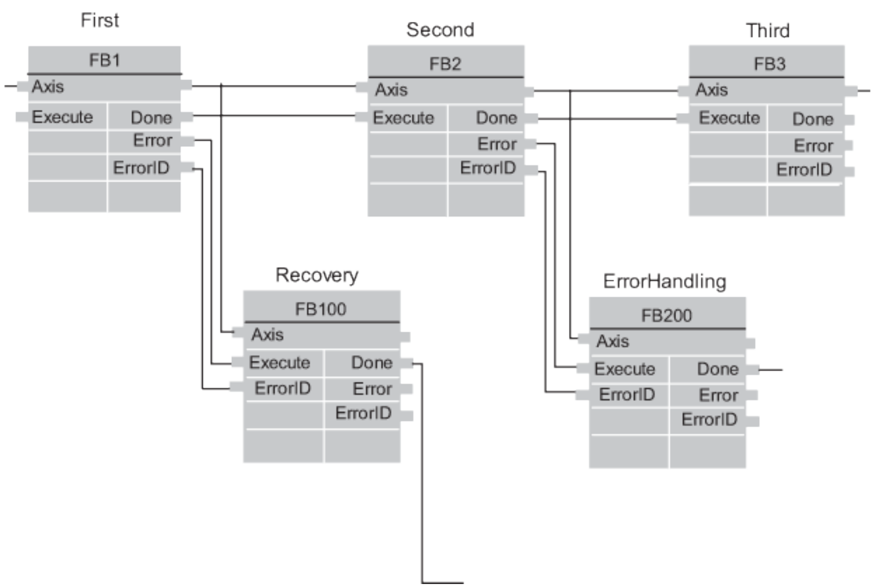
All access to the drive/motion control is via Function Blocks. Internally these Function Blocks provide basic error checking on the input data.

If the device itself has an error, it can be read using the MC_ReadAxisError block.

Function Blocks with centralized error handling



Function blocks with decentralized error handling



Function block interface

General rules

The following table provides general rules about the interface of the Motion Control function blocks.

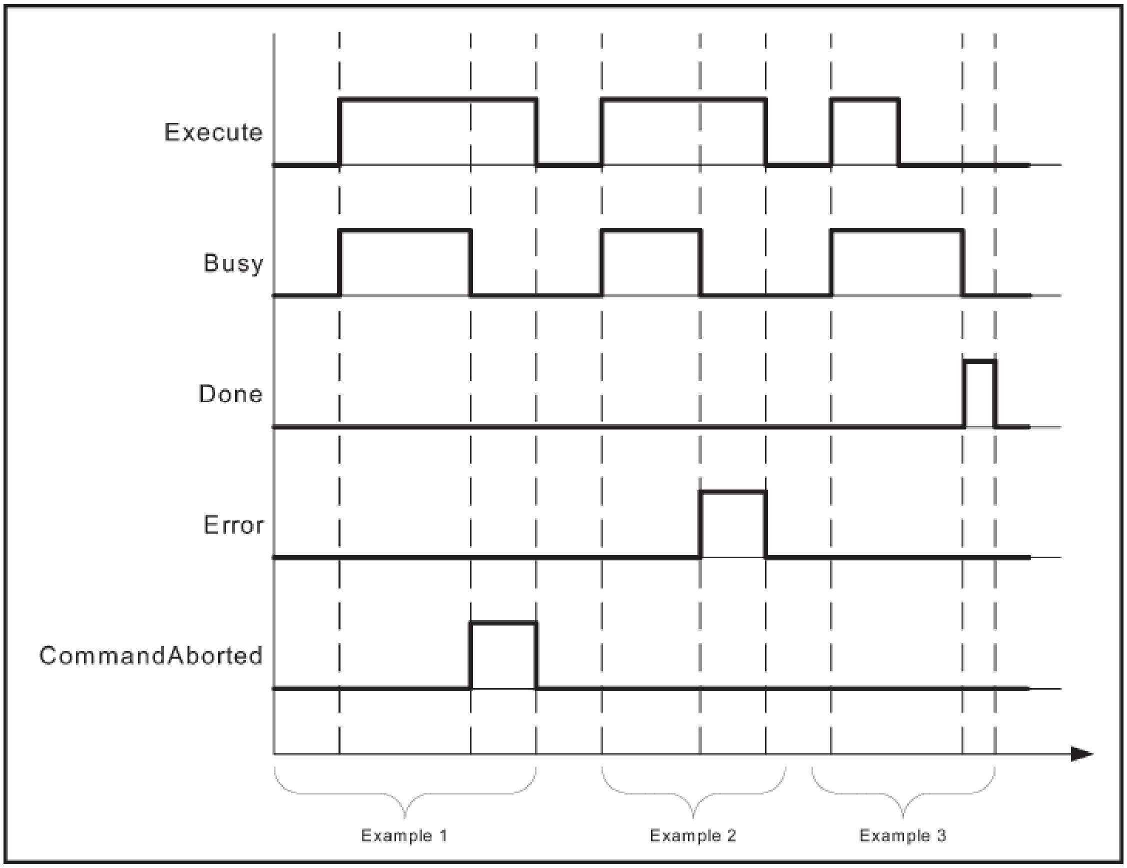
Rule applies to	Rule
Output exclusivity	When 'Execute' is true, the outputs ‘Busy’, ‘Done’, ‘Error’, and ‘CommandAborted’ are mutually exclusive.
Output status	The 'Done', 'InGear', 'InSync', 'InVelocity', 'Error', 'ErrorID' and 'CommandAborted' outputs are reset with the falling edge of 'Execute'. However, the falling edge of 'Execute' does not stop or even influence the execution of the actual FB. The corresponding outputs are set for at least one cycle if the situation occurs, even if execute was reset before the FB completed. If an instance of a FB receives a new 'Execute' before it finishes (as a series of commands on the same instance), the FB won’t return any feedback, like ‘Done’ or ‘CommandAborted’, for the previous action.
Input parameters	The parameters are read at the rising edge of the 'Execute' input. To modify any parameter, it is necessary to change the input parameter(s) and trigger the 'Execute' again.
Missing input parameters	According to IEC 61131-3, if any parameter of a function block input is missing (“open”) then the

	value from the previous invocation of this instance will be used. In the first invocation the default value is applied.
Position versus distance	'Position' is a value defined within a coordinate system. 'Distance' is a relative measure, the difference between two positions.
Sign rules	The 'Velocity', 'Acceleration', 'Deceleration' and 'Jerk' are always positive values. 'Position' and 'Distance' can be positive or negative.
Error Handling Behavior	All blocks have two outputs, which deal with errors that can occur while executing that Function Block. These outputs are defined as follow:
	Error: Rising edge of 'Error' indicates that an error occurred during the execution of the Function Block. **ErrorID:** Error number - see the Error Code List at the end of the manual.
	'Done', 'InVelocity', 'InGear', and 'InSync' indicate successful completion, so these signals are logically exclusive to “Error”. Types of errors: • Function Block Error (e.g. parameters out of range, state machine violation attempted) • Communication Error • Amplifier/Axis Error Instance errors do not always result in an axis error (forcing the axis to 'StandStill'). The error outputs of the relevant FB are reset with falling

	edge of 'Execute'.
Behavior of Done output	The “Done” output (as well as 'InGear', 'InSync', ..) is set when the commanded action has been completed successfully. With multiple Function Blocks working on the same axis in a sequence, the following applies: When one movement on an axis is interrupted with another movement on the same axis without having reached the final goal, 'Done' of the first FB will not be set.
Behavior of CommandAborted output	'CommandAborted' is set when a commanded motion is interrupted by another motion command. The reset-behavior of 'CommandAborted' is like that of 'Done'. When 'CommandAborted' occurs, the other output-signals such as 'InVelocity' are reset.
Inputs exceeding application limits	If a FB is commanded with parameters which result in a violation of application limits, the instance of the FB generates an error. The consequences of this error for the axis are application specific and thus should be handled by the application program.
Behavior of Busy output	'Busy' output indicates that the FB is not finished. 'Busy' is SET at the rising edge of 'Execute' and RESET when one of the outputs 'Done', 'Aborted', or 'Error' is set. It is recommended that this FB should be kept in the active loop of the application program for at least as long as 'Busy' is true, because the

	outputs may still change. For one axis, several Function Blocks might be busy, but only one can be active at a time. Exceptions are 'MC_SuperImposed' and 'MC_Phasing', where more than one FB related to one axis can be active.
Output ‘Active’	The 'Active' output is available on Function Blocks with buffering capabilities. This output is set at the moment the function block takes control of the axis. For unbuffered mode the outputs “Active” and “Busy” can have the same value.
Enable and Valid Status	The 'Enable' input is coupled to a 'Valid' output. 'Enable' is level sensitive, and 'Valid' shows that a valid set of outputs is available at the FB. The 'Valid' output is TRUE as long as a valid output value is available and the 'Enable' input is TRUE. The relevant output values are refreshed while the input 'Enable' is TRUE. If there is a FB error, the output is not valid (“Valid” set to FALSE). When the error condition disappears, the values will reappear and 'Valid' output will be set again.

The behavior of the “Execute” / “Done” style FBs is as follows:

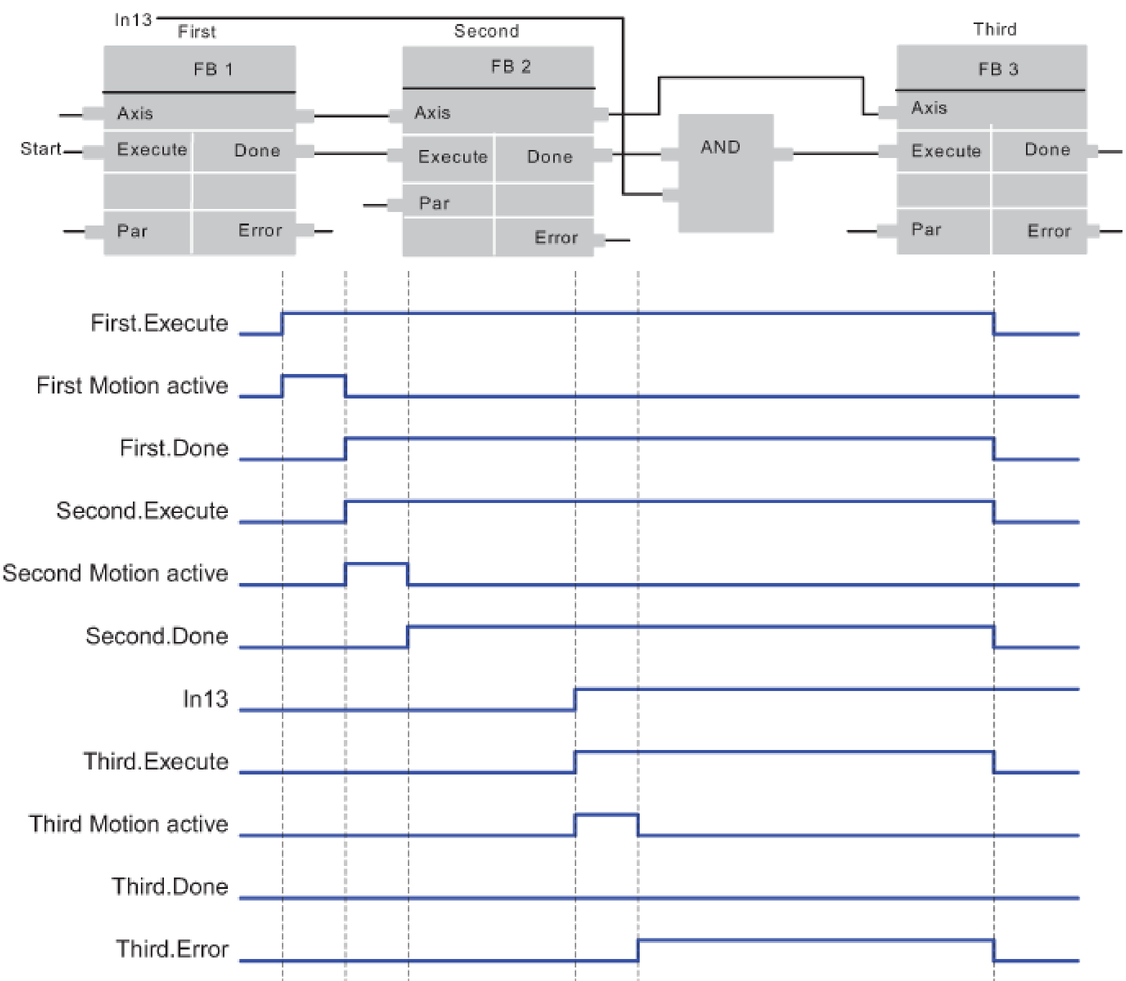


Why the command input is edge sensitive

The “Execute” input always triggers the function with its rising edge. New input values may be commanded during execution of a previous command because the inputs are only read once. The 'Done' output can be used to trigger the next part of the movement. The example given below is intended to explain the behavior of the Function Block execution.

The figure illustrates the sequence of three Function Blocks, 'First', 'Second' and 'Third', controlling the same axis. These three Function Blocks could be for instance various absolute or relative move commands. When “First” has completed, the output 'First.Done' triggers 'Second.Execute'. The output 'Second.Done' AND “In13” trigger 'Third.Execute'.

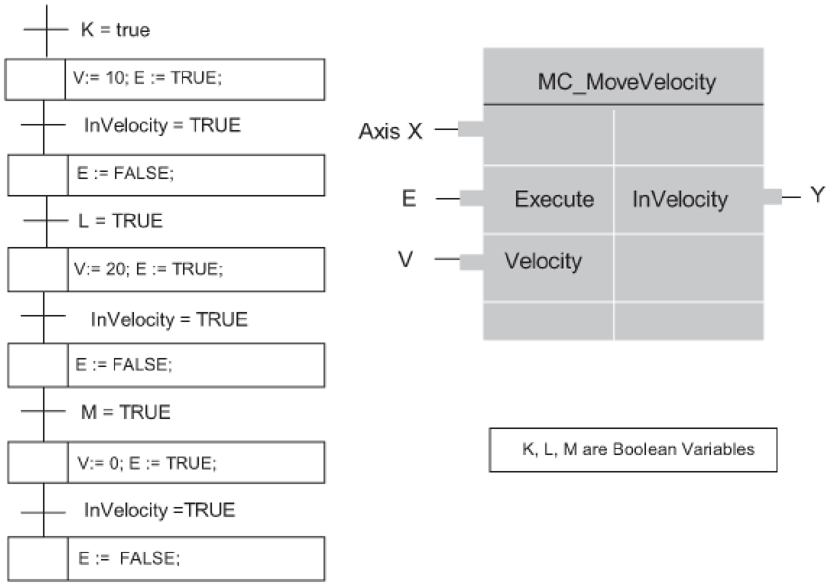
Function blocks to perform a complex movement



Example 1: Same Function Block instance controls different motions of an axis

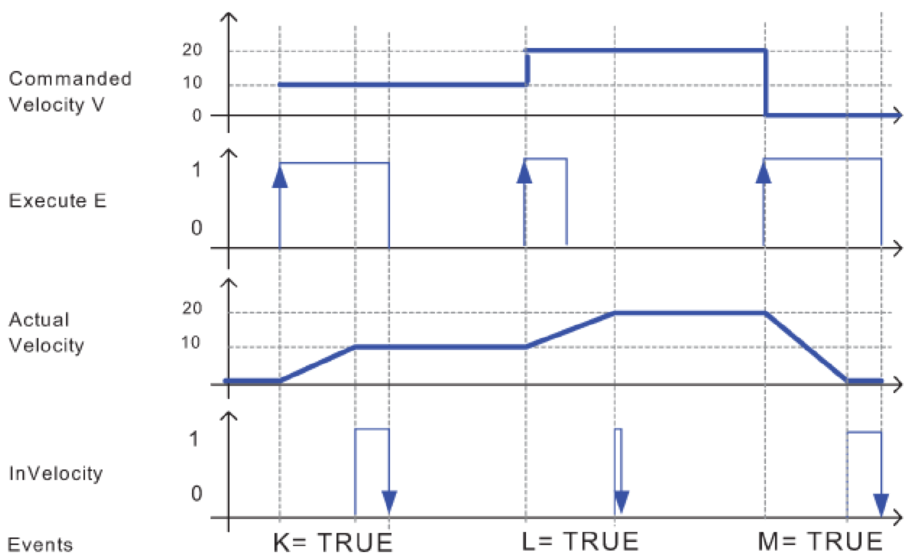
The figure below shows an example where the Function Block FB1 is used to control “AxisX” with three different values of Velocity. In a Sequential Function Chart (SFC) the velocity 10, 20, and 0 is assigned to V. To trigger the Execute input with a rising edge the variable E is stepwise set and reset.

Single FB usage with a SFC



The following timing diagram explains how it works.

Timing diagram for a usage of a single FB

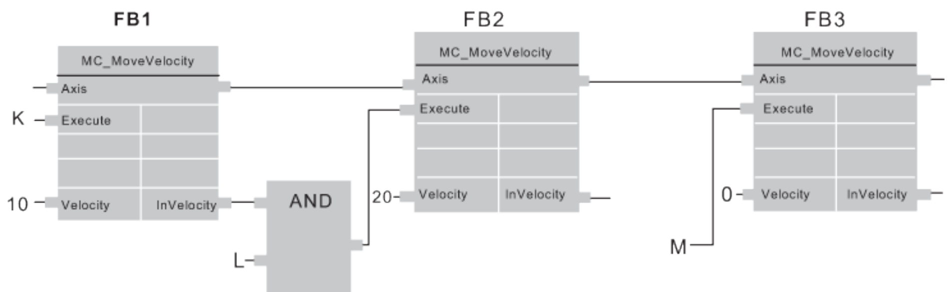


Note: The second InVelocity is set for only one cycle because the Execute has gone low before the Actual Velocity equals Commanded Velocity.

Example 2: Different FB instances control the motions of an axis

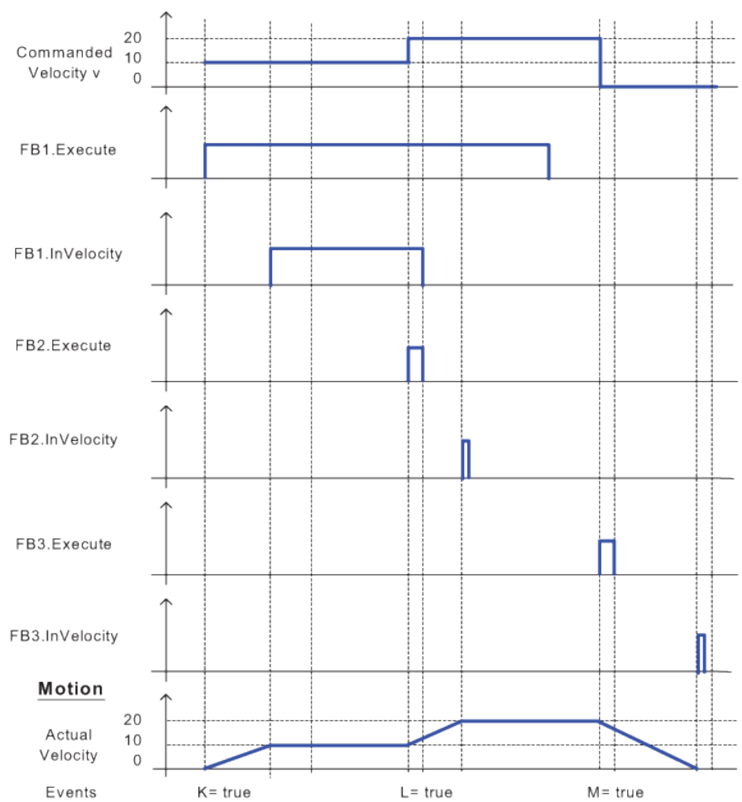
Different instances related to the same axis can control the motions on an axis. Each instance will then be responsible for one part of the global profile.

Cascaded Function Blocks



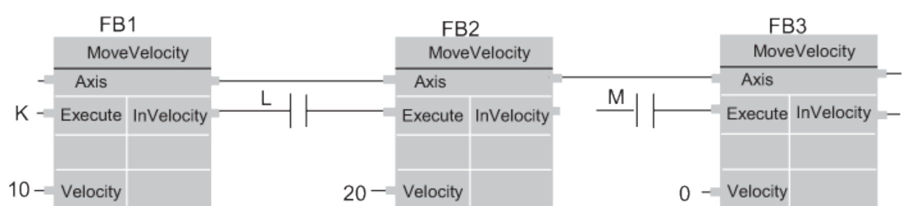
The timing diagram:

Cascaded Function Blocks timing diagram



A corresponding solution written in LD looks like:

Cascaded Function Blocks with LD



Aborting Versus Buffered Modes: Input BufferMode

Some of the FBs provide the input 'BufferMode'. By setting this input, the FB can either be run in "non-buffered mode" (default behavior) or in buffered mode. The transition behavior (blending) between two motions can be set by defining when the FB starts its action. The difference between these two modes is as follows:

- A Function Block in **non-buffered mode** is applied immediately, even when this interrupts a motion which is currently executed.
- A Function Block in **buffered mode** is not executed until the current FB has finished the motion it is currently executing and indicates this by setting the corresponding output (Done or InPosition or InVelocity - see table below).
- Up to 16 motion blocks can be buffered before error 4369 would be generated. Axis Parameter 1600 indicates the number of buffered motion blocks.

Possible options for the buffered mode

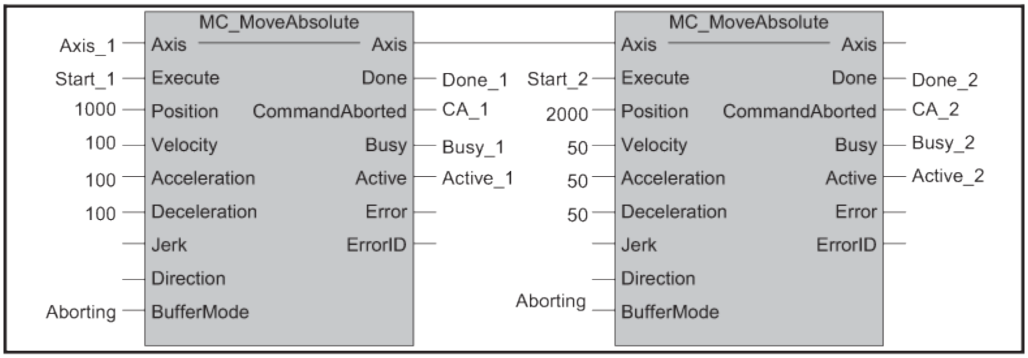
The input BufferMode must be connected with a INT data type which can have the following values:

Buffer mode	Short description Important note: The meaning of each value may vary depending on the FB(s) involved. For this reason, please also refer to the individual parameter descriptions!	Input value at BufferMode *
Aborting	This is the Default mode. The FB aborts an ongoing motion and the command affects the axis immediately.	INT#0
Buffered	The FB affects the axis as soon as the previous movement is complete. The axis will stop between the movements.	INT#1
BlendingLow	The FB controls the axis after the	INT#2

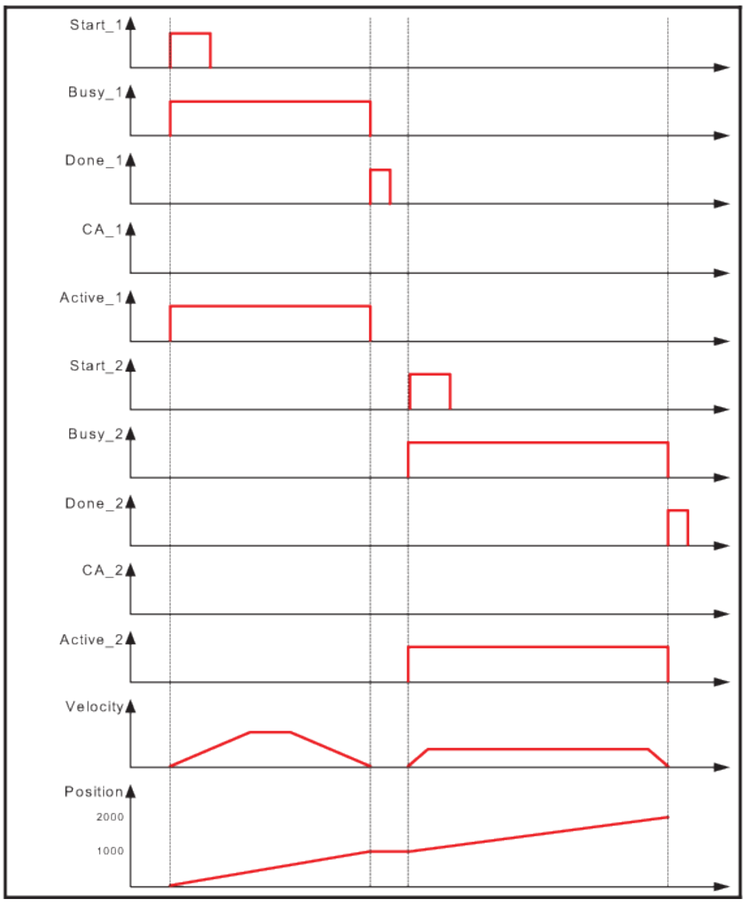
	previous FB has finished, but the axis will not stop between the movements. The velocity is blended with the lowest velocity of both commands.	
BlendingPrevious	The FB controls the axis after the previous FB has finished (equivalent to buffered), but the axis will not stop between the movements. Blending with the velocity of the previous move.	INT#3
BlendingNext	The FB controls the axis after the previous FB has finished, but the axis will not stop between the movements. Blending with velocity of this (next) function.	INT#4
BlendingHigh	The FB controls the axis after the previous FB has finished (equivalent to buffered), but the axis will not stop between the movements. Blending with highest velocity of the previous and this (next) function.	INT#5

Example 1: Standard behavior of 2 following absolute movements

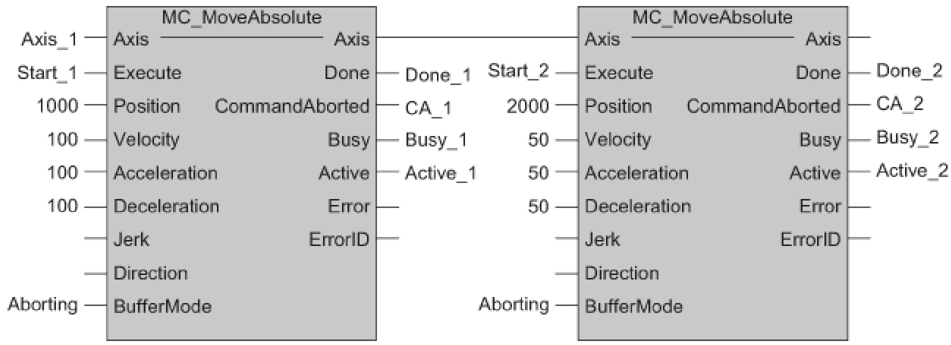
Basic example with two MC_MoveAbsolute on same axis



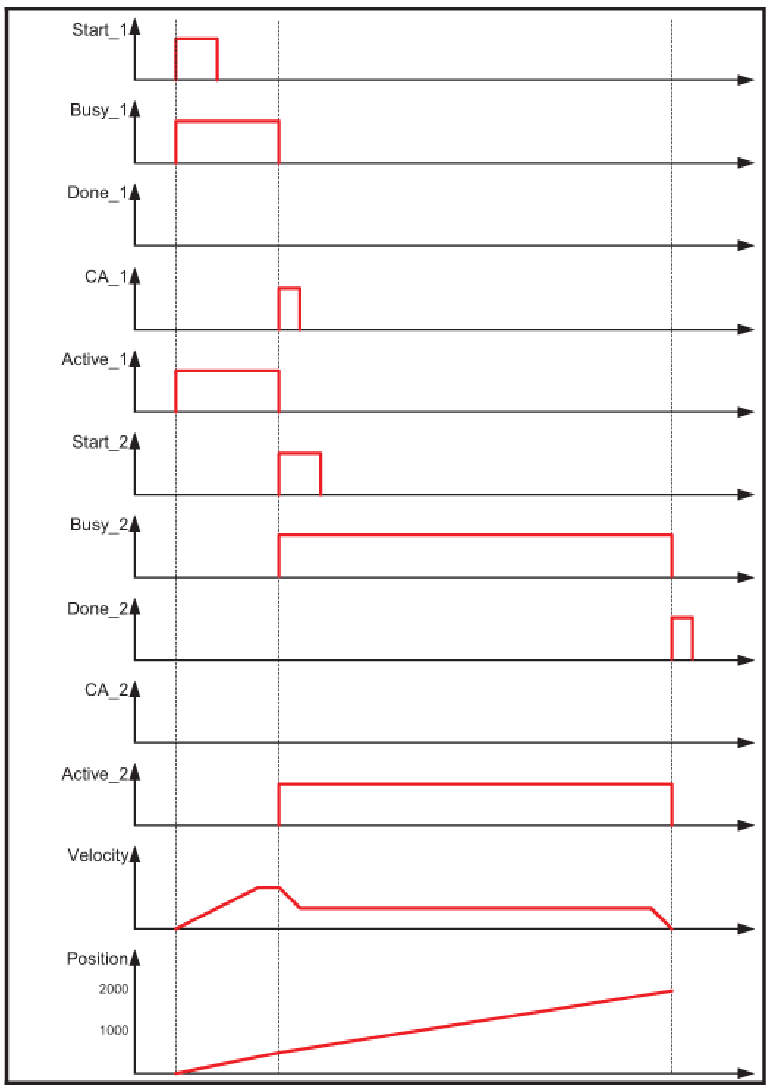
Timing diagram for example above without interference between FB1 and FB2 (Aborting Mode)



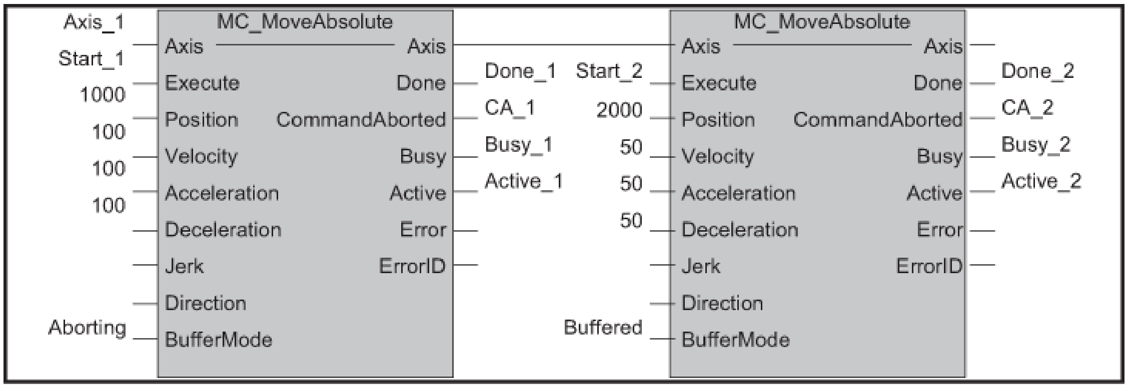
Example 2: Aborting motion



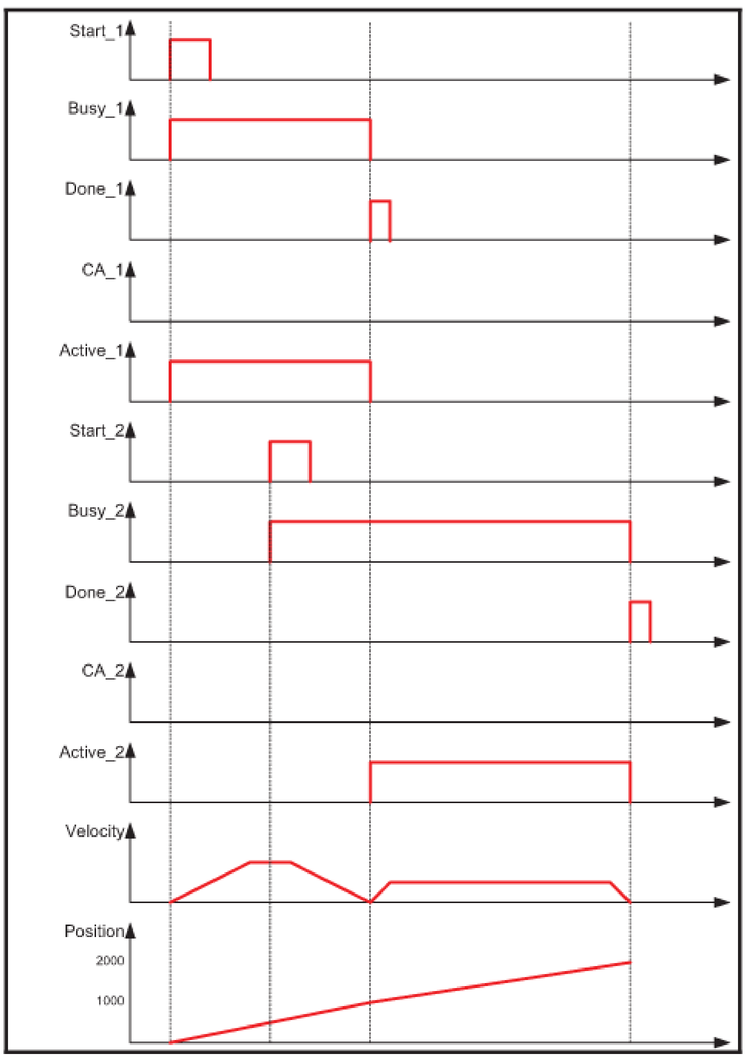
Timing diagram for example above with FB2 interrupting FB1 (Aborting Mode)



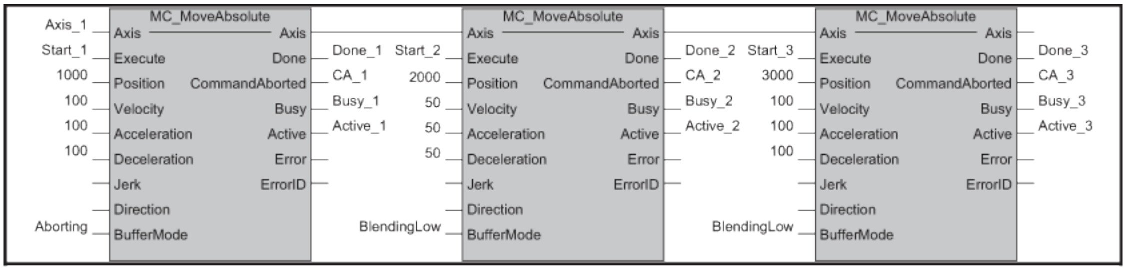
Example 3: Buffered motion



Timing diagram for example above in Buffered Mode
(Stopping to velocity 0 and starting FB2 at that point without delay)

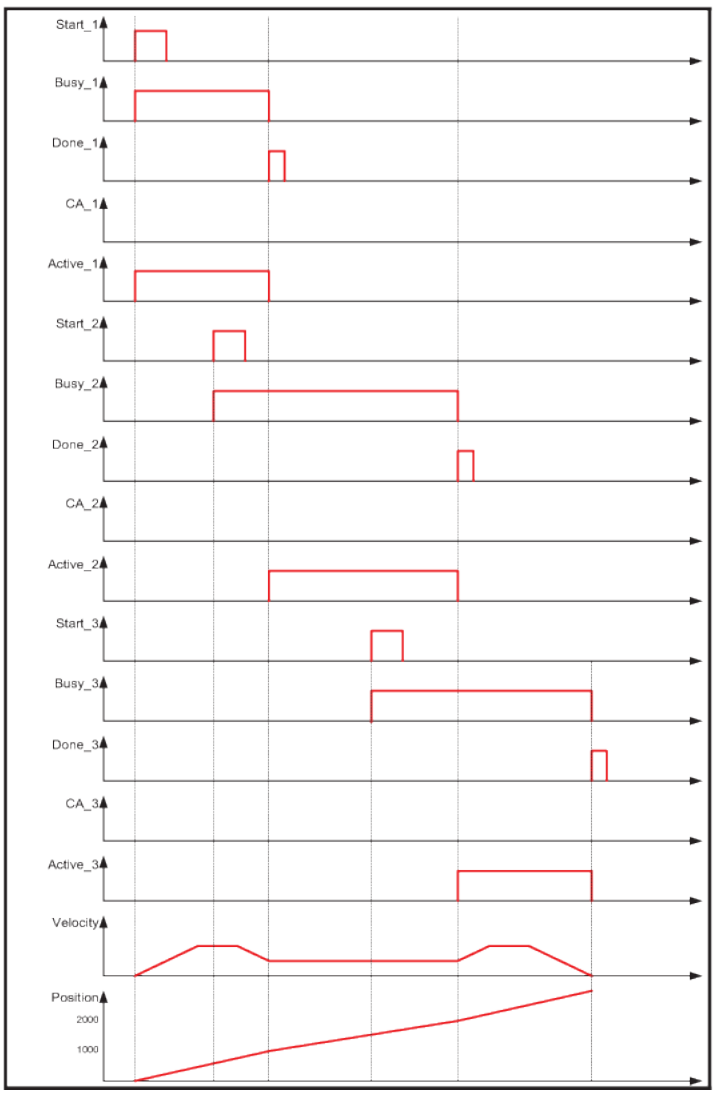


Example 4: BlendingLow motion

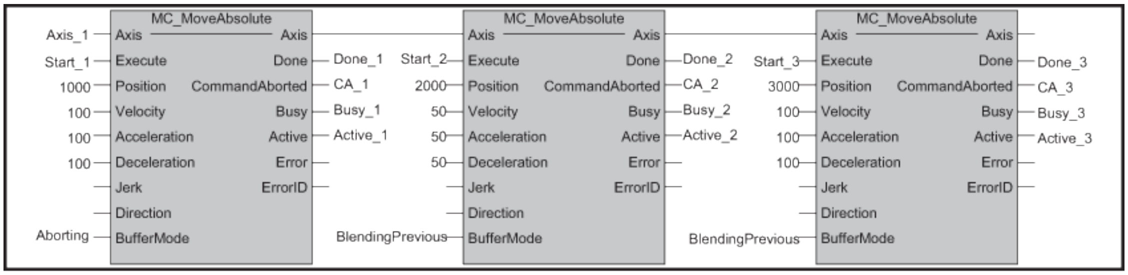


Timing diagram for example above with mode BlendingLow

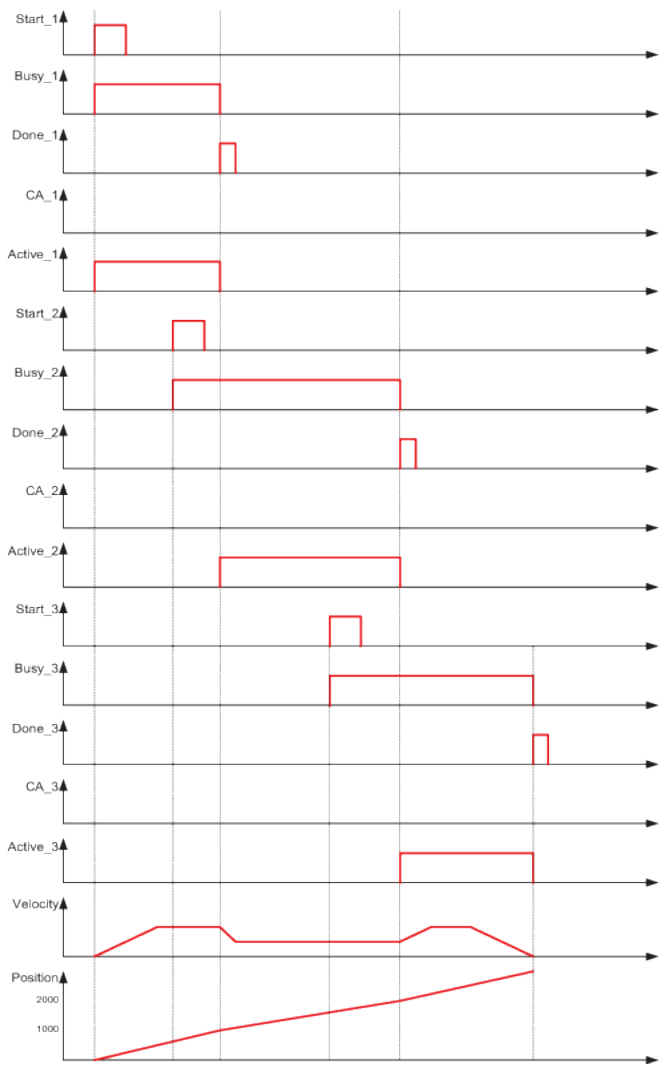
(Using lowest velocity (=velocity 2) from final position of FB1 until final position of FB2)



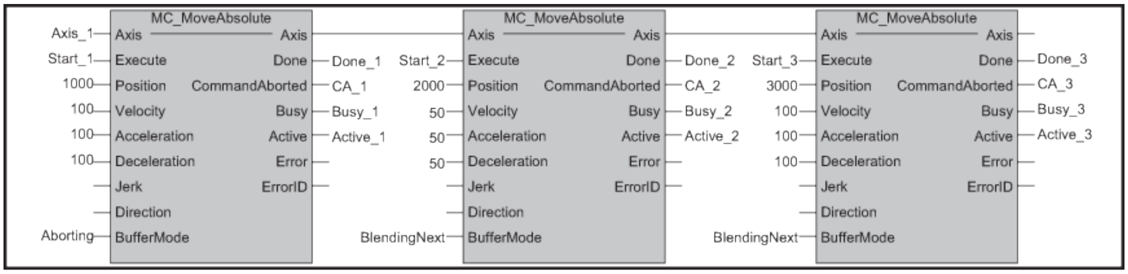
Example 5: BlendingPrevious motion



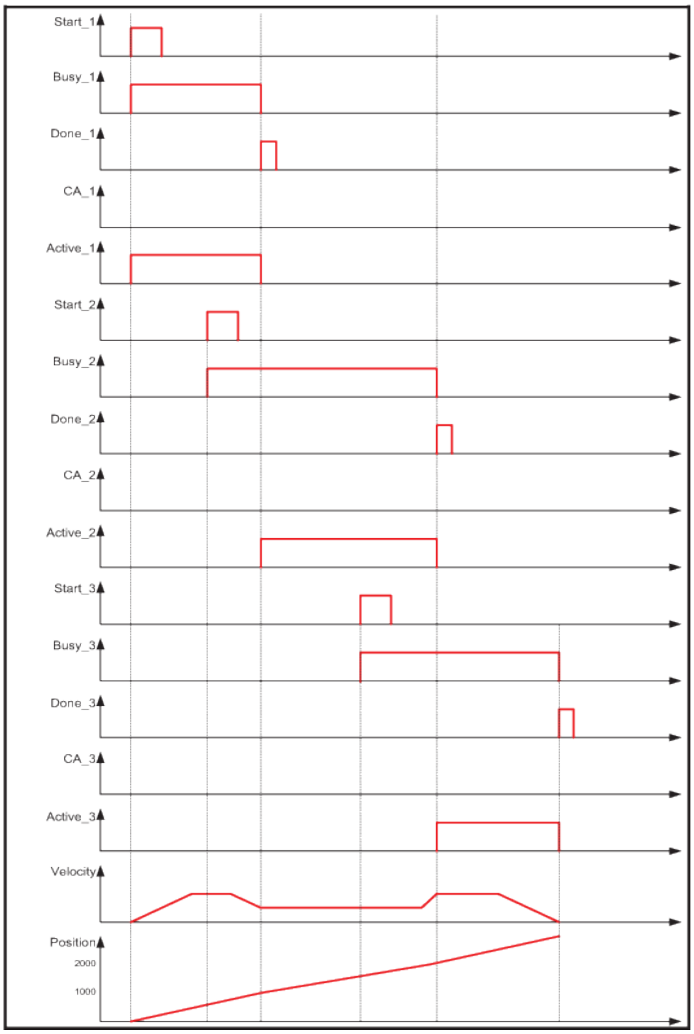
Timing diagram for example above with mode Merging1
(Uses velocity FB1 at final position FB1)



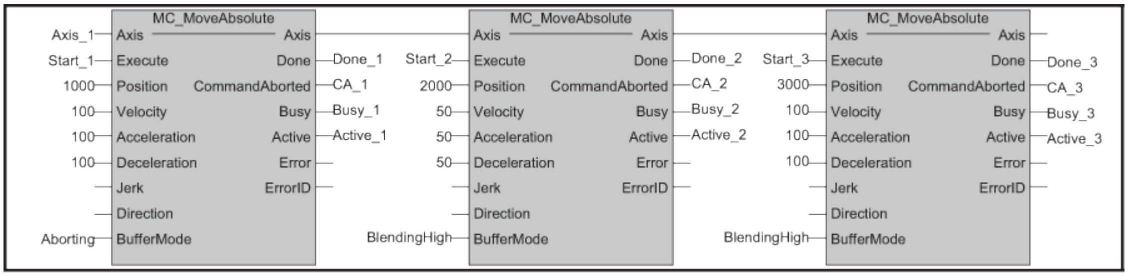
Example 6: BlendingNext motion



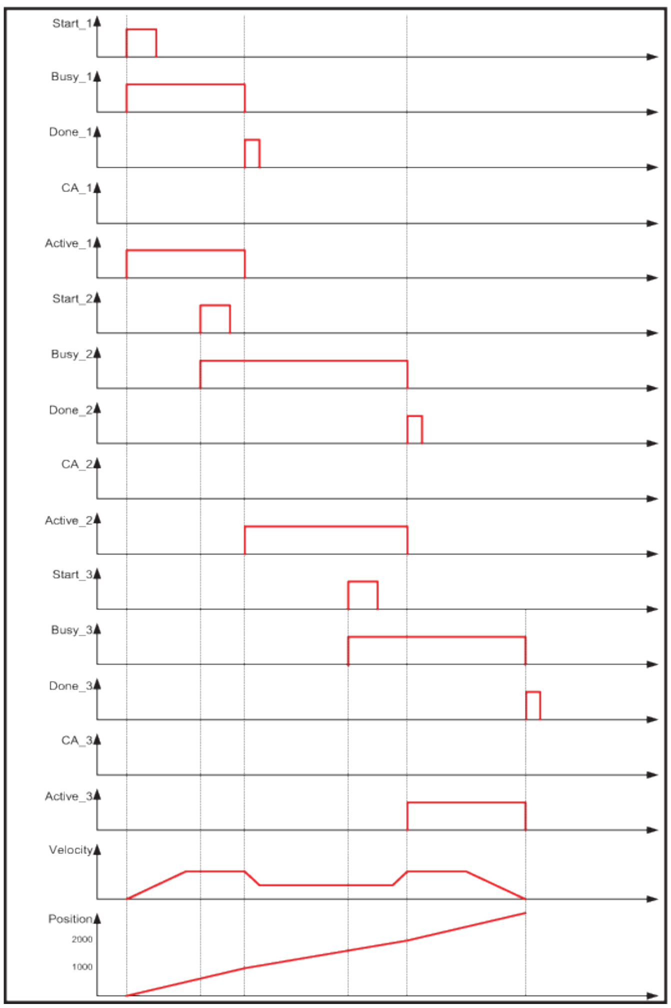
Timing diagram for example above with mode BlendingNextMotion



Example 7: BlendingHigh motion



Timing diagram for example above with mode BlendingHighMotion



Rules for the definition of Motion Control function blocks according to PLCopen

The input/output variables of the function blocks mandatory according to the PLCopen Standard are marked with the letter '**B**' in the defined tables in the definition of the function blocks.

Input/output variables marked with the letter '**E**' are optional, i.e. they can be implemented but are not mandatory.

Vendor specific input / output variables, i.e. added by the vendor, are marked with the letter '**V**'.

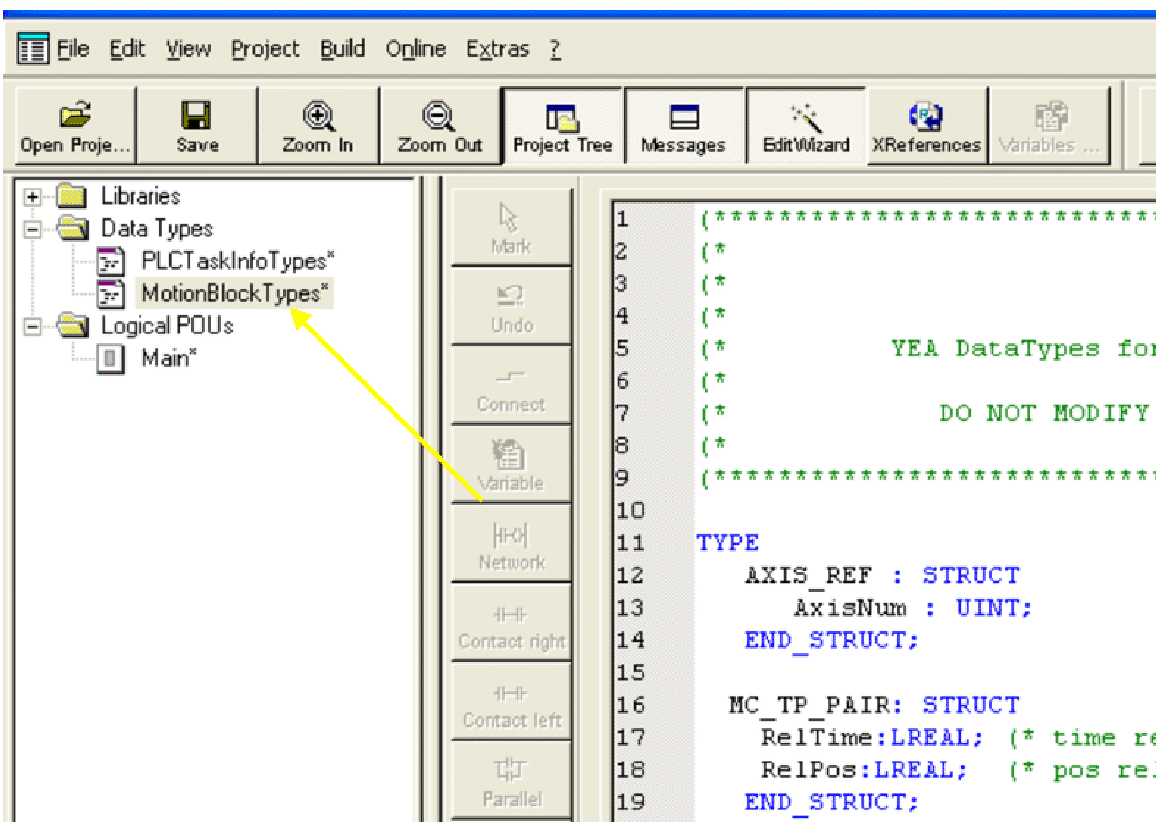
According to the IEC 61131-3 specification, the input variables may be unconnected or not parameterized by the user. In this case, the function block will use the value from the previous invocation of the function block instance, or in case of the first invocation, the initial value will be used.

Data Types

Data Types

A data type can be any simple or complex set of data consisting of multiple data types.

The following data types are supplied by Yaskawa as part of the PLCopenPlus firmware library and will appear in the project tree when a new project is created. The DataType file is named below.



Data Type: AXIS_REF

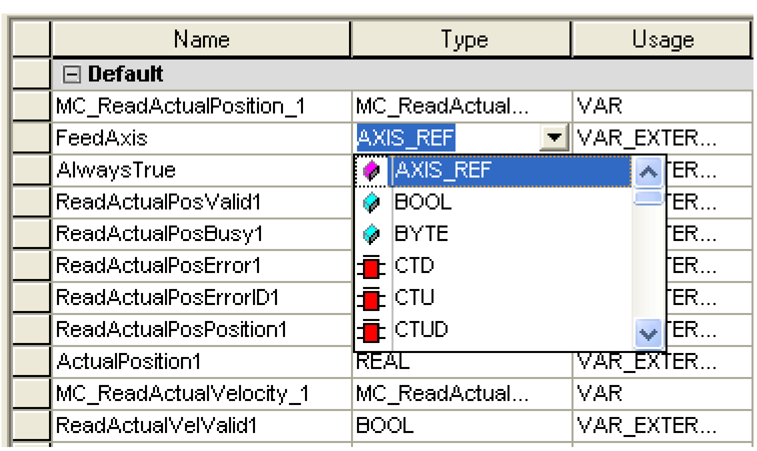
The AXIS_REF data type identifies an axis and thus provides the interface to the hardware or virtual axes. AXIS_REF is used as VAR_IN_OUT in all Motion Control Function Blocks described in this Online help. It is represented as an input and an output connected by a horizontal line in the graphical representation of a function block.

The value of AxisNum is determined by the logical axis number assigned in the Hardware Configuration. See the Configuration tab under each axis.

Data Type Declaration

```
TYPE

    AXIS_REF:STRUCT

        AxisNum:UINT;

    END_STRUCT;

END_TYPE
```

Variable Declaration Example



Code Example

```
AxisX.Number:=UINT#0;
MCMoveAbsoluteX(Axis:=AxisX, Execute:=FALSE);
AxisX:=MCMoveAbsolutX.Axis;
AxisY.Number:=UINT#0;
MCMoveAbsoluteY(Axis:=AxisY, Execute:=FALSE);
AxisX:=MCMoveAbsolutY.Axis;
```

Data Type: CONTINUOUS_REF

This datatype is for use with the Y_ProbeContinuous function block

Data Type Declaration

```
CONTINUOUS_LATCH_RECORD : STRUCT
   ValueCyclic : LREAL;          (* Cyclic latch value (rotary
   modulus)*)
   ValueNonCyclic : LREAL;       (* Non-cyclic latch value *)
   InputID : INT;                (* Input signal ID corresponding
   to the latch data
                                    Indicates C-Channel, EXT1,
   EXT2, EXT3 *)
   PatternIndex : UINT;          (* Signal pattern array index *)
   PatternCount : UINT;          (* Signal pattern repeat count
   *)
   Reserved : UINT;
END_STRUCT;

LATCH_BUFFER_TYP : ARRAY(0..127) OF CONTINUOUS_LATCH_RECORD

CONTINUOUS_REF : STRUCT
   BufferSize : UINT;            (* Maximum number of
   registration marks that will be
                                    tracked by the application at
   any one time *)
   BufferLevel : UINT;           (* Number of registration marks
   in the buffer and
                                    not yet processed by the
   application *)
```

```
  StorePointer : UINT;          (* Array index of the
  LATCH_BUFFER_TYP last stored by

                                   Y_ProbeContinuous *)

  UsePointer : UINT;            (* Array index of the next
  LATCH_BUFFER_TYP to be

                                   used by the application *)

  Buffer : LATCH_BUFFER_TYP;    (* Array of continuous latch
  data *)
END_STRUCT;
```

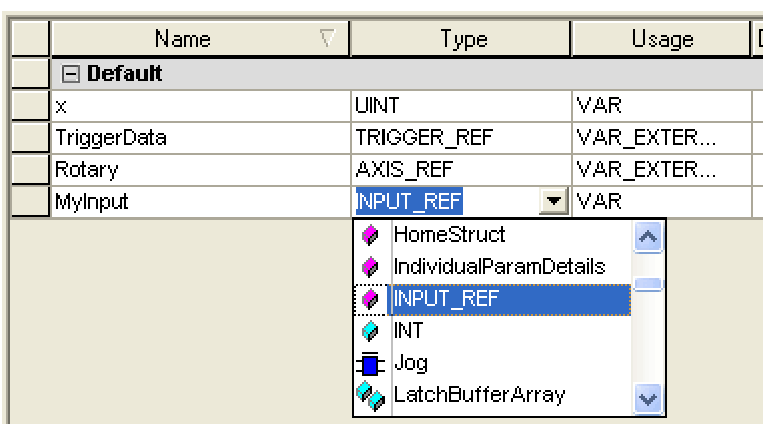
Data Type: INPUT_REF

This datatype is for use with the MC_ReadDigitalInput function block

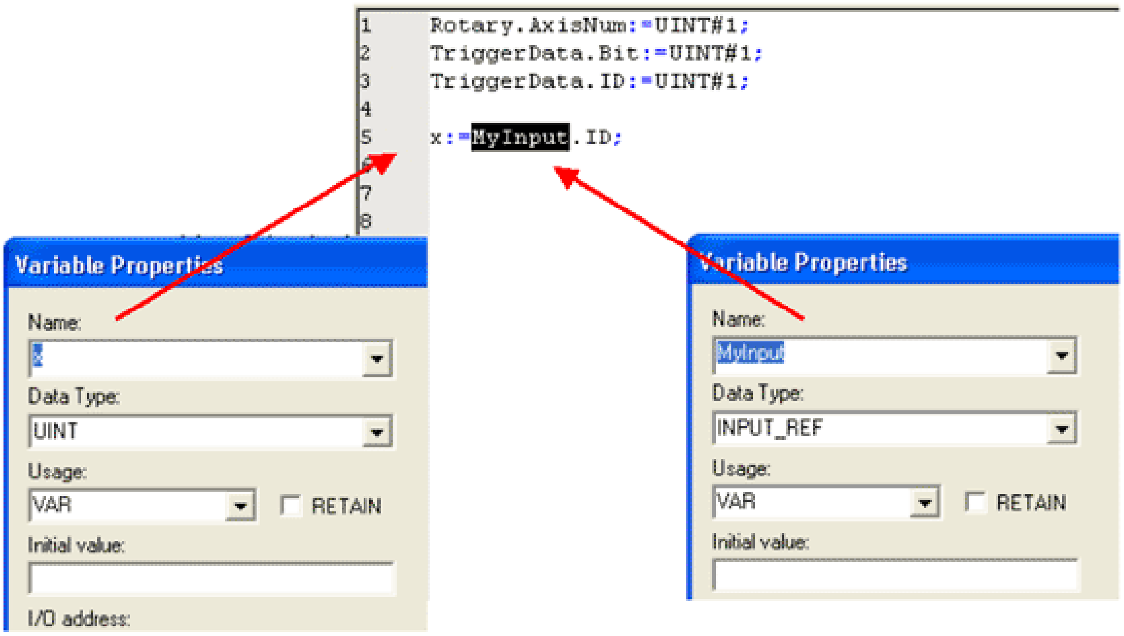
Data Type Declaration

```
TYPE

(* Inputs and outputs are referenced via a variable of the
type INPUT_REF or OUTPUT_REF *)

     INPUT_REF: STRUCT

          ID: UINT; (* Mapping may be required for drive
inputs and C-pulse. These inputs must not been neglected *)

     END_STRUCT;

END_TYPE
```

Variable Declaration Example



Code Example



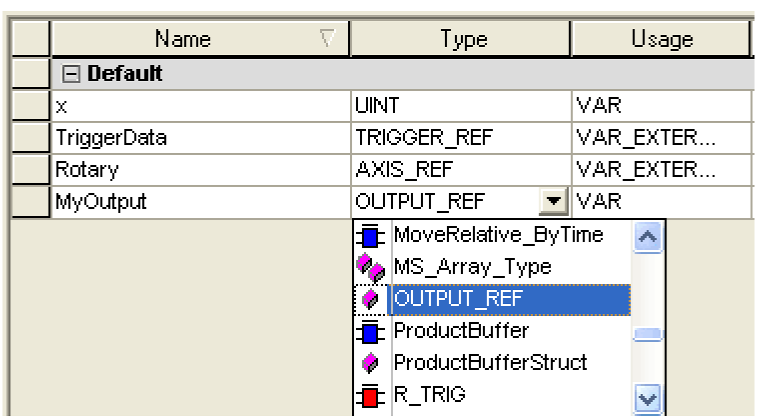
Data Type: OUTPUT_REF

This data type is for use with the MC_WriteDigitalOutput function block.

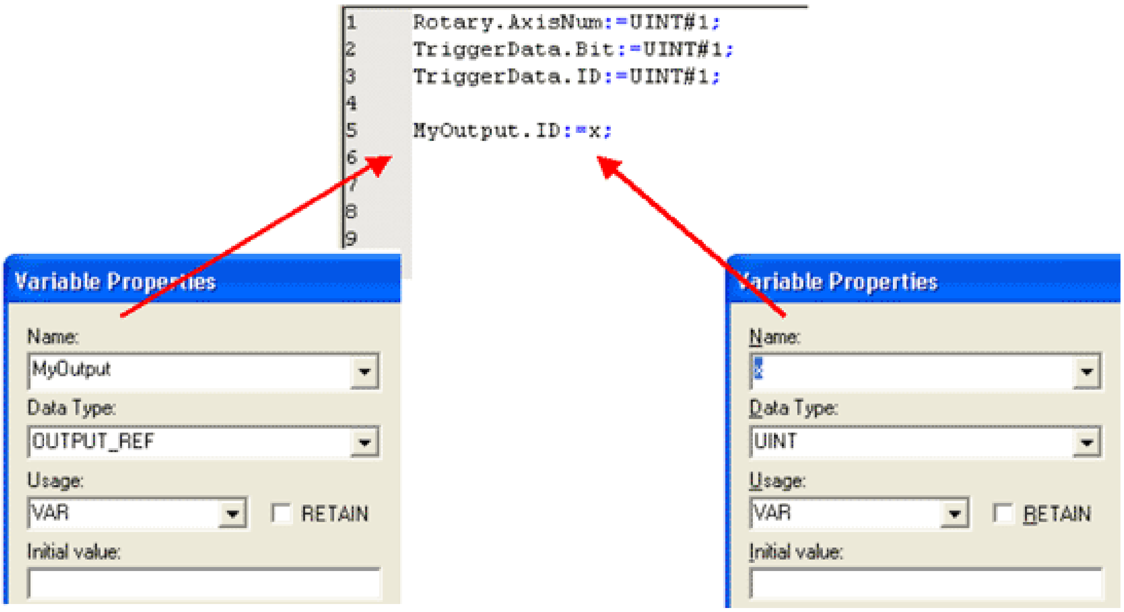
Data Type Declaration

```
TYPE

(* Inputs and outputs are referenced via a variable of the
type INPUT_REF or OUTPUT_REF *)

     OUTPUT_REF: STRUCT

          ID: UINT;    (* The user may output to memory or
hardware. *)

     END_STRUCT;

END_TYPE
```

Variable Declaration Example



Code Example



Data Type: PATTERN_REF

This datatype is for use with the Y_ProbeContinuous function block

Data Type Declaration

```
PATTERN_ARRAY_TYP : ARRAY(0..7) OF UINT;

PATTERN_REF : STRUCT

   PatternSize : UINT;        (* Number of sensors that will
   operate in a repeating

                                 pattern.  Sent to Sigma-5 Pn850
   *)

   PatternCount : UINT;       (* Number of times the pattern
   repeats until the FB

                                 will be done.  UINT#0 =
   infinite.  Sent to Sigma-5

                                 Pn 851 *)

   PatternArray : PATTERN_ARRAY_TYPE;

                              (* Array of signal ID pattern,
   indicating C Channel,

                                 EXT1, EXT2, EXT3.  Sent to Sigma-
   5 Pn853 and Pn854 *)
END_STRUCT;
```

Data Type: PrmStruct

This datatype is for use with the Y_ReadMultipleParameters function block

Data Type Declaration

```
Params : STRUCT
   Number : UINT;            (*  The parameter number to read  *)
   Reserved : UDINT;
   Value : LREAL;            (*  The value of the parameter  *)
END_STRUCT;


ParamList : ARRAY[0..99] OF Params;


PrmStruct : STRUCT
   LastParam : INT;            (*  Indicates the last parameter
   in the list  *)
   Reserved : UDINT;
   ParamData : ParamList;      (*  The array of parameter numbers
   and values  *)
END_STRUCT;
```

Data Type: RTC_STRUCT

This datatype is for use with the Y_SetRTC function block

Data Type Declaration

```
RTC_Struct:STRUCT
  Year:INT;
  Month:INT;
  Day:INT;
  Hour:INT;
  Minute:INT;
  Second:INT;
  Millisecond:INT;
END_STRUCT;
```

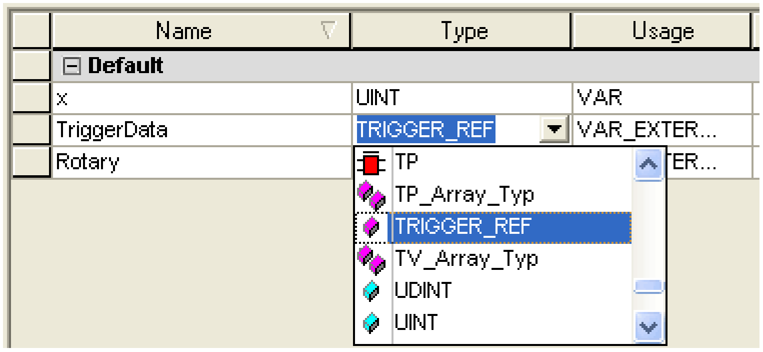
Data Type: TRIGGER_REF

This data type is for use with the MC_TouchProbe and MC_AbortTrigger function blocks.

Data Type Declaration

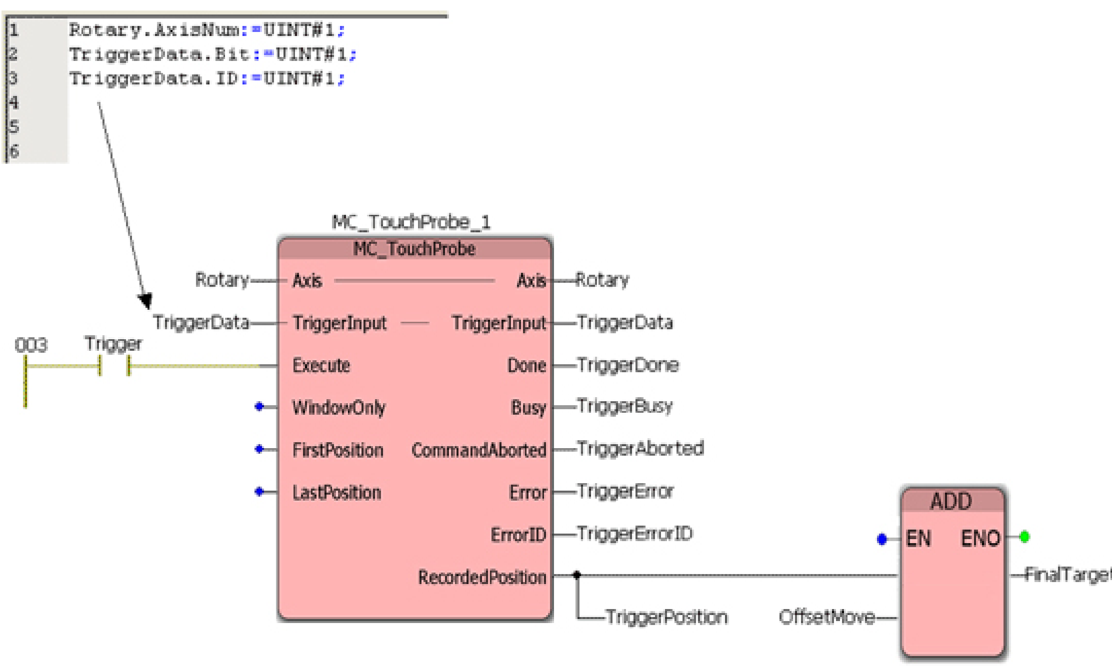
```
TYPE

(* MC_TouchProbe requires a trigger referenced via a variable
of the type TRIGGER_REF *)

     Detection_Pattern:(Rising_Edge, Falling_Edge);    (*  Not
used  *)


     TRIGGER_REF: STRUCT

          Input: INPUT_REF;

          Bit: UINT;

          Pattern: DETECTION_PATTERN;

          ID: UINT; (* Unique identification of the trigger;
used for MC_AbortTrigger *)

     END_STRUCT;

END_TYPE
```

Variable Declaration Example



The following chart details the correct values for the TRIGGER_REF structure based on the hardware latch to be detected.

Code Example



Data Type: Y_DISENGAGE_DATA

This data type is for use with the Y_CamOut function block.

Data Type Declaration

```
TYPE

   Y_Disengage_Data : STRUCT

      EndMode          : INT;    (* Possible values are
described in Y_DisengageMethod *)

      RampOut          : INT;    (* Reserved for future use  *)

      RampOutData1     : LREAL;  (* Reserved for future use  *)

      RampOutData2     : LREAL;  (* Reserved for future
use  *)

      RampOutData3     : LREAL;  (* Reserved for future
use  *)

      RampOutData4     : LREAL;  (* Reserved for future
use  *)

   END_STRUCT;

END_TYPE;

Y_DisengageMethod: (AtPosition, Immediate, EndOfProfile);

                     (* Immediate and EndofProfile Reserved for
future use  *)
```

Data Type: Y_ENGAGE_DATA

This data type is for use with the Y_CamIn function block.

Data Type Declaration

```
TYPE
   Y_Engage_Data : STRUCT
      StartMode      : INT;    (* Possible values are
described in Y_EngageMethod *)
      MasterRelative : BOOL;
      SlaveAbsolute  : BOOL;
      RampIn         : INT;    (* Reserved for future use
*)
      RampInData1    : LREAL;  (* Reserved for future use
*)
      RampInData2    : LREAL;  (* Reserved for future use
*)
      RampInData3    : LREAL;  (* Reserved for future use
*)
      RampInData4    : LREAL;  (* Reserved for future use
*)
   END_STRUCT;
END_TYPE;
Y_EngageMethod: (AtPosition, Immediate, Linked);
```

Data Type: Y_MS_CAM_STRUCT

This data type is for use with the Y_CamStructSelect, Y_ReadCamTable, and Y_WriteCamTable function blocks. Y_MS_CAM_STRUCT consists of the sub-structures found below. Refer to the Internally Created Cam Data diagram in the Cam Data Management section.

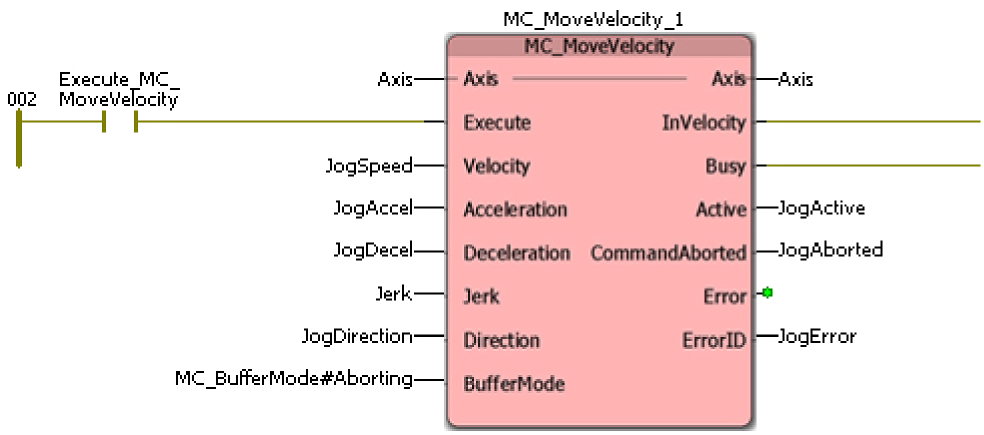
Data Type Declaration

```
TYPE

    Y_CAM_HEADER:STRUCT

        TableType:INT;      (*  INT#1 = Master/Slave
pair  *)

        Reserved1:UINT;

        DataSize:UDINT;     (*  Size of cam table in bytes.
There are 16 bytes

                               (8 Master/8 Slave)per
Y_MS_PAIR.  For example, if

                                your CAM profile has 360
data pairs, then the data

                            size is 360 pairs x 16 bytes =
5760 bytes  *)

    END_STRUCT;


    Y_MS_PAIR: STRUCT

        Master:LREAL;       (*  Master position  *)

        Slave:LREAL;        (*  Slave position   *)

    END_STRUCT;
```

```
    Y_MS_HEADER:STRUCT
        SlaveIncremental:BOOL;
        MasterIncremental:BOOL;
        Reserved1:UINT;
        Reserved2:UINT;
        Reserved3:UINT;
    END_STRUCT;

    MS_Array_Type:ARRAY[0..512] OF Y_MS_PAIR;

    Y_MS_CAM_STRUCT:STRUCT
        Header:Y_CAM_HEADER;
        MS_Header:Y_MS_HEADER;
        MS_Data:MS_Array_Type;
    END_STRUCT;
END_TYPE
```

Enumerated Types

Some blocks accept an enumerated type, which is a keyword (or constant) representing a value which will configure the operation of the function block. Enumerated types are equivalent to zero-based integers. Therefore, the first value equates to zero, the second to 1, etc. The format for enumerated types is as follows: ENUM:(0, 1, 2...) as displayed in the example below (MC_BufferMode#Aborting).



MC_BufferMode:(Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, BlendingHigh)

MC_Detection_Pattern: (Rising_Edge, Falling_Edge)

MC_Direction: (Positive_Direction, Shortest_Way, Negative_Direction, Current_Direction)

(* MC_Direction#ShortestWay and MC_Direction#Current_Direction are designed for use in applications where the Load Type is configured as a rotary or modularized axis. Additionally, MC_Direction#Current_Direction only applies if an existing move is in progress and another function block such as MC_MoveAbsolute or MC_MoveRelative is executed. Once the axis is at StandStill, using MC_Direction_CurrentDirection will default to the positive direction. *)

MC_SwitchMode:(On, Off, EdgeOn, EdgeOff, EdgeSwitchPositive, EdgeSwitchNegative) (* Only MC_SwitchMode#EdgeOn is supported *)

Y_AdjustMode: (MasterDistance, ElapsedTime, WithinRange)

- If AdjustMode=Y_AdjustMode#MasterDistance, then the cam adjustment starts immediately, and completes when the master has travelled the specified distance. If MasterDistance is 0.0, then the cam adjustment finishes in the same scan it starts.

- If AdjustMode=Y_AdjustMode#ElapsedTime, then the cam adjustment starts immediately, and completes within the specified time. If time=0.0, then the adjustment completes in the same scan it starts.

- If AdjustMode=Y_AdjustMode#WithinRange, then the cam adjustment starts when the master is crosses the StartPosition, and completes when the master reaches the EndPosition. If the master position is already between StartPosition and EndPosition, then the adjustment starts immediately, but still completes at the EndPosition, which means that the correction speeds may be higher.

Y_RampIn:(None, Accel, SCurve) - Reserved for future use.

Y_RampOut: Reserved for future use.

Y_EngageMethod:(AtPosition, Immediate, Linked): This enumerated type is reserved for Y_CamIn

Y_DisengageMethod:(AtPosition, Immediate, EndOfProfile): This enumerated type is reserved for Y_CamOut

Function Block List

This online help provides information about the function blocks which can be used for motion control. The function blocks are divided into single-axis and multi-axis motion blocks and administrative function blocks which do not generate a movement.

- Single-Axis Motion Function Blocks
- Single-Axis Administrative Function Blocks
- Multi-Axis Motion Function Blocks
- Multi-Axis Administrative Function Blocks
- Homing Function Blocks

Function Block	Support	Short description
Single-Axis Motion Function Blocks		
MC_AccelerationProfile	None	Commands an activation of a positioning task as an array which describes the acceleration of an axis depending on the time.
MC_GroupSyncOut	None	
MC_Halt	Future	

MC_Home	None	Obsolete function block. Please use Part 5 Homing Function Blocks to perform Homing Functions (i.e. MC_Step...)
MC_MoveAbsolute	Ver. 1.0	Commands a controlled motion of the axis at a specified absolute position.
MC_MoveAdditive	Future	Commands a controlled motion of a specified relative distance additional to the original commanded position in the discrete motion state.
MC_MoveContinuous	Future	
MC_MovePath	None	
MC_MoveRelative	Ver. 1.0	Commands a controlled motion of a specified distance relative to the actual

		position at the time of the execution.
MC_MoveSuperImposed	Ver. 1.0	Commands a controlled motion of a specified relative distance additional to an existing motion.
MC_MoveVelocity	Ver. 1.0	Commands a never ending controlled motion at a specified velocity.
MC_PathGearIn	None	
MC_Stop	Ver. 1.0	Commands a controlled motion stop of an axis.
MC_PositionProfile	Future	Commands an activation of a positioning task as an array which describes the positions of an axis depending on the time.
MC_TorqueControl	Ver. 1.0	
MC_VelocityProfile	Future	Commands an activation of a

		positioning task as an array which describes the velocity of an axis depending on the time.
Y_HoldPosition	Ver. 1.0.5.1	Puts the servo in position mode and freezes the profiler.
Single-Axis Administrative Function Blocks		
MC_AbortTrigger	Ver. 1.0	Aborts function blocks which are connected to trigger events.
MC_DigitalCamSwitch	Refer to the PLCopen Toolbox available on www.yaskawa.com	Provides output control based on a position input and speed.
MC_Power	Ver. 1.0	Sets or resets the enabling for an axis.
MC_ReadActualPosition	Ver. 1.0	Reads the actual position of the axis.
MC_ReadActualTorque	Ver. 1.0	Reads the actual torque of the axis.
MC_ReadActualVelocity	Ver. 1.0	Reads the actual velocity of the

		axis.
MC_ReadAxisError	Ver. 1.0	Indicates an axis error and allows to read the error.
MC_ReadBoolParameter	Ver. 1.0	Reads the axis parameters of the data type BOOL.
MC_ReadDigitalInput	Future	Function block not necessary to read inputs.
MC_ReadDigitalOutput	Future	Function block not necessary to set outputs.
MC_ReadParameter	Ver. 1.0	Reads the axis parameters.
MC_ReadStatus	Ver. 1.0	Returns the status of the axis with respect to the motion currently in progress.
MC_Reset	Ver. 1.0	Acknowledges an existing error message.
MC_SetOverride	Future	Sets the values of override for the whole axis, and all functions that are working on that axis.

MC_SetPosition	Ver. 1.0	Sets the current position of an axis to a new position and thus shifts the coordinate system.
MC_TouchProbe	Ver. 1.0	Record an axis position at a trigger event.
MC_WriteBoolParameter	Ver. 1.0	Writes the axis parameters of the data type BOOL.
MC_WriteDigitalOutput	Future	Writes a value to the output referenced by the argument 'Output'. Function block not necessary to write outputs.
MC_WriteParameter	Ver. 1.0	Writes the axis parameters.
Y_ClearAlarms	Ver. 1.0	Clears non-axis-related controller alarms
Y_DirectControl	Ver. 1.0.7.4	Allows direct access to any of three possible control modes

		available on the MECHATROLINK network servo control system.
Y_ReadAlarm	Ver. 1.0	Reads non-axis-related controller alarms
Y_ReadDriveParameter	Ver. 1.0.5.1	Reads a parameter from the associated motor driver
Y_VerifyParameters	Ver. 1.1	Compares parameters in the drive with those stored in the controller
Y_WriteDriveParameter	Ver. 1.0.5.1	Writes a parameter from the associated motor driver
Y_WriteParameters	Ver. 1.1	Sends parameters stored in the controller to the drive
Multi-Axis Motion Function Blocks		
MC_GearIn	Ver. 1.0	Activates an electronic velocity gearing between a slave and

		master axis.
MC_GearInPosition	Ver. 1.0	Commands a gear ratio between the position of the slave and master axes from the synchronization point onwards.
MC_GearOut	Ver. 1.0	Deactivates the electronic velocity gearing between a slave and master axis.
MC_GroupHalt	None	
MC_GroupHome	None	
MC_GroupStop	None	
MC_MoveCircularAbsolute	None	
MC_MoveCircularRelative	None	
MC_MoveDirectAbsolute	None	
MC_MoveDirectRelative	None	
MC_MoveLinearAbsolute	None	
MC_MoveLinearRelative	None	
MC_MovePathSynchronized	None	
MC_MovePositionDirectRelative	None	
MC_Phasing	Future	Creates a phase shift in the master position of a slave axis.

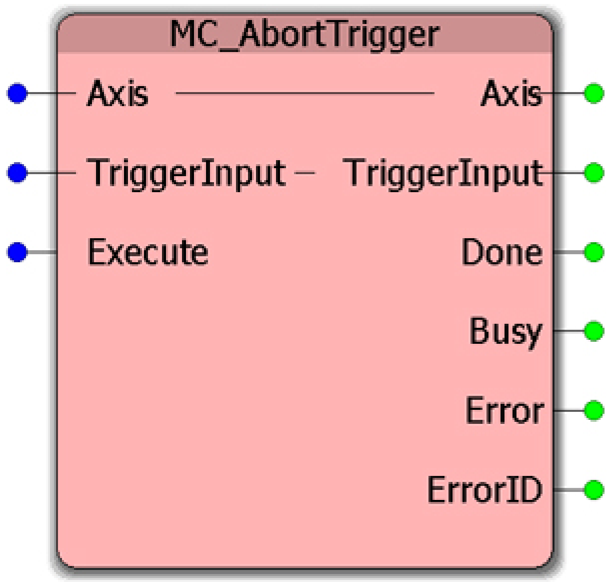
MC_TrackConveyorBelt	None	
MC_TrackRotaryTable	None	
Y_CamFileSelect	Ver. 1.1	
Y_CamIn	Ver. 1.1	Activates the coupling between master and slave axis.
Y_CamOut	Ver. 1.1	Deactivates the coupling of the slave axis with the master axis.
Y_CamScale	Ver. 1.1	Multiplication factor applied to the slave data
Y_CamShift	Ver. 1.1	
Y_CamStructSelect	Ver. 1.1	Loads cam data from the application program into motion memory
Y_ReadCamTable	Ver. 1.1	Copies cam data from motion memory into the application program
Y_ReleaseCamTable	Ver. 1.1	Frees motion memory and CamTableID

Y_ResetMechatrolink	Ver. 1.0.5.1	Restarts the MECHATROLINK network
Y_SlaveOffset	Ver. 1.1	Adds an offset to the slave data
Y_WriteCamTable	Ver. 1.1	Copies cam data from the application program to the motion memory
Multi-Axis Administrative Function Blocks		
MC_AddAxisToGroup	None	
MC_CamTableSelect	None	See Y_CamTableSelect
MC_GroupDisable	None	
MC_GroupEnable	None	
MC_GroupReadActualAcceleration	None	
MC_GroupReadActualPosition	None	
MC_GroupReadActualVelocity	None	
MC_GroupReadConfiguration	None	
MC_GroupReadError	None	
MC_GroupReadStatus	None	
MC_GroupReset	None	
MC_GroupSetOverride	None	
MC_GroupSetPosition	None	
MC_PathSelect	None	
MC_RemoveAxisFromGroup	None	

MC_SetCartesianTransform	None	
MC_SetCoordinateTransform	None	
MC_SetDynCoordTransform	None	
MC_SetKinTransform	None	
MC_SyncAxisToGroup	None	
MC_UngroupAllAxes	None	
Homing Function Blocks		
MC_AbortPassiveHoming	Future	
MC_FinishHoming	Ver. 1.0	Transfers an axis from 'Homing' state to 'Standstill' state.
MC_StepAbsolute	None	This function is not required with Yaskawa absolute encoders.
MC_StepAbsSwitch	Future	
MC_StepBlock	Future	
MC_StepDirect	Future	
MC_StepLimitSwitch	Ver. 1.0	Performs homing by searching for a limit switch.
MC_StepReferenceFlyingRefPulse	Future	
MC_StepReferenceFlyingSwitch	Future	
MC_StepRefPulse	Ver. 1.0	Performs homing by searching for a Zero pulse.

Function Blocks for Motion Control

MC_AbortTrigger



The Function Block aborts function blocks which are associated with trigger events (e.g. MC_TouchProbe).

Parameters

Parameter		Data type	Description
VAR_IN_OUT			
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).
E	TriggerInput	TRIGGER_REF	Reference to the trigger signal

			source. See MC_TouchProbe	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset	

			when 'Execute' or 'Enable' goes low.

Notes

The following chart details the correct values for the TRIGGER_REF structure based on the hardware latch to be detected.

Related Function Blocks

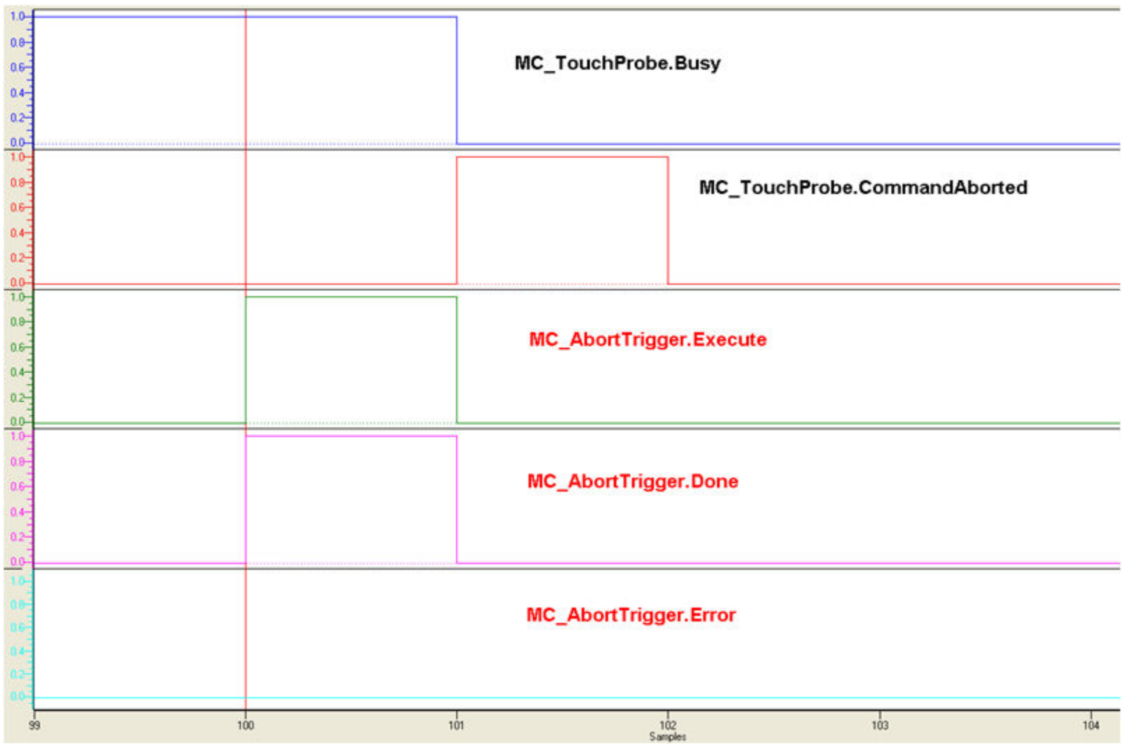
MC_TouchProbe

Error Description

ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4630	Trigger or pattern reference is not valid
57620	The structure size does not match.

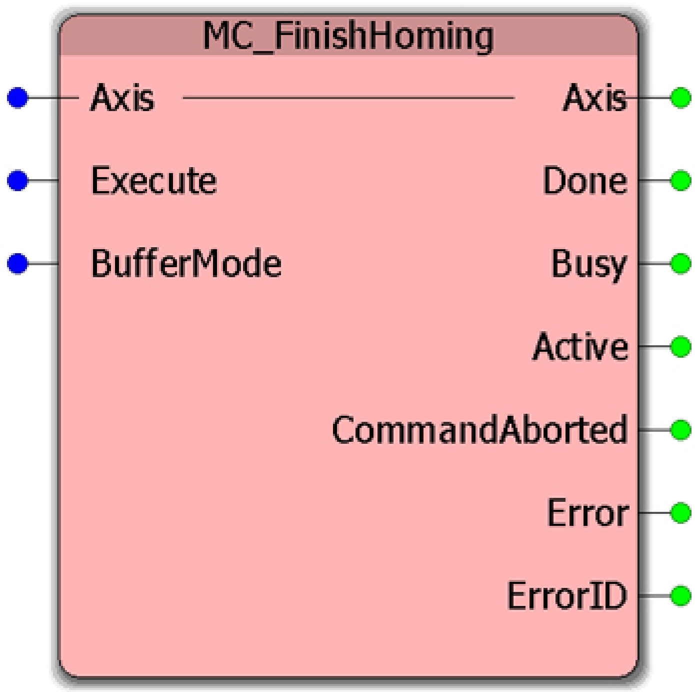
Example

The following example shows how MC_AbortTrigger cancels (Aborts) a busy MC_TouchProbe.



Timing Diagram for MC_AbortTrigger

MC_FinishHoming



This FB transfers an axis from the ‘Homing’ state to the ‘StandStill’ state. It does not perform any movement. This block is necessary after the user builds a homing procedure containing any number of MC_StepXXXX homing blocks (See Notes).

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	False

E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

This block is not necessary if the last homing block executed is MC_StepRefPulse, MC_StepDirect, or MC_StepAbsolute because these blocks will change the motion state back to 'Standstill' when complete.

This block is only necessary if the following homing blocks are last in a homing sequence:

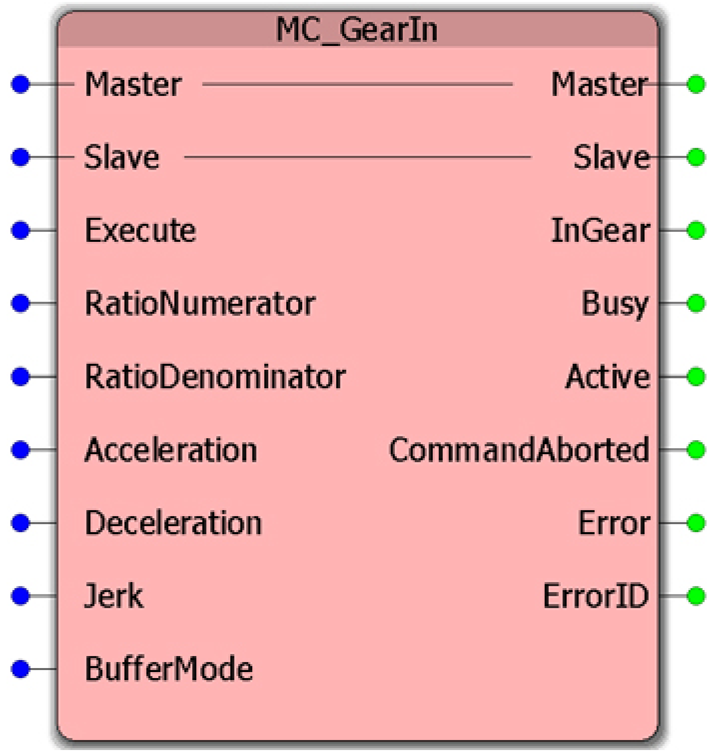
- MC_StepAbsSwitch

- MC_StepLimitSwitch
- MC_StepBlock.

Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4641	Buffer mode does not correspond to a valid enumeration value.
4893	The specified external axis may not be used. A physical axis is required
57620	The structure size does not match.

MC_GearIn



This Function Block commands a velocity ratio between the master and slave axes. If the gearing mode is executed while the master is already in motion, position synchronization will not be achieved unless some other method (MC_MoveSuperimposed or MC_GearInPos) is employed.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Master	AXIS_REF	A logical reference to the master axis	
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and	FALSE

			re-trigger the execute input.	
B	RatioNumerator	DINT	Gear ratio numerator	DINT#0
B	RatioDenominator	DINT	Gear ratio denominator	DINT#1
E	Acceleration	LREAL	Value of the acceleration in user units/second^2 (acceleration is applicable with same sign of torque and velocity)	LREAL#0.0
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	InGear	BOOL	Set high upon successful completion of the function. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

1. The slave accelerates up to the ratio of the master velocity and become "InGear" when ratio is reached. Compensation for any position relationship lost during the acceleration phase is not provided during synchronization. Use MC_GearInPos when maintaining a position relationship is important.

2. The gearing ratio can be changed while MC_GearIn is running, using a consecutive MC_GearIn command or re-triggering the 'Execute' input without the necessity to MC_GearOut first.

3. The InGear output is set the first time the specified ratio is reached.

Related Function Blocks

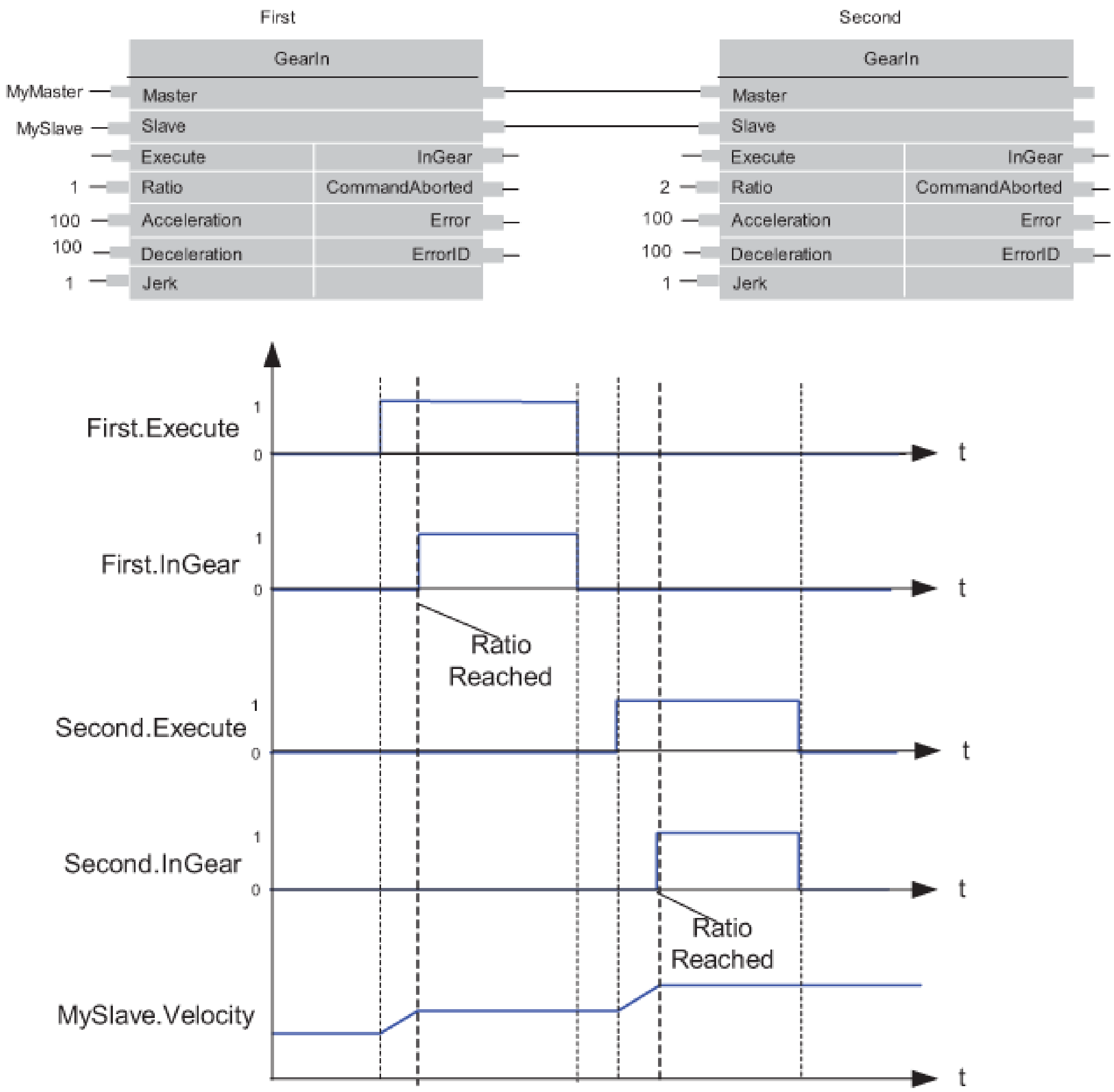
MC_GearOut: disengages the Slave axis from the master axis.

Error Description

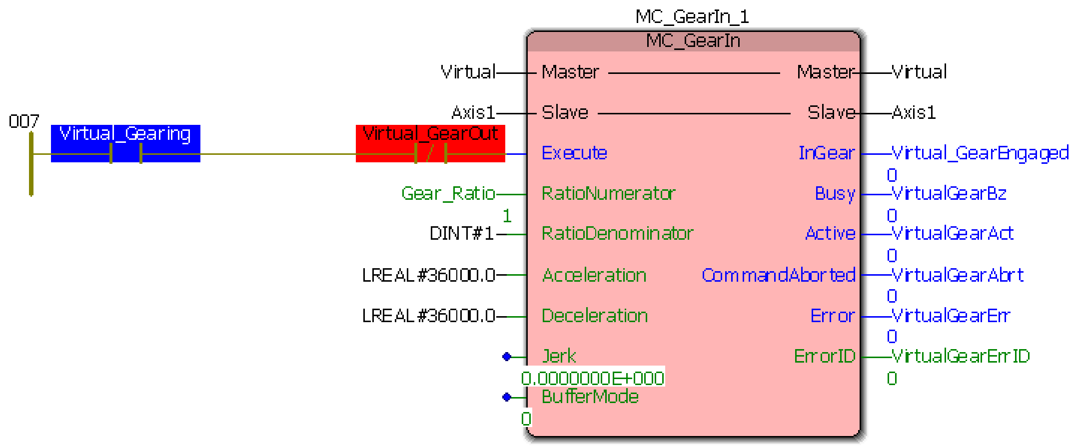
ErrorID	Meaning
0	No Error
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion

	block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4626	The master slave relationship is defined. A slave cannot be a master to another axis.
4641	Buffer mode does not correspond to a valid enumeration value.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
4666	Denominator is zero.
4667	Jerk is less than or equal to zero
4891	The slave axis can not be the same as the master axis.
57620	The structure size does not match.

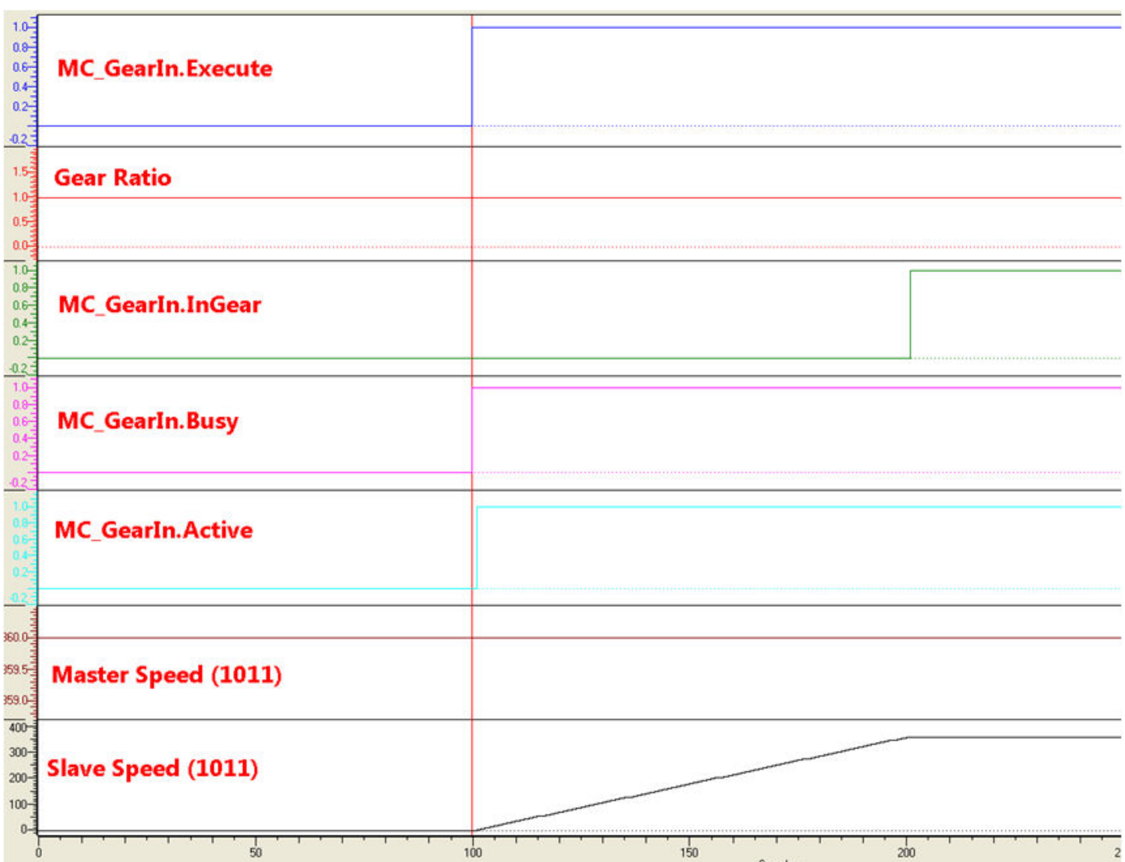
Timing Diagram



Example

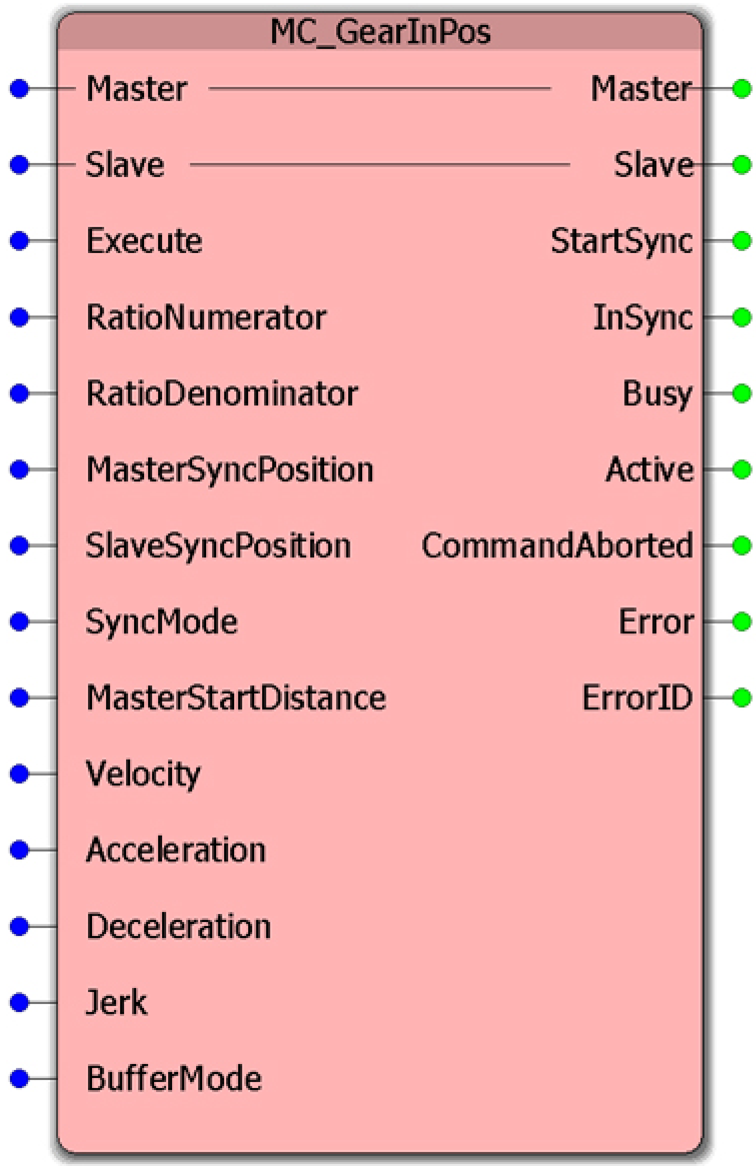


MC_GearIn_1
MC_GearIn
007
Virtual_Gearing
Virtual_GearOut
Virtual
Master
Master
Virtual
Axis1
Slave
Slave
Axis1
Execute
InGear
Virtual_GearEngaged
0
Gear_Ratio
RatioNumerator
Busy
VirtualGearBz
1
0
DINT#1
RatioDenominator
Active
VirtualGearAct
0
LREAL#36000.0
Acceleration
CommandAborted
VirtualGearAbrt
0
LREAL#36000.0
Deceleration
Error
VirtualGearErr
0
Jerk
ErrorID
VirtualGearErrID
0.0000000E+000
0
BufferMode
0



MC_GearIn.Execute
Gear Ratio
MC_GearIn.InGear
MC_GearIn.Busy
MC_GearIn.Active
Master Speed (1011)
Slave Speed (1011)
Samples

MC_GearInPos



This Function Block commands a gear ratio between the master and slave axes. Position synchronization is achieved over a defined region of travel for both master and slave.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Master	AXIS_REF	A logical reference to the master axis	
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	RatioNumerator	DINT	Gear ratio numerator	DINT#0
B	RatioDenominator	DINT	Gear ratio denominator	DINT#1
B	MasterSyncPosition	LREAL	Master Position at which the axes are synchronized	LREAL#0.0
B	SlaveSyncPosition	LREAL	Slave position at which the axes are synchronized	LREAL#0.0
E	SyncMode	INT	Reserved for future use	INT#0
E	MasterStartDistance	LREAL	Master Distance for synchronization procedure. See Note Below	LREAL#0.0
E	Velocity	LREAL	Maximum Velocity allowed by the slave during 'StartSync' to the 'InSync' event	LREAL#0.0
E	Acceleration	LREAL	Acceleration limit while attempting to Engage	LREAL#0.0
E	Deceleration	LREAL	Deceleration limit while attempting to Engage	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered	MC_BufferMode#Aborting

			MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	
VAR_OUTPUT				
E	StartSync	BOOL	The slave has started to synchronize, but not yet synchronized with the master	
B	InSync	BOOL	Set high when the slave first synchronizes with the master. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
B	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- If the master is stationary when the MC_GearInPos function is executed, it will remain busy until motion occurs.
- Only one SyncMode is supported: MC_SyncMode#Acc_Vel_Dec uses the input parameters Acceleration, Velocity, & Deceleration to make a move to the SlaveSyncPosition. The slave may attain synchronization early if these parameters are set higher than optimally required. If these parameters will not allow the slave to engage by the time the master reached the MasterSyncPosition, an error will result.
- MasterStartDistance and MasterSyncPosition are in units of the specified master.

- MasterDistance is a relative distance from the desired synchronization point. The slave will start the synchronization process when the master is within this range of the MasterSyncPosition.

Error Description

ErrorID	Meaning
0	No Error
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4626	The master slave relationship is defined. A slave cannot be a

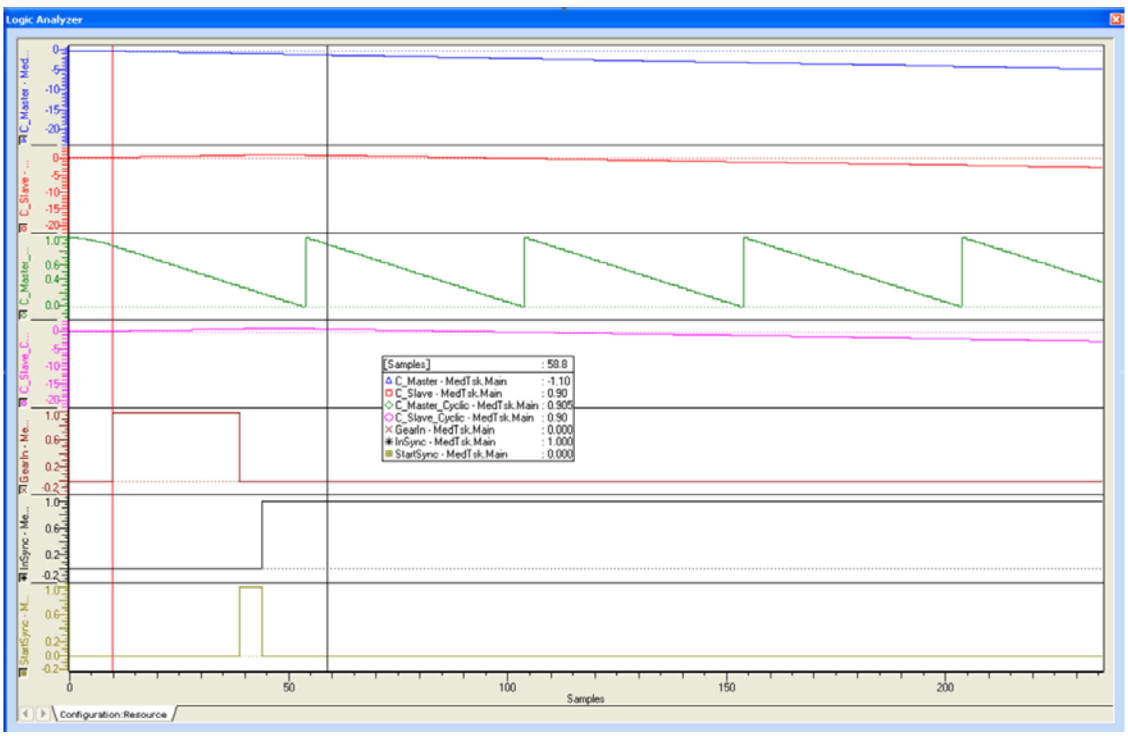
	master to another axis.
4641	Buffer mode does not correspond to a valid enumeration value.
4647	The synch mode does not correspond to a valid enumeration value.
4657	Distance parameter is less than or equal to zero.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
4666	Denominator is zero.
4669	Engage position is outside the cam table domain.
4889	The engage phase exceeded the distance limit. Slave axis could not attain the target position and velocity within the user specified master distance.
4891	The slave axis can not be the same as the master axis.
57620	The structure size does not match.

Usage Matrix Chart

Examples

Scenario 22:

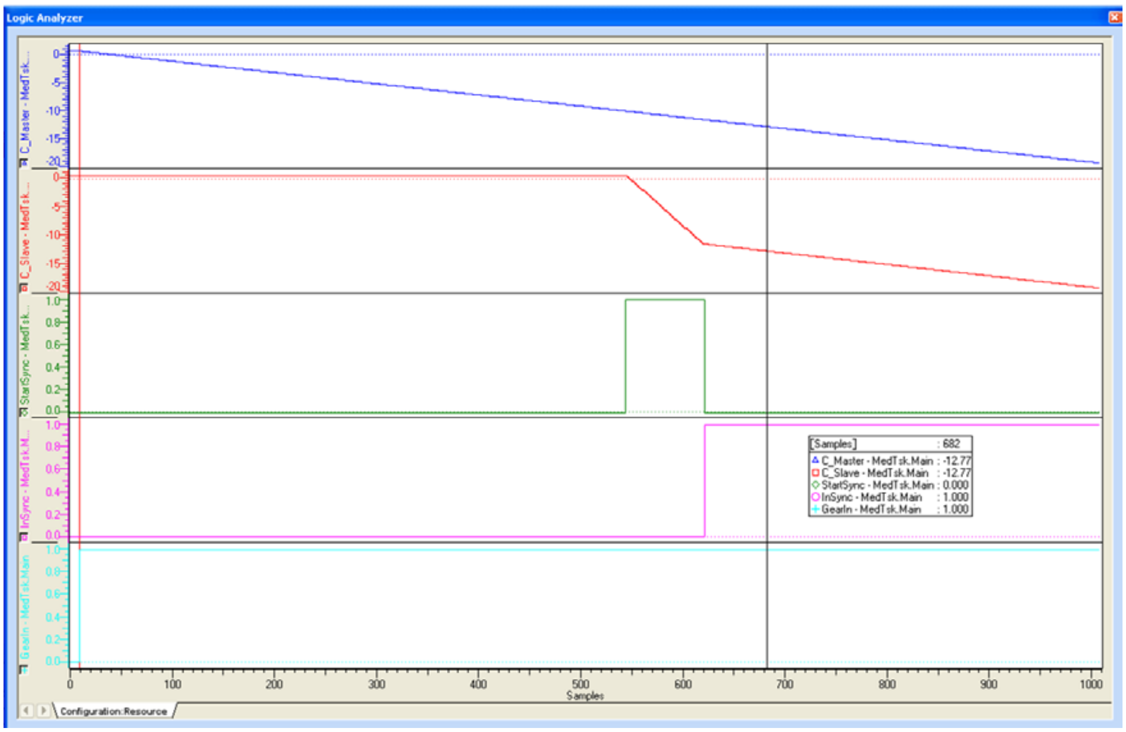
A logic analyzer plot of the commanded position (Prm 1010) of the master and the slave before and after' InSync' in is shown below.



Scenario 30:

A logic analyzer plot of the commanded position (Prm 1010) of the master and the slave before and after' InSync' in is shown below.

Recall that the requirements for this scenario state that the MasterStartDistance and the MasterSyncDistance must be negative.



Different examples of MC_GearInPos

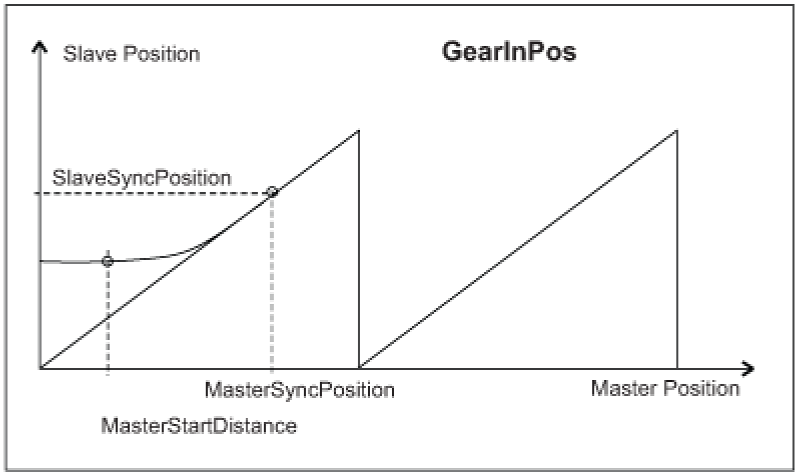


Figure 10.1

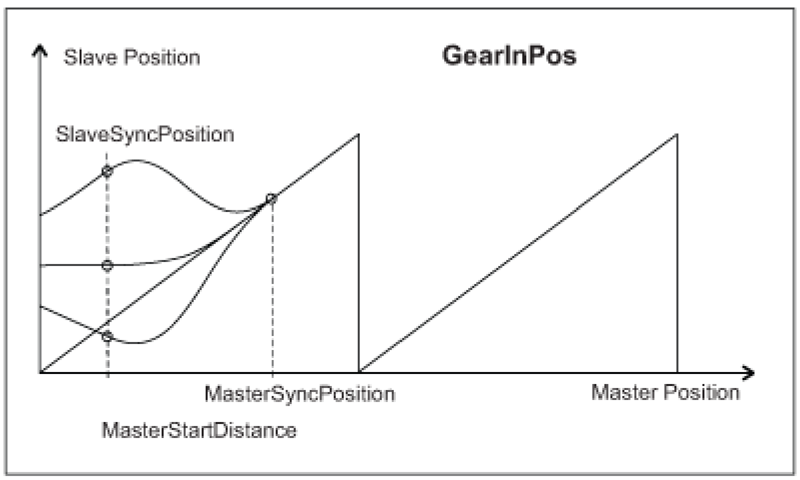


Figure 10.2

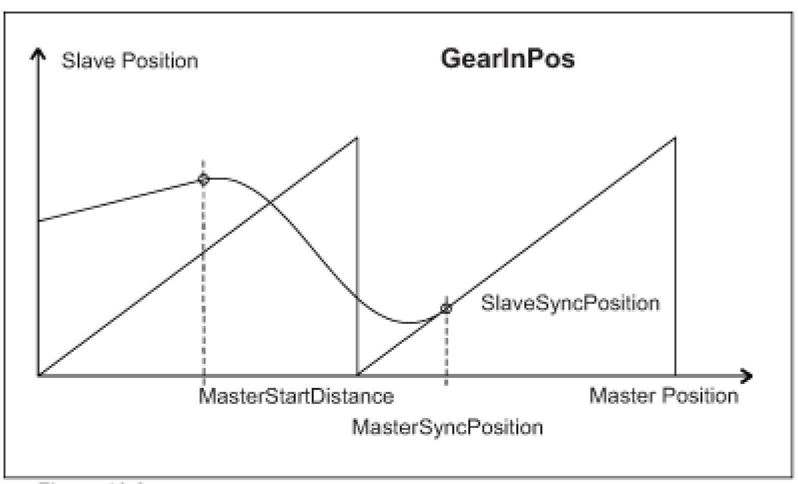
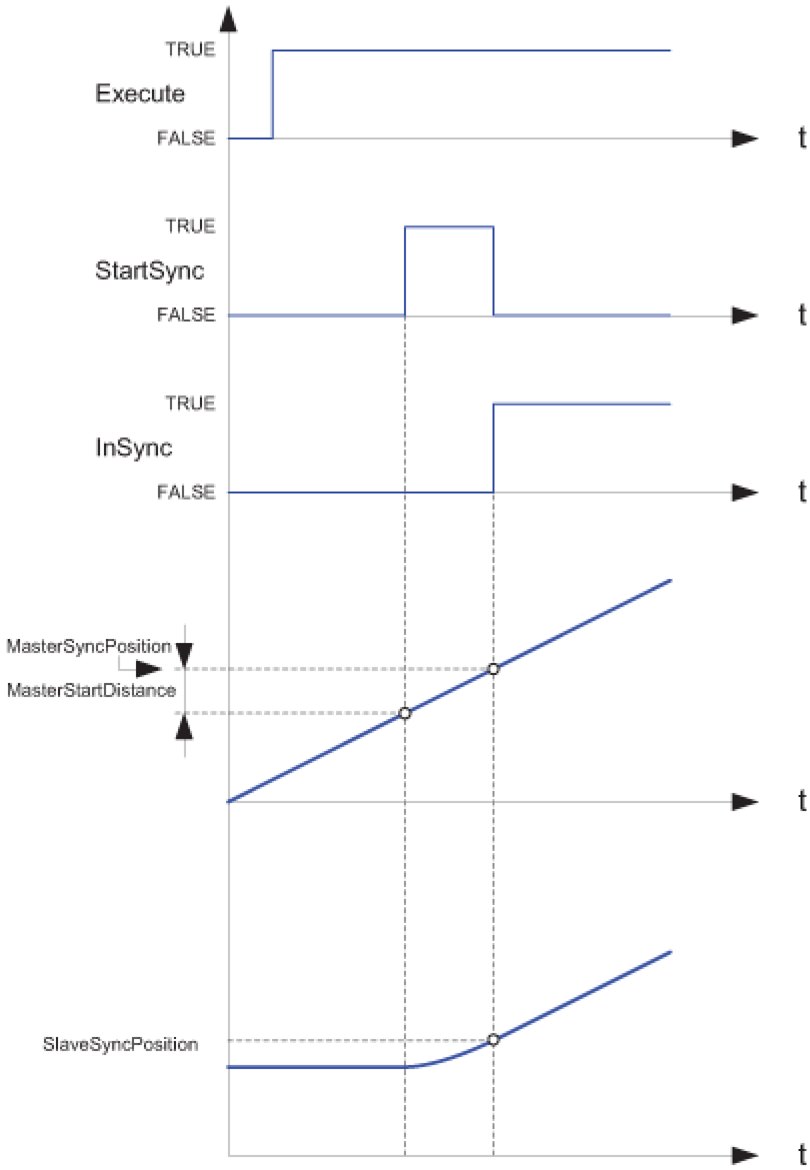


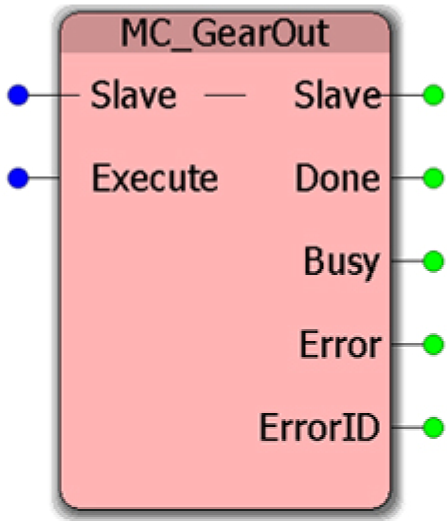
Figure 10.3

Timing Diagram

Timing Diagram of MC_GearInPos



MC_GearOut



This Function Block disengages the Slave axis from the Master axis. The slave will continue to move at the last commanded velocity.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset	

			when 'Execute' or 'Enable' goes low.

Notes

It is assumed that this command is followed by another command, for instance MC_Stop, MC_GearIn, or any other command. If there is no new command, the default condition will be to maintain the last geared velocity.

Related Function Blocks

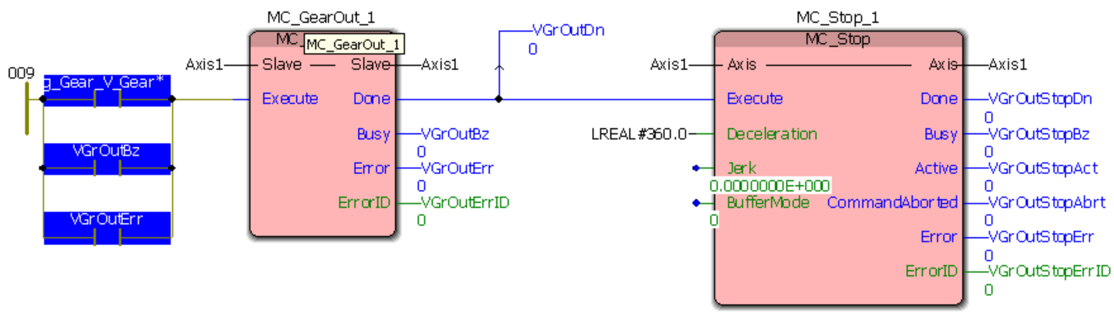
MC_GearIn: Commands a velocity ratio between the master and slave axes.

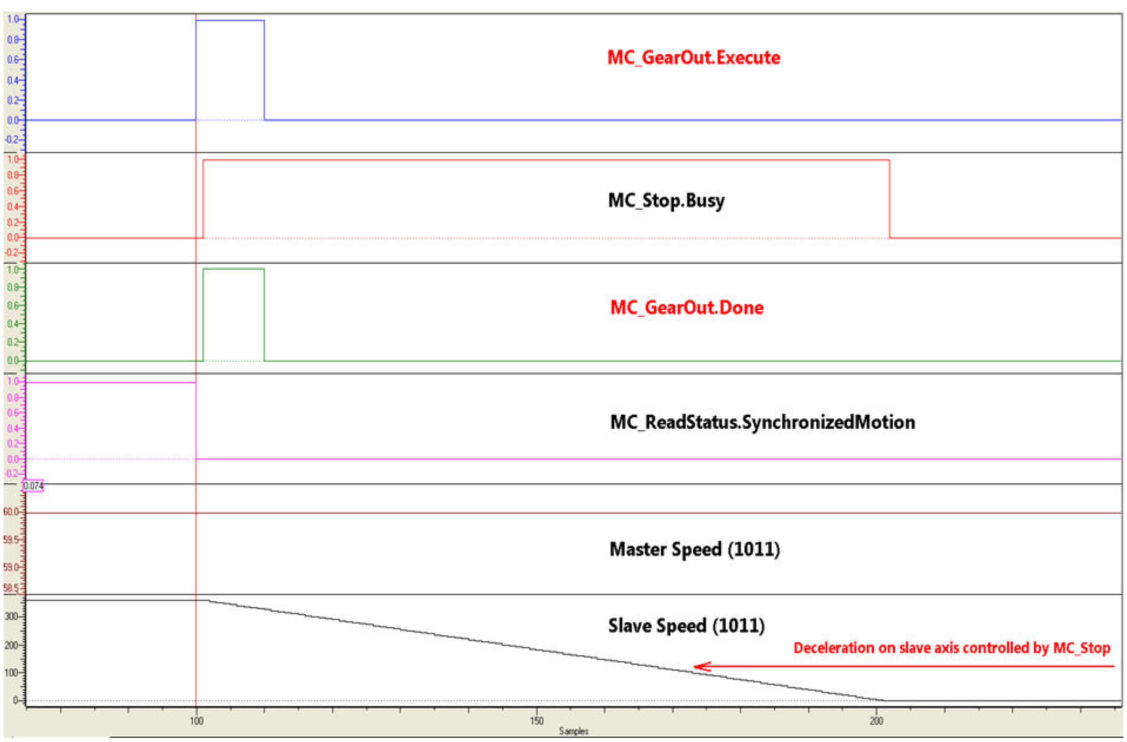
Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4376	The master slave relationship can not be modified because the master axis has not been set yet.
4404	Can not execute MC_GearOut because axis is not in gear
57620	The structure size does not match.

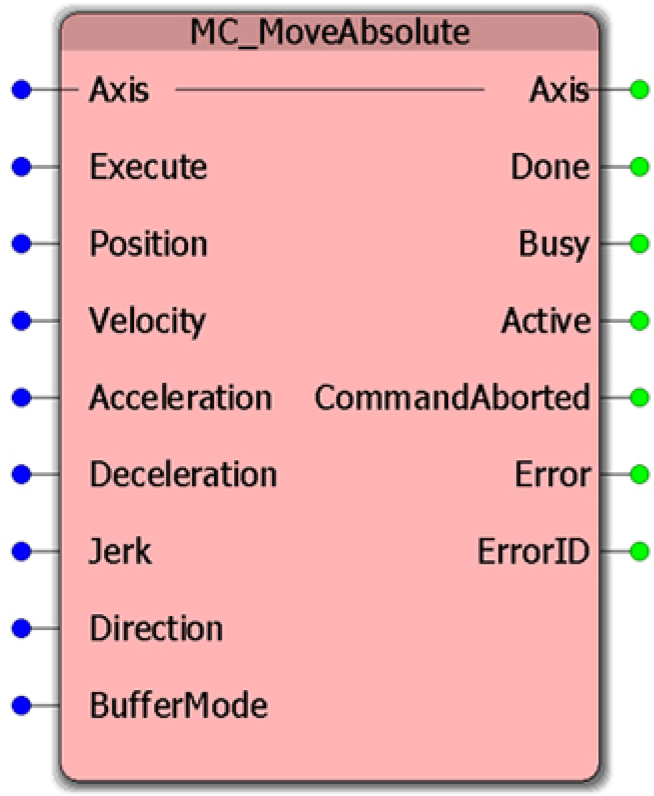
Example

In the example shown below, an MC_Stop function block is used to bring the slave axis to a standstill after the gear relation is broken using MC_GearOut





MC_MoveAbsolute



This Function Block commands a controlled motion to the specified absolute position.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE

B	Position	LREAL	A positive or negative value within the coordinate system in user units.	LREAL#0.0
E	Velocity	LREAL	Absolute value of the velocity in user units/second	LREAL#0.0
E	Acceleration	LREAL	Value of the acceleration in user units/second^2 (acceleration is applicable with same sign of torque and velocity)	LREAL#0.0
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
E	Direction	MC_Direction	Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction	MC_Direction#Positive_Direction
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	

E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

• The absolute position, as with all other inputs, can be updated while in motion by retriggering the Execute input.

• This action completes with velocity zero if no further blocks are pending.

• Regarding the use of the 'Direction' input:

  • If there is only one mathematical solution to reach the commanded position (like in linear systems), the value of the input Direction is ignored.

  • For rotary axis - valid absolute position values are in the range of the machine cycle. It is possible to specify a relative move of more than one machine cycle using MC_MoveRelative. When motion is complete, the position will be reported as somewhere between 0 and machine cycle.

  • The Enum type MC_Direction#Shortest_Way will cause motion through the shortest route. The controller will decide based on the current position when the function block is executed.

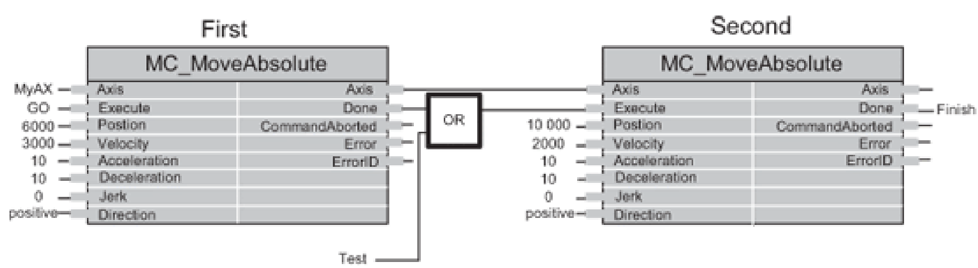
• For further information about the Done output, Profile Complete, and Motion Complete, see the Determining when motion is complete section.

Error Description

ErrorID	Meaning
0	No Error
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4658	Velocity parameter is less than or equal to zero.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
4641	Buffer mode does not correspond to a valid enumeration

	value.
4642	Direction does not correspond to a valid enumeration value.
4667	Jerk is less than or equal to zero
4378	The function block is not applicable for the external axis specified
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4381	Motion aborted due to axis alarm.
4893	The specified external axis may not be used. A physical axis is required
57617	Instance object is NULL.
57620	The structure size does not match.

Example



Timing Diagram

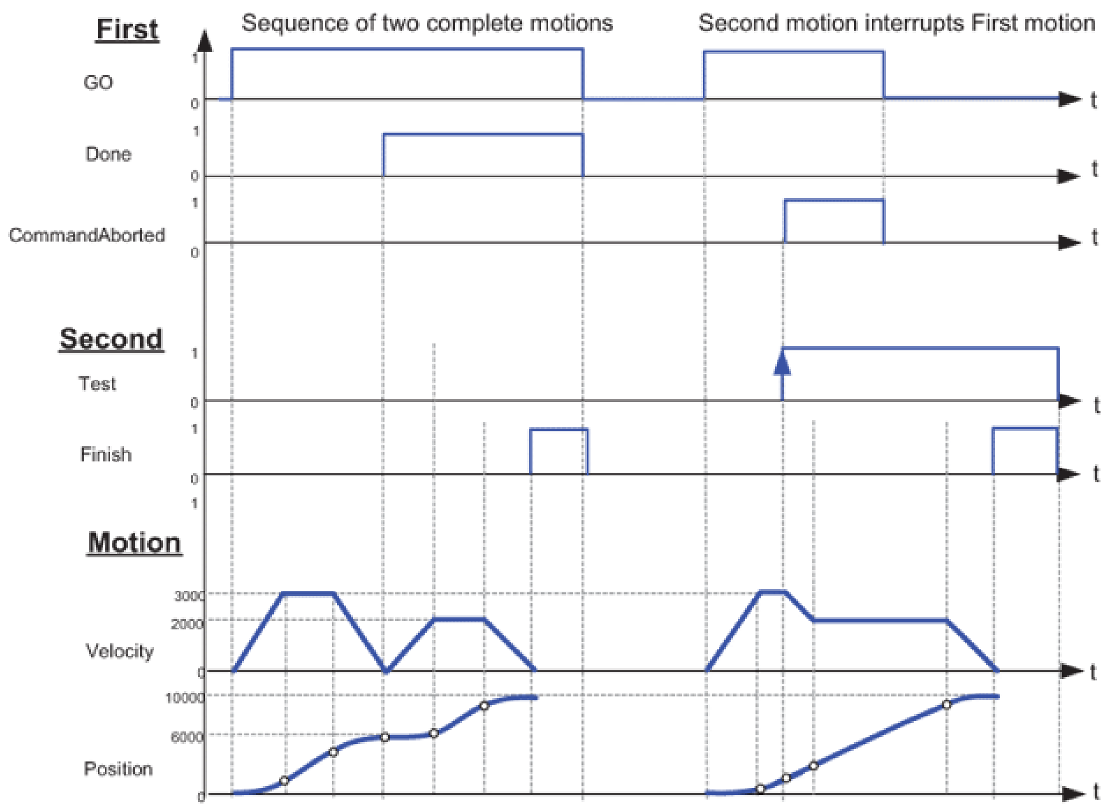
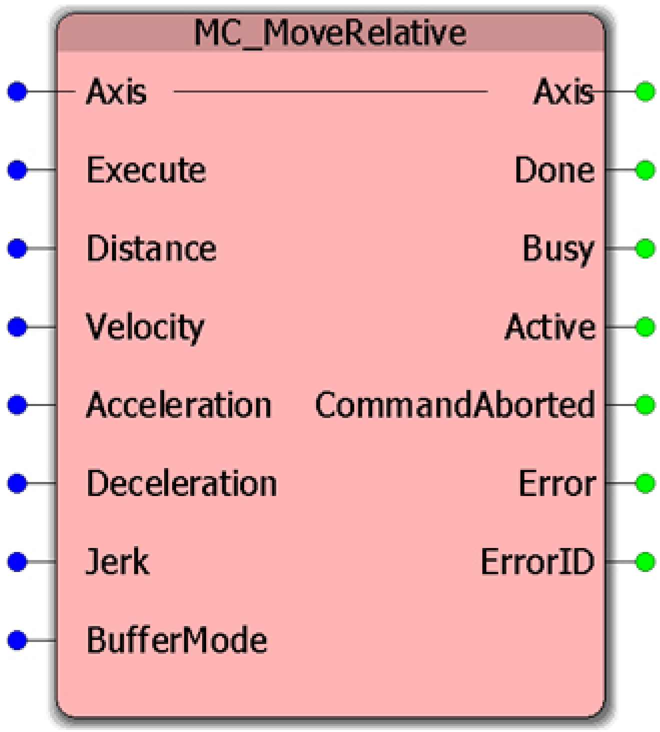


Figure 19: Timing diagram for MC_MoveAbsolute

Note to figure: the examples are based on two instances of the Function Block: instance "First" and "Second".

MC_MoveRelative



This Function Block commands a controlled motion of the specified distance relative to the commanded position at the time of the execution.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	Distance	LREAL	Incremental distance (in user units)	LREAL#0.0

E	Velocity	LREAL	Absolute value of the velocity in user units/second	LREAL#0.0
E	Acceleration	LREAL	Value of the acceleration in user units/second^2 (acceleration is applicable with same sign of torque and velocity)	LREAL#0.0
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• This action completes with zero velocity if no further function blocks are pending.

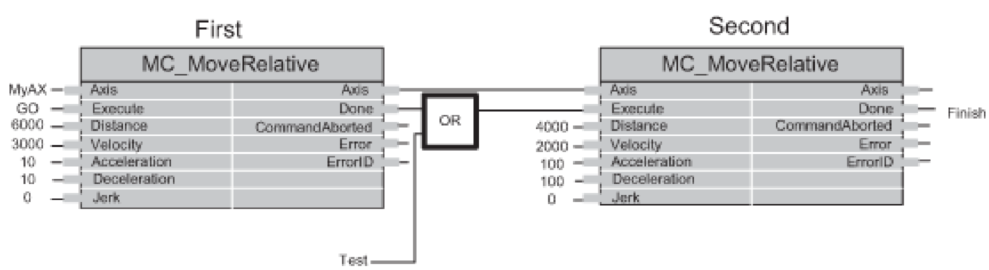
• For further information about the Done output, Profile Complete, and Motion Complete, see the Determining when motion is complete section.

Error Description

ErrorID	Meaning
0	No Error
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis

	number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4641	Buffer mode does not correspond to a valid enumeration value.
4642	Direction does not correspond to a valid enumeration value.
4658	Velocity parameter is less than or equal to zero.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
[4667]()	Jerk is less than or equal to zero
[4893]()	The specified external axis may not be used. A physical axis is required
[57620]()	The structure size does not match.

Example



Timing Diagram

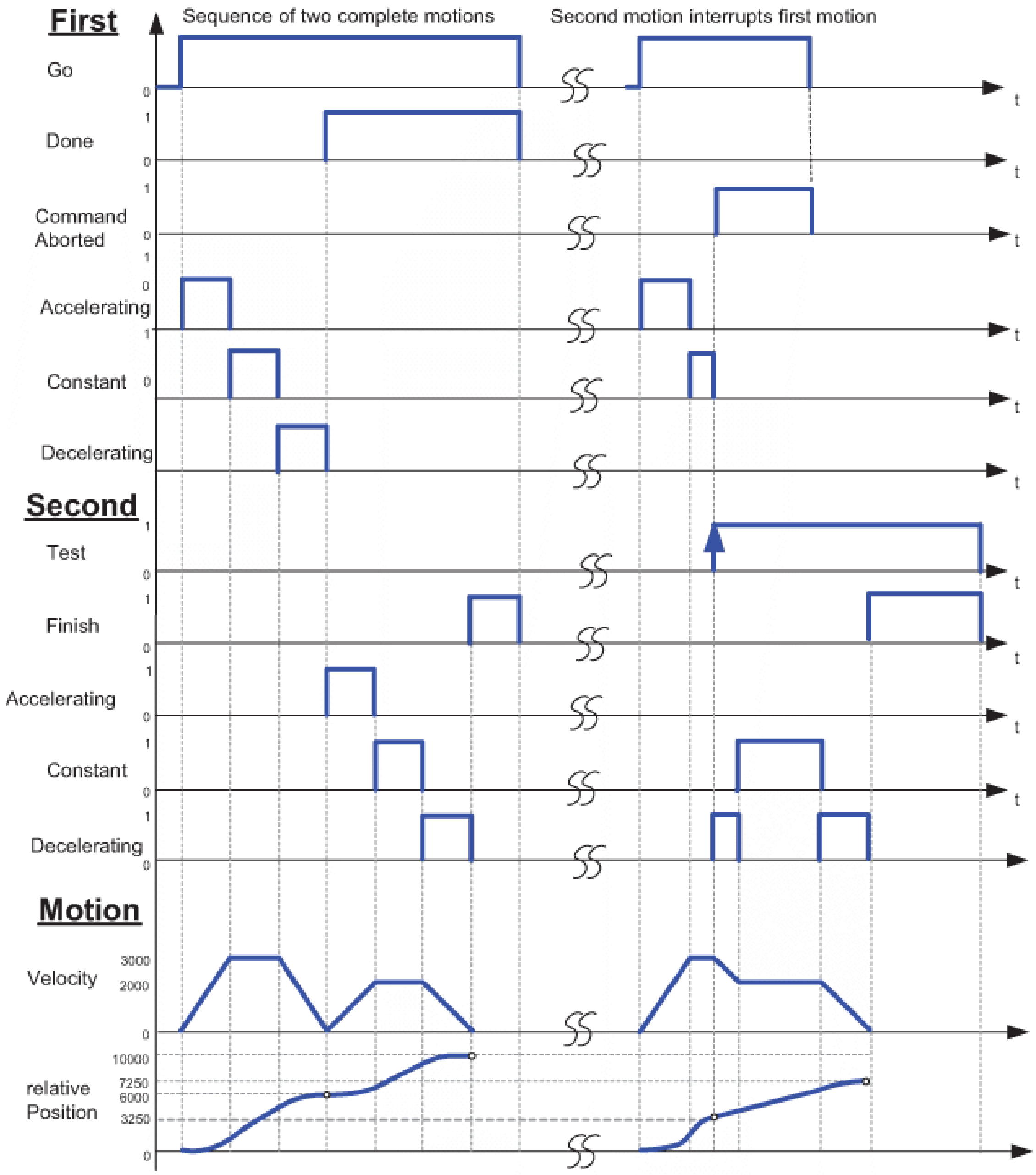
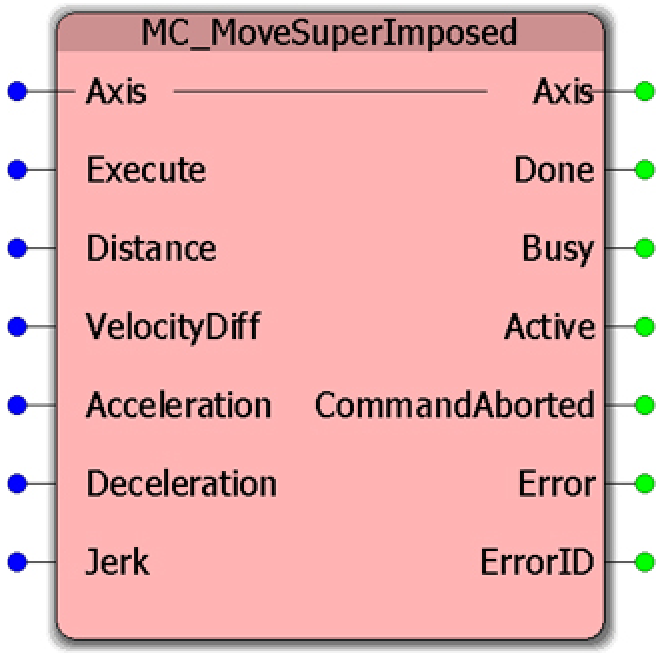


Figure 20: Timing diagram for MC_MoveRelative

MC_MoveSuperImposed



This Function Block commands a controlled motion of the specified relative distance additional to an existing motion. The existing Motion is not interrupted, but is superimposed by the additional motion.

Parameters

Parameter		Data type	Description
VAR_IN_OUT			
B	Axis	[AXIS_REF]()	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).

VAR_INPUT				Default
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	Distance	LREAL	Incremental distance that is to be superimposed (in user units)	LREAL#0.0
E	VelocityDiff	LREAL	Value of the maximum velocity difference to the ongoing motion (not necessarily reached)	LREAL#0.0
E	Acceleration	LREAL	Value of the acceleration in user units/second^2 (acceleration is	LREAL#0.0

			applicable with same sign of torque and velocity)	
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	

E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

- When MC_MoveSuperImposed is active, any other command in aborting mode except MC_MoveSuperImposed will abort both motion commands: both the MC_MoveSuperImposed and the underlying motion command. In any other Buffer mode, the underlying motion command is not aborted.

• If MC_MoveSuperImposed is active and another MC_MoveSuperImposed is commanded, only the on-going MC_MoveSuperImposed command is aborted, and replaced by the new MC_MoveSuperImposed, the underlying motion command continues.

• In the 'StandStill' motion state, MC_MoveSuperimposed acts like MC_MoveRelative.

• The values of Acceleration, Deceleration, and Jerk are additional values to the on-going motion, and not absolute ones. With this, the underlying FB always finishes its job in the same period of time regardless of whether a MC_MoveSuperimposed FB takes place concurrently.

• When used while gearing, MC_MoveSuperimposed acts on the slave axis, while MC_Phasing acts on the master side, as seen from the slave.

• The output “Active” has a different behavior as in buffered FBs.

Related Function Blocks

MC_GearInPos: Slave Axis position adjustment made to achieve position synchronization over a defined region of travel for both master and slave.

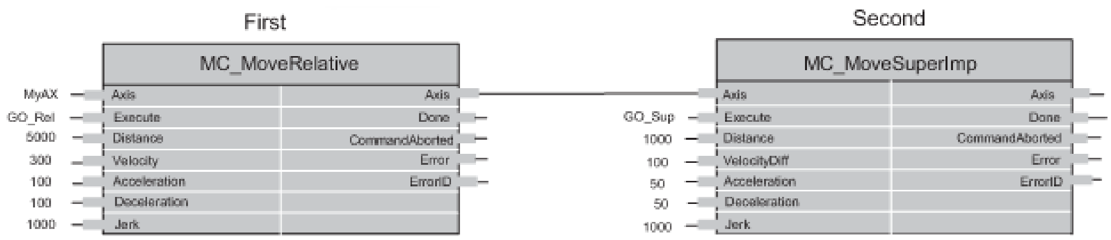
Y_CamShift: Dynamically modifies the master - slave relationship by adding a perceived offset to the master position, effectively causing the slave to advance or retard from the originally specified synchronization data in the cam data table.

Error description

ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4658	Velocity parameter is less than or equal to zero.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
4667	Jerk is less than or equal to zero
4378	The function block is not applicable for the external axis specified
4893	The specified external axis may not be used. A physical axis is

	required
57619	The structure pointer check sum is invalid.
57620	The structure size does not match.

Example 1



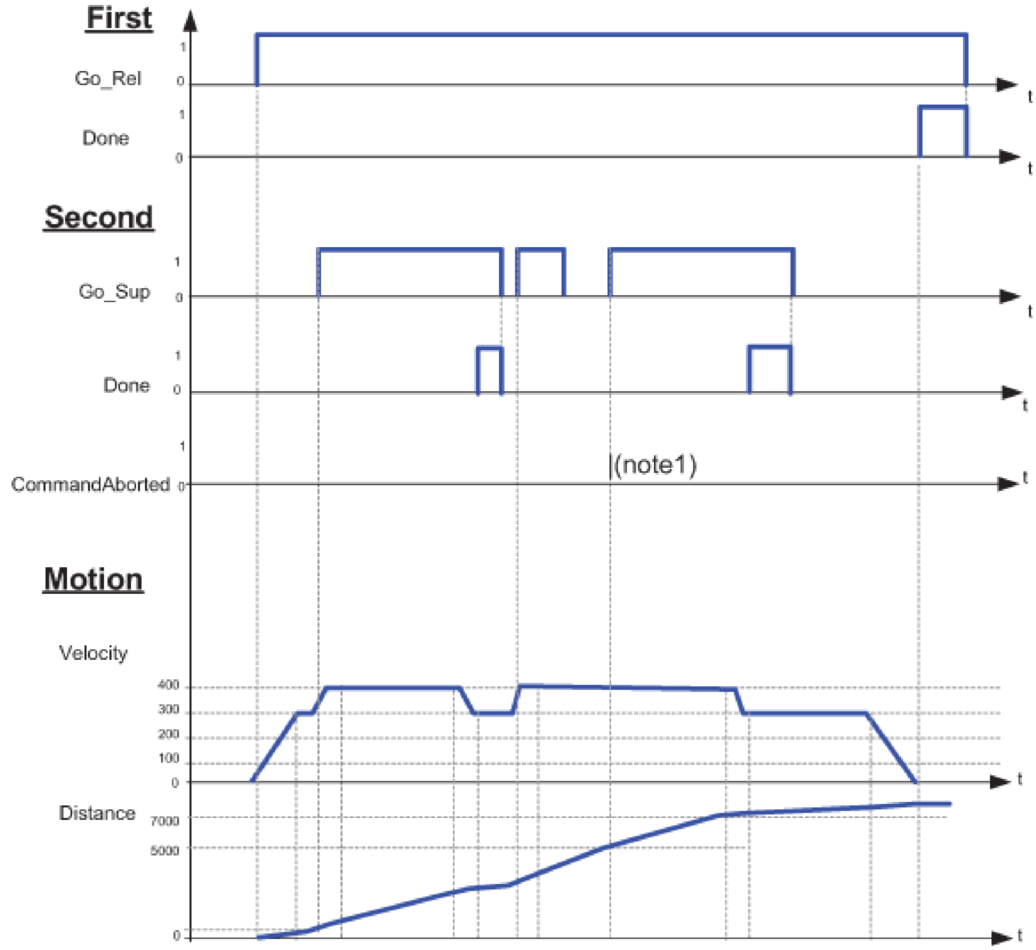
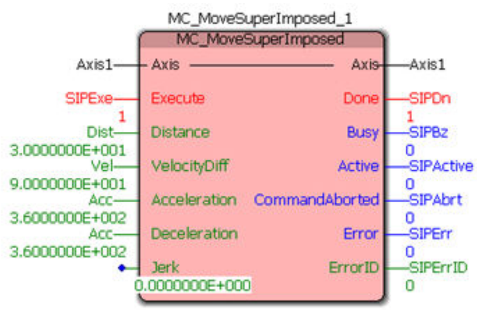


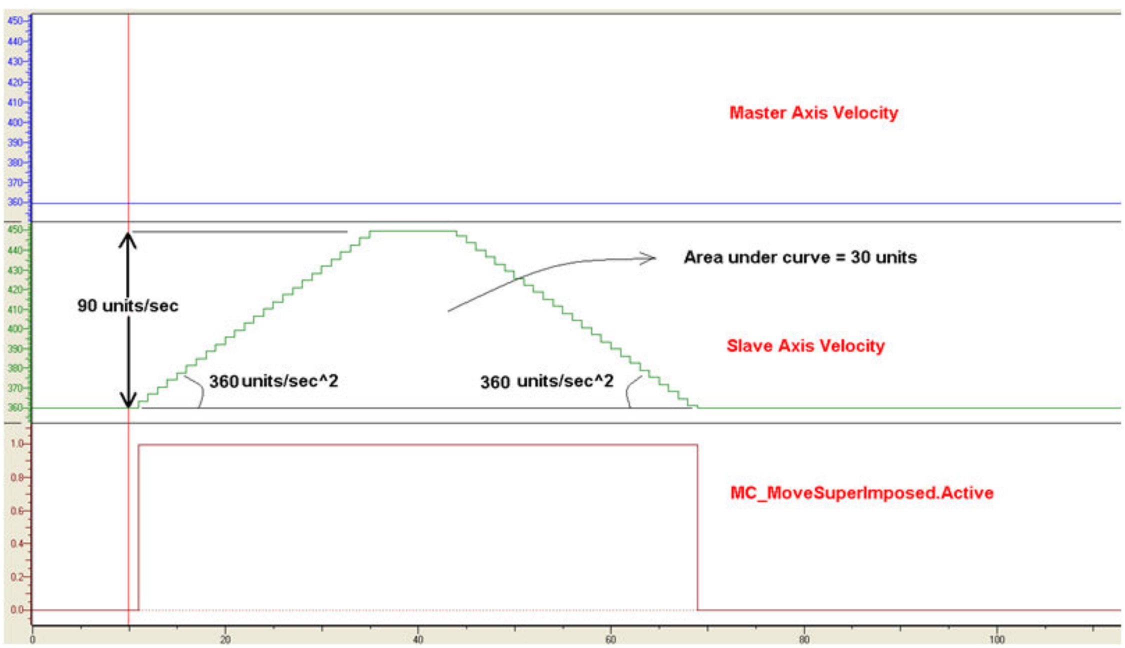
Figure 22: Timing diagram for MC_MoveSuperimposed

Note 1: the CommandAborted is not visible here, because the new command works on the same instance (see general rules 2.3.1)
Note 2: the end position is between 7000 and 8000, depending on the timing of the aborting of the second command set for the MC_MoveSuperimposed

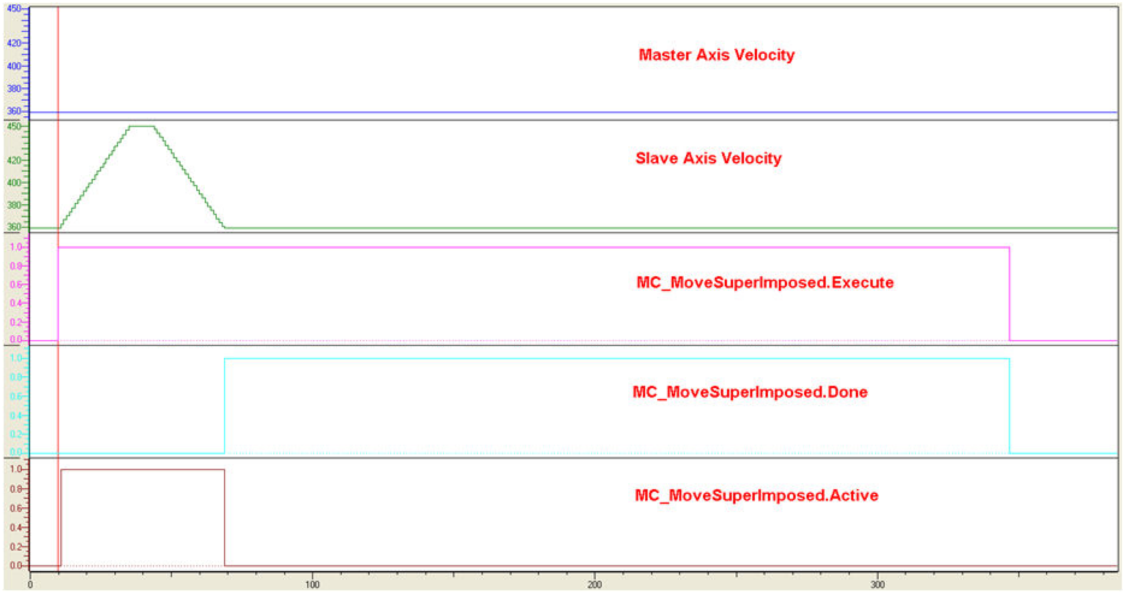
Example 2

Consider two axes, a master and a slave that need to be electronically geared to each other at a certain ratio. If a minor adjustment needs to be made on the slave axis to bump the axis ahead or retard it by a certain distance, MC_SuperImposed can be used to make that adjustment. The example shown here is that of an adjustment of 30 degrees being made to the slave axis which was geared to a master axis running at 360 degree/sec velocity with a 1:1 ratio.





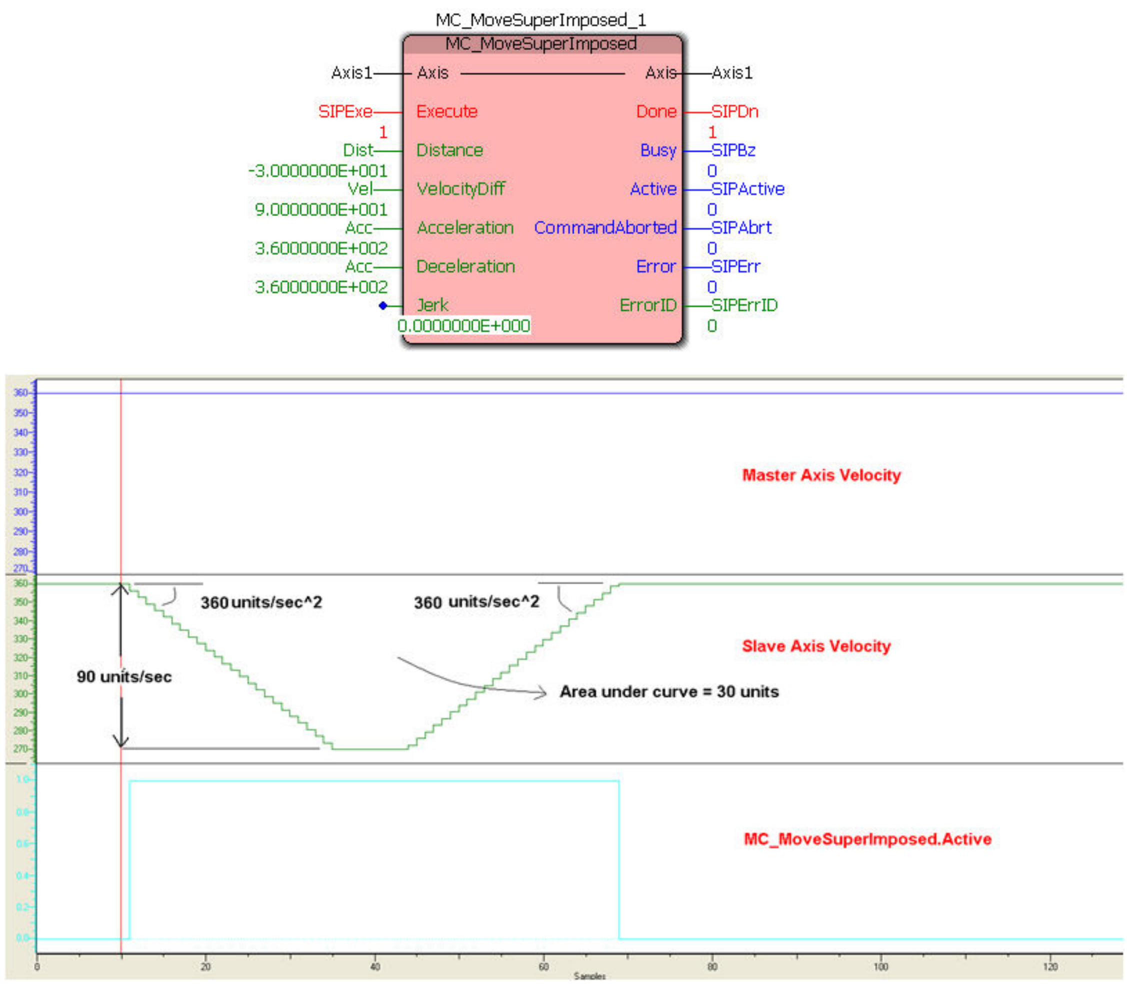
30 degree adjustment made while motion was continuing.



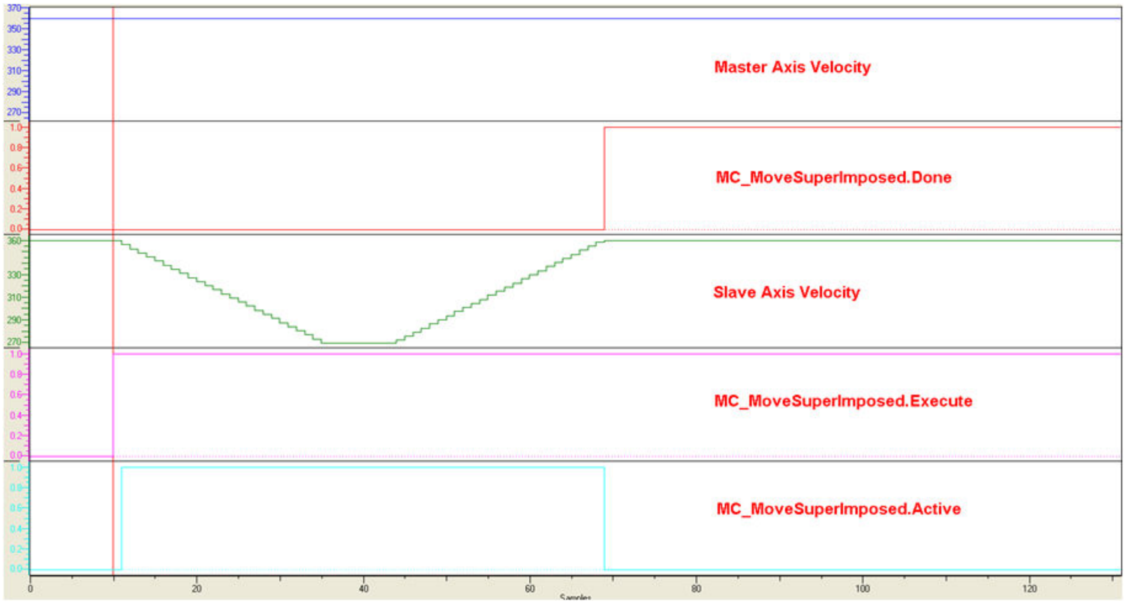
Timing diagram for MC_MoveSuperImposed.

Example 3

The example shown here is that of an adjustment of -30 degrees being made to the slave axis which was geared to a master axis running at 360 degree/sec velocity with a 1:1 ratio.

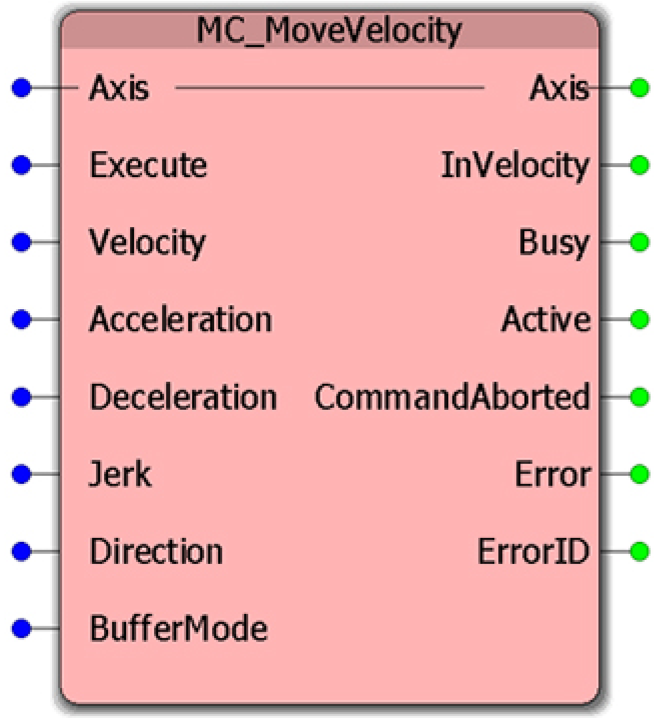


-30 degree adjustment made while motion was continuing.



Timing diagram for MC_MoveSuperImposed.

MC_MoveVelocity



This Function Block commands a never ending controlled motion at the specified velocity.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
E	Velocity	LREAL	Absolute value of the velocity in	LREAL#0.0

			user units/second	
E	Acceleration	LREAL	Value of the acceleration in user units/second^2 (acceleration is applicable with same sign of torque and velocity)	LREAL#0.0
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
E	Direction	MC_Direction	Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction	MC_Direction#Positive_Direction
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	InVelocity	BOOL	Set high upon successful completion of the function. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	

B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

• To stop motion, use MC_Stop

• The output 'InVelocity' will be reset when the block is aborted by another block or at the falling edge of 'Execute'.

• In combination with MC_MoveSuperimposed, the output 'InVelocity' stays TRUE once the velocity setpoint of the axis has reached the commanded velocity.

Related Function Blocks

Jog : Make the axis jog at different speeds without having to re-fire the execute input bit. Available in the PLCopen Toolbox library

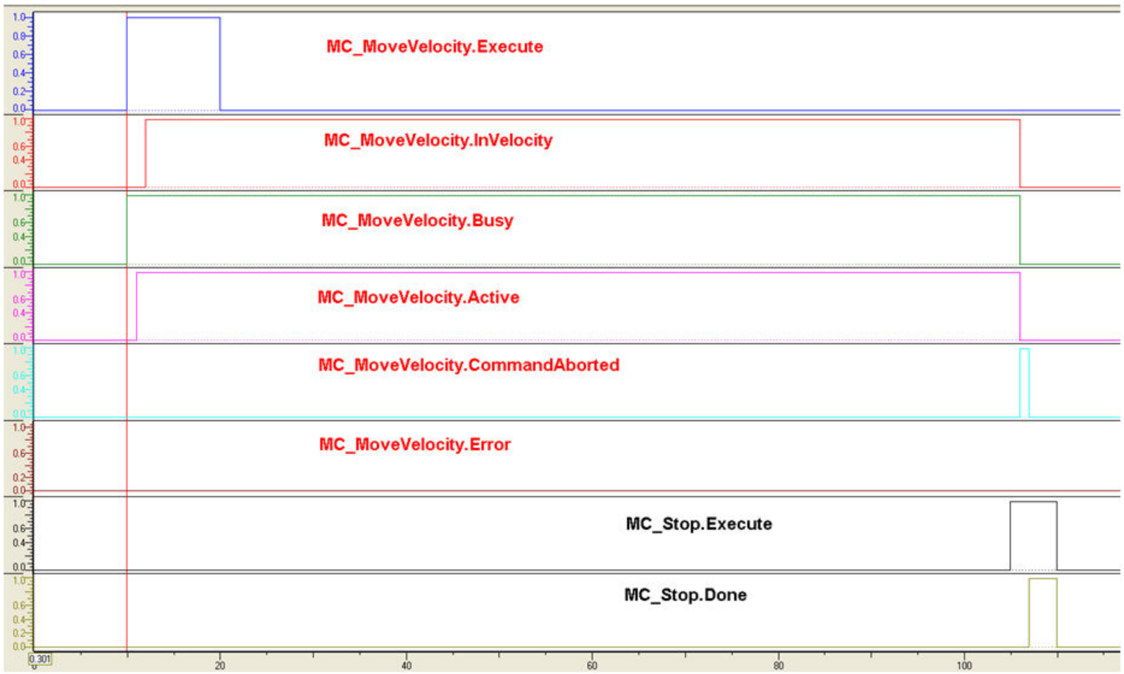
MC_Stop: Stops an axis in motion by bringing the axis to a standstill state.

Error Description

ErrorID	Meaning
0	No Error
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause

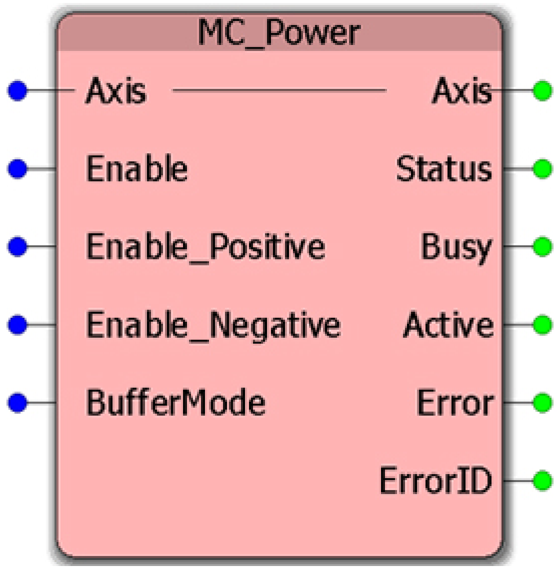
	motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4641	Buffer mode does not correspond to a valid enumeration value.
4642	Direction does not correspond to a valid enumeration value.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
4665	Velocity parameter is negative.
4667	Jerk is less than or equal to zero
57620	The structure size does not match.

Example



Timing Diagram for MC_MoveVelocity

MC_Power



This Function Block enables or disables the axis.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Axis	[AXIS_REF]()	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
E	Enable_Positive	BOOL	Permits motion in a positive direction. An error is generated if positive motion is commanded when this input is FALSE. - Not Supported	FALSE
E	Enable_Negative	BOOL	Permits motion in a negative direction. An error is generated if	FALSE

			negative motion is commanded when this input is FALSE. - Not Supported	
E	BufferMode	MC_BufferMode	Not supported. The behavior is as if MC_BufferMode#Aborting is set.	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Status	BOOL	Actual state of the axis, TRUE=Enabled, FALSE=Disabled.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- If the MC_Power FB is called with the 'Enable' true while being in 'Disabled', this either leads to 'Standstill' motion state if there is no error in the axis, or to ErrorStop if an Error exists.

- MC_Power is different than other block using the Enable/Valid method. Since MC_Power actively enables or disables the servo continuously, the Active output is always TRUE, and the Error and ErrorID outputs are always output. Do not include more than one MC_Power function block per axis in the IEC application program.

- When MC_Power is called with 'Enable' false, the axis goes to 'Disabled' motion state from every state including 'ErrorStop'.

- If the controller detects that the command position deviates significantly from the feedback position, the controller will post an alarm causing motion to stop. If while this alarm is active, the drive is power cycled, the controller will not re-enable the drive.

• The Active output will be TRUE if the servo axis is under control of MC_Power. If the Enable input bit is FALSE, the Active output indicates that the MC_Power is controlling the disabled state of the axis. If the Enable input bit is TRUE and the Status and Active bits are TRUE, MC_Power is controlling the enabled state of the axis.

• An example of recommended servo enable procedure is shown in the MPiec Programming Best Practices Guideline. (section 7.5)

Related Function Blocks

MC_ReadAxisError: Reports axis specific warnings or alarms.

MC_Reset: Clears axis specific alarms.

PLCopen Toolbox - AxisControl: Combines MC_Power, MC_ReadAxisError, and MC_Reset to Enable/disable an axis along with providing axis warnings and alarms as outputs.

Y_ReadAlarm: Reports general controller alarms NOT related to a specific axis.

Y_ClearAlarms: Clears general controller alarms NOT related to a specific axis.

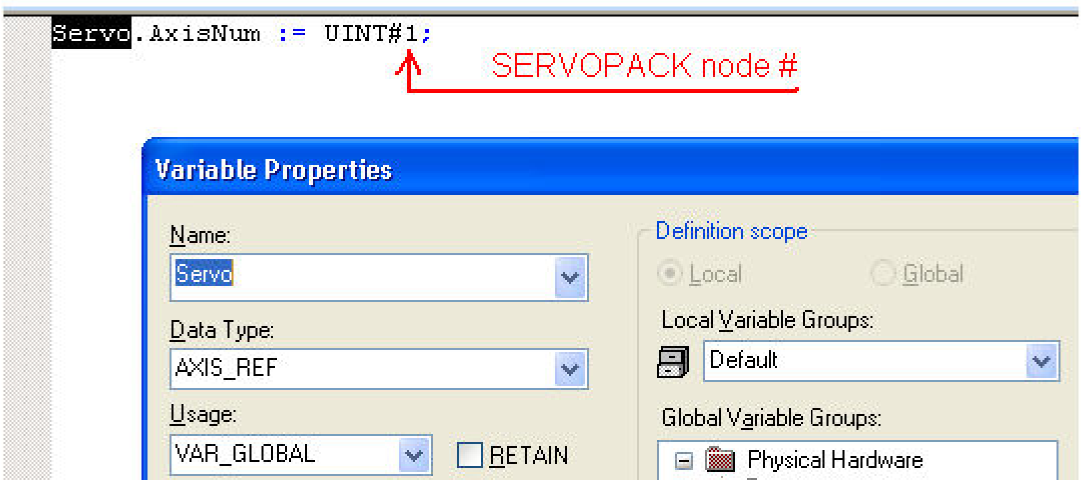
Error Description

ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4371	The servo drive failed to enable or disable. Check the amplifier wiring for L1 / L2 / L3
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4399	The L1 / L2 / L3 power inputs on the drive may not be supplied with power, possibly due to an E-Stop condition.
4400	The safety input (HBB on the CN8 connector) is preventing the drive from enabling.
4414	MECHATROLINK Communications to the drive was disrupted. Execute MC_Reset to restore the connection.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant

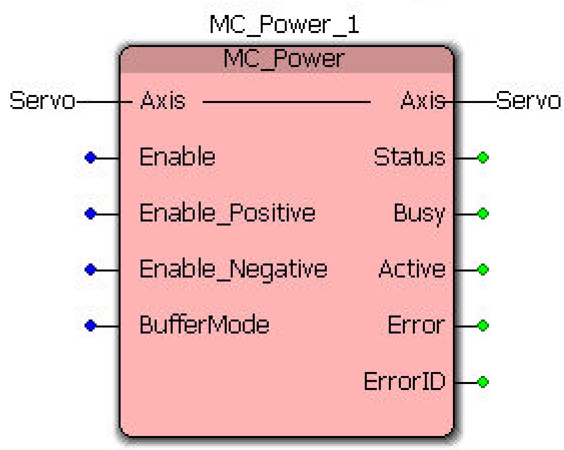
	POUs.
4641	Buffer mode does not correspond to a valid enumeration value.
4893	The specified external axis may not be used. A physical axis is required
4894	The specified virtual axis may not be used with this function block.
57617	Instance object is NULL.
57620	The structure size does not match.
61713	An internal motion kernel command failed. This error could be caused by inserting multiple MC_Power function blocks in the program for the same axis. Only one MC_Power function per axis is required. (Do not include more than one.)

Example

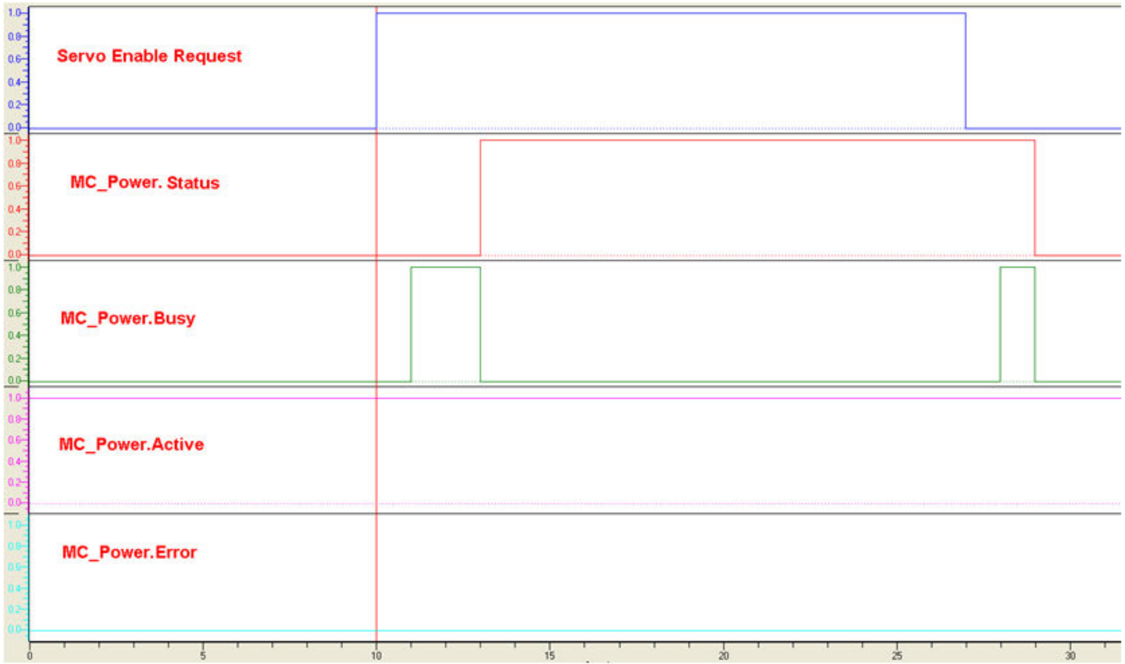
INITIALIZATION ROUTINE



INSTANTIATION of MC_Power

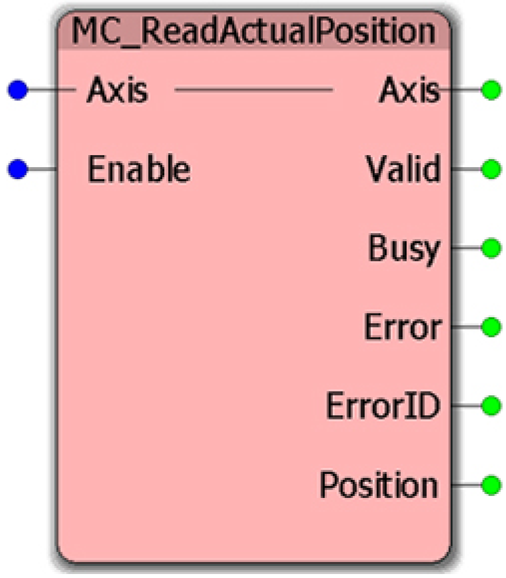


Initializing an axis and using MC_Power to enable the axis



Timing diagram for MC_Power

MC_ReadActualPosition



This Function Block returns the actual position of the axis in user units as configured in the Hardware Configuration.

Parameters

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when	

			'Execute' or 'Enable' goes low.
B	Position	LREAL	A positive or negative value within the coordinate system in user units.

Notes

The actual position is also available using MC_ReadParameter 1000. For additional parameters related to the axis position, check the Axis Parameter List.

Related Function Blocks

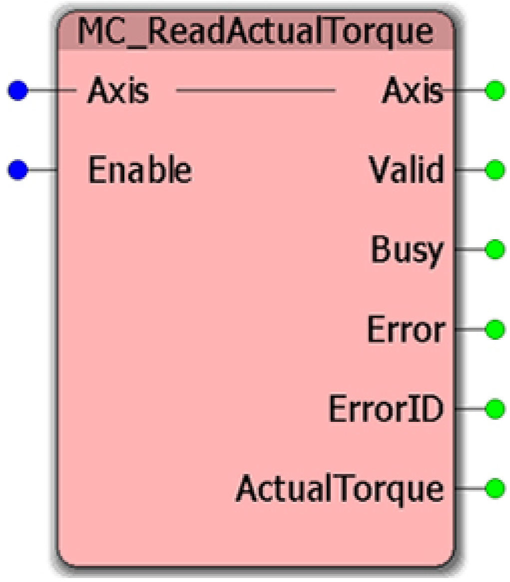
MC_ReadParameter: Returns the value of an axis-specific controller side parameter. Controller parameter 1005 gives the value of the cyclic position for rotary axes. Controller parameter 1006 gives the non cyclic position for rotary axes.

ReadAxisParameters: Reads axis parameters including feedback position and stores it under an AxisParameterStruct type variable. Available in PLCopenToolbox.

Error Description

ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
57620	The structure size does not match.

MC_ReadActualTorque



This Function Block returns the value of the actual torque or force in percentage of rated torque. Note that peak torque will vary from 200 to over 300% based on the specific amplifier model.

Parameters

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	

E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.
B	ActualTorque	LREAL	The value of the actual torque or force in percentage of rated torque.

Notes

The output ActualTorque is a signed value. The actual torque is also available using MC_ReadParameter 1004.

Related Function Blocks

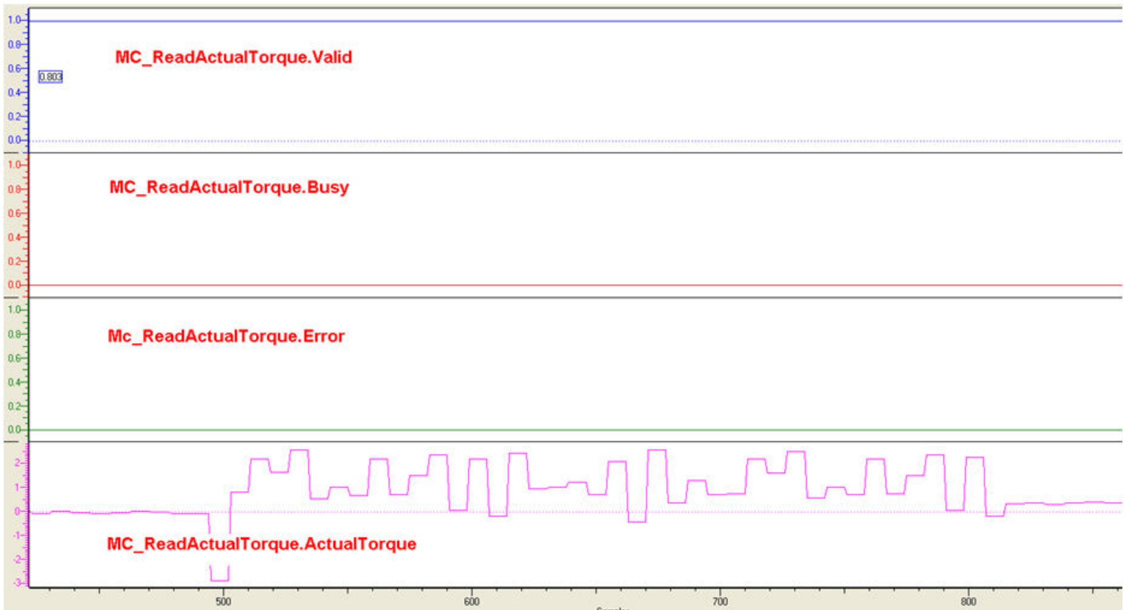
ReadAxisParameters: Reads axis parameters including actual torque or force (exerted by the axis) and stores it under an AxisParameterStruct type variable. Available in PLCopenToolbox.

MC_ReadParameter: returns the value of an axis-specific controller side parameter. Controller parameter 1004 gives the actual torque exerted by the axis.

Error Description

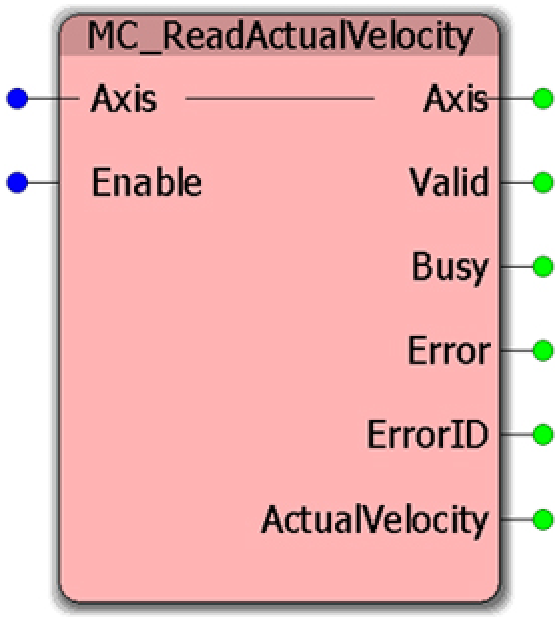
ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
57620	The structure size does not match.

Example



Timing Diagram for MC_ReadActualTorque for a slow move of an axis

MC_ReadActualVelocity



This Function Block returns the value of the actual velocity of the axis in user units / sec as configured in the Hardware Configuration.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	[AXIS_REF]()	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the	

			function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.
B	ActualVelocity	LREAL	The value of the actual velocity

Notes

The output 'ActualVelocity' is a signed value. The actual velocity is also available using MC_ReadParameter 1001. For additional parameters related to the axis velocity, check the Axis Parameter List.

Related Function Blocks

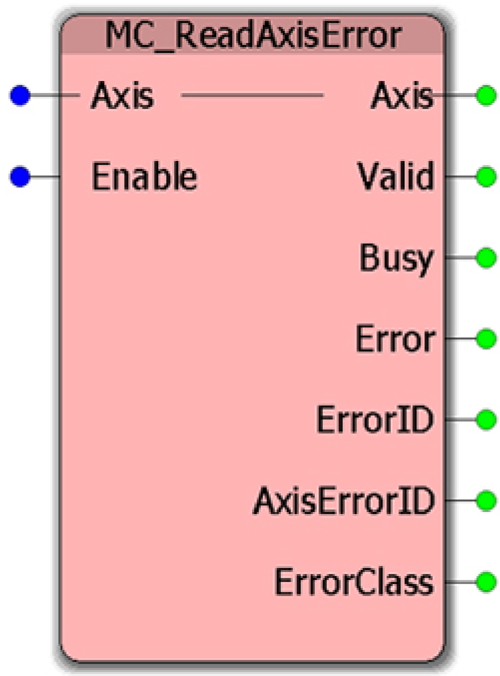
ReadAxisParameters: Reads axis parameters including feedback velocity and stores it under an AxisParameterStruct type variable. Available in PLCopenToolbox.

MC_ReadParameter: returns the value of an axis-specific controller side parameter. Controller parameter 1001 gives the feedback velocity of the axis.

Error Description

ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
57620	The structure size does not match.

MC_ReadAxisError



This Function Block reports axis alarms and warnings not related to the PLCopen Function Blocks. The ErrorClass output designates the source of the alarm or warning. The AxisErrorID output contains the error code.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset	

			if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
B	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.
B	AxisErrorID	UINT	The value of the axis error. Errors are generated by either the drive or the controller, based on the value of ErrorClass.
E	ErrorClass	UINT	See the Notes section below for a detailed description.

Notes

If ErrorClass has a value of 16#3302, 16#3303, 16#4302, or 16#4403, then the source of the problem is the amplifier. Sigma alarms are documented in the Sigma Series user manuals. Please refer to the following manuals for details regarding servo amplifier errors to look up the alarm code shown at AxisErrorID output:

- Sigma II with NS115: SIEPC71080001, see section 9.3
- Sigma III: YEA-SIA-S800-11, see section 10.1.4
- Sigma-5 with rotary motor: SIEPS8000046, see Section 9.1
- Sigma-5 with linear motor: SIEPS8000048, see Section 8.1.1

If ErrorClass is some value other than 16#3302, 16#3303, 16#4302, or 16#4403, the source of the problem is on the controller side. Refer to the Controller Alarm ID List.

Related Function Blocks

MC_Power: Enables/Disables an axis.

MC_Reset: Clears axis specific alarms.

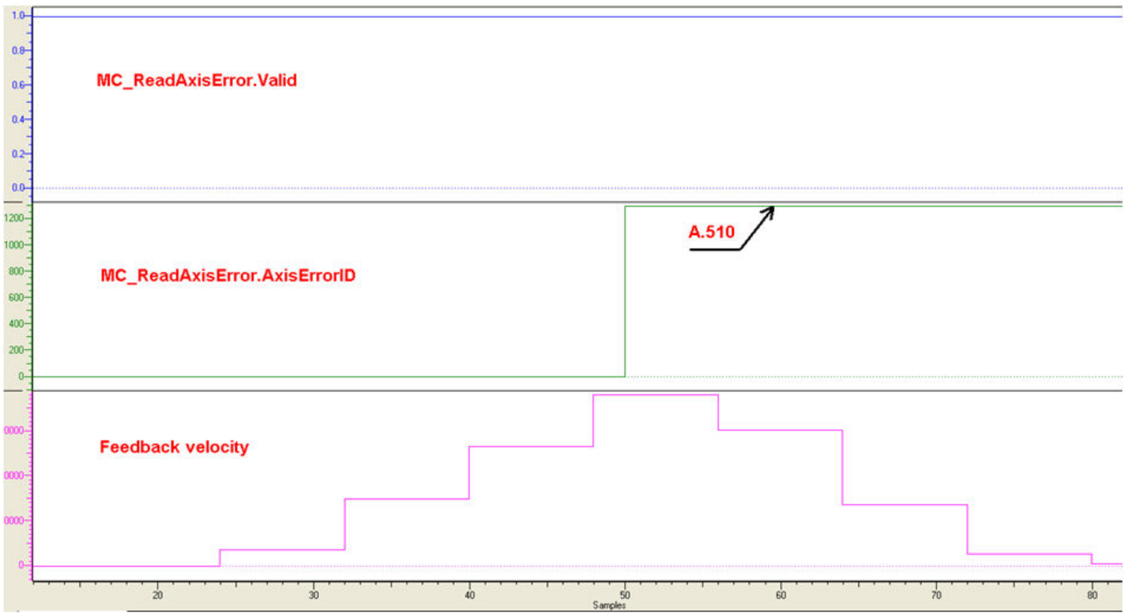
Y_ReadAlarm: Reports controller alarms.

PLCopen Toolbox - AxisControl: Combines MC_Power, MC_ReadAxisError, MC_Reset, and Y_ReadAlarm to Enable/disable an axis along with providing axis warnings and alarms as outputs.

Error Description

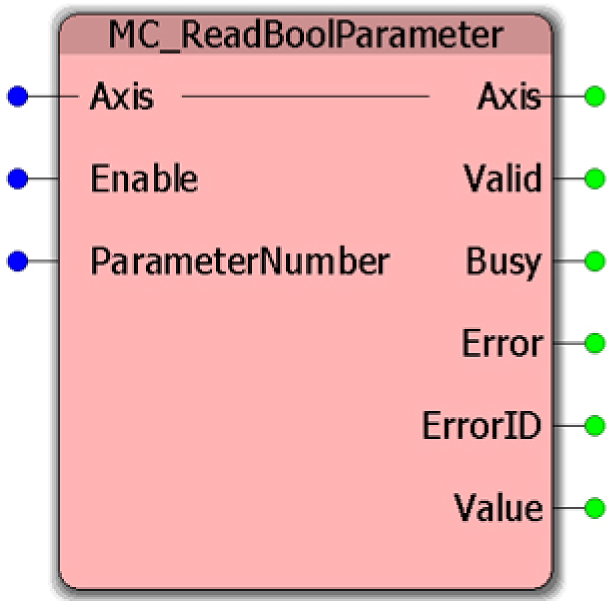
ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
57620	The structure size does not match.

Example



Timing diagram for an overspeed condition reflected in MC_ReadAxisError

MC_ReadBoolParameter



This Function Block reads the value of an axis specific boolean parameter and is for controller-side parameters only. For amplifier side parameters, refer to Y_ReadDriveParameter.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
B	ParameterNumber	UINT	Number of the Parameter in the controller	UINT#0
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input,	

			and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.
B	Value	BOOL	The drive parameter value

Notes

Refer to parameters with BOOL Data Type in the Axis Parameter List.

Related Function Blocks

MC_ReadParameter: returns the value of an axis-specific controller side parameter.

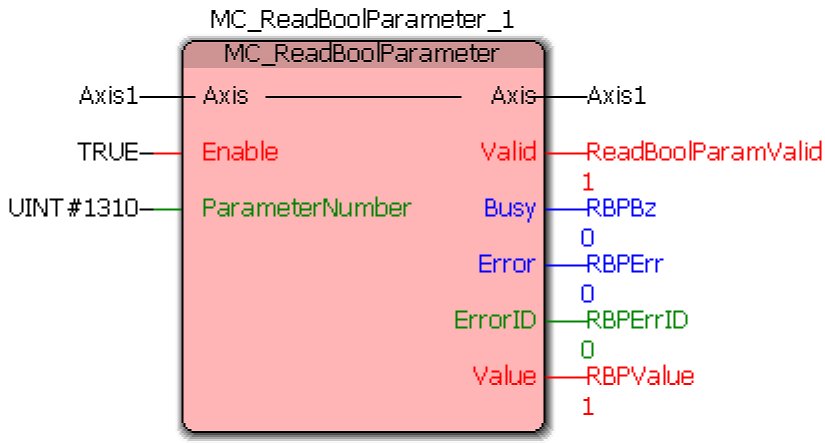
MC_WriteBoolParameter: writes the value of an axis specific boolean parameter and is for controller-side parameters only.

Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4403	The High Speed Output functionality is only available on external encoders.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis

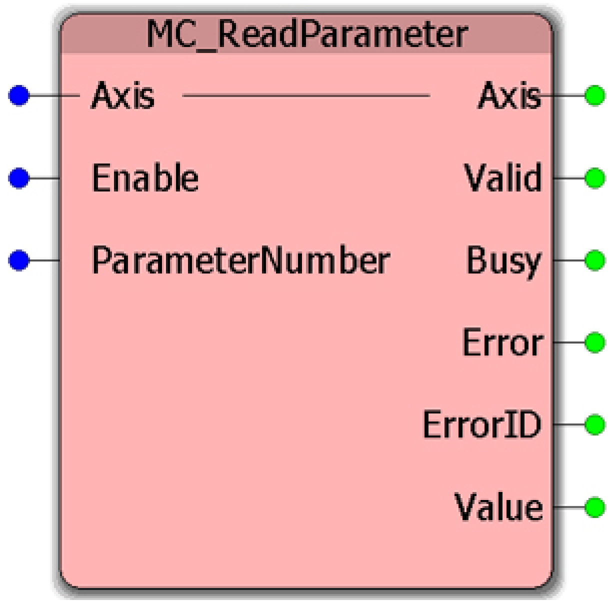
57620	The structure size does not match.

Example



Instance of MC_ReadBoolParameter reading boolean controller parameter 1310. In this case controller parameter 1310 is enabled (RBPValue = 1).

MC_ReadParameter



This Function Block returns the value of an axis-specific controller side parameter. For amplifier side parameters, refer to Y_ReadDriveParameter.

For a complete list of controller side parameters, refer to parameters with LREAL Data Type in the Axis Parameter List.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
B	ParameterNumber	UINT	Controller parameter number. Refer to parameters with LREAL Data Type in the Axis Parameter List.	UINT#0

VAR_OUTPUT			
B	Valid	BOOL	Indicates that the outputs of the function are valid.
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.
B	Value	LREAL	The drive parameter value

Notes

Refer to parameters with LREAL Data Type in the Axis Parameter List.

Related Function Blocks

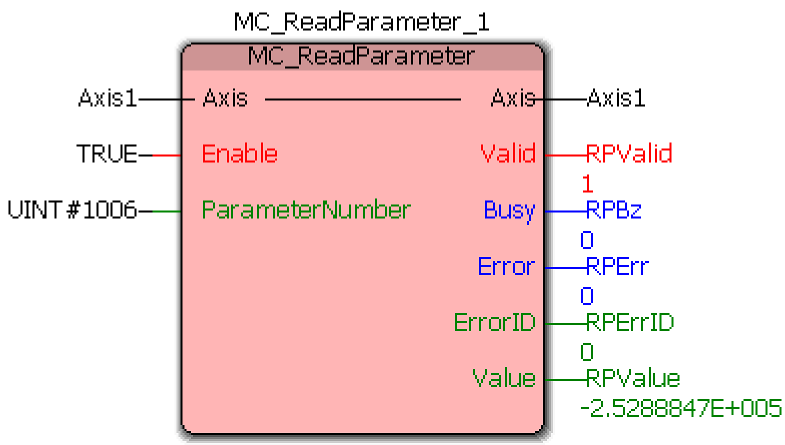
MC_WriteParameter: writes the value of an axis-specific parameter and is for controller side parameters only.

Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4402	The scan compensation delay parameter 1305 is only valid for external encoders.
4403	The High Speed Output functionality is only available on external encoders.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.

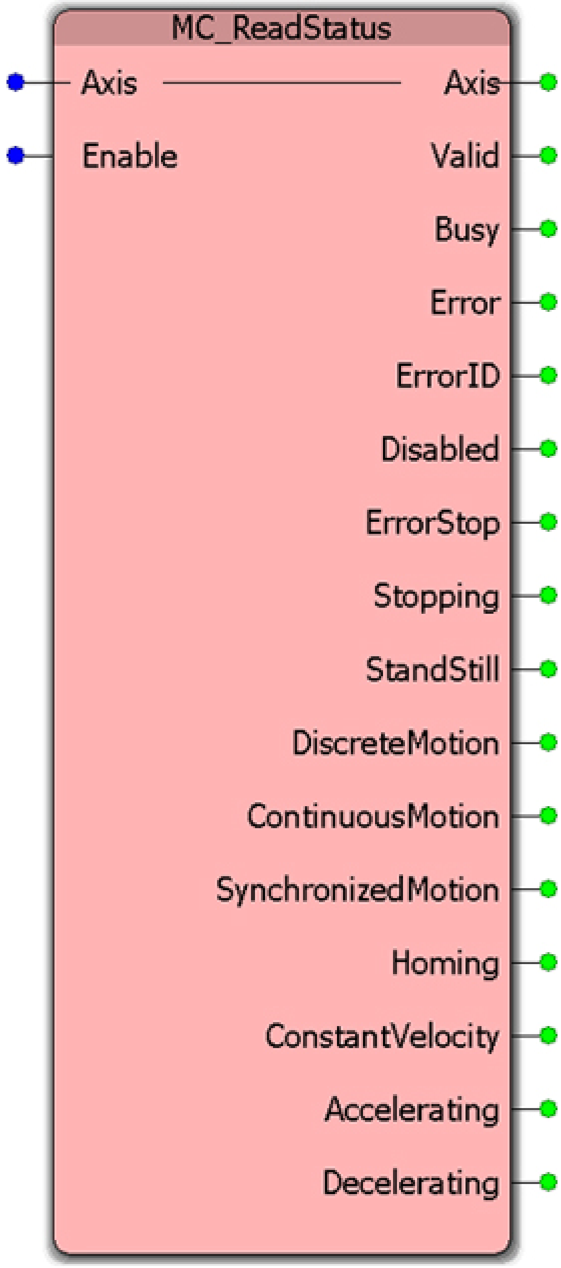
4648	The parameter number does not exist for the specified axis
4676	The time value must be within 0 to 10 MECHATROLINK cycles.
57617	Instance object is NULL.
57620	The structure size does not match.

Example



Instance of MC_ReadParameter reading non cyclic actual position of an axis

MC_ReadStatus



This Function Block returns in detail the status of the axis with respect to motion currently in progress. The status reflects the Motion State Diagram and other motion related attributes.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
B	ErrorStop	BOOL	See the state diagram	
B	Disabled	BOOL	See the state diagram	
B	Stopping	BOOL	See the state diagram	
B	StandStill	BOOL	See the state diagram	
B	DiscreteMotion	BOOL	See the state diagram	
B	ContinuousMotion	BOOL	See the state diagram	
E	SynchronizedMotion	BOOL	See the state diagram	
E	Homing	BOOL	See the state diagram	
E	ConstantVelocity	BOOL	Motor moves with constant velocity	
E	Accelerating	BOOL	Increasing energy of the motor	
E	Decelerating	BOOL	Decreasing energy of the motor	

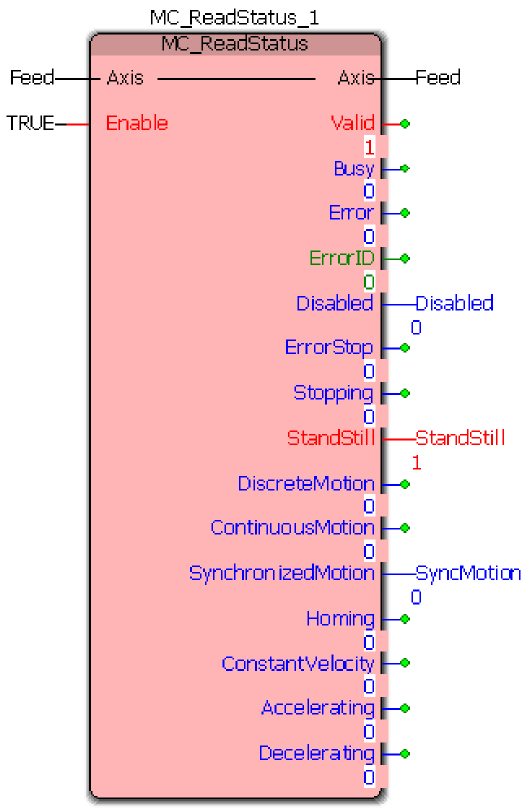
Notes

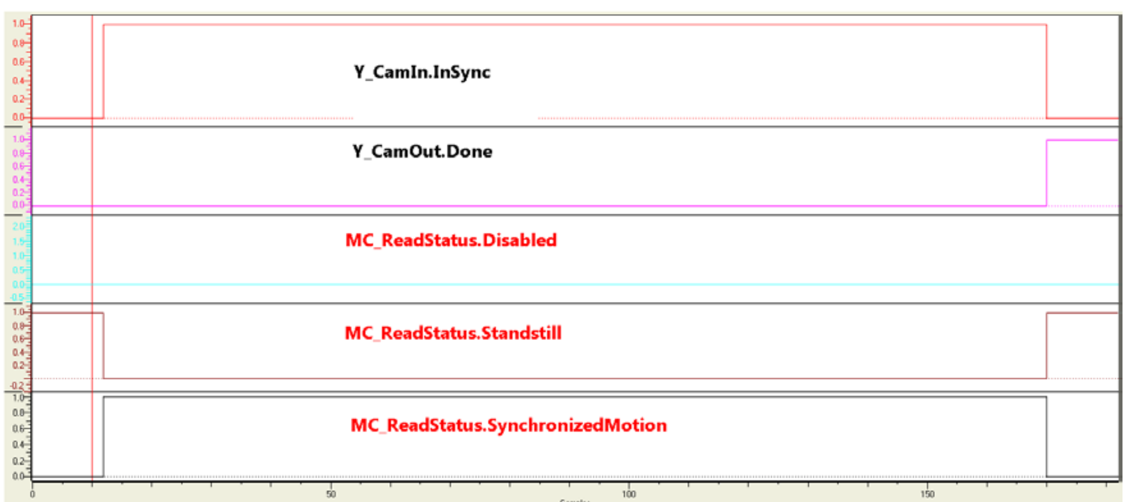
None

Error description

ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4378	The function block is not applicable for the external axis specified
57620	The structure size does not match.

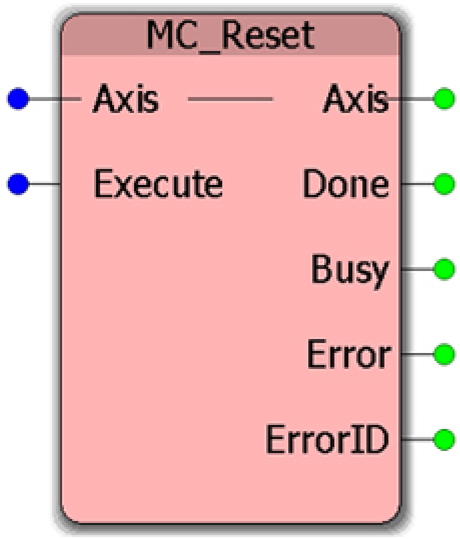
Example





Timing diagram showing the outputs of MC_ReadStatus before, during and after a camming operation

MC_Reset



This function block will reset axis related alarms on either the controller or amplifier side.

Parameters

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Axis	[AXIS_REF]()	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	

E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
B	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

Referring to the Motion State Diagram, this function block makes the transition from the ErrorStop to StandStill state by resetting axis-related errors.

Axis specific alarms are those whose ErrorClass starts with 16#3302, 16#3303, 16#4302, or 16#4403. Use MC_ReadAxisError to determine whether the alarm is axis specific, or a controller alarm in general.

Some Sigma alarms cannot be cleared without power cycle. MC_Reset does not verify that alarms are cleared before setting the Done output. It returns Done when the attempt to clear is complete. Use MC_ReadAxisError to check if the axis still has an alarm/error after MC_Reset is done. There may be more than one alarm active for the axis.

For those alarms or situations when a servopack power must be cycled, the Y_ResetMechatrolink function block will cause a soft restart of the amplifier, eliminating the need to physically cycle power.

Please refer to the following manuals for details regarding servo amplifier errors:

- Sigma II with NS115: SIEPC71080001, see section 9.3

- Sigma III: YEA-SIA-S800-11, see section 10.1.4
- Sigma-5 with rotary motor: SIEPS8000043, see Section 6.1
- Sigma-5 with linear motor: SIEPS8000044, see Section 6.1

Related Function Blocks

MC_Power: Enables/Disables an axis.

MC_ReadAxisError: Reports axis warnings or alarms.

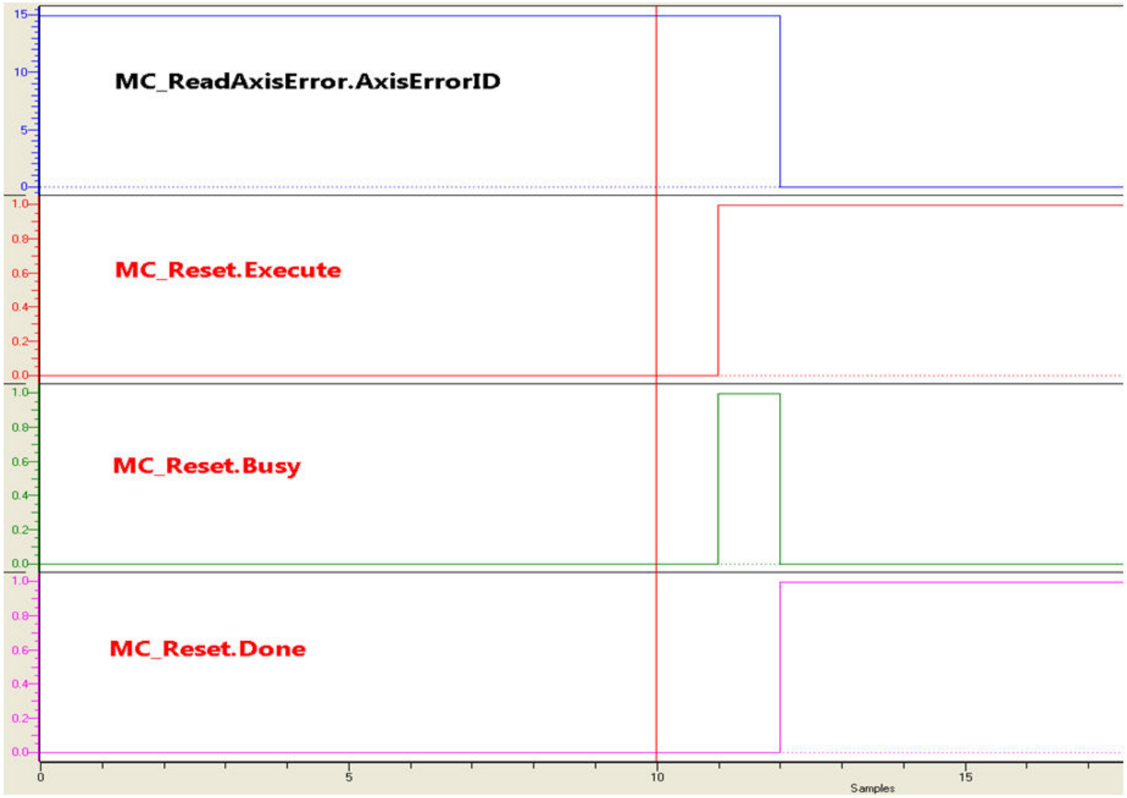
Y_ReadAlarm: Reports controller alarms.

PLCopen Toolbox - AxisControl: Combines MC_Power, MC_ReadAxisError, MC_Reset, and Y_ReadAlarm to Enable/disable an axis along with providing axis warnings and alarms as outputs.

Error description

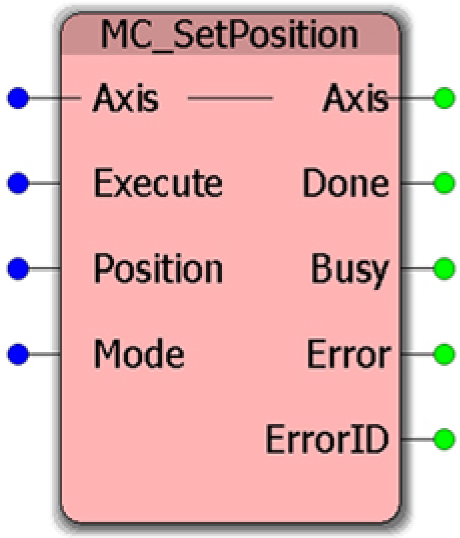
ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
45332	Sending clear alarms command to servo drive failed.
57620	The structure size does not match.

Example



Timing diagram for MC_Reset

MC_SetPosition



This Function Block shifts the coordinate system of an axis by changing both the commanded position as well as the actual position of an axis with the same value without any movement caused.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	Position	LREAL	A positive or negative value within the coordinate system in user units.	LREAL#0.0
E	Mode	BOOL	RELATIVE =True, ABSOLUTE = False (Default). This refers to the coordinate	FALSE

			system, not the motor type.	
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• If Mode=FALSE, the position input value is treated as an ABSOLUTE position, and the axis position is defined as such. If Mode=TRUE, then the value of the Position input is added to the current commanded position, the sum of which becomes the new commanded position without any motion occurring.

• If the specified axis has a Sigma Series absolute encoder, the absolute encoder offset is calculated and stored in the controller's battery-backed RAM when this block is executed. The next time the system is powered up, the absolute encoder position is read, then automatically adjusted to reflect the desired position for the machine. It is not necessary to physically move the motor to a zero point and reset the absolute encoder.

• If Y_ResetAbsoluteEncoder was executed, a power cycle is required before MC_SetPosition can effectively store the absolute encoder offset. Y_ResetMechatrolink can be used to soft power cycle the network and servopacks.

• A slave axis will jump when changing the master's position. Add program logic to avoid this situation.It is recommended not to use

MC_SetPosition while the axis is in motion. Make sure the axis is disabled or in standstill state before executing MC_SetPosition.

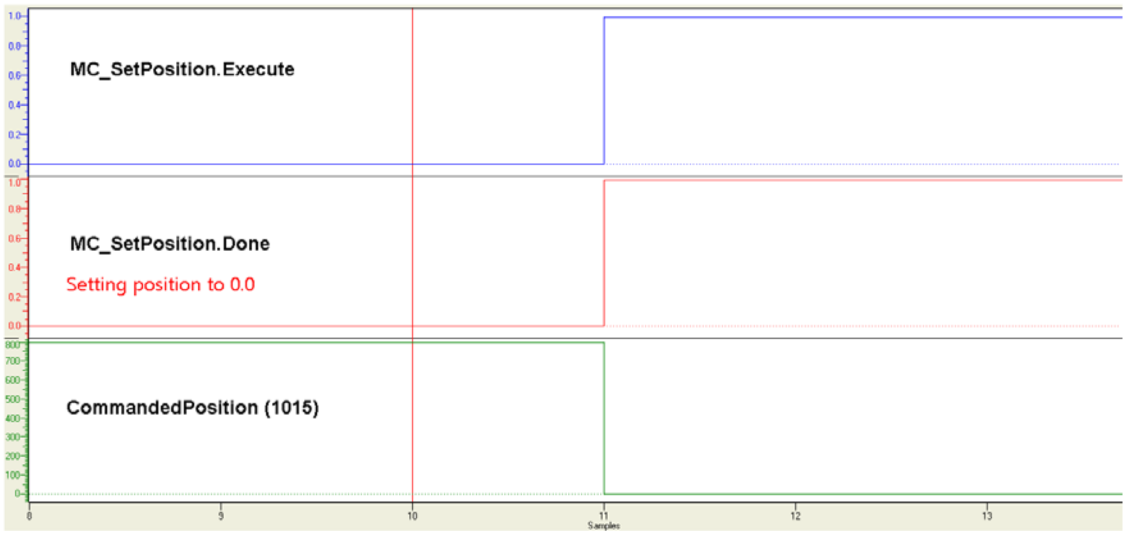
• An error will be generated if executed on an external encoder axis that has the High Speed Output (Coincidence output function) enabled.

Error Description

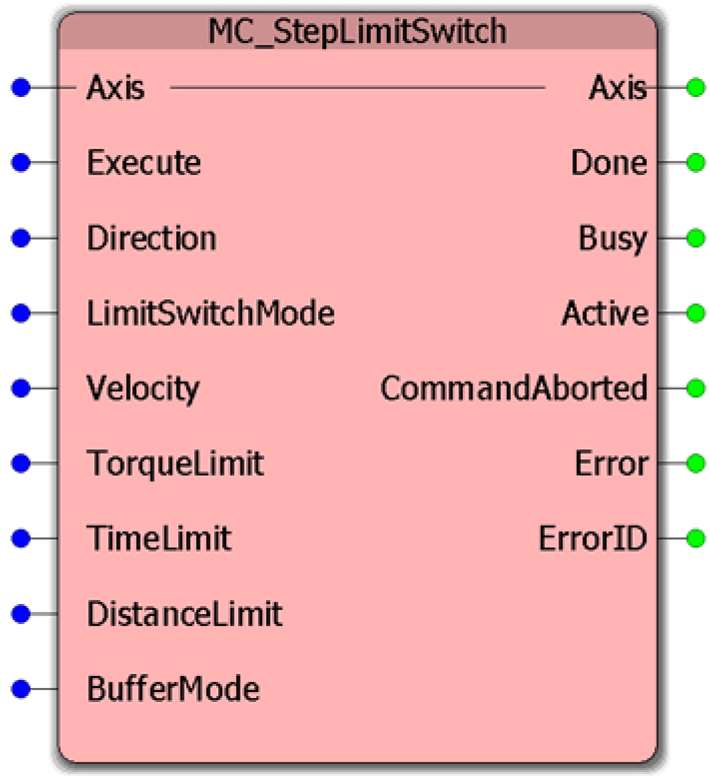
ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4380	MC_SetPosition can not be executed while the axis is moving.
4382	When the axis is in rotary mode, and the MC_SetPosition tries to set a position that is equal to or greater than the MachineCycle, this error is generated, and the position is not set.
4390	Position cannot be defined while the axis is the cam master of other axes.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4646	Mode does not correspond to a valid enumeration value.
57620	The structure size does not match.

Example

Timing diagram when MC_SetPosition sets position to zero.



MC_StepLimitSwitch



This function Block performs a homing function by searching for a limit switch. Logic is built-in to account for cases in which the limit is already hit. See the example below.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	[AXIS_REF]()	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-	FALSE

			trigger the execute input.	
E	Direction	MC_Direction	Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction	MC_Direction#Positive_Direction
E	LimitSwitchMode	MC_SwitchMode	Sensor condition to finalize MC_StepLimitSwitch: Only MC_SwitchMode#EdgeOn is supported. MC_SwitchMode#On = When sensor is ON MC_SwitchMode#Off = When sensor is OFF MC_SwitchMode#EdgeOn = When Off to On transition in sensor MC_SwitchMode#EdgeOff = When On to Off transition in sensor	MC_SwitchMode#EdgeOn
E	Velocity	LREAL	Absolute value of the velocity in user units/second	LREAL#0.0
E	TorqueLimit	LREAL	Maximum torque or force [in % of rated torque].The amplifier's Torque Limits (Pn402 and Pn403) will override the torque limit set by this value if they are lower than the TorqueLimit setting.	100% of Rated Torque
E	TimeLimit	LREAL	Time limit for homing to complete (in seconds).	LREAL#0.0 (no time limit)
E	DistanceLimit	LREAL	Maximum distance the axis can travel in search of home sensor.	LREAL#0.0
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext	MC_BufferMode#Aborting

			MC_BufferMode#BlendingHigh	
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- This function block acts in conjunction with the Sigma Servo's P-OT and N-OT functions. Refer to the servo amplifier parameters Pn50A and Pn50B for P-OT and N-OT configurations.

- For proper operation, it is recommended to set Pn001.1 to 1 to maintain the servo lock state. The function block cannot complete successfully if the servo is disabled during the process.

- TimeLimit, DistanceLimit, and TorqueLimit have a default value of 0.0, which means an infinite limit.

- It is recommended to disable controller velocity feedforward component (1310) for smooth servo motion while homing using MC_StepLimitSwitch.

  - Sigma II with NS115: SIEPC71080001, see Appendix B.

  - Sigma III: YEA-SIA-S800-11, see section 7.5.2

  - Sigma-5 with rotary motor: SIEPS80000046, see Section 10.1

- Sigma-5 with linear motor: SIEPS8000048, see Section 9.1

• Pn50A and Pn50B can be set in the MotionWorks IEC Configuration.

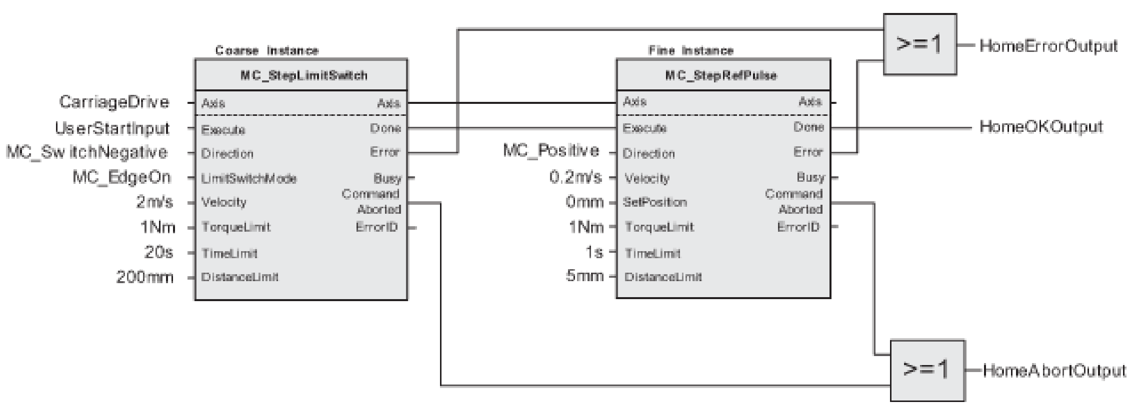
• For further information about the Done output, Profile Complete, and Motion Complete, see the Determining when motion is complete section.

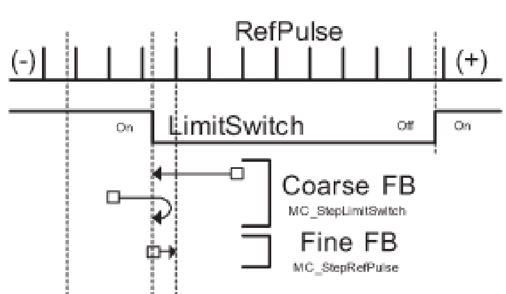
Error Description

ErrorID	Meaning
0	No error.
1	Time limit exceeded.
2	Distance limit exceeded.
3	Torque limit exceeded.
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4379	A homing sequence is already in progress.
4381	Motion aborted due to axis alarm.
4383	Axis must be commanded at standstill when homing is attempted..
4391	The function block can not be used with a virtual axis.
4625	Axis ID does not correspond to an axis configured on the

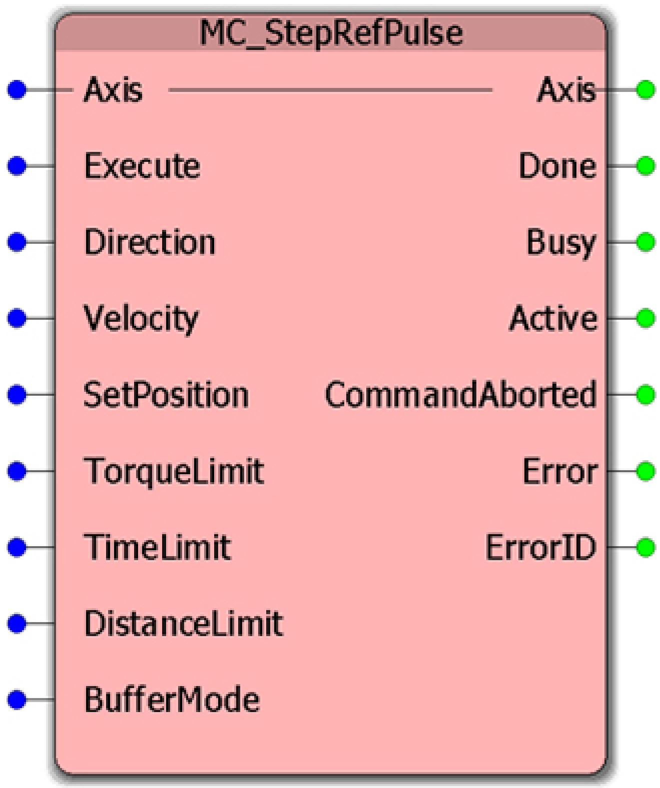
	system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4642	Direction does not correspond to a valid enumeration value.
4646	Mode does not correspond to a valid enumeration value.
4658	Velocity parameter is less than or equal to zero.
4897	The drive's model number or type does not match the parameter file.
57620	The structure size does not match.

Example





MC_StepRefPulse



This function Block performs homing by searching for Zero pulse (also called Marker or reference pulse) in the encoder on all Sigma Series rotary servos.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and	FALSE

			re-trigger the execute input.	
E	Direction	[MC_Direction]()	Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction	MC_Direction#Positive_Direction
E	Velocity	LREAL	Absolute value of the velocity in user units/second	LREAL#0.0
E	SetPosition	LREAL	Value of the absolute position [u] to be set when homing is done. The reference	LREAL#0.0
E	TorqueLimit	LREAL	Maximum torque or force [in % of rated torque].The amplifier's Torque Limits (Pn402 and Pn403) will override the torque limit set by this value if they are lower than the TorqueLimit setting.	LREAL#0.0
E	TimeLimit	LREAL	Time limit for homing to complete (in seconds).	LREAL#0.0
E	DistanceLimit	LREAL	Maximum distance the axis can travel in search of home sensor.	LREAL#0.0
E	BufferMode	[MC_BufferMode]()	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block	

			takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

- It is recommended to use velocity equivalent to 60 RPM or less to find the C channel.
- For further information about the Done output, Profile Complete, and Motion Complete, see the Determining when motion is complete section.

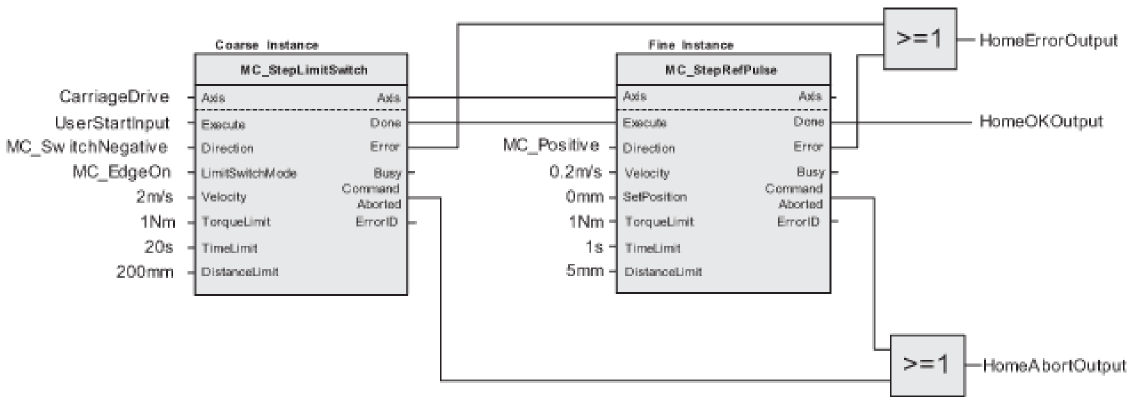
Error Description

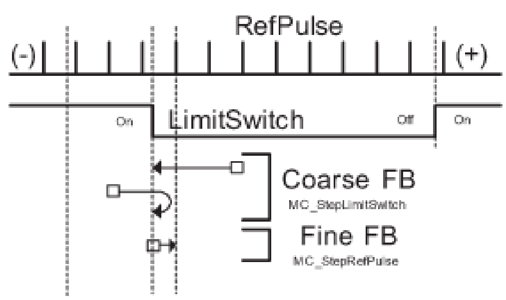
ErrorID	Meaning
0	No error
1	Time limit exceeded
2	Distance limit exceeded
3	Torque limit exceeded
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.

YASKAWA

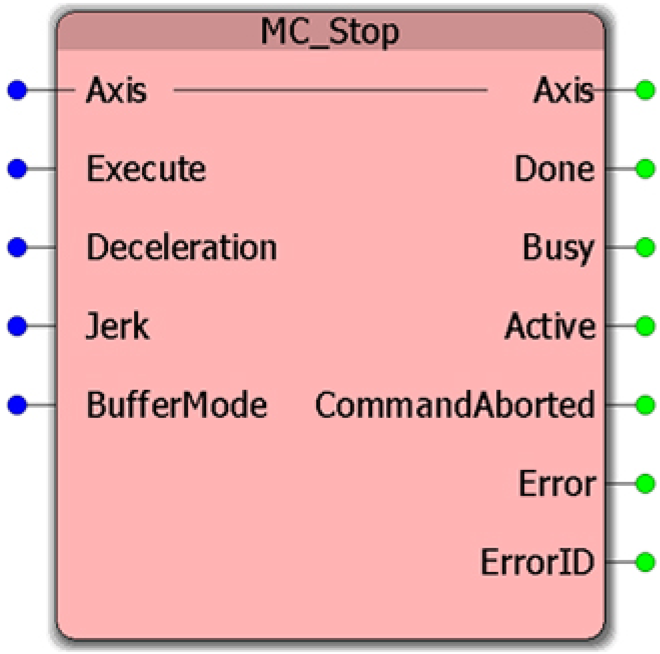
4378	The function block is not applicable for the external axis specified
4379	A homing sequence is already in progress.
4381	Motion aborted due to axis alarm.
4382	When the axis is in rotary mode, and the MC_SetPosition tries to set a position that is equal to or greater than the MachineCycle, this error is generated, and the position is not set.
4383	Axis must be commanded at standstill when homing is attempted.
4390	Position cannot be defined while the axis is the cam master of other axes.
4391	The function block can not be used with a virtual axis.
4396	Axis latch function already in use.
4397	Over travel limit still ON after attempting to move away from it.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4642	Direction does not correspond to a valid enumeration value.
4646	Mode does not correspond to a valid enumeration value.
4658	Velocity parameter is less than or equal to zero.
57620	The structure size does not match.
61713	An internal motion kernel command failed. This error could be caused by inserting multiple MC_Power function blocks in the program for the same axis. Only one MC_Power function per axis is required. (Do not include more than one.)

Example


Coarse Instance
MC_StepLimitSwitch
Fine Instance
MC_StepRefPulse
CarriageDrive
UserStartInput
MC_SwitchNegative
MC_EdgeOn
2m/s
1Nm
20s
200mm
Axis
Execute
Direction
LimitSwitchMode
Velocity
TorqueLimit
TimeLimit
DistanceLimit
Done
Error
Busy
CommandAborted
ErrorID
MC_Positive
0.2m/s
0mm
1Nm
1s
5mm
SetPosition
>=1
HomeErrorOutput
HomeOKOutput
>=1
HomeAbortOutput


RefPulse
(-)
(+)
On
LimitSwitch
Off
On
Coarse FB
MC_StepLimitSwitch
Fine FB
MC_StepRefPulse

MC_Stop



This Function Block commands a controlled motion stop and transitions the axis to the 'Stopping' state. While the axis is in the 'Stopping' state, no other FB can perform motion on the same axis. Other blocks will generate the error, 4370, MotionProhibited. After the axis has reached zero velocity, the Done output is set to TRUE. The axis remains in the 'Stopping' state as long as 'Execute' is still TRUE or zero velocity is not yet reached. When 'Done' is TRUE and 'Execute' is FALSE, the axis goes to the 'StandStill' state.

Parameters

Parameter		Data type	Description
VAR_IN_OUT			
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).
VAR_INPUT			**Default**

B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk $[u/s^3]$. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use.	LREAL#0.0
E	BufferMode	MC_BufferMode	Not supported. The behavior is as if MC_BufferMode#Aborting is set.	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- When 'Execute' is high, the axis remains in the 'Stopping' state and may not execute any other command. Other function blocks will generate the error, 4370, MotionProhibited, if executed.

- While MC_Stop.Execute = TRUE, the axis will be in the stopping state and new moves should be prohibited.

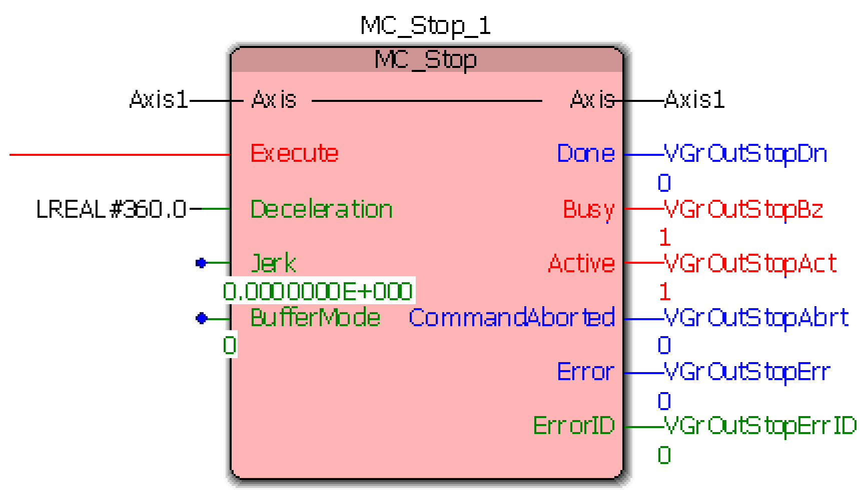
- For further information about the Done output, Profile Complete, and Motion Complete, see the Determining when motion is complete section.

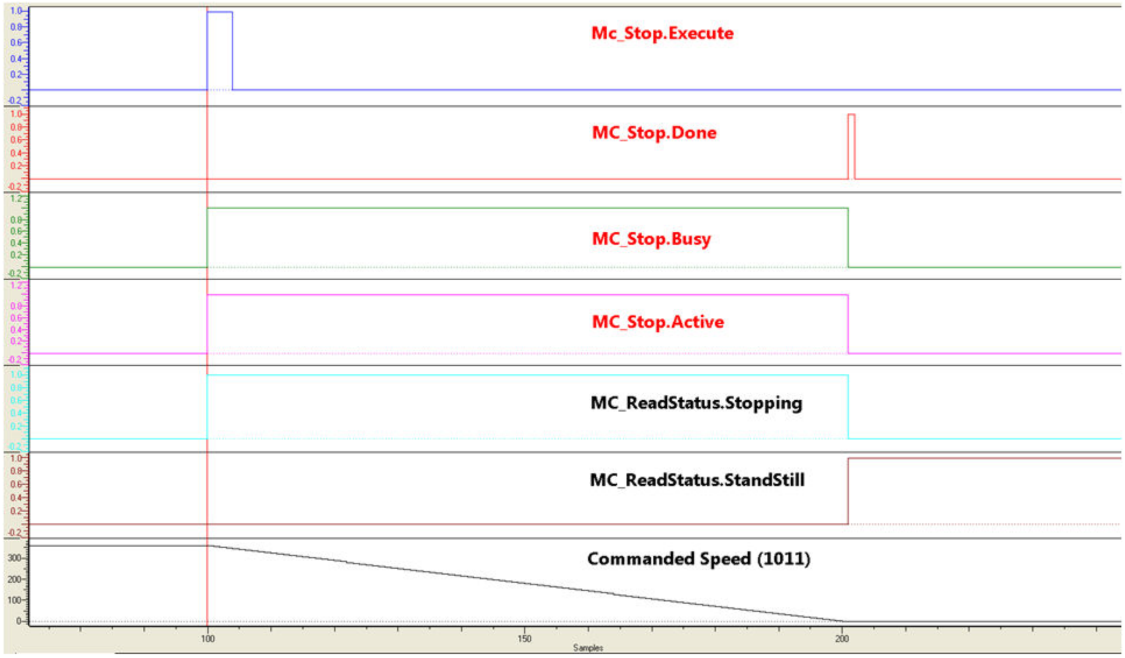
Error Description

ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is

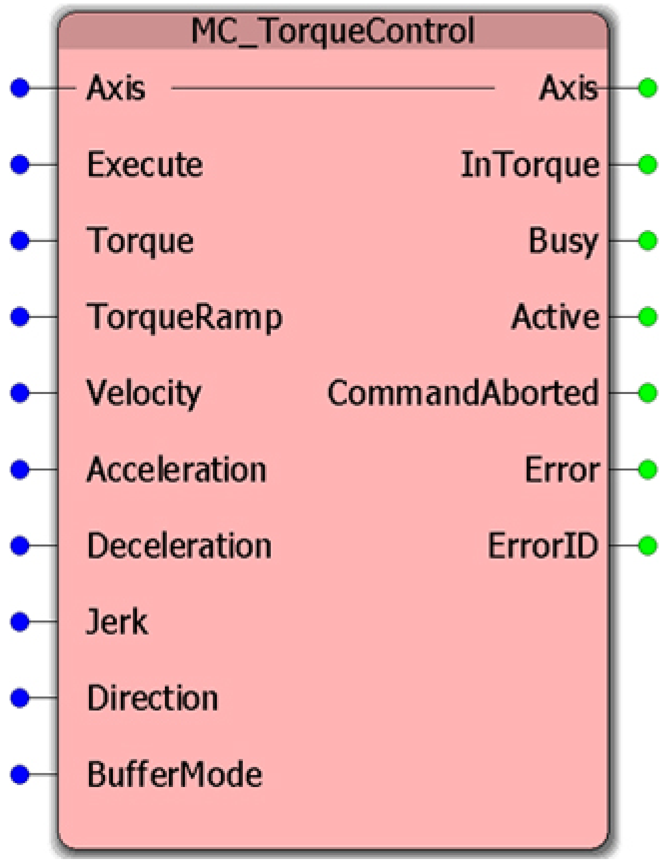
	properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4660	Deceleration is less than or equal to zero.
4893	The specified external axis may not be used. A physical axis is required
57620	The structure size does not match.

Example




Mc_Stop.Execute
MC_Stop.Done
MC_Stop.Busy
MC_Stop.Active
MC_ReadStatus.Stopping
MC_ReadStatus.StandStill
Commanded Speed (1011)
Samples

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.

Parameters

	Parameter	Data type	Description	Default
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	Torque	LREAL	Value of the torque (in percentage of rated torque) The Torque input ultimately specifies the maximum torque that can be applied. If the initial command torque is less than the Torque input, the command torque is increased according to the TorqueRamp input. Similarly, if the initial command torque is greater than the Torque input, the command torque is decreased according to the TorqueRamp input. Once the commanded torque equals the Torque input, the command torque will not change.	LREAL#0.0
E	TorqueRamp	LREAL	The rate at which the set value of the torque or force is achieved (%/s). Example: if Torque is 3.0 and TorqueRamp is 1.0, it will take 3.0 seconds for the set torque to be achieved.	LREAL#0.0
E	Velocity	LREAL	Absolute value of the velocity in user units/second The Velocity input along with the Acceleration and Deceleration inputs determines the velocity limit. If the initial velocity limit is less then the Velocity input, then the	LREAL#0.0

			velocity limit is increased according to the Acceleration input. Similarly, if the initial velocity limit is greater than the Velocity input, then the velocity limit is decreased according to the Deceleration input. Once the velocity limit equals the Velocity input, the velocity limit will not change.	
E	Acceleration	LREAL	Value of the acceleration in user units/second^2 (acceleration is applicable with same sign of torque and velocity)	LREAL#0.0
E	Deceleration	LREAL	Value of the deceleration in user units/second^2 (deceleration is applicable with opposite signs of torque and velocity)	LREAL#0.0
E	Jerk	LREAL	Value of the Jerk [u/s^3]. Value of the jerk in user units/second^3. Jerk not supported . Reserved for future use. (Not supported until future version)	LREAL#0.0
E	Direction	MC_Direction	Specifies the direction of motion. Allowable modes are positive_direction, shortest_way, negative_direction, current_direction. MC_Direction#Positive_Direction MC_Direction#Shortest_Way MC_Direction#Negative_Direction MC_Direction#Current_Direction	MC_Direction#PositiveDirection
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				

B	InTorque	BOOL	Setpoint value of torque or force is reached for the first time
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

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.

5. If the velocity limit is reached, then the actual torque will often be much lower than the command torque. Since the command torque does not change after equaling the Torque input, the actual torque response due to sudden changes in loads is based on the servo drive characteristics, and the rate of change of the actual torque may exceed the TorqueRamp input.

6. If the Torque input is positive, and the Direction input equals MC_Direction#negative_direction, then the Torque input is negated. However, for compatibility with previous versions, the Direction input is ignored if the Torque input is negative since the reverse direction is implied.

7. When a soft position limit is exceeded, MC_TorqueControl will be aborted and the axis will be switched into position mode. In this case, the axis will decelerate according to ServoPack parameters Pn80D, Pn80E, Pn80F and Pn827.

Torque Input	Direction Input	Axis Direction of Motion
Positive	Positive	Positive
Positive	Negative	Negative
Negative	Negative	Negative
Negative	Positive	Negative

Related Function Blocks

Y_DirectControl: Control mode 3 on Y_DirectControl block available from the Y_Motion firmware library allows the user to specify a commanded torque every scan.

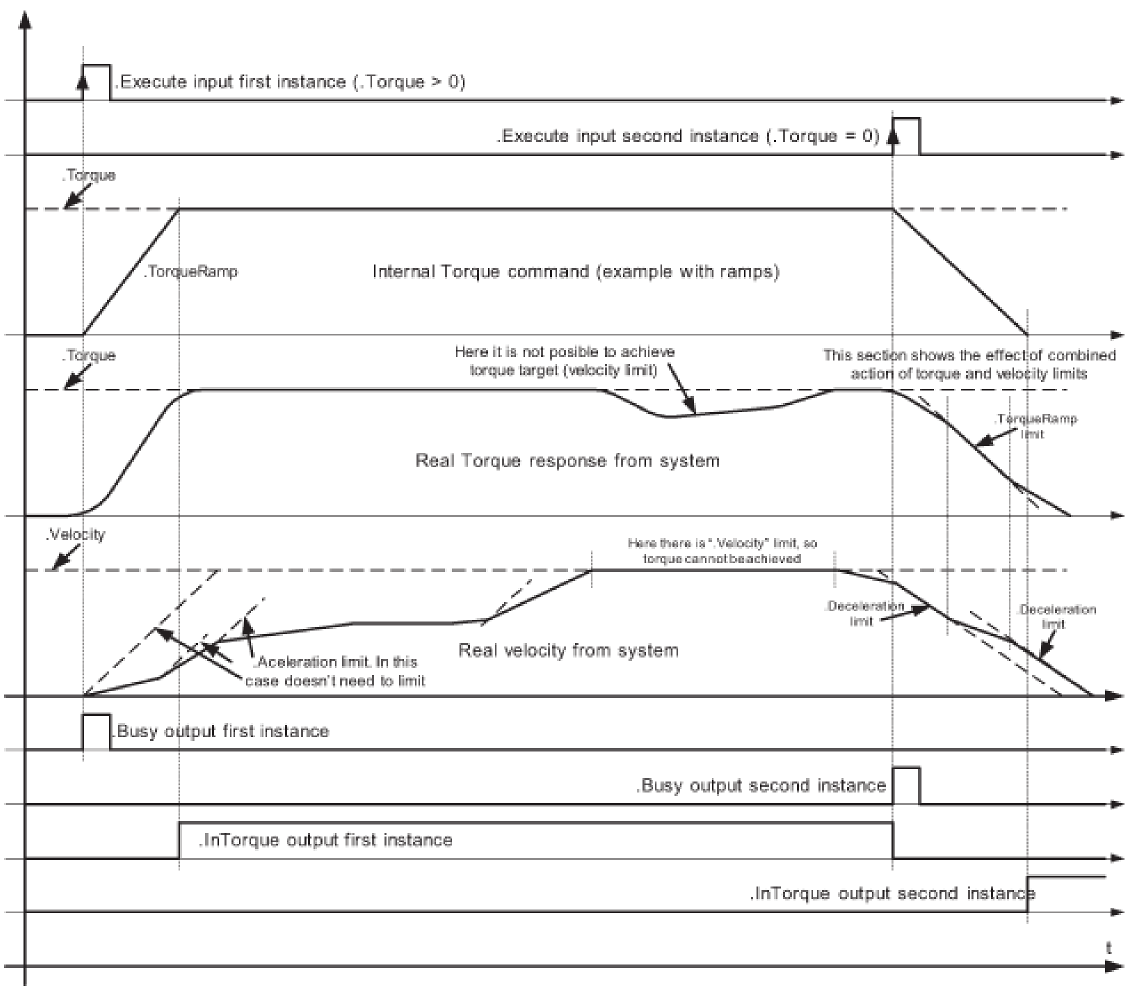
Error Description

ErrorID	Meaning
0	No Error
4369	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.

4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4413	The Stepper axis does not support the mode of motion commanded
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4641	Buffer mode does not correspond to a valid enumeration value.
4642	Direction does not correspond to a valid enumeration value.
4658	Velocity parameter is less than or equal to zero.
4659	Acceleration is less than or equal to zero.
4660	Deceleration is less than or equal to zero.
4661	Torque is less than or equal to zero.
4668	The parameter number does not exist for the specified axis
57620	The structure size does not match.

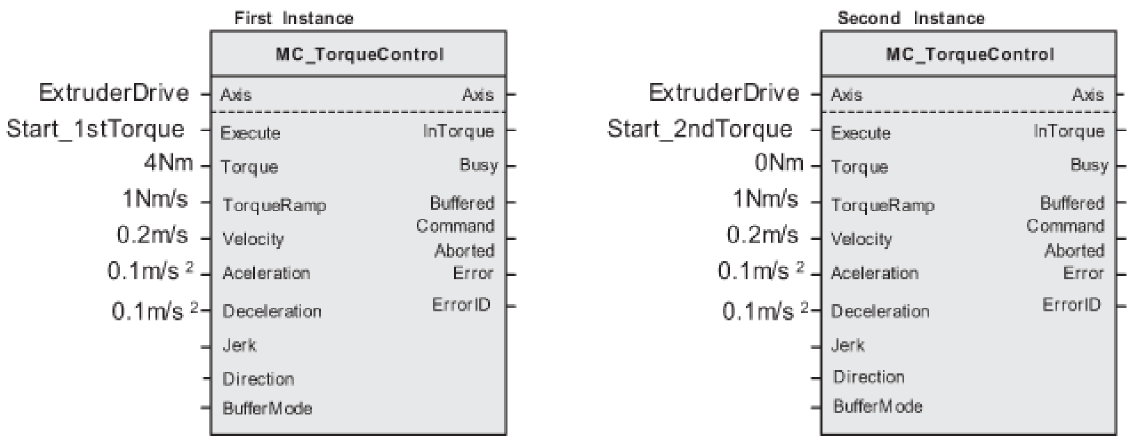
Example

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

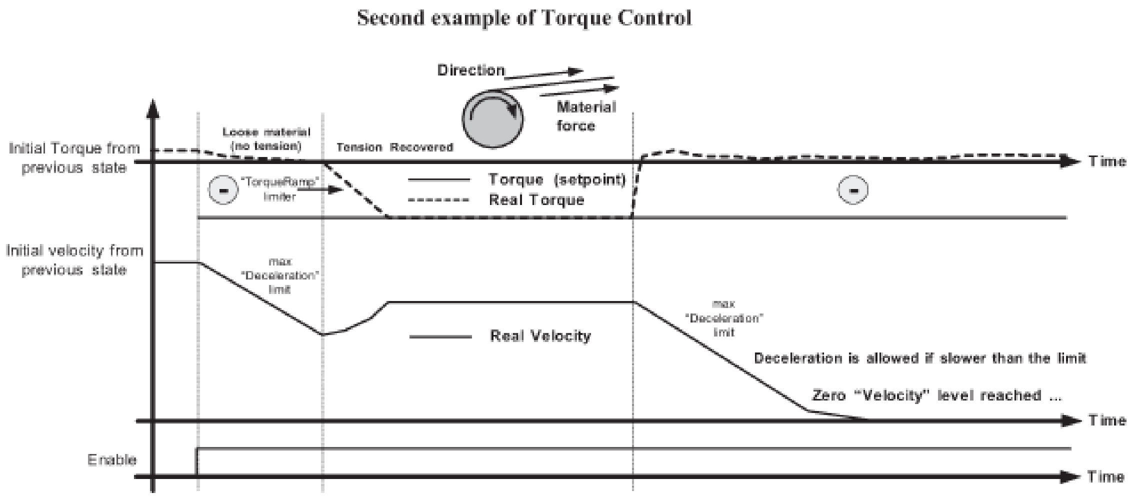


This example could be implemented in a Function Block Diagram as follows:

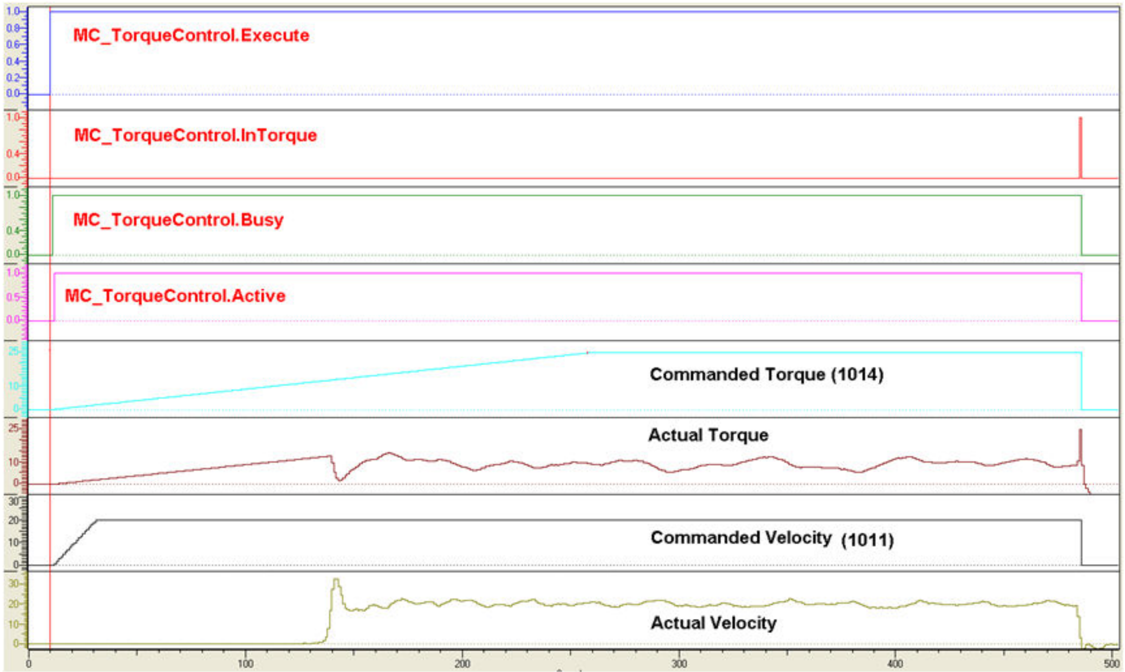
Example of Torque Control



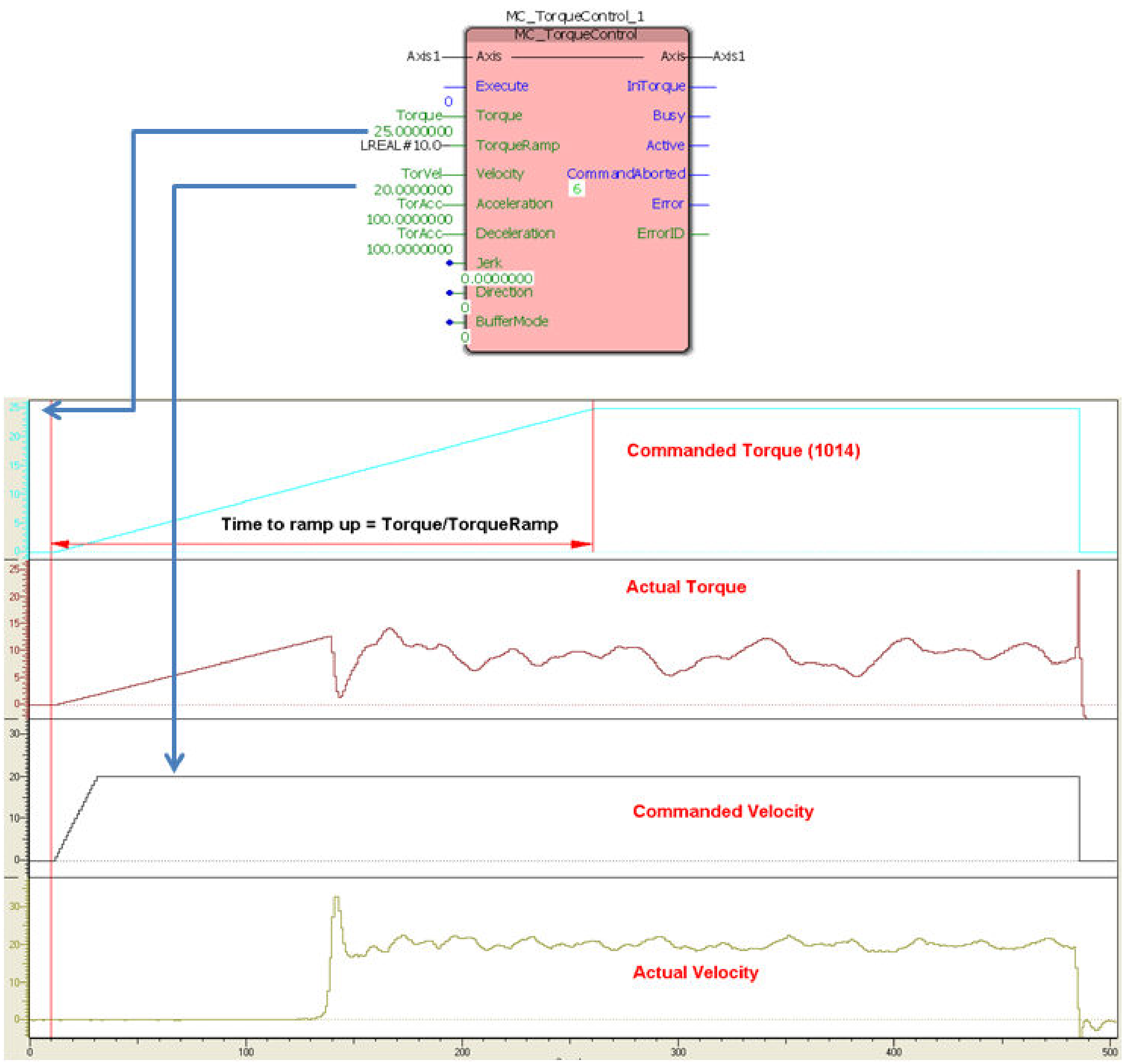
With the second example we use opposite signs for Direction & Torque (e.g. Retention or brake control). (In the FB: +Direction –Torque). It is like an unwinding application with torque on the material, and a break in the material. When the material breaks, as shown in the middle of the picture, this 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.



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.

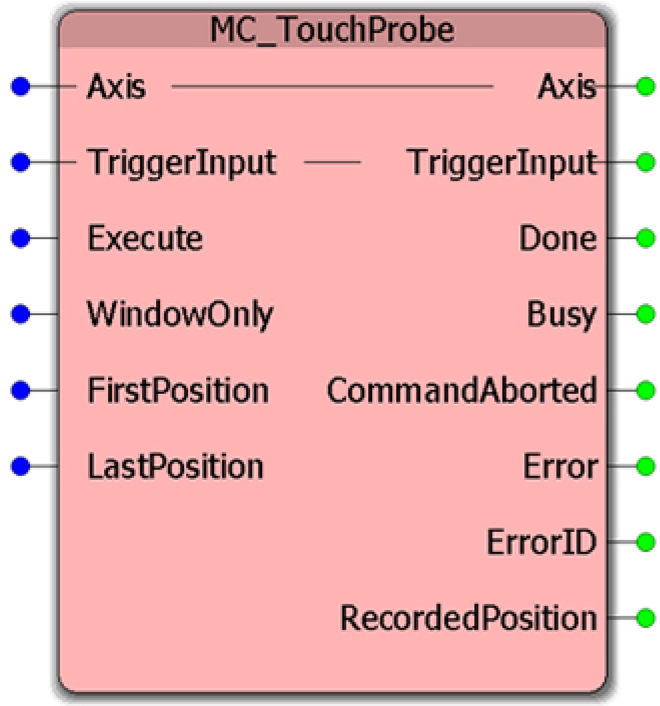


Example of MC_TorqueControl being applied to move an axis to a hard stop



Sample torque and velocity profiles for an instance of MC_TorqueControl

MC_TouchProbe



The function block will output the axis position when a trigger event occurs. This function block is specifically designed to work with the high speed hardware latches on servopacks and LIO option cards. If using the MP3200iec and MECHATROLINK III servopacks, it is possible to execute two MC_Touchprobe function blocks simultaneously to capture the rising and falling edge of a single sensor, or measure the phase difference between two sensors. See the chart below for configuring the TRIGGER_REF structure to capture the position for the desired input.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
E	TriggerInput	TRIGGER_REF	Reference to the trigger signal source.	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
E	WindowOnly	BOOL	This feature is not currently supported.	FALSE
E	FirstPosition	LREAL	This feature is not currently supported.	LREAL#0.0
E	LastPosition	LREAL	This feature is not currently supported.	LREAL#0.0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
B	RecordedPosition	LREAL	Position where trigger event occurred (in user units [u])	

Notes

1. RecordedPosition will reflect the configuration of the axis, meaning that if the axis is configured as rotary type (cyclic) then the RecordedPosition will be modularized to fit within the Machine Cycle. To use the unmodularized latch on an axis configured for rotary mode, use

MC_ReadParameter #1031 [LatchPositionNonCyclic] after the Done output comes on.

2. Refer to the MP2000Siec Hardware Manual or MP2600iec Hardware Manual for specifications regarding the local I/O.

3. Refer to the appropriate servo manual for latch performance data on those devices:

- Sigma II with NS115: SIEPC71080001, see section 9.3
- Sigma III: YEA-SIA-S800-11, see section 10.1.4
- Sigma-5 with rotary motor: SIEPS8000043, see Section 6.1
- Sigma-5 with linear motor: SIEPS8000044, see Section 6.1

4. The following chart details the correct values for the TRIGGER_REF structure based on the hardware latch to be detected.

5. If using the MP3200iec, two latches can be recorded simultaneously. Firmware version 2.5.0 or higher is required.

Device	Signal	Hardware Pin #	Software Default Variable Name	TRIGGER_REF			
				Bit	ID	Input Input_Ref ID	Pattern
				UINT	UINT	UINT	ENUM
LIO-01	Encoder C Channel	A3/B3	n/a	0	User specified. For use with MC_AbortTrigger	Not used, it is implied by AXIS_REF	For future use
	DI-01	A22	M□□_DI_01	1			
LIO-02	Encoder C Channel	A3/B3	n/a	0			
	DI-01	A22	M□□_DI_01	1			
LIO-06	Encoder C Channel	35	n/a	0			
	DI-01	39	M□□_DI_01	1			
MP2600	External C Channel	35	n/a	0			
	Cn13 DI-01	39	MO1_DI_01	1			
SGDH	C Channel	n/a	n/a	0			
	EXT1	44	AX□□_SI4_EXT1	1			
	EXT2	45	AX□□_SI5_EXT2	2			
	EXT3	46	AX□□_SI6_EXT3	3			
SGDS	C Channel	n/a	n/a	0			
	EXT1	10*	AX□□_SI4_EXT1	1			
	EXT2	11*	AX□□_SI5_EXT2	2			
	EXT3	12*	AX□□_SI6_EXT3	3			
SGDV	C Channel	n/a	n/a	0			
	EXT1	10*	AX□□_SI4_EXT1	1			
	EXT2	11*	AX□□_SI5_EXT2	2			
	EXT3	12*	AX□□_SI6_EXT3	3			

□ denotes the node or slot number

* denotes the default pin, which can be changed by setting Pn 511 in the drive.

Note that the Hardware Pin numbers listed for the servopacks above are default pin assignments which can be changed according to the value of drive parameter Pn 511.

Related Function Blocks

MC_AbortTrigger: Aborts function blocks associated with trigger events.

Y_ProbeContinuous: Uses the continuous latch mode supported by the Sigma-5 amplifier.

PLCopen Toolbox - ProductBuffer: Uses MC_TouchProbe and provides an array of recorded latch positions.

Cam Toolbox - CamSlave_FeedToLength: For camming applications that index a slave forward in one direction and require on the fly adjustments of the actual index length based on a sensor input.

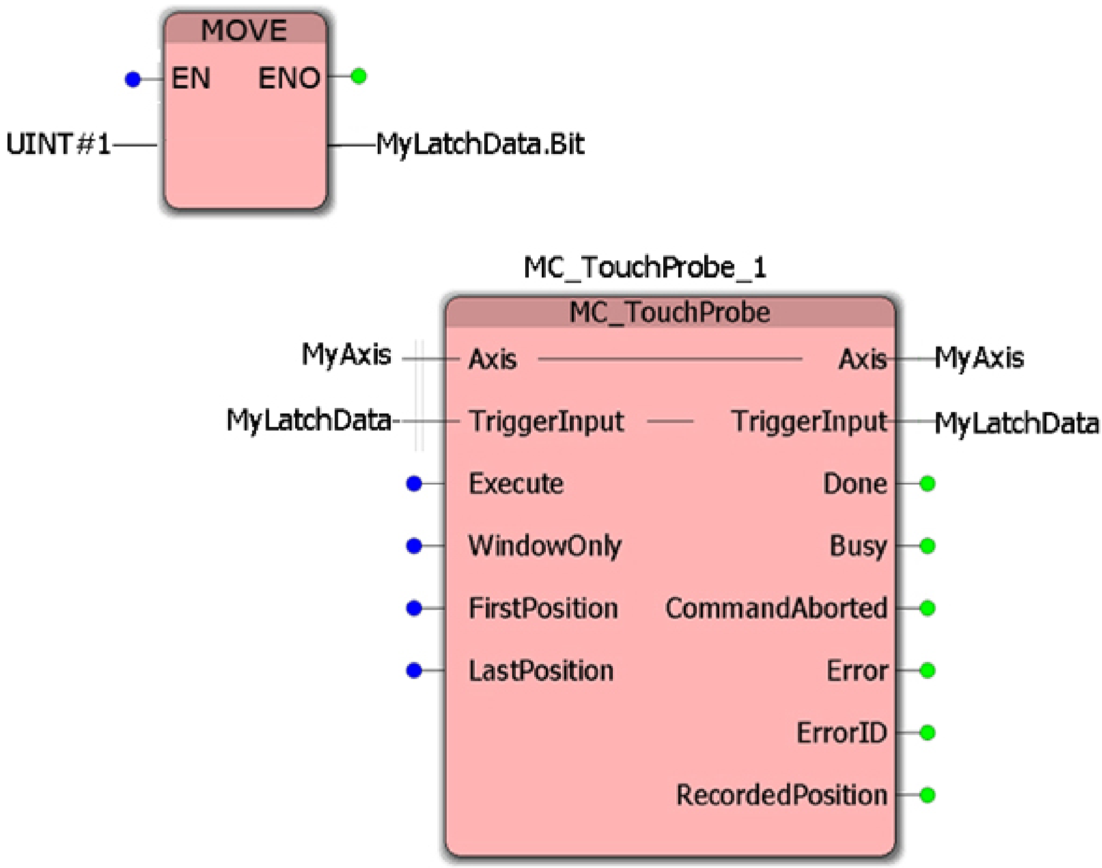
Cam Toolbox - CamSlave_PullToLength: For camming applications where the slave mechanism pulls material forward but the mechanism has a reciprocating stroke.

Error description

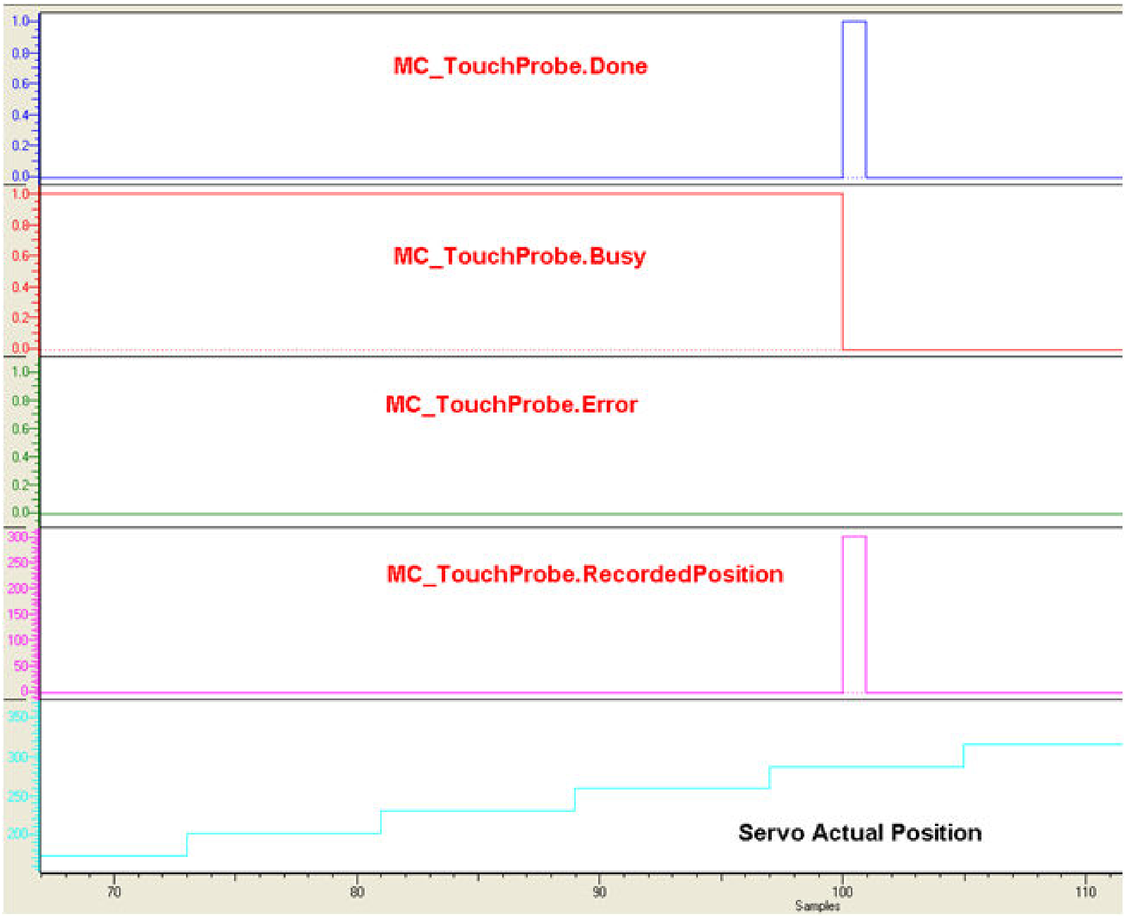
ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4396	Axis latch function already in use.
4406	Continuous Latch Mode is only supported on Sigma V servopacks.
4624	Invalid Structure Value
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4630	Trigger or pattern reference is not valid
4894	The specified virtual axis may not be used with this function block.
57620	The structure size does not match.

Example

If the following logic is programmed on the MP2300Siec controller with an SGDV servo, the MC_TouchProbe block will record the position of the servo axis (MyAxis) when the input connected to pin 10 is fired. Since only the bit field in the TRIGGER_REF structure is used, the following code is effective.

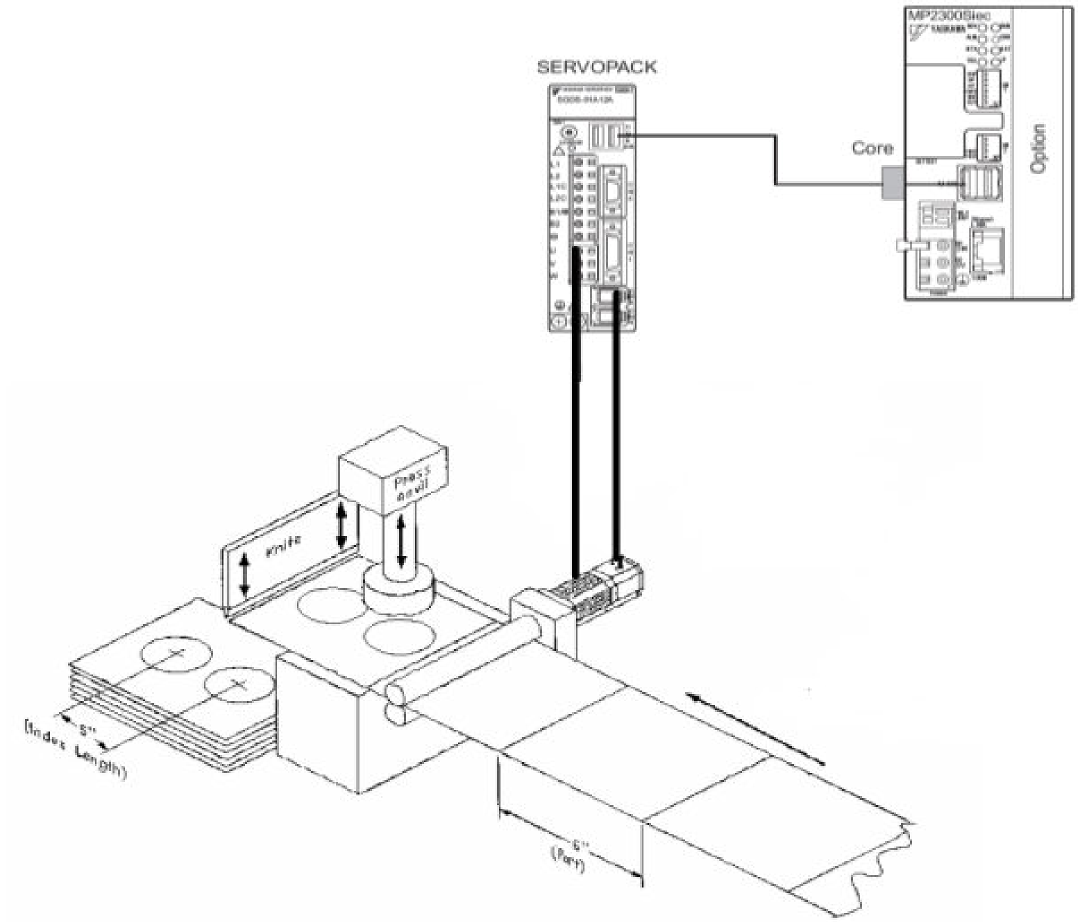


Configuring MC_TouchProbe

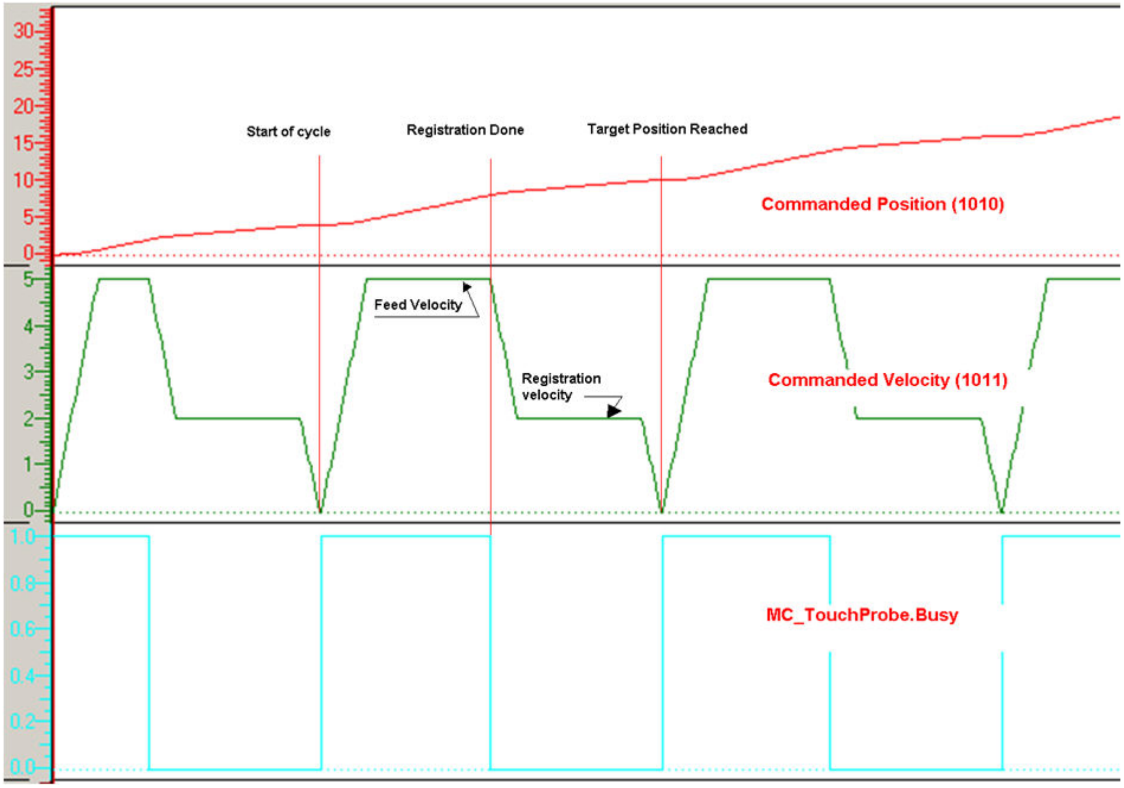


Timing Diagram for MC_TouchProbe

Application Example



Feed to length application for a punch press

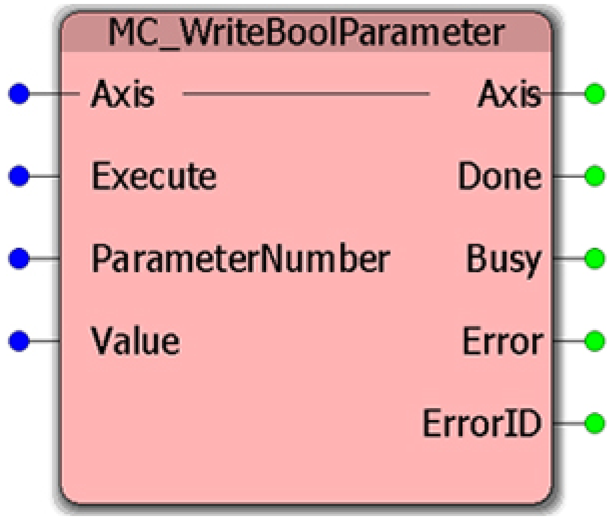


Timing Diagram for Feed to Length application

Example code manual for feed to length application using MC_TouchProbe: http://www.yaskawa.com/site/dmcontrol.nsf/(DocID)/NUNN-7C98L5?opendocument

Example code in MotionWorks IEC Express for feed to length application using MC_TouchProbe: http://www.yaskawa.com/site/dmcontrol.nsf/(DocID)/NUNN-7C98HJ?opendocument

MC_WriteBoolParameter



This Function Block writes the value of an axis specific parameter and is for controller-side parameters only. For amplifier side parameters, refer to Y_WriteDriveParameter.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	ParameterNumber	UINT	Number of the Parameter in the controller	UINT#0
B	Value	BOOL	The drive parameter value	FALSE

VAR_OUTPUT			
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

Refer to parameters with BOOL Data Type in the Axis Parameter List.

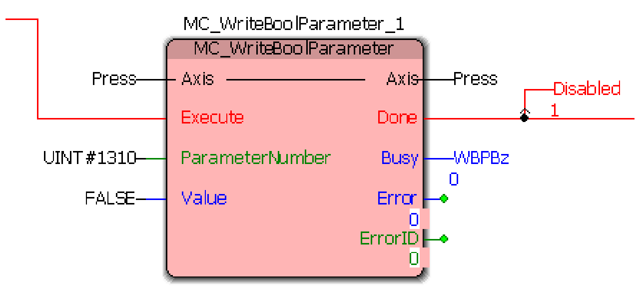
Error Description

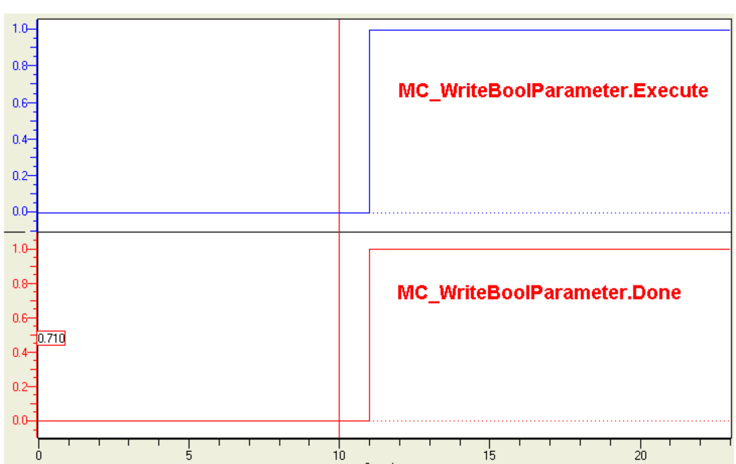
ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4403	The High Speed Output functionality is only available on external encoders.
4409	Parameter write already in progress.
4410	Parameter is read-only.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis

4675	Axis filter time constant out of range, or an attempt to change the value was made while the axis was enabled. (The axis must disabled to change the moving average time constant.)
4898	No filter configured for axis.
4899	Axis position compensation file not found.
4900	Invalid axis position compensation file format.
4901	Cannot enable/disable axis position compensation while servo on.
4902	Invalid compensation table wrap range.
57620	The structure size does not match.

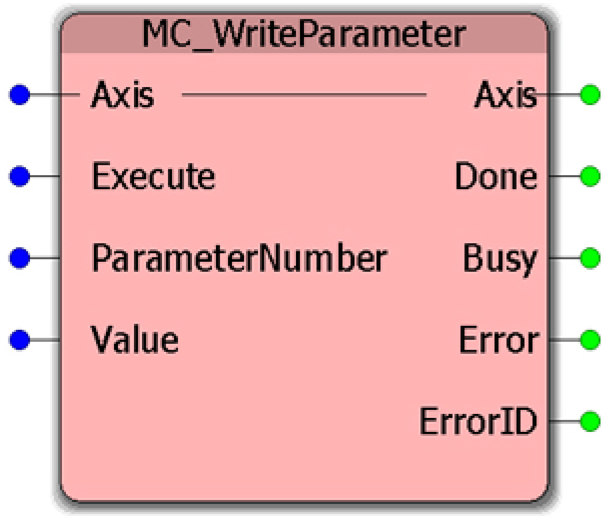
Example

Example of disabling controller velocity feedforward component.





MC_WriteParameter



This Function Block writes the value of an axis-specific parameter and is for controller side parameters only. For amplifier side parameters, refer to Y_WriteDriveParameter.

For a complete list of controller side parameters, refer to parameters with LREAL Data Type in the Axis Parameter List.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	ParameterNumber	UINT	Number of the Parameter in the controller	UINT#0

B	Value	LREAL	The drive parameter value	LREAL#0.0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

Refer to parameters with LREAL Data Type in the Axis Parameter List.

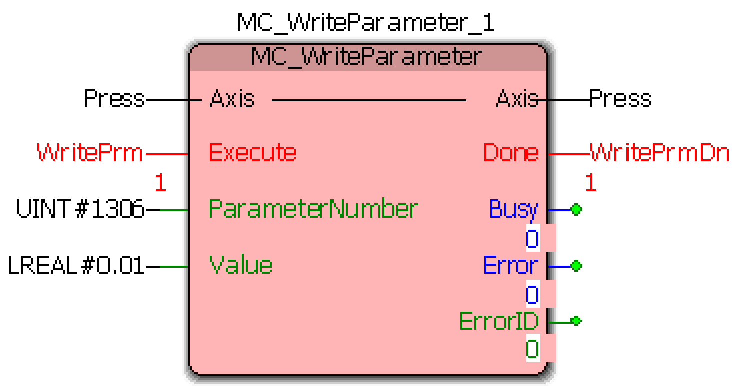
Error Description

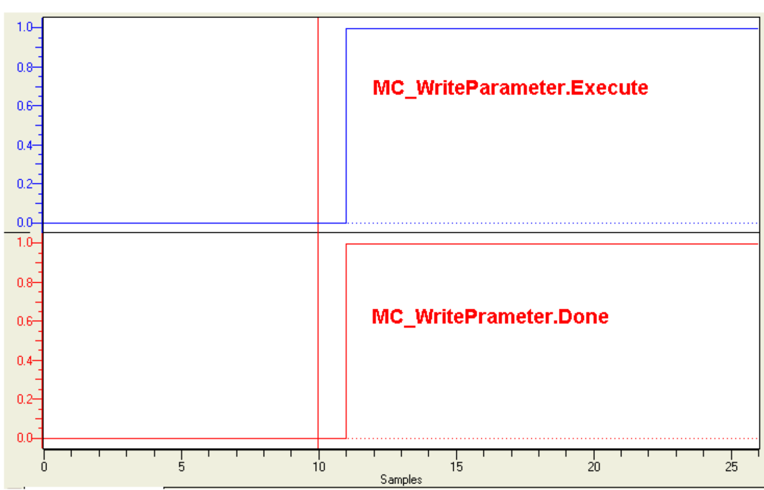
ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4402	The scan compensation delay parameter 1305 is only valid for external encoders.
4403	The High Speed Output functionality is only available on external encoders.
4410	Parameter is read-only.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis

4675	Axis filter time constant out of range, or an attempt to change the value was made while the axis was enabled. (The axis must disabled to change the moving average time constant.)
4676	The time value must be within 0 to 10 MECHATROLINK cycles.
4898	No filter configured for axis.
57620	The structure size does not match.

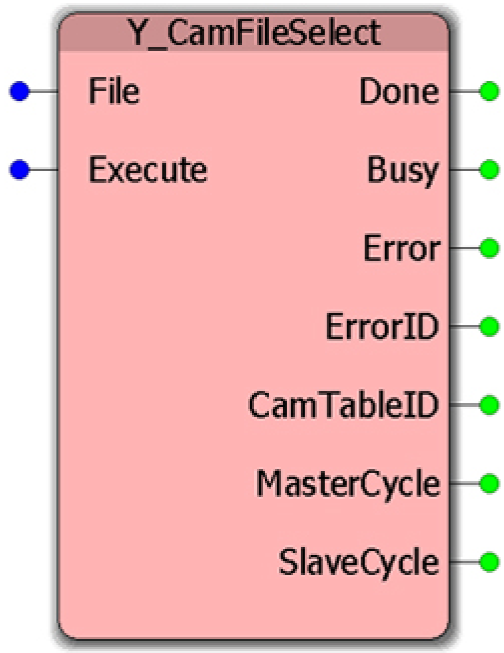
Example

Example of writing controller parameter 1306.





Y_CamFileSelect



This function block loads a cam table from a CSV file into the motion memory.

Parameters

Parameter		Data Type		Description
VAR_INPUT				Default
V	File	STRING	File name of cam table. See Notes and Example for supported format. See Configuring FileName Input for Y_CamFileSelect to see how files in non-default directories can be accessed.	(Empty String)
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input. The Execute input on the Y_CamFileSelect block should be interlocked with the busy output so that the Execute input will not "see" a rising edge while	FALSE

			the busy output is set.	
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
B	CamTableID	UINT	A reference to the cam memory in the motion engine.	
V	MasterCycle	LREAL	Difference between the last and first master positions in the table	
V	SlaveCycle	LREAL	Difference between the last and first slave positions in the table	

Notes

• Supported File Naming Convention: 8.3 format.

• See the section called User File Storage for memory sizes of all MPiec series controllers.

• Supported File format: .CSV file - simple master slave pairs with linear interpolation between the points

1. This file has an optional header with the following values:

    • 'MasterIncremental' (case insensitive): If 'TRUE' (case insensitive) or '1', then the master values are incrementally defined. In other words, each value represents an addition to the previous value. The default is false.

    • 'SlaveIncremental' (case insensitive): If 'TRUE' (case insensitive) or '1', then the slave values are incrementally defined. In other words, each value represents an addition to the previous value. The default is false.

• 'Rows' (case insensitive): Specifies the number of rows to read. Defining this value speeds up reading the file. This header parameter is optional.

2. If incrementally defined, the start of each table is assumed to be zero.

• Once the file is loaded into the motion memory, the CamTableID (and the cam table it refers to) will be valid until Y_ReleaseCamTable is executed or the controller power is cycled (rebooted). Only the CSV file may be stored in flash memory. The cam data transferred to the motion memory resides in RAM.

• To modify the existing cam data (CamTableID already obtained), use Y_ReadCamTable and Y_WriteCamTable in the application program.

• CamTableID can be used by more than one master/slave relationship. Modifying the cam table (via Y_ReadCamTable and Y_WriteCamTable) will affect all relationships.

• If a CamTableID is no longer needed, the application program should release the cam memory using Y_ReleaseCamTable.

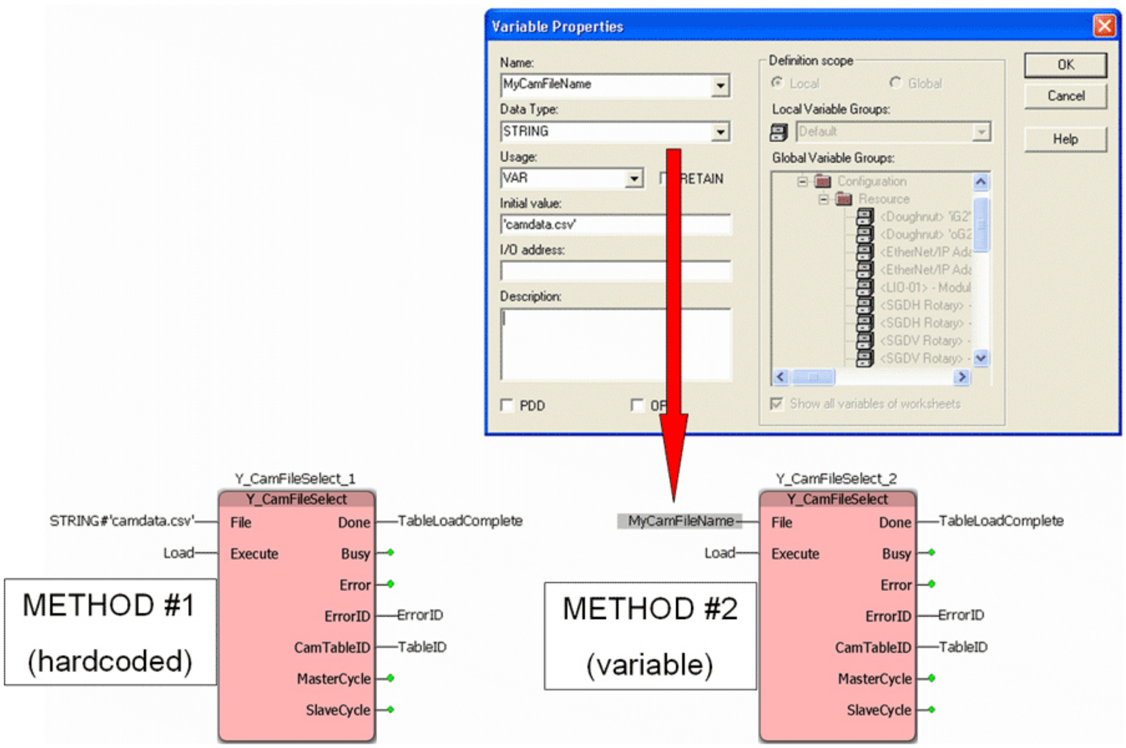
• Refer to Camming Overview for more information regarding cam file creation.

Error Description

ErrorID	Meaning
0	No Error
4377	File reading already in progress
4381	Motion aborted due to axis alarm.
4884	The specified cam file does not exist.
4885	Invalid header for the cam file. You must first populate the TableType and DataSize in the Y_MS_CAM_STRUCT before executing the function.
4886	The first (master) column must be either increasing or decreasing. If the master data is incremental, even the very first point cannot be zero.
4387	File reading already in progress
4895	Missing or unknown file extension

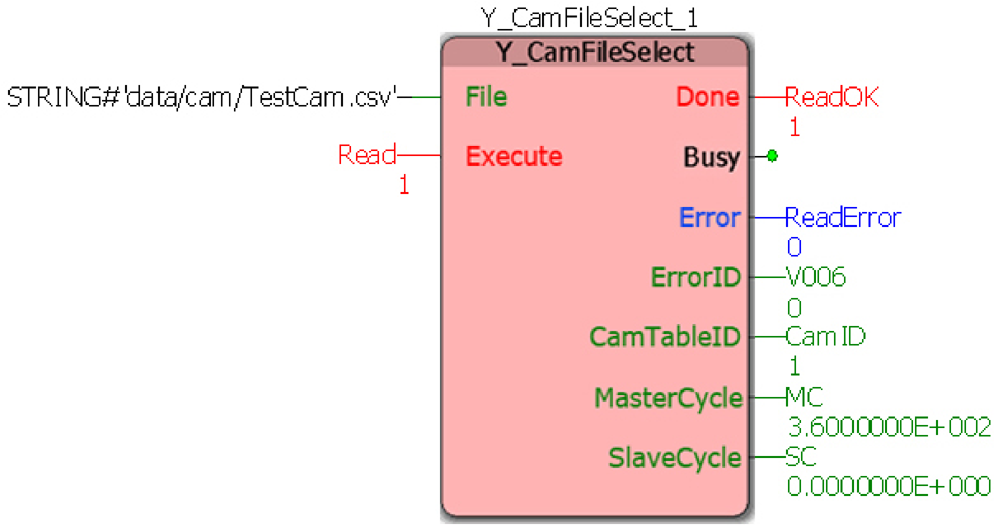
Example 1

Selecting a cam file that has been downloaded using the Resource Dialog window.

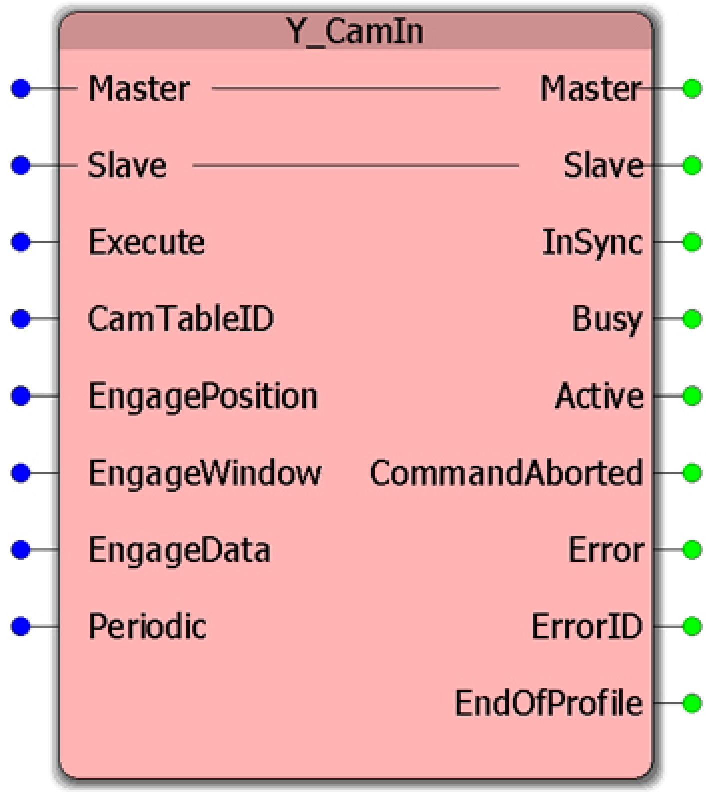


Example 2

Selecting a cam file that has been downloaded using the Hardware Configuration Online Utilities Menu.



Y_CamIn



This Function Block engages the axis in camming mode with the cam profile specified by CamTableID.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Master	AXIS_REF	A logical reference to the master axis	
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute	FALSE

			input.	
B	CamTableID	UINT	A reference to the cam memory in the motion engine.	UINT#0
E	EngagePosition	LREAL	The master position at which the slave starts following the master. Units are those of the cam master. See EngageData details below.	LREAL#0.0
V	EngageWindow	LREAL	The cam will engage at any master position from EngagePosition +/- (EngageWindow)/2. Units are those of the cam master.	1% of the CamMasterCycle
E	Periodic	BOOL	If Periodic is FALSE, the cam profile will be run just once. This eliminates the need to disengage the slave with Y_CamOut. When Periodic is TRUE, the cam profile will repeatuntil Y_CamOut or MC_Stop is executed.	FALSE
V	EngageData	Y_Engage_Data	Structure containing details about how the cam will engage. See EngageData details below.	All zeros in Y_Engage_Data structure
VAR_OUTPUT				
B	InSync	BOOL	Set high when the slave first synchronizes with the master. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
E	EndOfProfile	BOOL	Pulsed output signaling the cyclic end of a CAM Profile	

Notes

• The term 'CamMaster' is used in reference to a machine cycle derived from the selected cam table data. This may be different than the machine cycle of the master axis as defined in the Hardware Configuration. The input parameters of this function block, such as EngagePosition, refer to the CamMaster (Parameter 1502.)

• If the EngageWindow is too small, the CamMaster may pass through the engage window without ever being inside the window during the motion scan (MECHATROLINK or dual port RAM update interval.) This condition is not detected, and the slave may not engage. The slave would be stuck in CamState=1 (Parameter 1540.) See the Camming section for more information about setting the EngageWindow.

• For more information on the behavior of the MPiec Cam function, see the Cam Transition Matrix.

• This function block does not alter (abort, blend, etc.) any existing motion on the slave axis until the CamMaster is within the specified window. Once in this window, any existing motion is aborted. Exception: Y_Engage_Data.Immediate:=TRUE would abort any other motion immediately.

• If Periodic:=TRUE, the last master point in the cam table must equal the master cycle.

• If Periodic:=FALSE, the cam table data may represent a sub-region of the master cycle, but the engage position must be within the table domain.

• EngageData:

Note the various StartModes and their intended application.

Data Type	Value	Comments
StartMode	AtPosition	**(Default)** The slave will engage when the master position is within the range of [EngagePosition +/- (EngageWindow/2)]. MasterRelative is ignored. Use this setting for normal circumstances. The intended usage requires setting YCamIn.Execute:=TRUE at some point before the master and slave are to be synchronized. The motion engine, operating at the MECHATROLINK or dual port RAM update interval will monitor for the exact position to start the camming process.
	Immediate	Y_CamIn does not wait for the master position to reach the EngagePosition. The EngagePosition and the EngageWindow inputs are ignored. This mode is intended for use when the master is not moving, such as during fault or E-Stop recovery in the middle of a cam cycle. In this scenario, the slave may be moved to the equivalent cam position of the master, then the cam can be re-engaged immediately using MasterRelative:=FALSE to preserve the original synchronization. If MasterRelative=TRUE, then CamMasterShift (Parameter 1511) is adjusted so that the master position at the time YCamIn.Execute changes to TRUE corresponds to the start of the table domain. This scenario would change the synchronization between the master and slave. Immediate Mode is not recommended for application scenarios where the master is in motion, as a position drift or phase lag may be introduced.
	Linked	The new cam profile will be switched on the fly at the end of the current cam table. This mode is intended for use when cams with different Machine Cycles are to be run without stopping. Use the Linked mode or applications where the product size must be changed on the fly.
MasterRelative	FALSE	**(Default)** The absolute position of the cam master (Parameter 1512) is used as the cam

		master directly. Example: TDC of master is zero degrees, desired EngagePosition is zero degrees. Use an EngagePosition of 0.0 degrees. Another Example: Assume the machine is faulted with the master at 50% of the cam cycle and the slave at 40% of the cycle. (Assume master has traveled further than the slave because it coasted to a stop during E-Stop.) Using functions available in the Cam Toolbox, determine where the slave must be positioned to re synchronize it with the master, i.e. move the slave forward another 10% of it's cycle. Use StartMode:=Immediate and the cam operation will resume, retaining the original synchronization of the master and slave.
	TRUE	MasterMasterShift (Parameter 1511) is adjusted so that the master position at the time YCamIn.Execute changes to TRUE corresponds to the start of the table domain. This scenario changes the synchronization between the master and slave.
SlaveAbsolute	FALSE	**(Default)** An internal SlaveOffset is set to the slave's initial commanded position when the engage event occurs. The cam table data is effectively offset by the slaves initial position. Prior to Y_CamIn, the programmer should move the slave to a position that corresponds to the desired EngagePosition to ensure proper synchronization with the master. This mode ensures that the slave will not "jump" before starting to follow the master, but the slave motion region may be incorrect, causing it to hit an endstop if it's initial position is not confirmed.
	TRUE	The internal SlaveOffset is NOT adjusted when the engage event occurs. If the slave is not positioned the correct starting location, there will be an instantaneous change in position, possibly resulting in excessive following error or overspeed alarms, or mechanical damage. When using either method, ensure the slave is ready (at a proper position) to follow the master.

RampIn	RampInType#None	The slave will immediately track the cam profile, which might cause an instantaneous change in slave position, based on table data.
	RampInType#Accel	The slave will ramp in based on acceleration constraints. In this case, RampInData1 is the maximum velocity, RampInData2 is the maximum acceleration, RampInData3 is the maximum deceleration, and RampInData4 is the maximum jerk. While the slave is not tracking the master because of the accel applied, the CamState will be 2.
	RampInType#SCurve	The slave will ramp in using an S-Curve. In this case, RampInData1 is the distance over which engaging occurs. While the slave is not tracking the master because of the accel applied, the CamState will be 2.

RampIn is not currently supported.

Error description

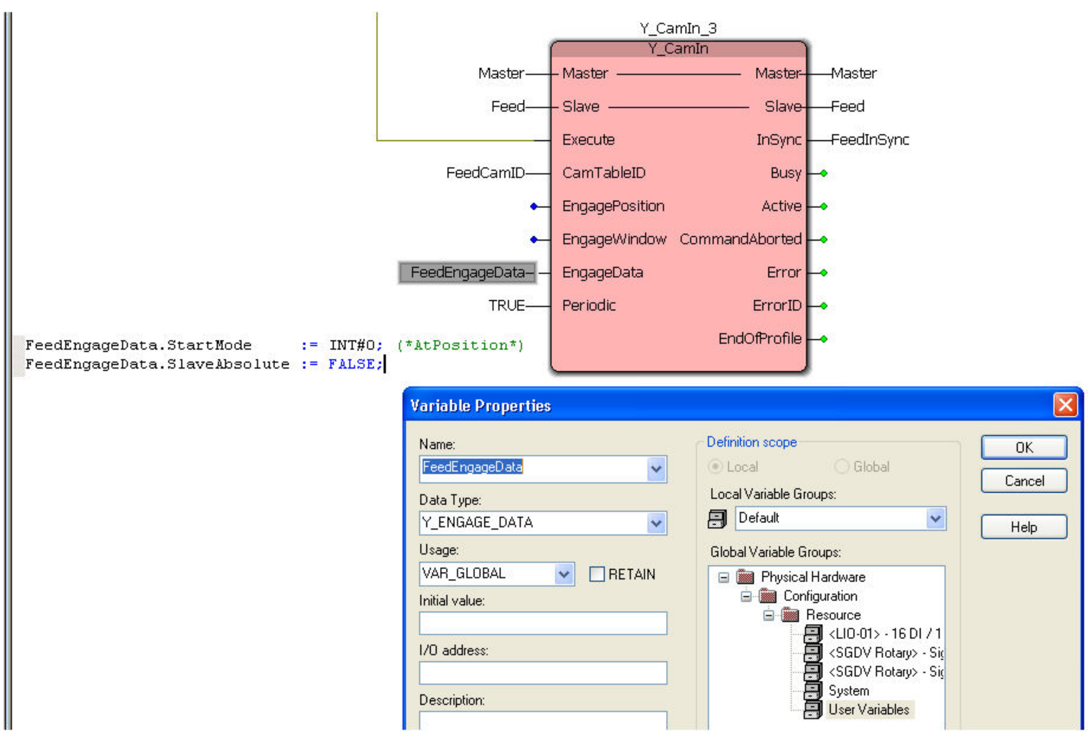
ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified

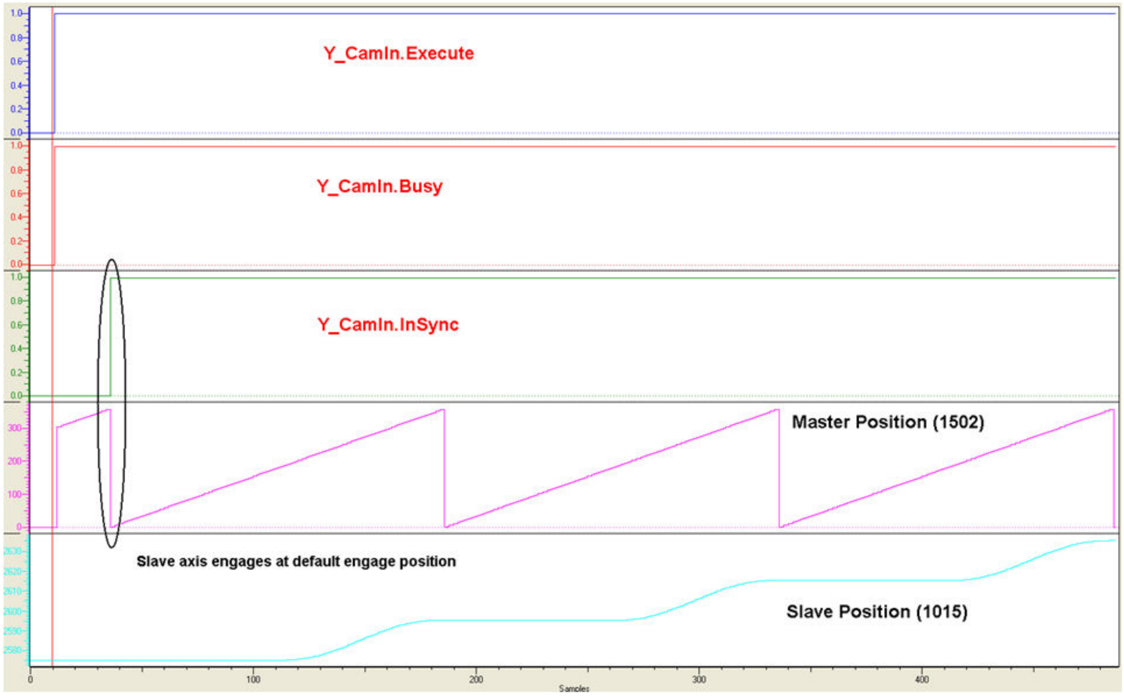
4381	Motion aborted due to axis alarm.
4394	More than 10 Y_CamIn, Y_CamOut, or MC_GearInPos function blocks for a given axis are active at the same time. Most likely the application program is not coded correctly, and the Execute input is being fired too frequently.
4395	Window parameters are outside of the cams Machine Cycle. (0 to Prm1502, the last master position in the active cam table.)
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4626	The master slave relationship is defined. A slave cannot be a master to another axis.
4633	Table size results in misaligned data. Refer to the help section "Internally Created Cam Data." A cam table will have a multiple of 16 bytes if created correctly.
4643	Start mode does not correspond to a valid enumeration value.
4669	Engage position is outside the cam table domain.
4670	Engage window is less than zero.
4887	CamTableID does not refer to a valid cam table.
4891	The slave axis can not be the same as the master axis.
57620	The structure size does not match. This error may occur because data passed to an 'Axis' input on a PLCopen function block is not an AXIS_REF. If you have included a data element into a user structure which includes an AXIS_REF, be sure that the input to the function block is entered correctly.
57874	Argument data is NULL. The EngageData input must be

	connected.
61713	An internal motion kernel command failed. This error could be caused by inserting multiple MC_Power function blocks in the program for the same axis. Only one MC_Power function per axis is required. (Do not include more than one.) On Y_CamIn, check the associated cam table for duplicated or decreasing master positions.

Example 1

Example 1 shown below illustrates cam engage with 'AtPosition' start mode.

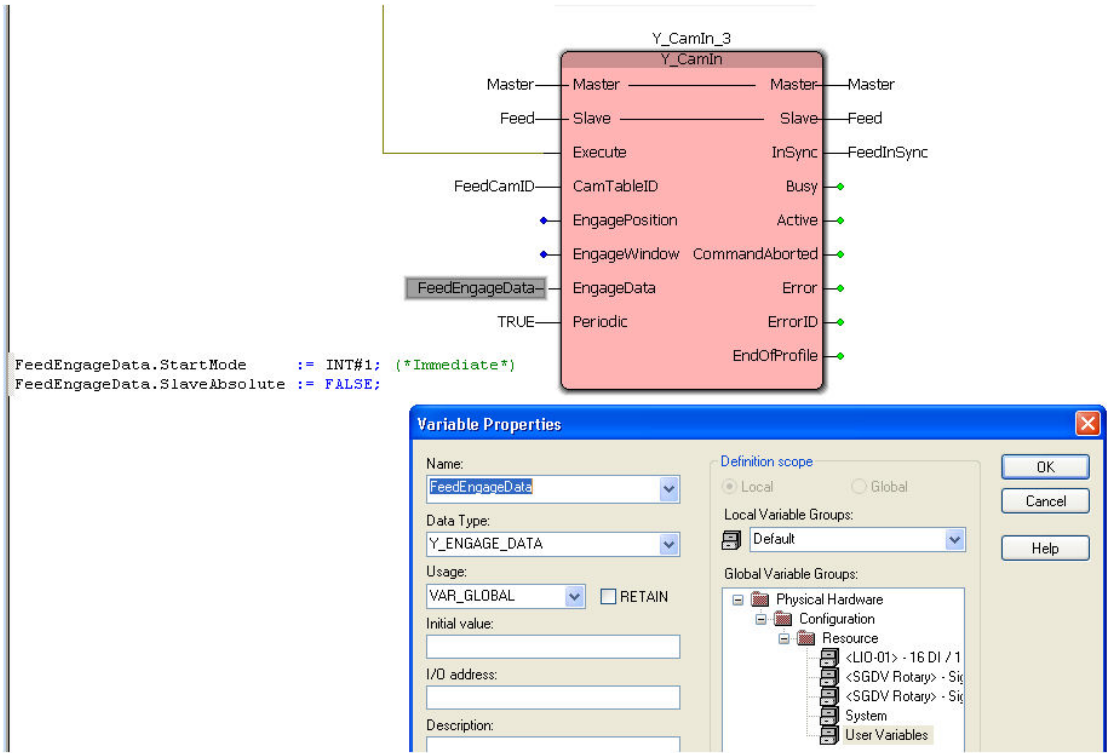


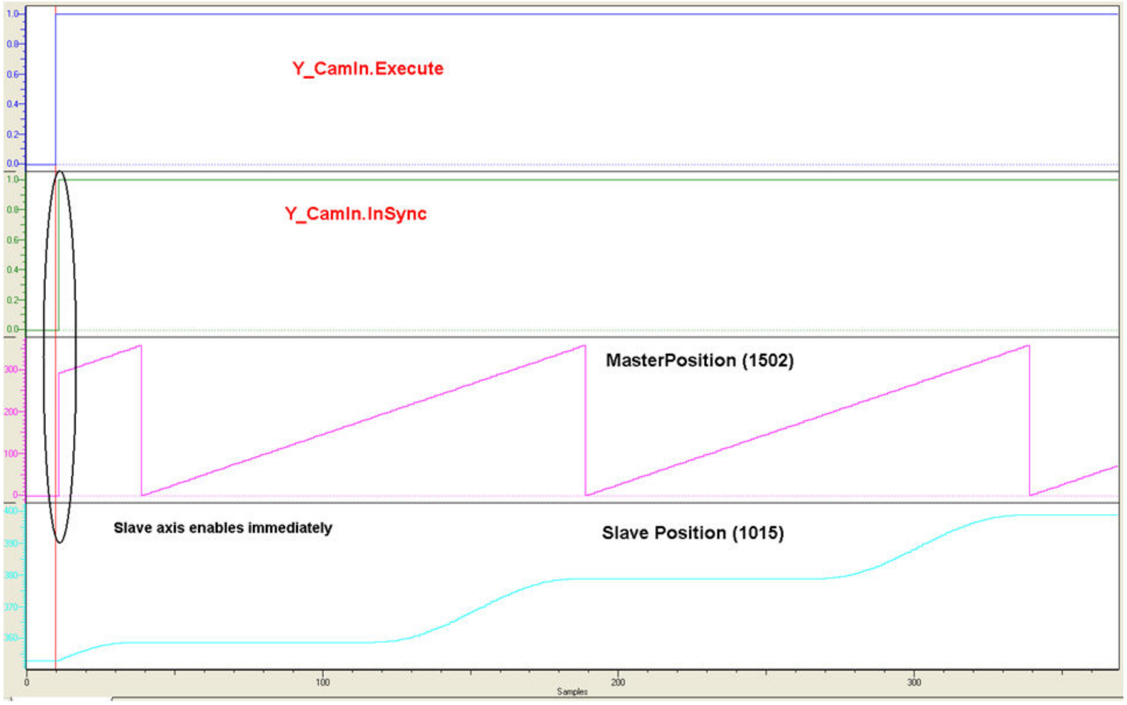


Notice that the function block executed before the actual master and slave started camming. This allows the Y_CamIn function block to be placed in a slower application task while the motion task running at the MECHATROLINK or dual port RAM update interval monitors for the EngagePosition.

Example 2

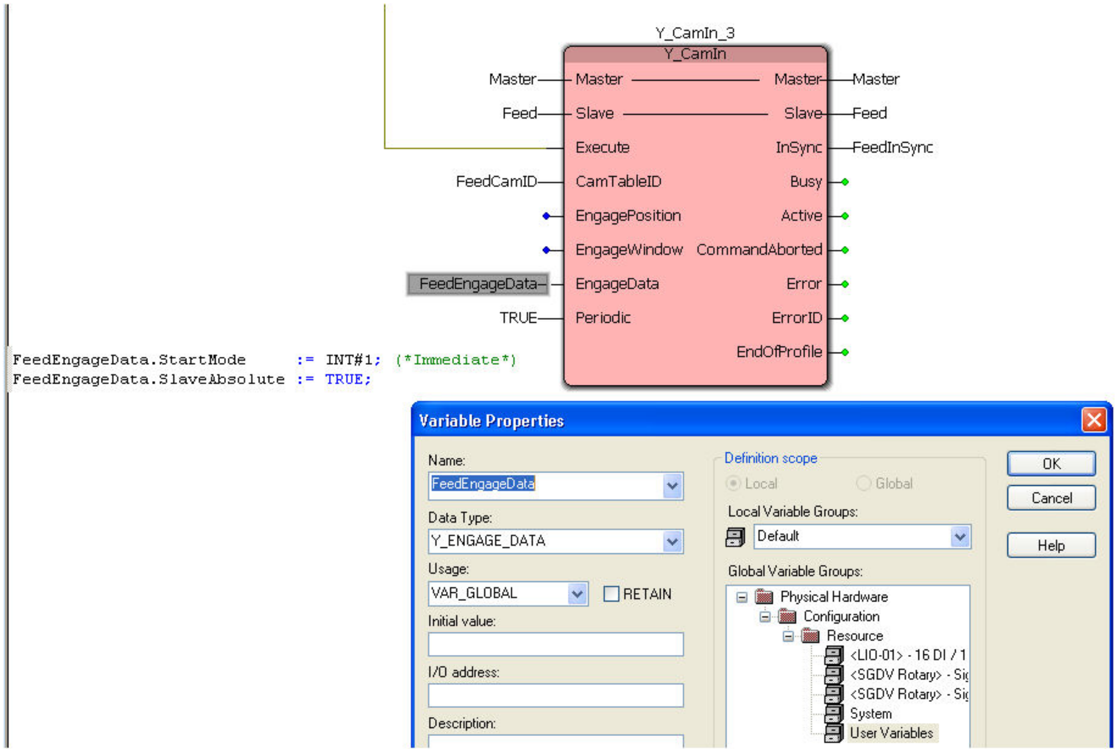
Example 2 shown below illustrates cam engage with 'Immediate' start mode.

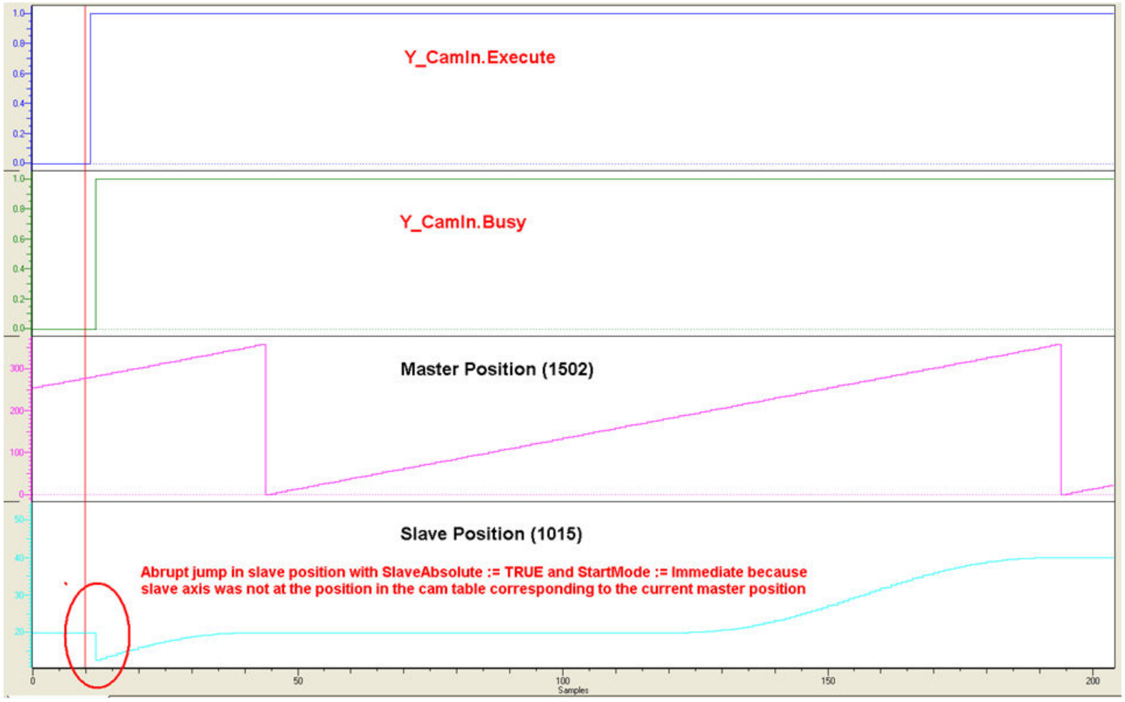



Y_CamIn.Execute
Y_CamIn.InSync
MasterPosition (1502)
Slave axis enables immediately
Slave Position (1015)
Samples

Example 3

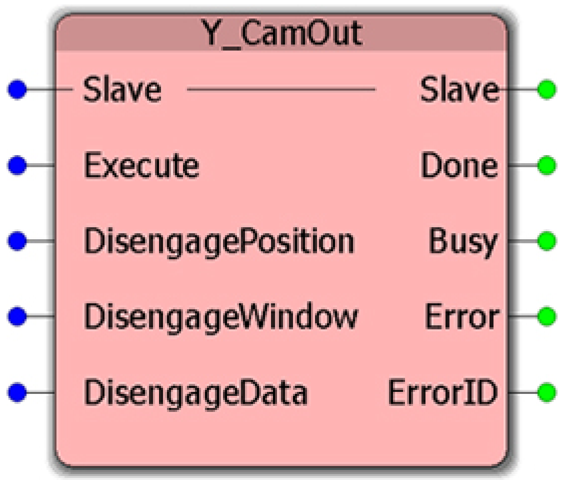
Example 3 shown below uses the 'SlaveAbsolute' engage mode and 'Immediate' start mode.




Y_CamIn.Execute
Y_CamIn.Busy
Master Position (1502)
Slave Position (1015)
Abrupt jump in slave position with SlaveAbsolute := TRUE and StartMode := Immediate because
slave axis was not at the position in the cam table corresponding to the current master position
Samples

YASKAWA

Y_CamOut



This Function Block disengages a Slave axis from its Master axis when the master is at the DisEngagePosition.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
V	DisengagePosition	LREAL	The master position at which the slave will stop following the master. Units are those of the cam master.	LREAL#0.0
V	DisengageWindow	LREAL	The slave will disengage at any master position from	1% of the Master Cycle

			DisengagePosition +/- (DisengageWindow)/2. Units are those of the cam master.	
V	DisengageData	Y_Disengage_Data	Structure containing details about how the cam will disengage.	All zeros in Y_Disengage_Data structure
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• Unlike the PLCopen standards describing MC_CamOut, the slave's final camming velocity is NOT held; the slave will stop and hold position at the disengage event.

• If Y_CamOut is executed when the axis is not associated to a cam master, there is no error; the Done output immediately becomes TRUE.

• For the DisengageData input, only Y_Disengage_Method#AtPosition is currently supported. To immediately stop the slave from being synchronized to the master, use MC_Stop for the slave axis.

• For more information on camming transitions, see the Cam Transition Matrix.

• See the Camming section for more information about the DisengageWindow.

• For further information about the Done output, Profile Complete, and Motion Complete, see the Determining when motion is complete section.

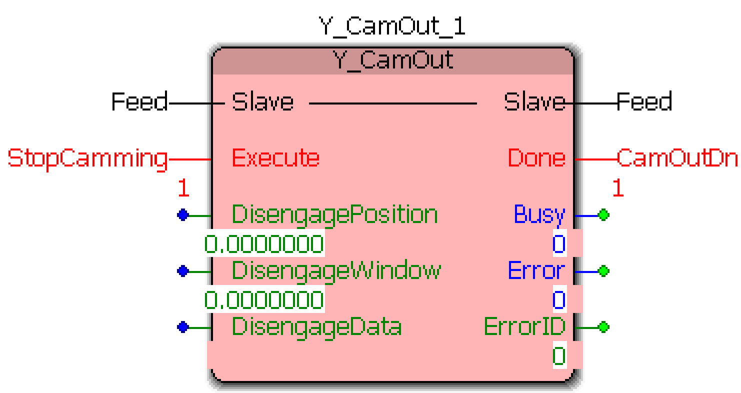
Error Description

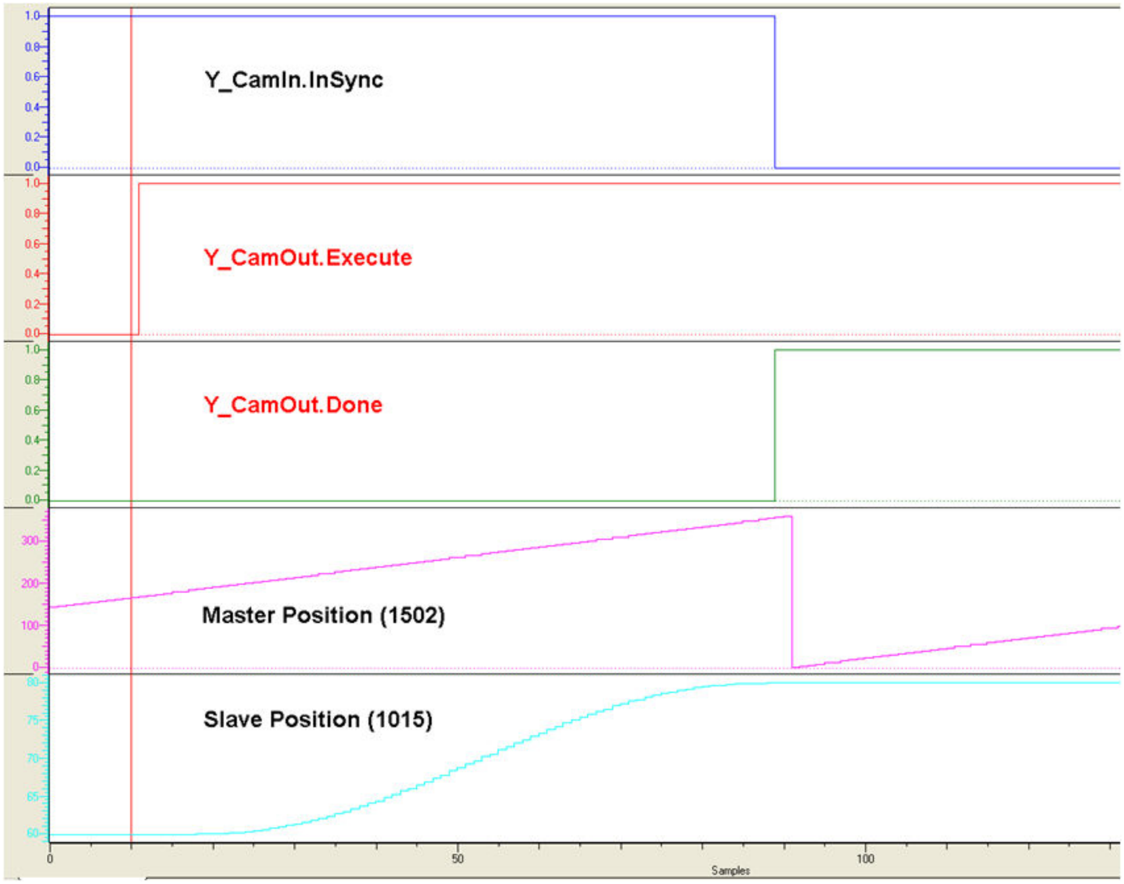
ErrorID	Meaning
0	No Error
4375	CamOut called while not camming.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4394	More than 10 Y_CamIn, Y_CamOut, or MC_GearInPos function blocks for a given axis are active at the same time. Most likely the application program is not coded correctly, and the Execute input is being fired too frequently.
4395	Window parameters are outside of the cams Machine Cycle. (0 to Prm1502, the last master position in the active cam table.)
4405	Y_CamOut was aborted.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4652	Y_CamOut only supports "AtPosition"

4671	Disengage position is outside the cam table domain.
4672	Negative Disengage Window
57620	The structure size does not match.
57874	Argument data is NULL. The EngageData input must be connected.

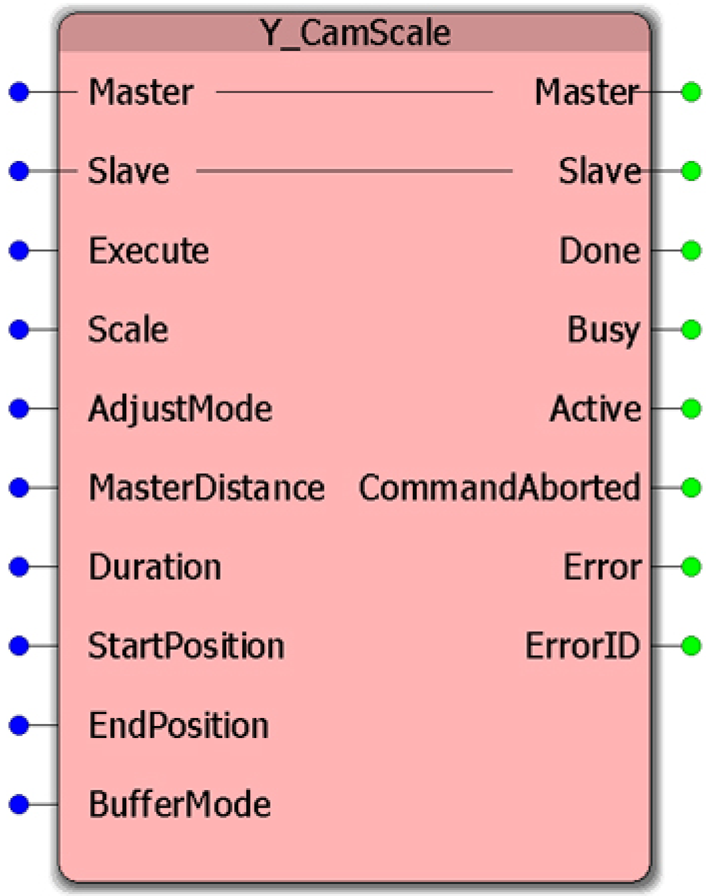
Example

The example shown below illustrates a Y_CamOut disengaging a cam relation at the default disengage position of 0.0.




Y_CamIn.InSync
Y_CamOut.Execute
Y_CamOut.Done
Master Position (1502)
Slave Position (1015)
Samples

Y_CamScale



This Function Block multiplies cam slave position data derived from the cam table by a scale factor.

Parameters

	Parameter	Data Type	Description	Default
VAR_IN_OUT				
B	Master	AXIS_REF	A logical reference to the master axis	
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and	FALSE

			the function is initiated. To modify an input, change the value and re-trigger the execute input.	
V	Scale	LREAL	Percentage	LREAL#0.0
V	AdjustMode	Y_AdjustMode	AdjustMode is an integer with the following values: Y_AdjustMode#MasterDistance: the scale starts immediately and completes when the master has travelled the specified distance. Y_AdjustMode#ElapsedTime: the scale starts immediately and completes within the specified time. Y_AdjustMode#WithinRange: the scale starts when the master is crosses the StartPosition and completes when the master reaches the EndPosition.	Y_AdjustMode#MasterDistance
V	MasterDistance	LREAL	Only used if AdjustMode = Y_AdjustMode#MasterDistance. This is the change in master position from when the function block first executes until the adjustment is complete. Units are those of the cam master.	LREAL#0.0
V	Duration	LREAL	Only used if AdjustMode = Y_AdjustMode#ElapsedTime. Units are seconds.	LREAL#0.0
V	StartPosition	LREAL	Only used if AdjustMode = Y_AdjustMode#WithinRange. The initial position of the master where it is possible to start making the adjustment. Units are those of the cam master.	LREAL#0.0
V	EndPosition	LREAL	Only used if AdjustMode = Y_AdjustMode#WithinRange. The final position of the master where the adjustment must be completed. Units are those of the cam master.	LREAL#0.0
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow,	MC_BufferMode#Aborting

			BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• The Scale amount is absolute. If the current scaling is at 110%, and this function block is executed with a Scale input parameter value of 115%, this function will increase the scaling an additional 5%.

• The underlying table is not affected; this function block only scales the result of cam table lookup.

• 100.00% scaling will cause no scaling of the cam data.

• This function uses a modified sine pattern to ‘meter in’ the change from 0 to 100% of the adjustment change required as described above.

• A master/slave relationship is defined the first time a Y_CamIn, Y_CamShift, Y_CamScale, or Y_SlaveOffset block completes (Done output is TRUE.) The "first time" is defined as power up or after completion of Y_CamOut. If the master/slave relationship is already defined, then it is checked for consistency, and if not correct, the block produces an error (Invalid master slave combination).

• Only BufferMode=MC_BufferMode#aborting or MC_BufferMode#buffered is supported. If MC_BufferMode#aborting, then the function block will abort any phase shifts, cam shifts, cam scaling that are currently taking place; it will not abort the underlying Cam or gearing. If MC_BufferMode#buffered, then the phase shift will occur after all previous phase shifts, cam shifts, cam scalings complete.

• For more information on cam scale behavior, see the Cam Transition Matrix.

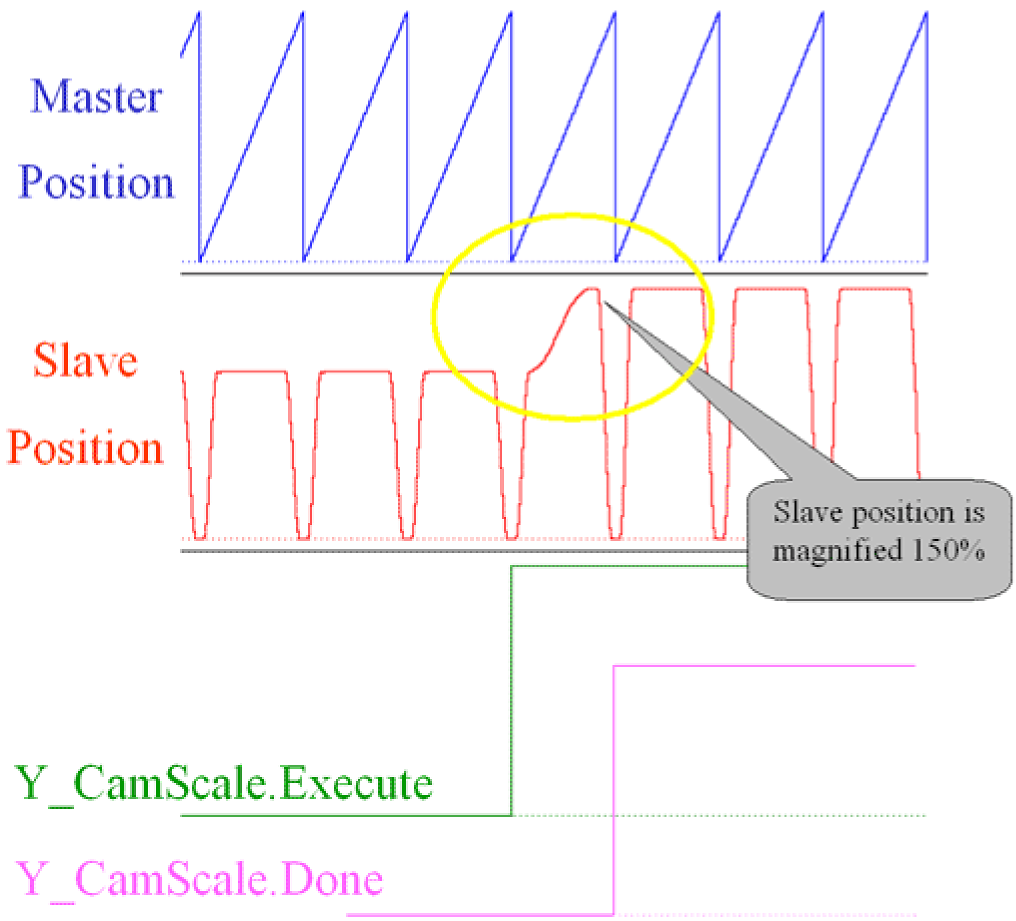
• For more information on how cam scale applies to camming, see the Camming Block Diagram.

Error Description

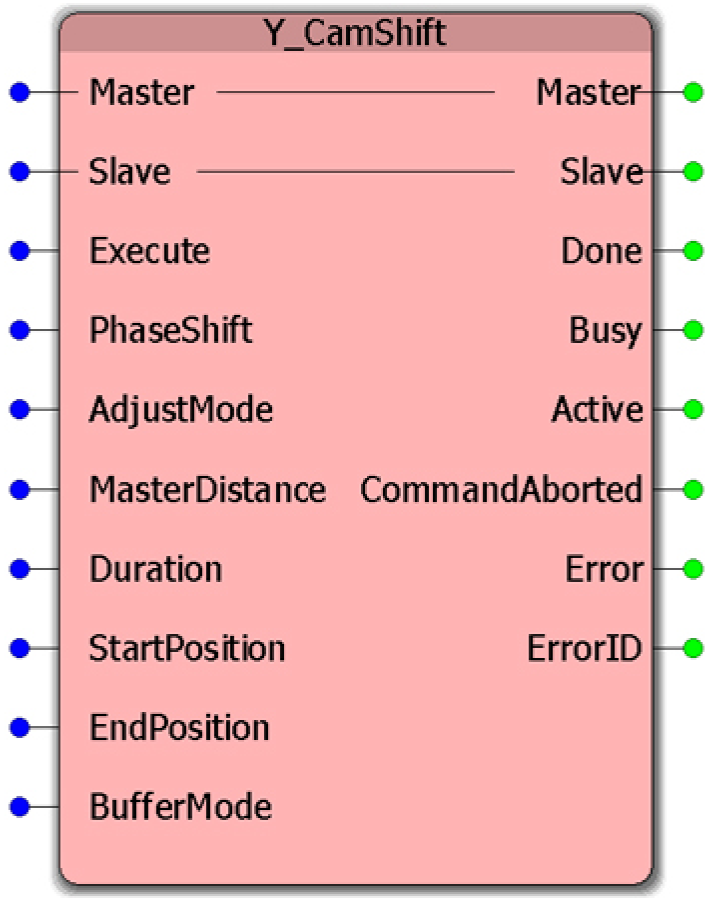
ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4374	Torque move prohibited while non-torque moves queued or in progress.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.

4626	The master slave relationship is defined. A slave cannot be a master to another axis.
4633	Table size results in misaligned data. Refer to the help section “Internally Created Cam Data.” A cam table will have a multiple of 16 bytes if created correctly.
4649	Invalid adjust mode
4657	Distance parameter is less than or equal to zero.
4663	Specified time was less than zero.
4673	StartPosition is outside of master's range.
4674	EndPosition is outside of master's range.
57620	The structure size does not match.

Timing Diagram



Y_CamShift



This Function Block dynamically modifies the master - slave relationship by adding a perceived offset to the master position, effectively causing the slave to advance or retard from the originally specified synchronization data in the cam data table.

Parameters

Parameter		Data Type	Description
VAR_IN_OUT			
B	Master	AXIS_REF	A logical reference to the master axis
B	Slave	AXIS_REF	A logical reference to the slave axis

VAR_INPUT				Default
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
V	PhaseShift	LREAL	The relative amount of adjustment required in Master reference units.	LREAL#0.0
V	AdjustMode	Y_AdjustMode	AdjustMode is an integer with the following values: Y_AdjustMode#MasterDistance: The adjustment starts immediately and completes when the master has travelled the specified MasterDistance. Y_AdjustMode#ElapsedTime: The adjustment starts immediately and completes within the specified Time. Y_AdjustMode#WithinRange: The adjustment starts when the master first crosses the StartPosition and completes when the master reaches the EndPosition.	Y_AdjustMode#MasterDistance
V	MasterDistance	LREAL	Only used if AdjustMode = Y_AdjustMode#MasterDistance. This is the change in master position from when the function block first executes until the adjustment is complete. Units are those of the cam master.	LREAL#0.0
V	Duration	LREAL	Only used if AdjustMode = Y_AdjustMode#ElapsedTime. Units are seconds.	LREAL#0.0
V	StartPosition	LREAL	Only used if AdjustMode = Y_AdjustMode#WithinRange. The initial position of the master where it is possible to start making the adjustment. Units are those of the cam master.	LREAL#0.0
V	EndPosition	LREAL	Only used if AdjustMode = Y_AdjustMode#WithinRange. The	LREAL#0.0

			final position of the master where the adjustment must be completed. Units are those of the cam master.	
E	BufferMode	MC_BufferMode	Defines the behavior of the axis - allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	MC_BufferMode#Aborting
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• The PhaseShift amount input is a relative shift from the current absolute shift value as stored in Parameter 1511, CamMasterShift.

• This function uses a modified sine pattern to 'meter in' the adjustment from the current adjustment to current + PhaseShift. The effects of multiple Y_CamShifts are cumulative.

• Only BufferMode=MC_BufferMode#aborting and MC_BufferMode#buffered are supported. If MC_BufferMode#aborting, then any phase shift, cam shift, cam scaling that are currently taking place; it will not abort the underlying Cam or gearing. If MC_BufferMode#buffered, then the phase shift will occur after all previous phase shifts, cam shifts, cam scalings are complete.

• The shift is allowed to occur over multiple cycles of the master if the application requires this. This is only possible in Y_AdjustMode#MasterDistance by setting MasterDistance to a value larger than the Master Machine Cycle, or with Y_AdjustMode#ElapsedTime, by setting the Time input larger then the time it takes for the machine to complete one cycle.

• For more information on cam shift behavior, see the Cam Transition Matrix.

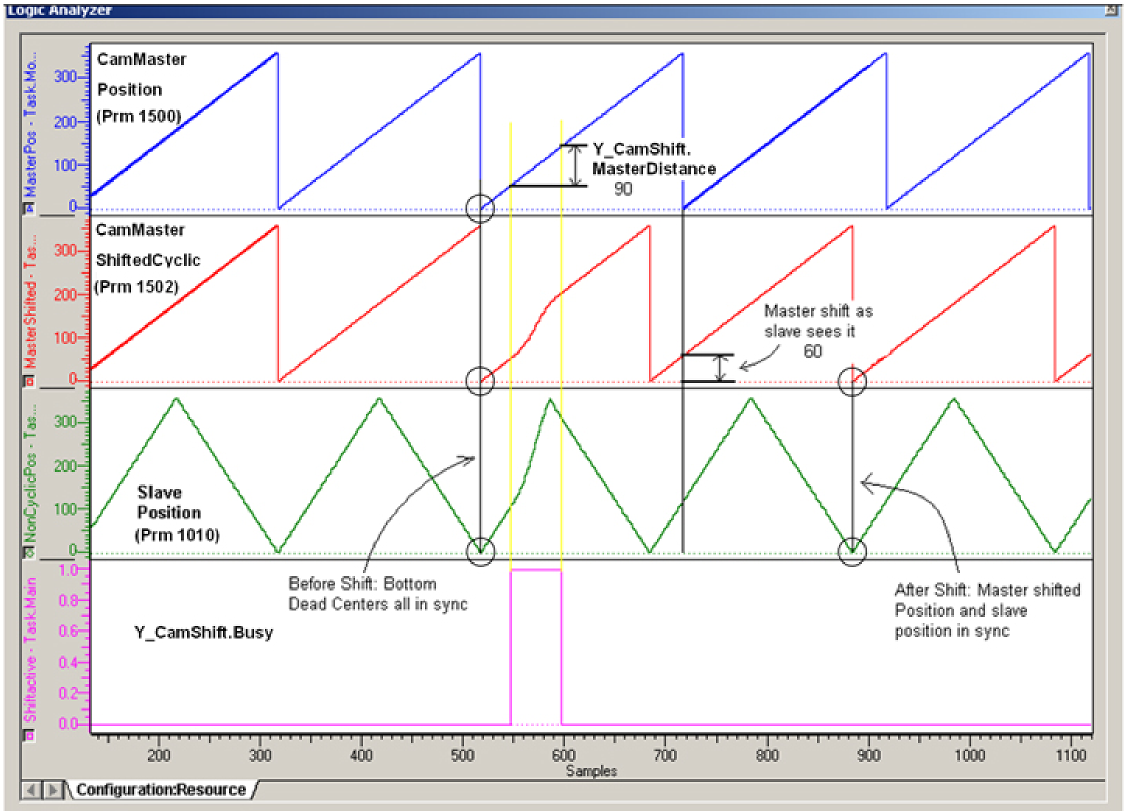
• For more information on how cam shift applies to camming, see the Camming Block Diagram.

Error Description

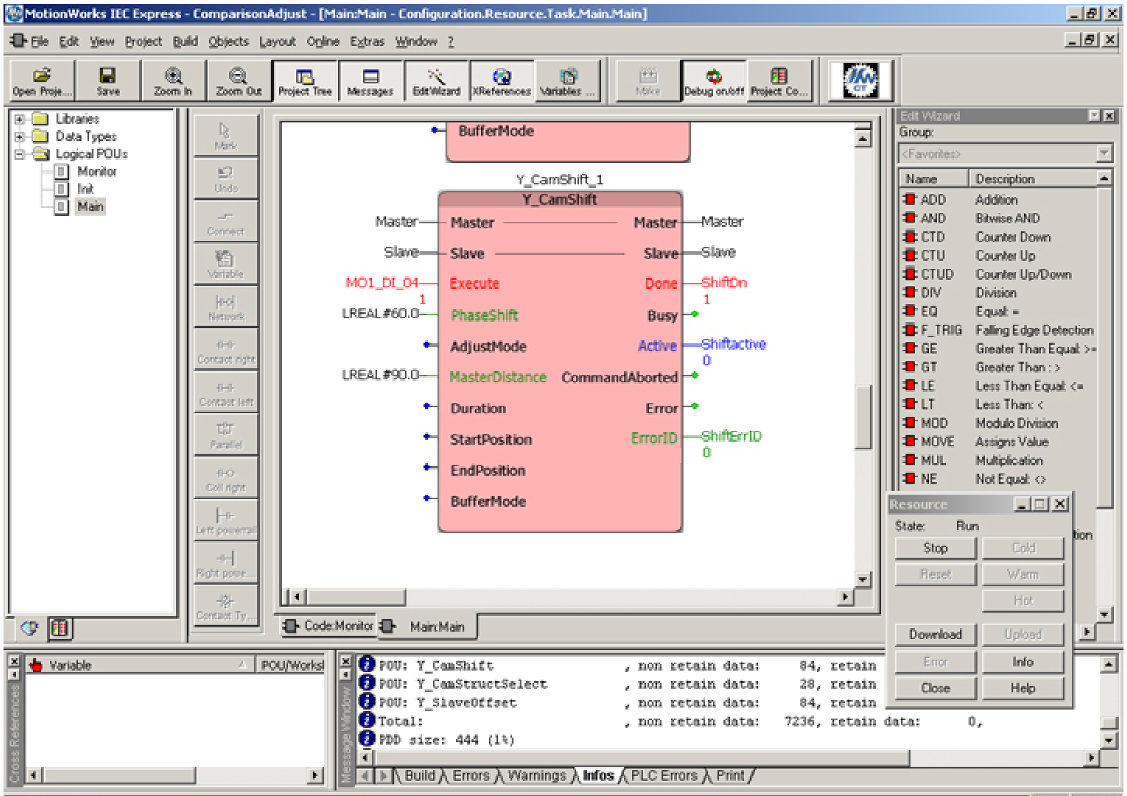
ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis.Also, a motion block may be attempting to abort an MC_TorqueControl move.

4374	Torque move prohibited while non-torque moves queued or in progress.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4398	The cam shift is not possible with EndPosition and current master position.This error occurs if the shift is greater than the distance to the end of the window. For example: shift = 90, window [180,360], and the master position = 300 when Y_CamShift.Execute=TRUE.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4626	The master slave relationship is defined. A slave cannot be a master to another axis.
4633	Table size results in misaligned data. Refer to the help section “Internally Created Cam Data.” A cam table will have a multiple of 16 bytes if created correctly.
4649	Invalid adjust mode
4657	Distance parameter is less than or equal to zero.
4663	Specified time was less than zero.
4673	StartPosition is outside of master's range.
4674	EndPosition is outside of master's range.
57620	The structure size does not match.

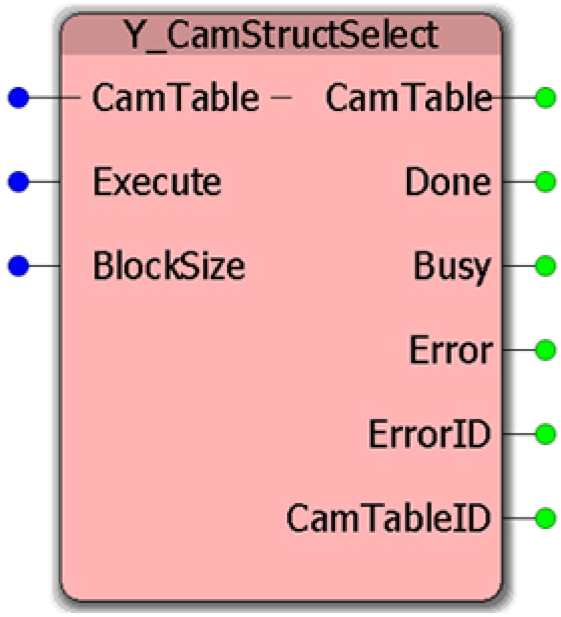
Example



Logic Analyzer
CamMaster
Position
(Prm 1500)
Y_CamShift.
MasterDistance
90
CamMaster
ShiftedCyclic
(Prm 1502)
Master shift as
slave sees it
60
Slave
Position
(Prm 1010)
Before Shift: Bottom
Dead Centers all in sync
After Shift: Master shifted
Position and slave
position in sync
Y_CamShift.Busy
Samples
Configuration:Resource


MotionWorks IEC Express - ComparisonAdjust - [Main:Main - Configuration.Resource.Task.Main.Main]
File Edit View Project Build Objects Layout Online Extras Window ?
Open Proje... Save Zoom In Zoom Out Project Tree Messages Edit Wizard XReferences Variables ... Make Debug on/off Project Co...
Libraries
Data Types
Logical POUs
Monitor
Init
Main
BufferMode
Y_CamShift_1
Y_CamShift
Master Master Master Master
Slave Slave Slave Slave
MO1_DI_04 Execute Done ShiftOn
1 1
LREAL#60.0 PhaseShift Busy
AdjustMode Active Shiftactive
0
LREAL#90.0 MasterDistance CommandAborted
Duration Error
StartPosition ErrorID ShiftErrID
0
EndPosition
BufferMode
Code:Monitor Main:Main
Edit Wizard
Group:
<Favorites>
Name Description
ADD Addition
AND Bitwise AND
CTD Counter Down
CTU Counter Up
CTUD Counter Up/Down
DIV Division
EQ Equal: =
F_TRIG Falling Edge Detection
GE Greater Than Equal: >=
GT Greater Than : >
LE Less Than Equal: <=
LT Less Than: <
MOD Modulo Division
MOVE Assigns Value
MUL Multiplication
NE Not Equal: <>
Resource
State: Run
Stop Cold
Reset Warm
Hot
Download Upload
Error Info
Close Help
Variable POU/Works
POU: Y_CamShift , non retain data: 84, retain
POU: Y_CamStructSelect , non retain data: 28, retain
POU: Y_SlaveOffset , non retain data: 84, retain
Total: , non retain data: 7236, retain data: 0,
PDD size: 444 (1%)
Build Errors Warnings Infos PLC Errors Print

Y_CamStructSelect



This function block loads a cam table from the application memory area to the motion memory area and returns a CamTableID to be referenced when activating the Cam function.

Parameters

Parameter		Data Type	Description	Default
VAR_IN_OUT				
B	CamTable	[Y_MS_CAM_STRUCT]()	Cam data structure	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
V	BlockSize	UDINT	Size of cam data in bytes copied per application task rate (if BlockSize is unconnected, then the full amount).	UINT#0 (Entire CamStruct)

VAR_OUTPUT			
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
B	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.
B	CamTableID	UINT	A reference to the cam memory in the motion engine.

Notes

• Loads a cam file from the application program memory into the motion kernel memory.

• To access cam data that has previously been assigned a CamTableID and resides in the motion kernel memory, use Y_ReadCamTable and Y_WriteCamTable.

• Each application task scan, the function block copies a portion of data from the application program memory to the motion kernel memory. The portion is determined by the BlockSize input. If BlockSize is 0, the entire structure is copied in one scan. If the Cam structure is too large and the scan time too small, a watchdog error may occur.

• Y_MS_CAM_STRUCT is any 'ANY' input, but the motion kernel memory checks that it starts with a valid Y_CAM_HEADER.

• The application programmer can adjust the size of the cam arrays by editing the Data Types worksheet.

• If a CamTableID is no longer needed, the application program should release the cam memory using Y_ReleaseCamTable.

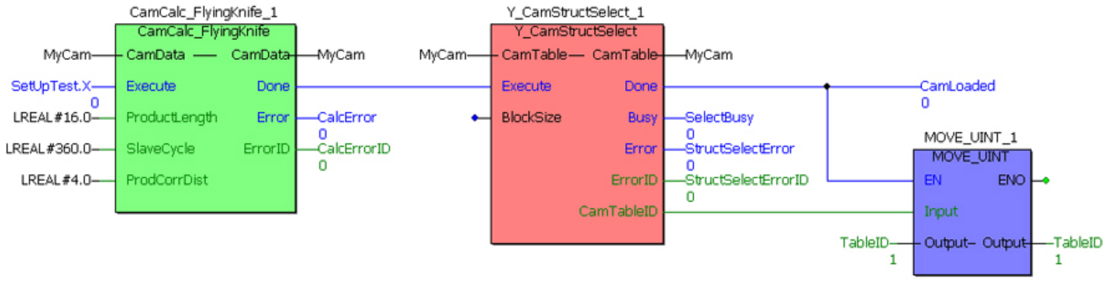
• Refer to Camming Overview for more information regarding cam file creation.

• The behavior of this function block has been modified in Firmware Release Version 1.2.3 to adhere to the PLCopen specification. Prior to that firmware release, the CamTableID was always output even after Execute was low.

Error Description

ErrorID	Meaning
0	No Error
4377	File reading already in progress
4381	Motion aborted due to axis alarm.
4387	Already copying cam data (If Execute transition to TRUE while Busy = TRUE)
4633	Table size results in misaligned data. Refer to the help section “Internally Created Cam Data.” A cam table will have a multiple of 16 bytes if created correctly.
4634	Buffer size results in misaligned data
4635	Table type is not supported

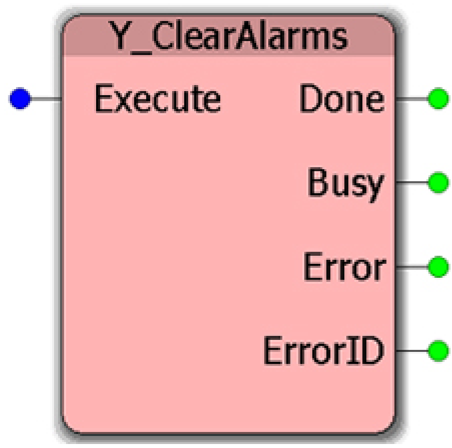
Example




```
Y_CamStructSelect_1
(
    CamTable:=XCamTable,
    Execute:=(Active AND NOT(PathGenerationErrors)) AND (F_TRIG_SelectX.Q OR Y_CamStructSelect_1.Busy OR Y_CamStructSelect_1.Error),
    BlockSize:=UDINT#1024
);
XCamTable:=Y_CamStructSelect_1.CamTable;

F_TRIG_SelectY(CLK:=Y_CamStructSelect_1.Done);
IF F_TRIG_SelectY.Q THEN
    PathID.XAxisTable:=Y_CamStructSelect_1.CamTableID;
    XSelectDone:=TRUE;
END_IF;
```

Note: The MOVE_UINT function block shown is available from the Math Toolbox on www.yaskawa.com/iectb.

Y_ClearAlarms



This Function Block clears controller based alarms that are not axis specific. To clear axis related alarms, use MC_Reset.

Parameters

Parameter		Data Type	Description	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

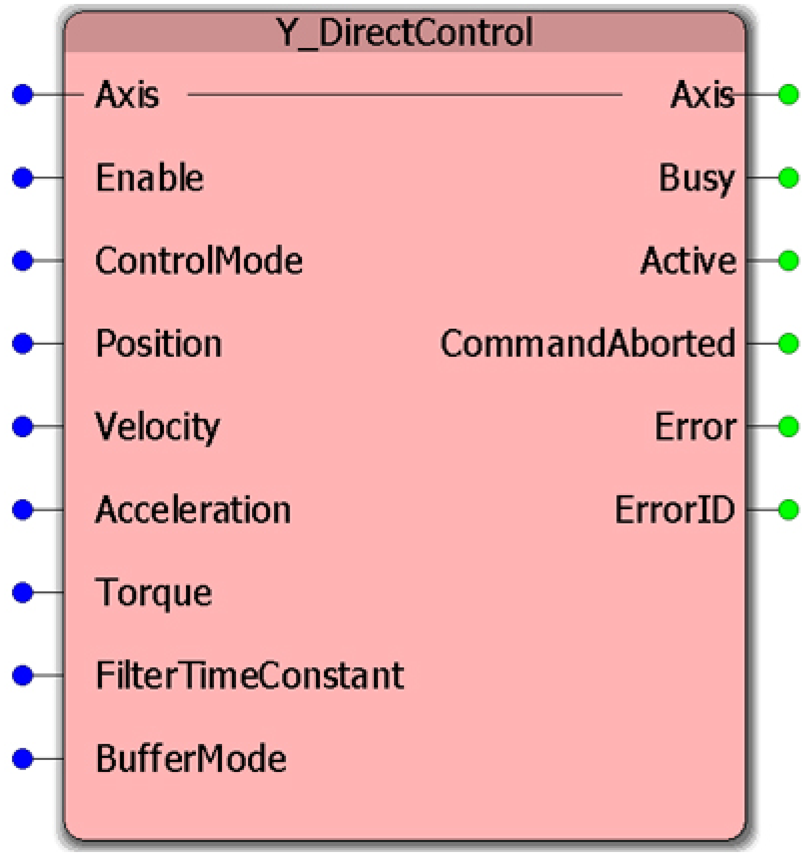
Notes

Refer to the Controller AlarmID List

Error Description

ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.

Y_DirectControl



This block allows direct access to any of three possible control modes available on the specified axis. Y_DirectControl makes it possible to perform open loop velocity control for winding applications. In position mode, the IEC application program can apply an algorithm to directly command the position at every scan.

Parameters

Parameter		Data type	Description
VAR_IN_OUT			
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).

VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
B	ControlMode	INT	ControlMode: 1=position, 2=velocity, 3=torque	INT#0
E	Position	LREAL	A positive or negative value within the coordinate system in user units.	LREAL#0.0
E	Velocity	LREAL	Velocity in user units/second. See notes below.	LREAL#0.0
E	Acceleration	LREAL	Not supported	--
E	Torque	LREAL	Value of the torque (in percentage of rated torque)	LREAL
E	FilterTimeConstant	LREAL	Moving average filter specified in seconds. See below for details.	LREAL#0.0 (No Filter)
V	BufferMode	MC_BufferMode	The behavior of the axis could be Aborting or Buffered MC_BufferMode#Aborting MC_BufferMode#Buffered	MC_BufferMode#Aborting
VAR_OUTPUT				
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- Use the appropriate input with the appropriate mode.

  In Position mode, Velocity input is the velocity feedforward. The Torque input is ignored.

  In Velocity Mode, the Torque input is the torque limit.

  In Torque Mode, the Velocity input is the velocity limit.

- The acceleration input is never used. The user must calculate the required command profile.

- The FilterTimeConstant is specified in seconds. This feature is applicable if the IEC application task rate in which this function is executing is slower than the Motion Engine update. (MECHATROLINK or Dual Port Ram.) If this time constant is set to the PLC scan time (e.g. 0.01 for a 10ms scan), then the filter calculates additional command data (an interpolated value) at each Motion Engine update, reaching the set point just before the next IEC application scan. If set to zero or not connected, the command position is changed to the user specified value only at the end of each IEC application task.

- If it is necessary to switch from Torque to Position mode, the Y_HoldPosition function block is useful in making that transition.

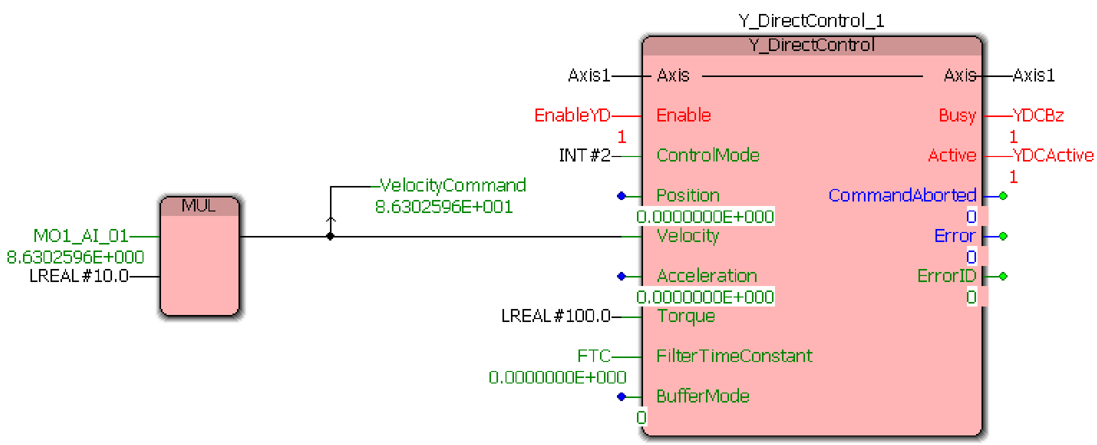
Error Description

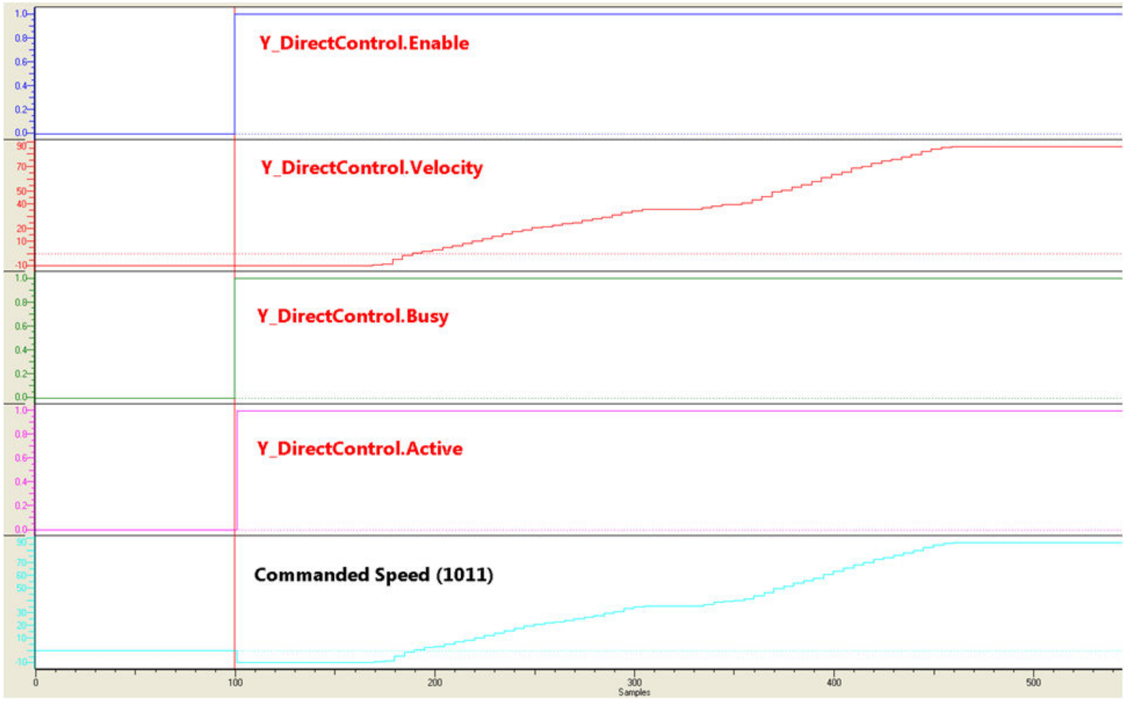
ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can

	override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
57874	Argument data is NULL. The EngageData input must be connected.

Example 1

Y_DirectControl configured for velocity mode.

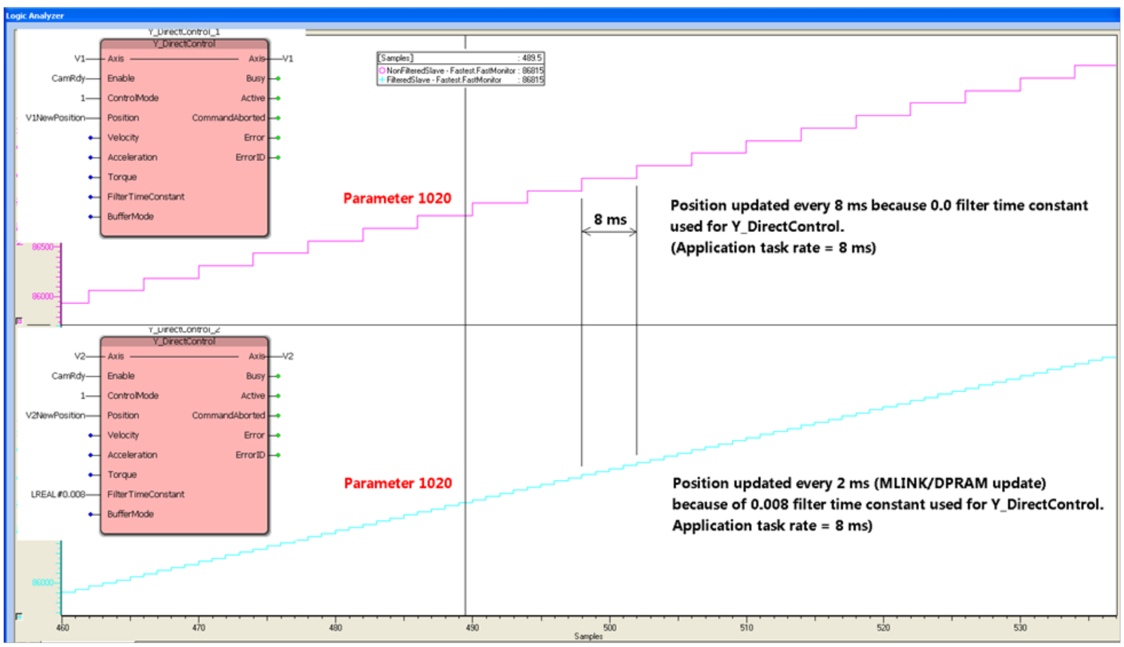



Y_DirectControl.Enable
Y_DirectControl.Velocity
Y_DirectControl.Busy
Y_DirectControl.Active
Commanded Speed (1011)
Samples

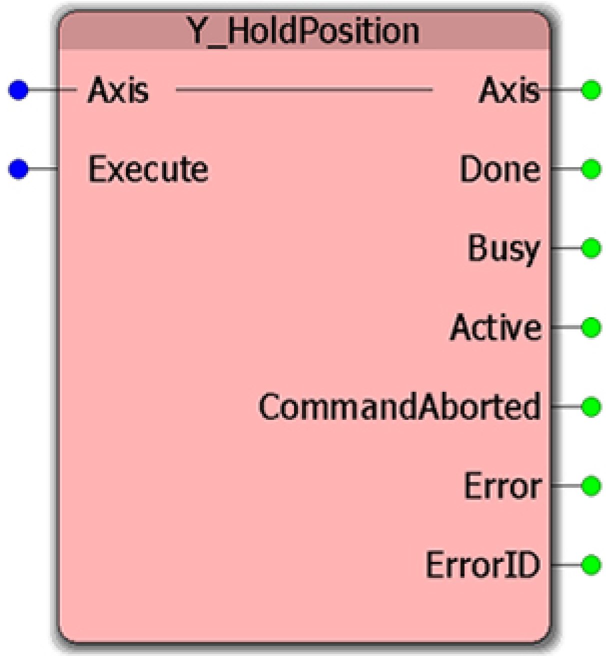
Example 2

The example below shows the effect of the filter time constant on the commanded position. The Y_DirectControl block is in position mode.

The IEC application task is running a 8mSec. The logic analyzer is synchronized with a 2ms task (The MECHATROLINK rate.)



Y_HoldPosition



This Function Block commands an immediate position hold with maximum deceleration and changes the axis state to ‘Stopping’. It aborts any ongoing FB execution. After the axis has held position, the Done output is set to TRUE immediately. As soon as ‘Done’ is SET, the axis goes to state ‘StandStill’.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-	FALSE

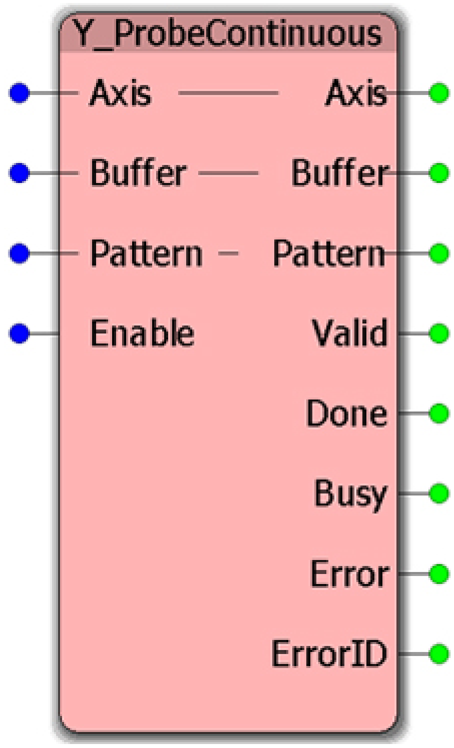
			trigger the execute input.	
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Error Description

ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis. Also, a motion block may be attempting to abort an MC_TorqueControl move.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.

4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4660	Deceleration is less than or equal to zero.
4893	The specified external axis may not be used. A physical axis is required
57620	The structure size does not match.

Y_ProbeContinuous



A “continuous latch mode” is supported by Sigma-5 servo amplifiers. In this mode, the servo amplifier will automatically re-arm the latch function to capture latches that may occur very close together, thus saving the round trip time required to retrieve the latch status and re-arm the latch from the controller. The controller will automatically store the latches into a buffer in the CONTINUOUS_REF data structure connected to the function block. Up to 8 latch events can be defined as a single pattern. The mode can be configured to operate once until a specific pattern has been captured, or infinitely.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
V	Buffer	CONTINUOUS_REF	Structure containing data for configuring and operating continuous latch mode. See example below for a pictorial description of the data.	
V	Pattern	PATTERN_REF	Defines the sequence of inputs that reflect the data to be captured.	
VAR_INPUT				Default
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

Controller firmware version 1.2.4.6 or higher is required to use this function block. This function block is located in the YMotion firmware library.

The physical hardware in the Sigma-5 servo amplifier can only store one latch at a time. However, the controller can store many of them into the CONTINUOUS_REF structure.

The Sigma-5 servo amplifier specification indicates the minimum interval between latches to be 500μs. Any latches that occur during the 500μs re-arming interval will be ignored.

If the PatternSize is greater than 1, only the latches that occur in the exact sequence specified by Pattern will be stored. Any other latches that occur out of sequence will be ignored.

Upon the rising edge of Enable, there will be a short time when the function is busy, but the outputs are not valid yet. This is the time when the amplifier Pns and the Latch mode enable function are sent to the amplifier.

This function block is a hybrid between an 'Execute' and an 'Enable' function block model. If PatternCount=0, latches will continue to be stored until the Enable input goes low. If PatternCount is non zero, then when the PatternCount has been reached, the Busy output goes false and Done becomes true.

At the rising edge of Enable, PatternSize, PatternCount, and Pattern will be checked to be within range and then sent to amplifier Pn's 850, 851, and 852~853 respectively.

Related Function Blocks

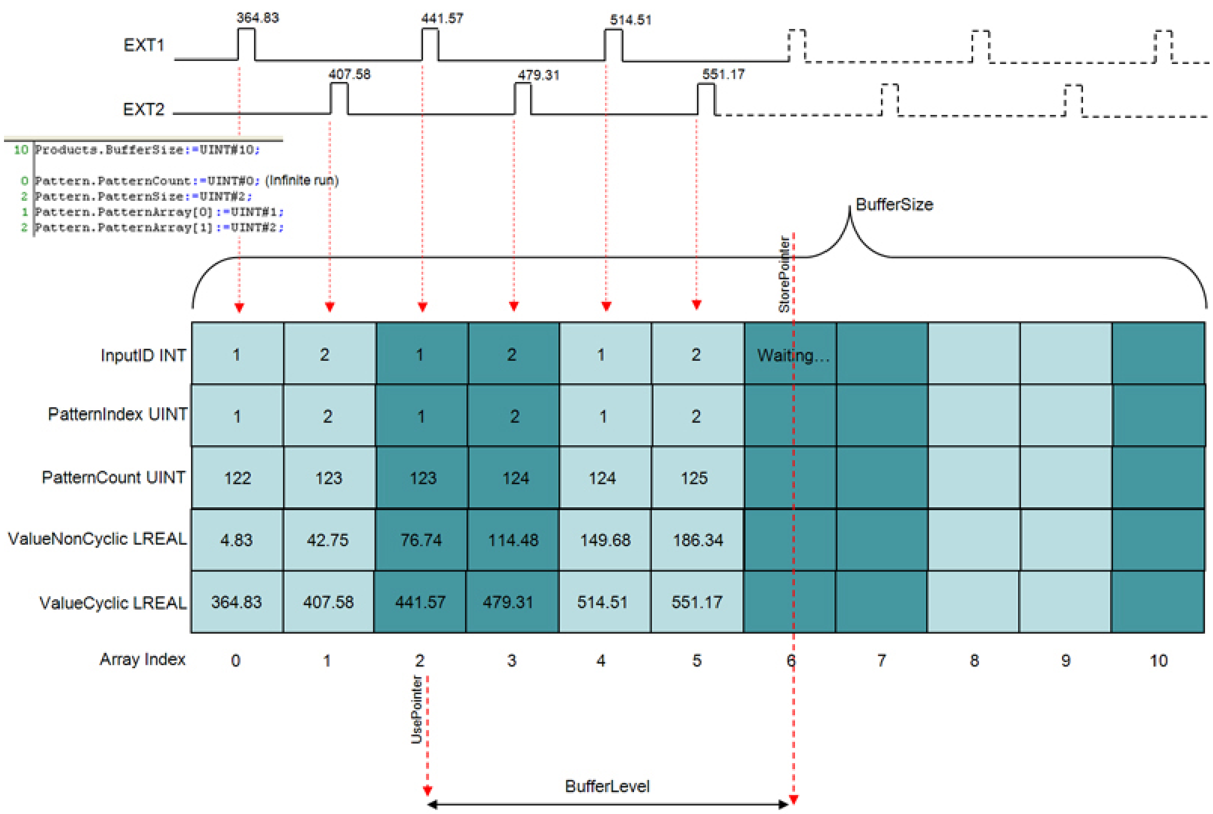
Cam Toolbox - CamSlave_FeedToLength2: For camming applications that index a slave forward in one direction and require on the fly adjustments of the actual index length based on a sensor input.

Error description

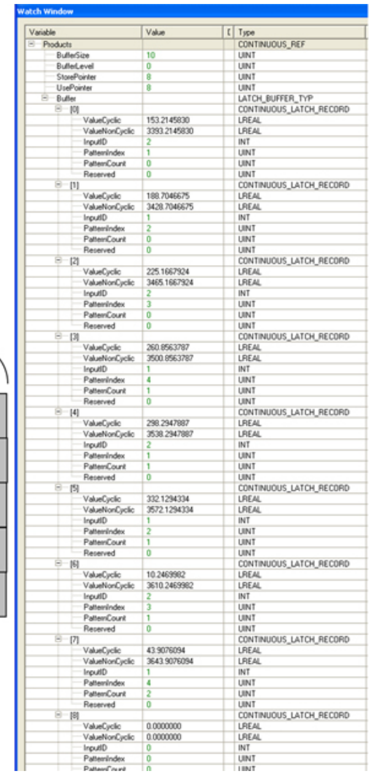
ErrorID	Meaning
0	No Error
4406	Continuous Latch Mode is only supported on Sigma V servopacks.
4407	Continuous latch buffer exhausted
4408	Invalid pattern size or count
4630	Trigger or pattern reference is not valid
4638	User Buffer Full.
4677	Array size is too large
4678	Buffer array index out of range
57620	The structure size does not match.

Examples

Y_ProbeContinuousOperation – Example 1



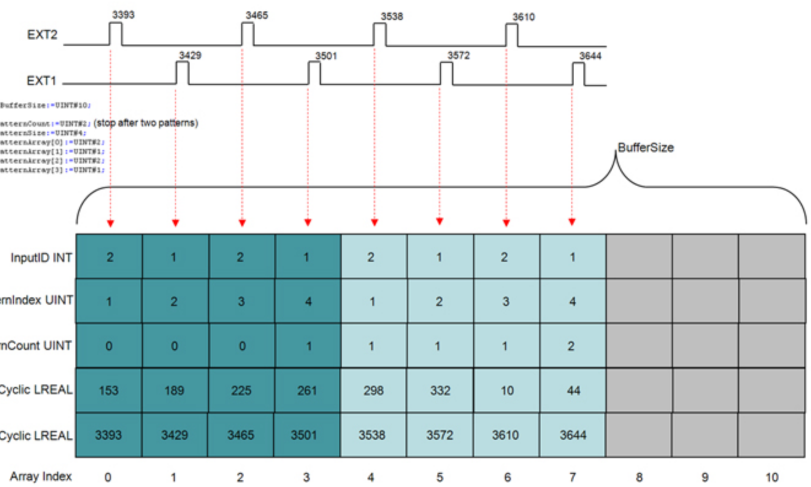
	0	1	2	3	4	5	6	7	8	9	10
InputID INT	1	2	1	2	1	2	Waiting…				
PatternIndex UINT	1	2	1	2	1	2					
PatternCount UINT	122	123	123	124	124	125					
ValueNonCyclic LREAL	4.83	42.75	76.74	114.48	149.68	186.34					
ValueCyclic LREAL	364.83	407.58	441.57	479.31	514.51	551.17					



Watch Window

Variable	Value	[	Type
Products			CONTINUOUS_REF
BufferSize	10		UINT
BufferLevel	0		UINT
StorePointer	8		UINT
UsePointer	8		UINT
Buffer			LATCH_BUFFER_TYP
[0]			CONTINUOUS_LATCH_RECORD
ValueCyclic	153.2145830		LREAL
ValueNonCyclic	3393.2145830		LREAL
InputID	2		INT
PatternIndex	1		UINT
PatternCount	0		UINT
Reserved	0		UINT
[1]			CONTINUOUS_LATCH_RECORD
ValueCyclic	188.7046675		LREAL
ValueNonCyclic	3428.7046675		LREAL
InputID	1		INT
PatternIndex	2		UINT
PatternCount	0		UINT
Reserved	0		UINT
[2]			CONTINUOUS_LATCH_RECORD
ValueCyclic	225.1667924		LREAL
ValueNonCyclic	3465.1667924		LREAL
InputID	2		INT
PatternIndex	3		UINT
PatternCount	0		UINT
Reserved	0		UINT
[3]			CONTINUOUS_LATCH_RECORD
ValueCyclic	260.8563787		LREAL
ValueNonCyclic	3500.8563787		LREAL
InputID	1		INT
PatternIndex	4		UINT
PatternCount	1		UINT
Reserved	0		UINT
[4]			CONTINUOUS_LATCH_RECORD
ValueCyclic	298.2947887		LREAL
ValueNonCyclic	3538.2947887		LREAL
InputID	2		INT
PatternIndex	1		UINT
PatternCount	1		UINT
Reserved	0		UINT
[5]			CONTINUOUS_LATCH_RECORD
ValueCyclic	332.1294334		LREAL
ValueNonCyclic	3572.1294334		LREAL
InputID	1		INT
PatternIndex	2		UINT
PatternCount	1		UINT
Reserved	0		UINT
[6]			CONTINUOUS_LATCH_RECORD
ValueCyclic	10.2469982		LREAL
ValueNonCyclic	3610.2469982		LREAL
InputID	2		INT
PatternIndex	3		UINT
PatternCount	1		UINT
Reserved	0		UINT
[7]			CONTINUOUS_LATCH_RECORD
ValueCyclic	43.9076094		LREAL
ValueNonCyclic	3643.9076094		LREAL
InputID	1		INT
PatternIndex	4		UINT
PatternCount	2		UINT
Reserved	0		UINT
[8]			CONTINUOUS_LATCH_RECORD
ValueCyclic	0.0000000		LREAL
ValueNonCyclic	0.0000000		LREAL
InputID	0		INT
PatternIndex	0		UINT
PatternCount	0		UINT
[illegible]			

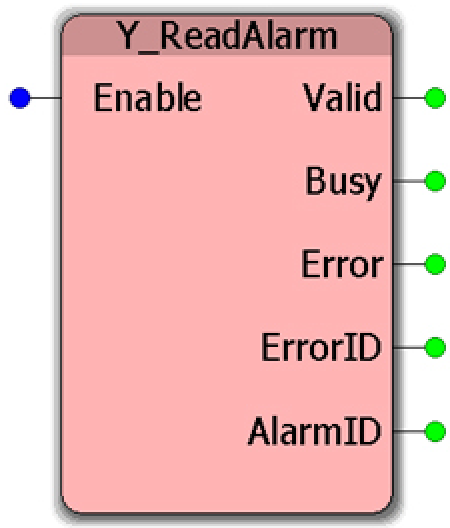
Y_ProbeContinuousOperation – Example 2




```
10 Products.BufferSize:=UINT#10;

2 Pattern.PatternCount:=UINT#2; (stop after two patterns)
4 Pattern.PatternSize:=UINT#4;
2 Pattern.PatternArray[0]:=UINT#2;
1 Pattern.PatternArray[1]:=UINT#1;
2 Pattern.PatternArray[2]:=UINT#2;
1 Pattern.PatternArray[3]:=UINT#1;
```

InputID INT	2	1	2	1	2	1	2	1			
PatternIndex UINT	1	2	3	4	1	2	3	4			
PatternCount UINT	0	0	0	1	1	1	1	2			
ValueNonCyclic LREAL	153	189	225	261	298	332	10	44			
ValueCyclic LREAL	3393	3429	3465	3501	3538	3572	3610	3644			
Array Index	0	1	2	3	4	5	6	7	8	9	10

Y_ReadAlarm



This Function Block reports controller-specific alarms that are not axis related. The Function Block Y_ClearAlarms clears alarms reported by this block.

Parameters

Parameter		Data Type	Description	Default
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
V	AlarmID	UDINT	This output provides the Controller Alarm ID. This output is reset when execute goes low.	

Notes

Refer to the Controller AlarmID List for a comprehensive list of alarm codes. Axis specific alarms are reported by MC_ReadAxisError.

Related Function Blocks

MC_Power: Enables/Disables an axis.

MC_ReadAxisError: Reports axis warnings or alarms.

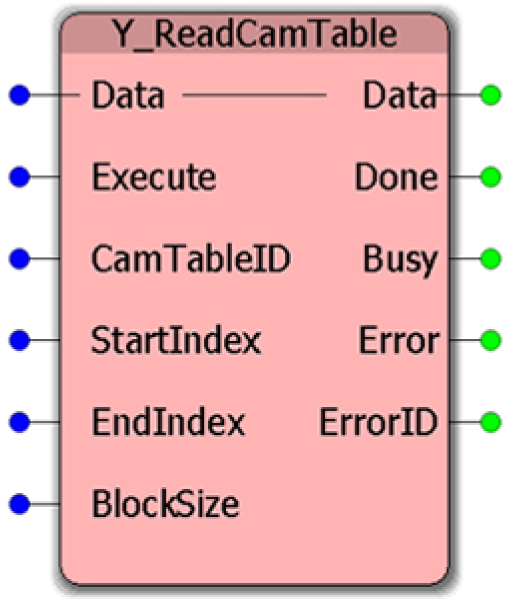
MC_Reset: Clears axis specific alarms.

PLCopen Toolbox - AxisControl: Combines MC_Power, MC_ReadAxisError, MC_Reset, and Y_ReadAlarm to Enable/disable an axis along with providing axis warnings and alarms as outputs.

Error Description

ErrorID	Meaning
0	No Error
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.

Y_ReadCamTable



This Function Block copies a cam table from the motion memory into the application program memory.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Data	Y_MS_CAM_STRUCT	Cam data structure	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	CamTableID	UINT	A reference to the cam memory in the motion engine.	UINT#0
V	StartIndex	UDINT	Index into cam table in bytes (as used with Y_Cam_Struct)	UDINT#0
V	EndIndex	UDINT	Index into cam table in bytes (as used with Y_Cam_Struct). 0 is interpreted as the maximum index.	UDINT#0

V	BlockSize	UDINT	Size of cam data in bytes copied per application task rate (if BlockSize is unconnected, then the full amount).	UDINT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low..	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• This function block requires that a cam file was previously loaded with Y_CamFileSelect or Y_CamStructSelect.

• You must first populate the TableType and DataSize before the function will execute without error. Remember that the Y_MS_CAM_STRUCT's DataSize element is in bytes, so multiply by 16 to account for the number of pairs expected (each LREAL is 8 bytes). If left at 0, the function will result with ErrorID 4885.

• When reading the cam table, this function block shall not exceed the EndIndex, the cam table size, or the number of elements in Data.

• If EndIndex=0, then it defaults to the cam table size.

• Each scan, the function block copies a portion of data from the motion memory area to the application program memory. The BlockSize input specifies the number of data pairs to transfer per scan. If BlockSize is 0, then the entire table is copied in one PLC scan. If the table is large and the task time is small, a watchdog error may result.

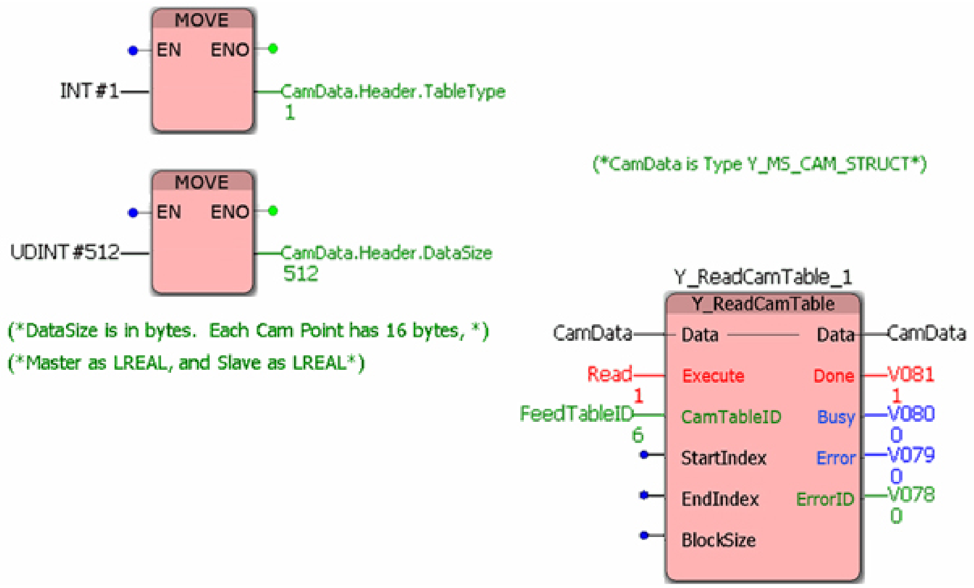
• Y_MS_CAM_STRUCT is any 'ANY' input, but the motion kernel memory checks that it starts with a valid Y_CAM_HEADER.

• Refer to the Internally Created Cam Data diagram in the Cam Data Management section.

Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4387	Already copying cam data (If Execute transition to TRUE while Busy = TRUE)
4633	Table size results in misaligned data. Refer to the help section “Internally Created Cam Data.” A cam table will have a multiple of 16 bytes if created correctly.
4635	Table type is not supported
4636	Invalid start index.
4637	Invalid end index
4885	Invalid header for the cam file. You must first populate the TableType and DataSize in the Y_MS_CAM_STRUCT before executing the function.
4887	CamTableID does not refer to a valid cam table.

Example

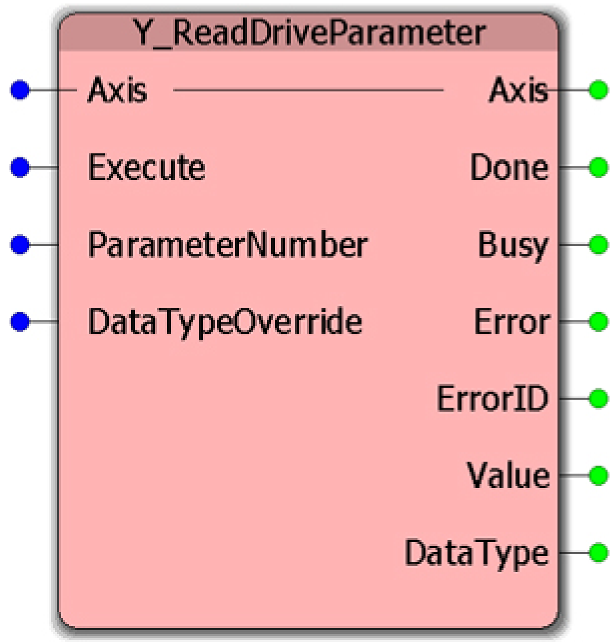


Watch Window

Variable	Value	Type	Instance
CamData		Y_MS_CAM_STRUCT	Configuration.Resource.Task.Main.CamData
Header		Y_CAM_HEADER	Configuration.Resource.Task.Main.CamData.Header
TableType	1	INT	Configuration.Resource.Task.Main.CamData.Header.TableType
Reserved1	0	UINT	Configuration.Resource.Task.Main.CamData.Header.Reserved1
DataSize	512	UDINT	Configuration.Resource.Task.Main.CamData.Header.DataSize
MS_Header		Y_MS_HEADER	Configuration.Resource.Task.Main.CamData.MS_Header
SlaveIncremental	0	USINT	Configuration.Resource.Task.Main.CamData.MS_Header.SlaveIncremental
MasterIncremental	0	USINT	Configuration.Resource.Task.Main.CamData.MS_Header.MasterIncremental
Reserved1	0	UINT	Configuration.Resource.Task.Main.CamData.MS_Header.Reserved1
Reserved2	0	UINT	Configuration.Resource.Task.Main.CamData.MS_Header.Reserved2
Reserved3	0	UINT	Configuration.Resource.Task.Main.CamData.MS_Header.Reserved3
MS_Data		MS_Array_Type	Configuration.Resource.Task.Main.CamData.MS_Data
[0]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[0]
Master	0.0000000E+000	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[0].Master
Slave	0.0000000E+000	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[0].Slave
[1]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[1]
Master	3.6000000E+002	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[1].Master
Slave	3.4602080E-001	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[1].Slave
[2]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[2]
Master	7.2000000E+002	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[2].Master
Slave	1.3840830E+000	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[2].Slave
[3]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[3]
Master	1.0800000E+003	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[3].Master
Slave	3.1141870E+000	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[3].Slave
[4]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[4]
Master	1.4400000E+003	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[4].Master
Slave	5.5363320E+000	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[4].Slave
[5]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[5]
Master	1.8000000E+003	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[5].Master
Slave	8.6505190E+000	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[5].Slave
[6]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[6]
Master	2.1600000E+003	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[6].Master
Slave	1.2456750E+001	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[6].Slave
[7]		Y_MS_PAIR	Configuration.Resource.Task.Main.CamData.MS_Data[7]
Master	2.5200000E+003	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[7].Master
Slave	1.6955020E+001	LREAL	Configuration.Resource.Task.Main.CamData.MS_Data[7].Slave

Watch 1 | Watch 2 | Watch 3 | Watch 4

Y_ReadDriveParameter



This Function Block reads the specified parameter from the drive or amplifier of the specified axis. To read a controller side axis parameter, use [MC_ReadParameter].

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	[AXIS_REF]	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
V	ParameterNumber	UINT	Number of the Parameter in the drive. Note that the parameter numbers for the Sigma	UINT#0

			amplifiers are displayed in hex in all documentation. For consistency, the ParameterNumber can be entered in hex as shown in the example below.	
V	DataTypeOverride	INT	Enumeration with the following values: 0 = default (i.e., fetched from the parameter XML file.); 1 = UINT; 2 = UDINT; 3 = INT; 4 = DINT.	INT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
V	Value	DINT	The drive parameter value	
V	DataType	INT	Enumeration with the following values: 0 = default (i.e., fetched from the parameter XML file.); 1 = UINT; 2 = UDINT; 3 = INT; 4 = DINT.	

Notes

Use the links below to access the ServoPack manuals on www.yaskawa.com to view the list of parameters.

- Sigma II with NS115: SIEPC71080001, see Appendix B 1
- Sigma III: YEA-SIA-S800-11, see section 11.2.2
- Sigma-5 with rotary motor: SIEPS8000046, see Section 10.1.2
- Sigma-5 with linear motor: SIEPS8000048, see Section 9.1.2

In most cases, the drive parameters are 16 bit values and the DataType override is not necessary.

The parameter size (2 or 4 bytes) and sign is fetched from the default parameter XML files if DataTypeOverride is not connected.

- If the parameter is not found in this file, the a "NoDefaultParameterInfo" error will occur

- For all unsigned 32 bit parameters, the user is responsible for converting Value to UDINT using the DINT_TO_UDINT function since the value might be greater than 268435455. To assist the user in determining if this is need, the data type is specified as an output.

- DataType is an enumeration with the following values:

  1 = UINT

  2 = UDINT

  3 = INT

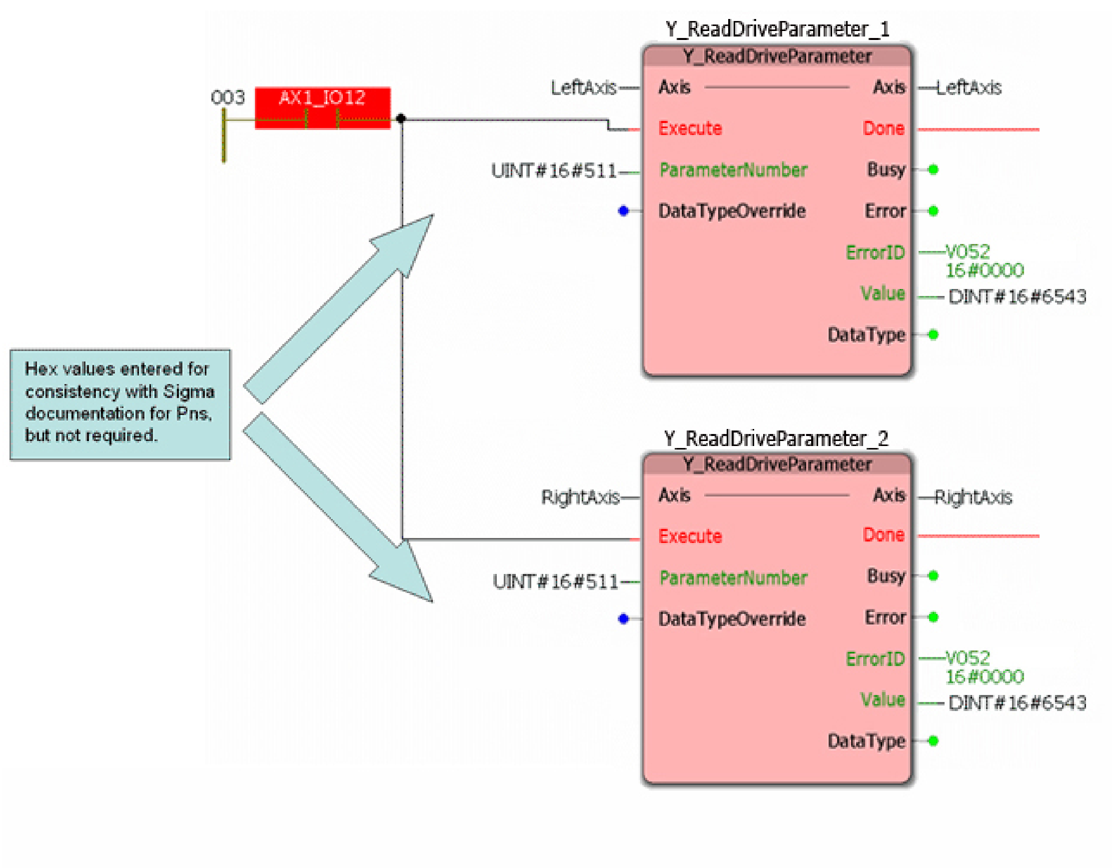
  4 = DINT

Error description

ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4401	The controller cannot communicate with the axis. It may be disconnected from the network.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical

	axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4892	Default drive parameter info is not available for this parameter.
57620	The structure size does not match.

Example



Y_ReadMultipleParameters

This function will read a number of controller parameters at once. The parameters must be an LREAL type. Populate the parameter numbers into the ParamStruct, and the function block will supply the values to the requested parameters.

Refer to parameters with LREAL Data Type in the Axis Parameter List.

Parameters

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
V	ParamStruct	PrmStruct	Structure containing a list of parameter numbers to be read and their corresponding values	
VAR_INPUT				**Default**
B	Enable	BOOL	The function will continue to execute while enable is held high.	FALSE
VAR_OUTPUT				
B	Valid	BOOL	Indicates that the outputs of the function are valid.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
V	ErrorPrm	UINT	If there was an error while attempting to read one of the parameters listed in the ParamStruct, this output will contain the offending parameter number.	

Notes

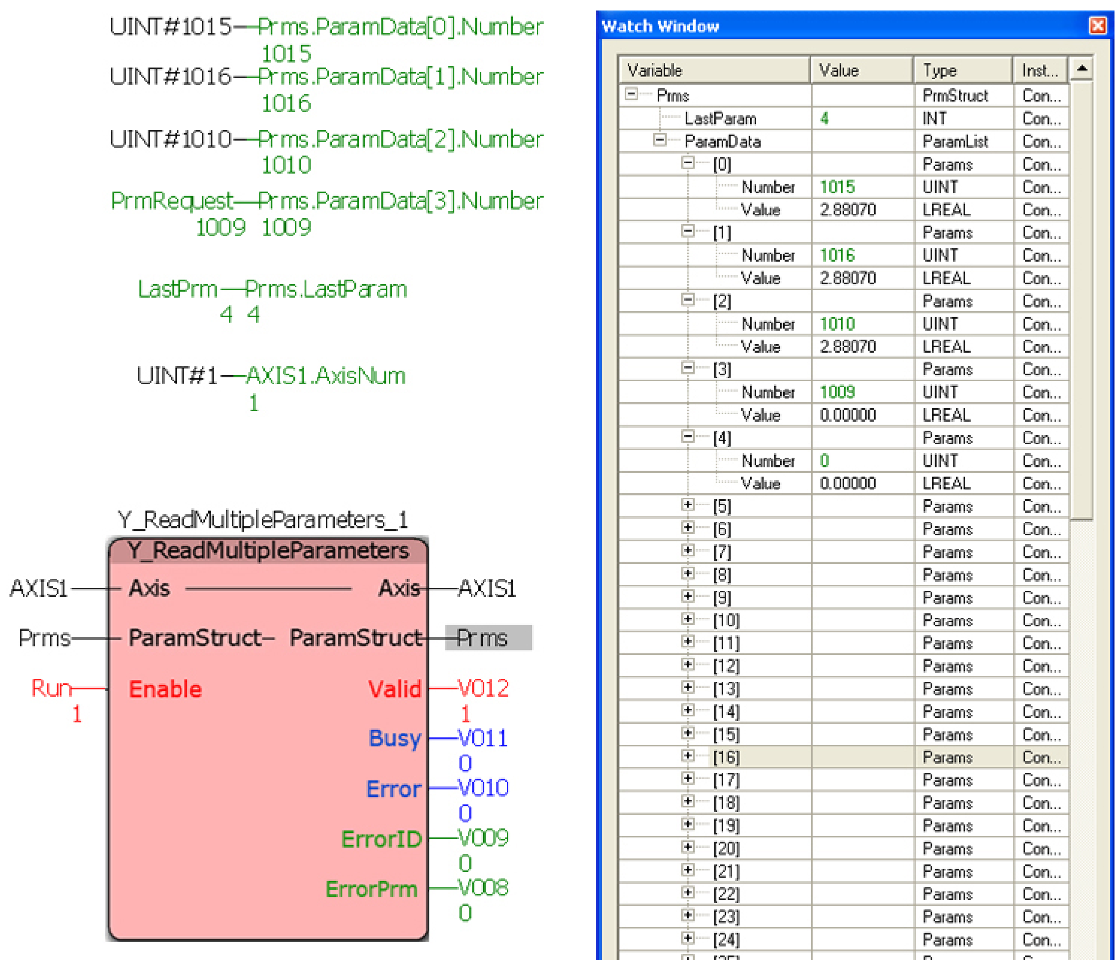
- Refer to parameters with LREAL Data Type in the Axis Parameter List. Only LREAL type parameters can be read with Y_ReadMultipleParameters.
- Firmware version 2.0.0 and it's YMotion firmware library is required to use this function block.
- PrmStruct.LastPrm is the quantity of parameters to be read, which will be one less than the last array index value, because the array is zero based.
- BOOL parameters cannot be read with this function block. Use MC_ReadBoolParameter.

Error Description

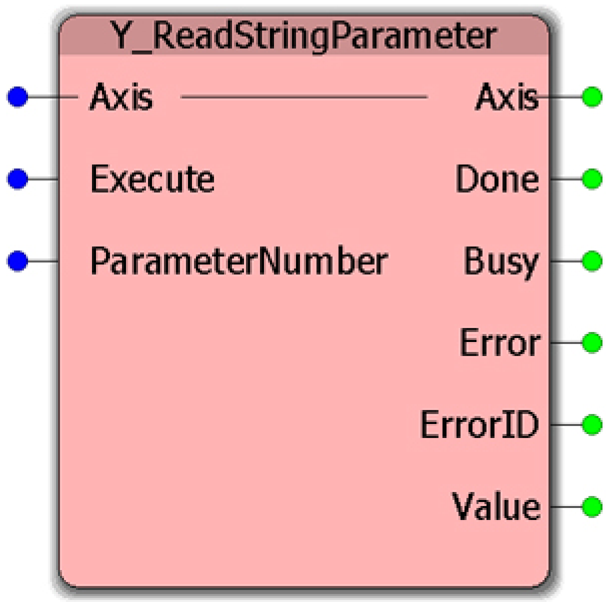
ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4402	The scan compensation delay parameter 1305 is only valid for external encoders.
4403	The High Speed Output functionality is only available on external encoders.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis

[4676]()	The time value must be within 0 to 10 MECHATROLINK cycles.
[57617]()	Instance object is NULL.
[57620]()	The structure size does not match.

Example



Y_ReadStringParameter



This Function Block returns the string value of an axis-specific parameter.

Refer to parameters with STRING Data Type in the Axis Parameter List. The two parameters available are AmplifierModel (1819) and MotorModel (1823).

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input,	FALSE

			change the value and re-trigger the execute input.	
V	ParameterNumber	UINT	Controller parameter number. Refer to parameters with STRING Data Type in the Axis Parameter List.	UINT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
V	Value	STRING	The drive parameter value	

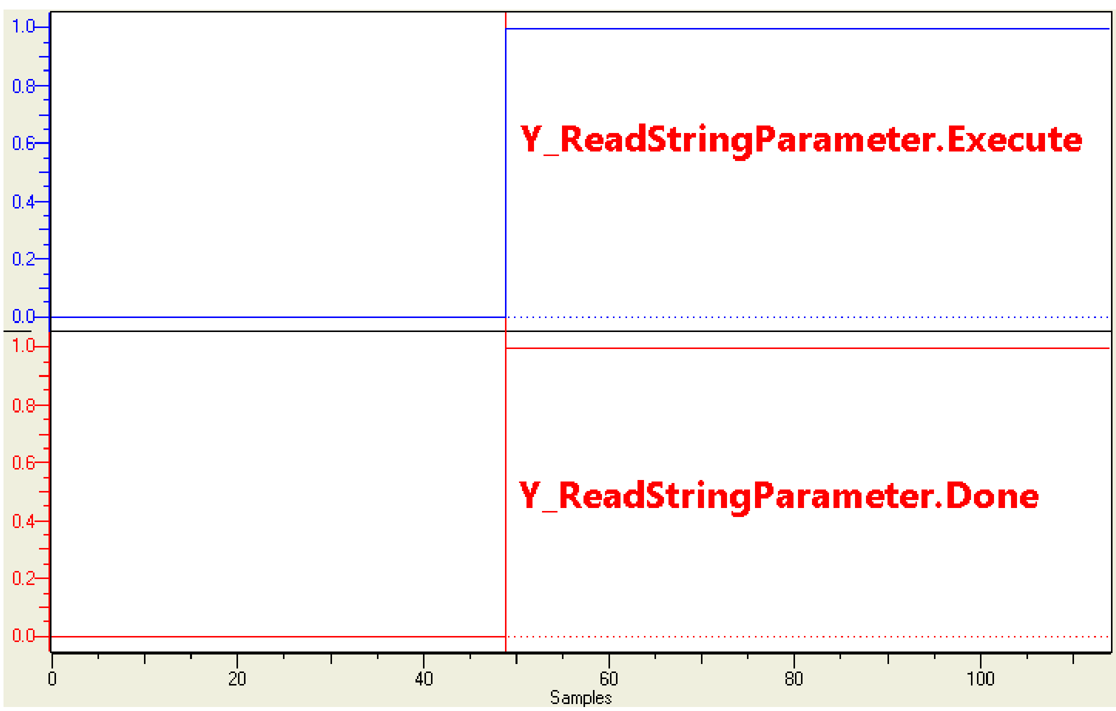
Error description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4402	The scan compensation delay parameter 1305 is only valid for external encoders.
4403	The High Speed Output functionality is only available on external encoders.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis

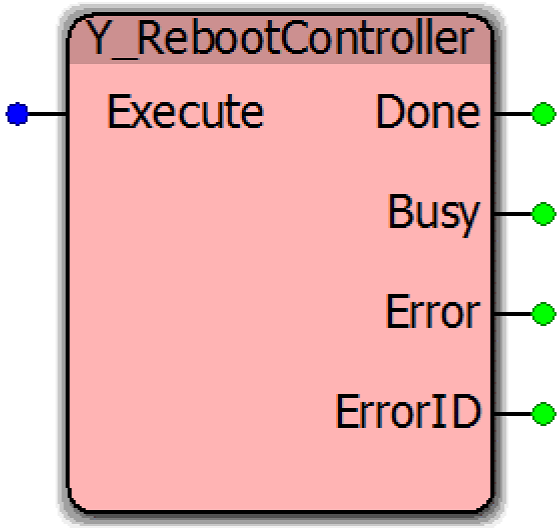
	number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis
4676	The time value must be within 0 to 10 MECHATROLINK cycles.
57617	Instance object is NULL.
57620	The structure size does not match.

Example

Using Y_ReadStringParameter to read the servopack model



Y_RebootController



This function block reboots all controller processes, and reinitializes the MECHATROLINK network. Nodes are temporarily disconnected from network and rediscovered afterward. This function block was introduced in firmware version 2.5 for use with the MP2600iec controller and will execute on all other controller models. The application is similar to Y_ResetMechatrolink, which is not supported on the MP2600iec. Resetting an option style amplifier causes its option card to reboot.

Parameters

Parameter		Data Type	Description	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	

E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

• This function block is useful for resetting an alarm condition which requires power cycle, such as an absolute encoder reset, in particular, on the MP2600iec.

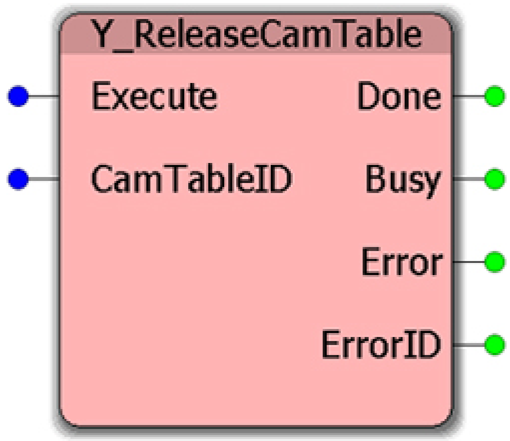
• Executing this function will start a complete system reboot, meaning that ethernet communications will also be restarted.

• This function requires firmware 2.5.0. The function block is included in the Y_Motion firmware library of MotionWorks IEC v2.5.0.

Error Description

ErrorID	Meaning
0	No Error
4415	Reboot already in progress
45337	Rebooting the controller is prohibited while an axis is enabled.

Y_ReleaseCamTable



This Function Block frees memory in the motion area currently allocated for a cam table.

Parameters

Parameter		Data Type	Description	Default
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	CamTableID	UINT	A reference to the cam memory in the motion engine.	UINT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset	

			when 'Execute' or 'Enable' goes low.

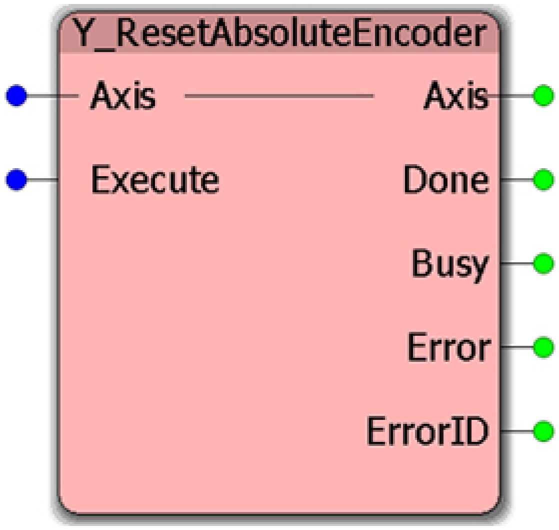
Notes

- After this function block is Done, the CamTableID is no longer valid.
- If the cam table is in use when this block executes, cam table memory is freed when camming completes and no error is generated.

Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4887	CamTableID does not refer to a valid cam table.

Y_ResetAbsoluteEncoder



This Function Block clears absolute encoder alarms caused by battery power loss, cable disconnection, etc. This function block is equivalent to the Fn008 servo amplifier function, which can be performed from the front panel of an SGDH amplifier or via SigmaWin, or via the MPiec controllers webserver page.

WARNING: After performing this function, the motor position will be cleared and must be re-established (see MC_SetPosition) to avoid mechanical damage to the machine.

With firmware 1.2.3 and higher, an A.CC0 alarm (multiturn disagreement alarm) can also be cleared.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
B	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

After successfully resetting the absolute encoder, servo power must be cycled. A soft reboot be accomplished by using the Y_ResetMechatrolink function block.

Perform the setup operation for the absolute encoder in the following circumstances:

- When starting the machine for the first time.
- When an encoder backup error (A.810) occurs.
- When an encoder checksum error (A.820) occurs.
- When the multi-turn data of absolute encoder is to be set to zero.

Please refer to the following manuals for more details regarding absolute encoder reset:

- Sigma II : YEA-SIA-S800-32.2, see section 5.7.4
- Sigma III: YEA-SIA-S800-11, see section 7.7.2
- Sigma-5 with rotary motor: SIEPS8000046, see Section 4.6.4

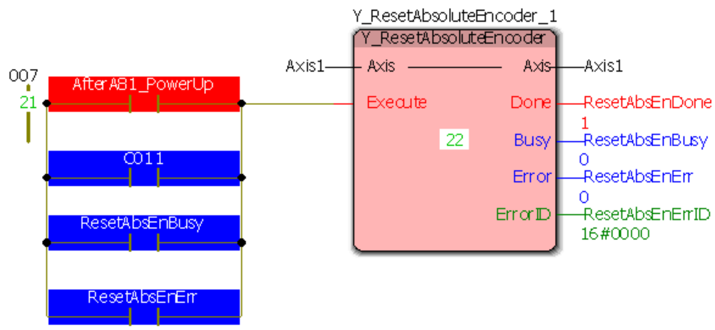
Error description

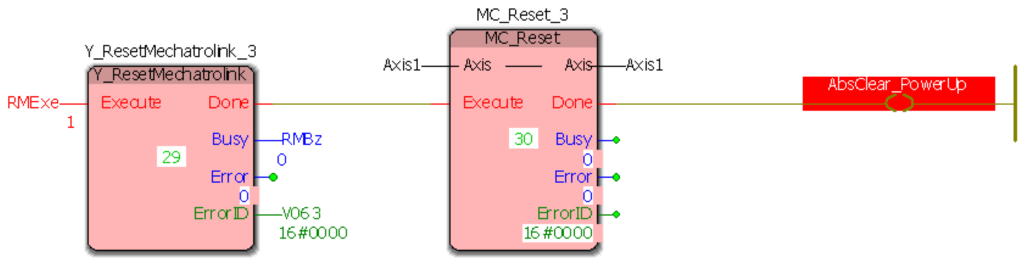
ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4401	The controller cannot communicate with the axis. It may be disconnected from the network.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
45335	Failed to initialize absolute encoder.
57620	The structure size does not match.
61713	An internal motion kernel command failed. This error could be caused by inserting multiple MC_Power function blocks in the program for the same axis. Only one MC_Power function per axis is required. (Do not include more than one.)

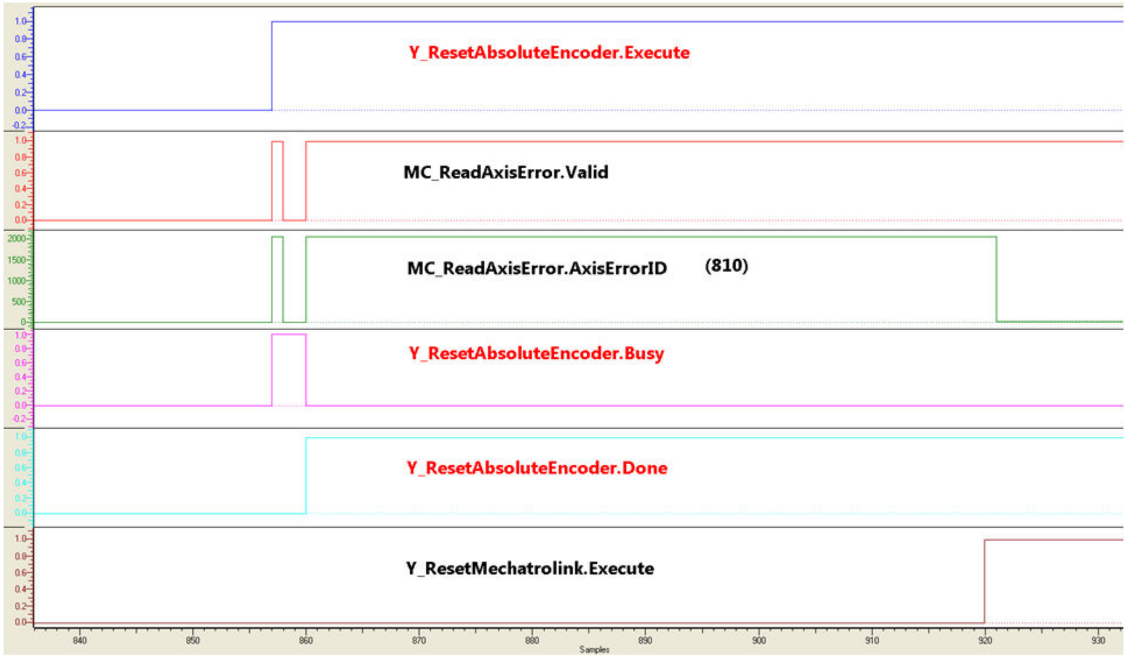
Example

Y_ResetAbsoluteEncoder is used to reset the A.810 alarm.

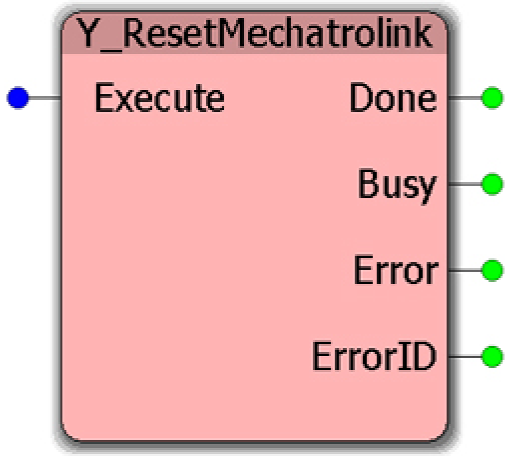
Y_ResetMechatrolink can be used on MECHATROLINK based servopacks to cycle power to the servopacks (required after an A.810 alarm).






Y_ResetAbsoluteEncoder.Execute
MC_ReadAxisError.Valid
MC_ReadAxisError.AxisErrorID
(810)
Y_ResetAbsoluteEncoder.Busy
Y_ResetAbsoluteEncoder.Done
Y_ResetMechatrolink.Execute

Y_ResetMechatrolink



This function block resets the MECHATROLINK network. Nodes are temporarily disconnected from network and rediscovered afterward. This function is identical to the web interface's "Reset ServoNet" button.

Parameters

Parameter		Data Type	Description	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• Check the Y_ReadAlarm function block to determine if any alarms related to MECHATROLINK exist.

• Executing Y_ResetMechatrolink will cause Sigma-5 ServoPacks to do a soft reboot, which will clear any alarms that would otherwise require power cycle. Use MC_Reset to clear the 000F alarm (temporary network disconnect alarm) after Y_resetMechatrolink is done executing.

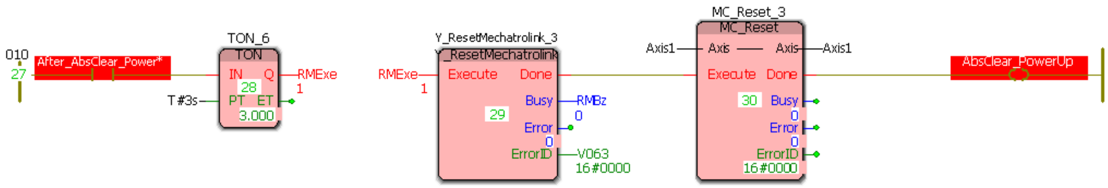
• Prior to firmware 2.0.0, Y_ResetMechatrolink cleared controller axis parameters 1310 and 1311 back to default as a side effect. If either of these features (S-Curve filter or Mechatrolink sub interpolation filter) were enabled by the application, re-write them after Y_ResetMechatrolink.

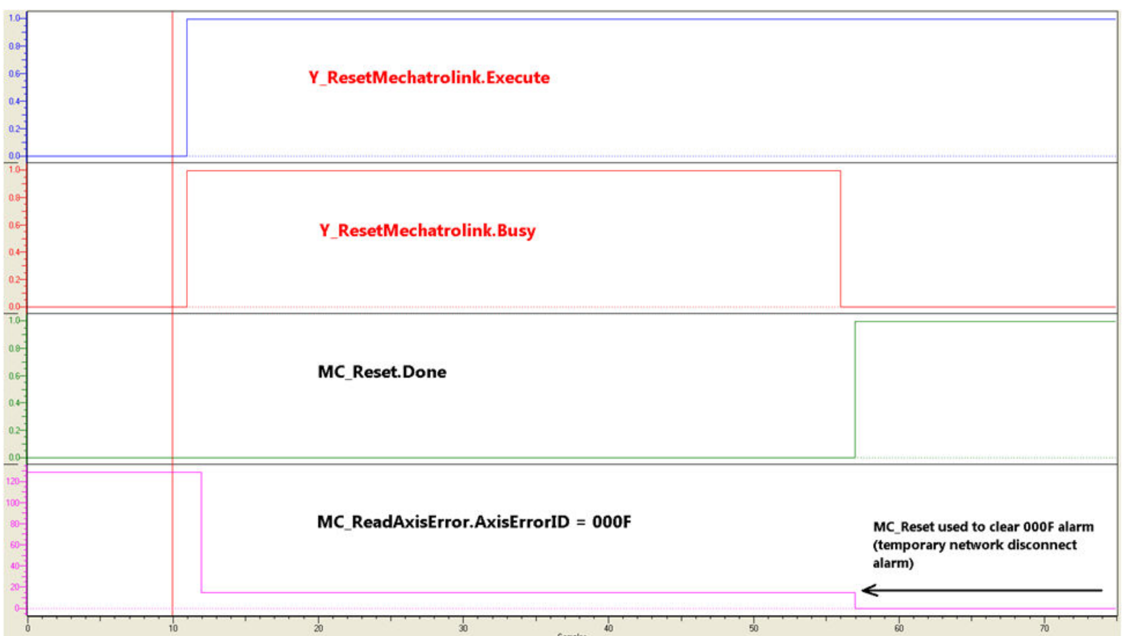
Error Description

ErrorID	Meaning
0	No Error
4386	MECHATROLINK reset is already in progress.
45334	Function cannot be utilized if there is a servo enabled or in motion on the network.

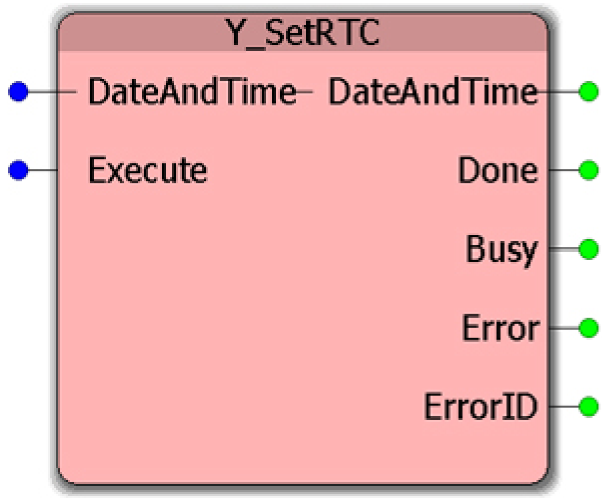
Example

In the example shown below, MC_Reset is used to clear the 000F alarm (temporary network disconnect alarm) after Y_ResetMechatrolink has completed executing.





Y_SetRTC



This Function Block allows the application program to set the controller's real time clock.

Parameters

	Parameter	Data type	Description	Default
VAR_IN_OUT				
B	DateAndTime	[RTC_Struct]()	Date and time	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	

E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

The real time clock can be read as a string using the RTC_S function block from the ProConOS firmware library.

Refer to the Yaskawa Toolbox for a function that provides the real time clock as an RTC_Struct data type

Error Description

ErrorID	Meaning
0	No Error
4679	Invalid date or time values entered.

Y_SlaveOffset

This Function Block applies an offset to the slave position. For use with cam mode.

Parameters

	Parameter	Data type	Description	
VAR_IN_OUT				
B	Master	AXIS_REF	A logical reference to the master axis	
B	Slave	AXIS_REF	A logical reference to the slave axis	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and	BOOL

			the function is initiated. To modify an input, change the value and re-trigger the execute input.	
V	Offset	LREAL	Absolute offset to be applied to the cam profile. Units are those of the slave.	LREAL#0.0
V	AdjustMode	Y_AdjustMode	AdjustMode is an integer with the following values: Y_AdjustMode#MasterDistance: the scale starts immediately and completes when the master has travelled the specified distance. Y_AdjustMode#ElapsedTime: the scale starts immediately and completes within the specified time. Y_AdjustMode#WithinRange: the scale starts when the master is crosses the StartPosition and completes when the master reaches the EndPosition.	Y_AdjustMode#MasterDistance
V	MasterDistance	LREAL	Only used if AdjustMode = Y_AdjustMode#MasterDistance. This is the change in master position from when the function block first executes until the adjustment is complete. Units are those of the cam master.	LREAL#0.0
V	Duration	LREAL	Only used if AdjustMode = Y_AdjustMode#ElapsedTime. Units are seconds.	LREAL#0.0
V	StartPosition	LREAL	Only used if AdjustMode = Y_AdjustMode#WithinRange. The initial position of the master where it is possible to start making the adjustment. Units are those of the cam master.	LREAL#0.0
V	EndPosition	LREAL	Only used if AdjustMode = Y_AdjustMode#WithinRange. The final position of the master where the adjustment must be completed. Units are those of the cam master.	LREAL#0.0
B	BufferMode	MC_BufferMode	Defines the behavior of the axis -	MC_BufferMode#Aborting

			allowable modes are Aborting, Buffered, BlendingLow, BlendingPrevious, BlendingNext, and BlendingHigh.- MC_BufferMode#Aborting MC_BufferMode#Buffered MC_BufferMode#BlendingLow MC_BufferMode#BlendingPrevious MC_BufferMode#BlendingNext MC_BufferMode#BlendingHigh	
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
E	Active	BOOL	For buffered modes, this output is set high at the moment the block takes control of the axis. For non buffered modes, the outputs Busy and Active have the same value.	
E	CommandAborted	BOOL	Set high if motion is aborted by another motion command or MC_Stop. This output is cleared with the same behavior as the Done output.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

- The Offsets provided by this function are not related to any initial offset the slave may have had when engaged with `Y_Start_Mode.SlaveAbsolute:=FALSE.`
- The Offset input absolute. If the current Offset is at 15mm, and this function block is executed with an Offset input parameter value of 22mm, this function will increase the Offset by an additional 7mm.
- The underlying table is not affected; this function block only adds an Offset after the result of cam table lookup.

• 0.0 Offset will cause no offset of the cam data, however the initial slave offset (discrepancy between slave commanded position and first data used from the table) when Y_Start_Mode.SlaveAbsolute:=FALSE will remain.

• This function uses a modified sine pattern to ‘meter in’ the change from 0 to 100% of the adjustment change required as described above.

• A master/slave relationship is defined the first time a Y_CamIn, Y_CamShift, Y_CamScale, or Y_SlaveOffset block completes (Done output is TRUE.) The "first time" is defined as power up or after completion of Y_CamOut. If the master/slave relationship is already defined, then it is checked for consistency, and if not correct, the block produces an error (Invalid master slave combination).

• Only BufferMode=MC_BufferMode#aborting or MC_BufferMode#buffered is supported. If MC_BufferMode#aborting, then the function block will abort any phase shifts, cam shifts, cam scaling that are currently taking place; it will not abort the underlying Cam or gearing. If MC_BufferMode#buffered, then the phase shift will occur after all previous phase shifts, cam shifts, cam scalings complete.

• For more information on slave offset behavior, see the Cam Transition Matrix.

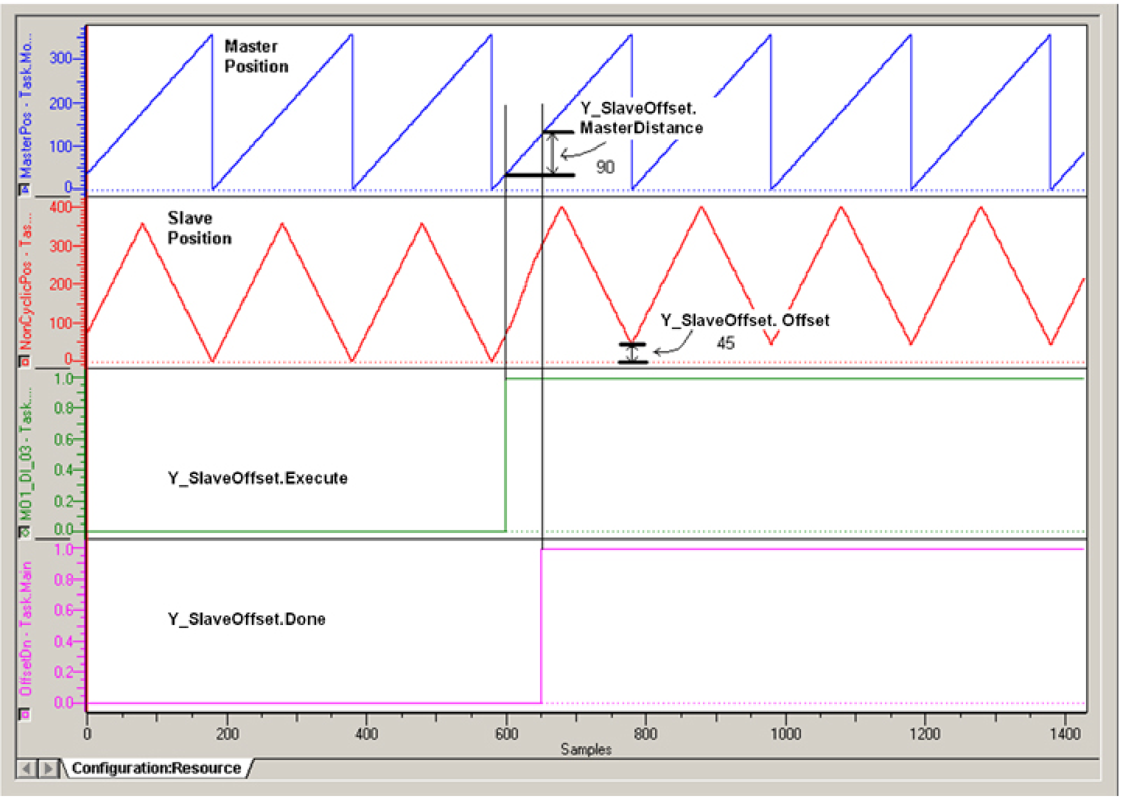
• For more information on how slave offset applies to camming, see the Camming Block Diagram.

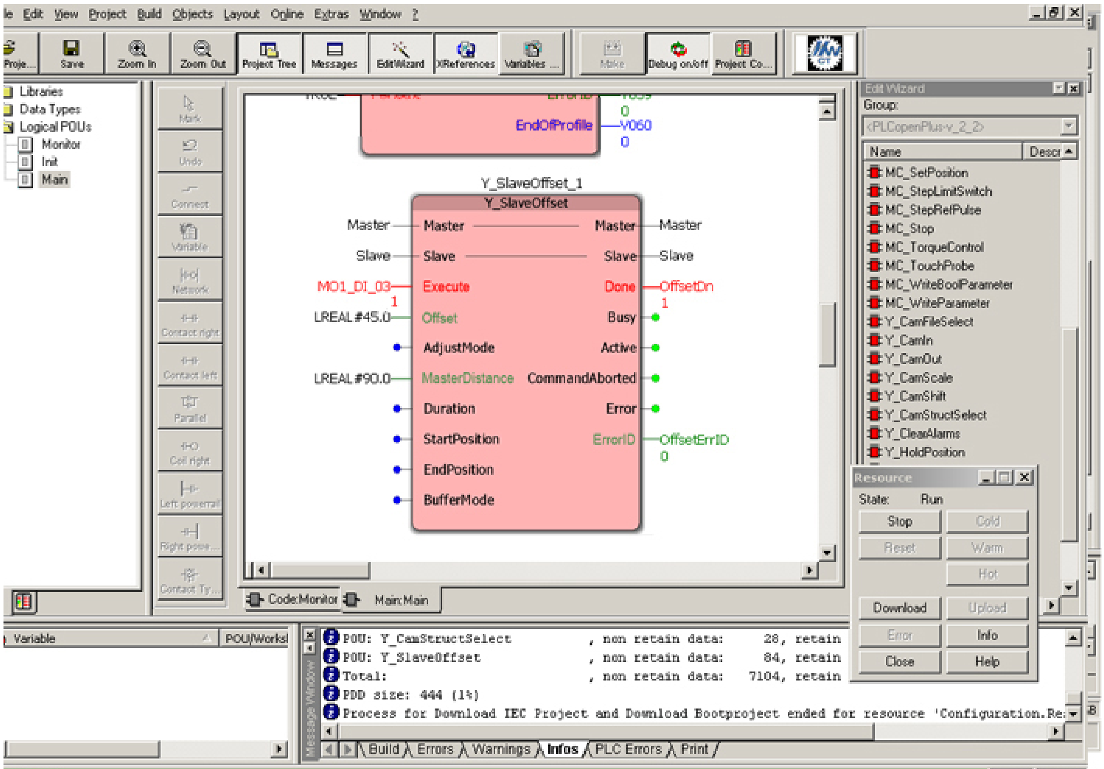
Error description

ErrorID	Meaning
0	No Error
4370	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis.Also, a motion block may be attempting to abort an MC_TorqueControl move.
4374	Torque move prohibited while non-torque moves queued or in progress.
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4625	The function block is not applicable for the external axis specified
4626	The master slave relationship is defined. A slave cannot be a master to another axis.
4633	Table size results in misaligned data. Refer to the help section “Internally Created Cam Data.” A cam table will have a multiple of 16 bytes if created correctly.
4649	Invalid adjust mode
4657	Distance parameter is less than or equal to zero.
4663	Specified time was less than zero.
4673	StartPosition is outside of master's range.

4674	EndPosition is outside of master's range.
57620	The structure size does not match.

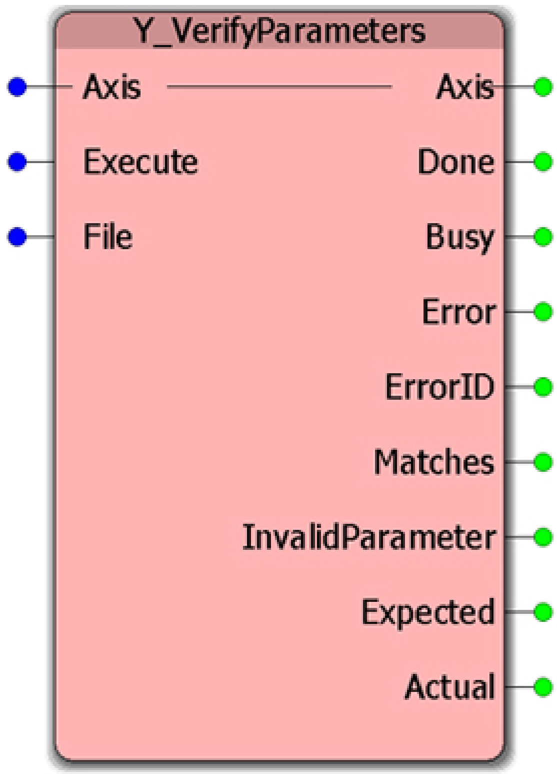
Example





Y_SlaveOffset_1
Y_SlaveOffset
Master
Slave
Execute
Offset
AdjustMode
MasterDistance
Duration
StartPosition
EndPosition
BufferMode
Done
Busy
Active
CommandAborted
Error
ErrorID
MO1_DI_03
LREAL#45.0
LREAL#90.0
OffsetDn
OffsetErrID
EndOfProfile
V060

Y_VerifyParameters



This Function Block compares the current parameters in the drive with the parameter file stored in the controller via the MotionWorks IEC Hardware Configuration.

Parameters

Parameter		Data type	Description	
VAR_IN_OUT				
B	Axis	[AXIS_REF]()	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated.	FALSE

			To modify an input, change the value and re-trigger the execute input.	
V	File	STRING	Optional, but if specified it is relative to /flash/user/driveParam/ on the controller. If the file name is not specified, then it defaults to "AXIS#DrivePn.xml", which is written to the controller when pressing Save from the Hardware Configuration.	See Description
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
E	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	
V	Matches	BOOL	Set to True when the current drive parameters match the parameter file.	
V	InvalidParameter	UINT	If Matches is False, this is set to the first drive parameter number that does not match.	
V	Expected	DINT	If Matches is False, Expected will contain the value in the parameter file.	
V	Actual	DINT	If Matches is False, Actual contains the actual value on the drive.	

Notes

If drive parameters are changed by the IEC application program using Y_WriteDriveParameter, Y_VerifyParameters will indicate that the parameter sets are different.

Related Function Blocks

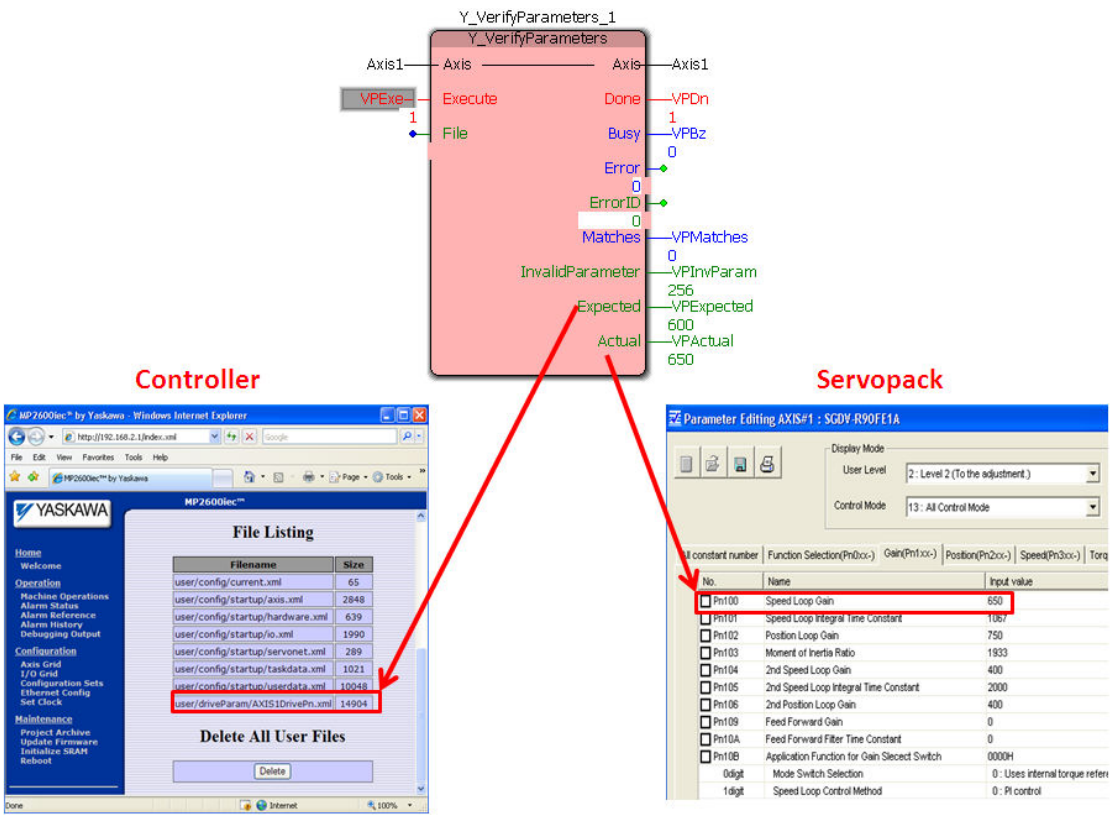
Y_WriteParameters: writes all parameters to the drive as stored in the controller via the MotionWorks IEC Configuration.

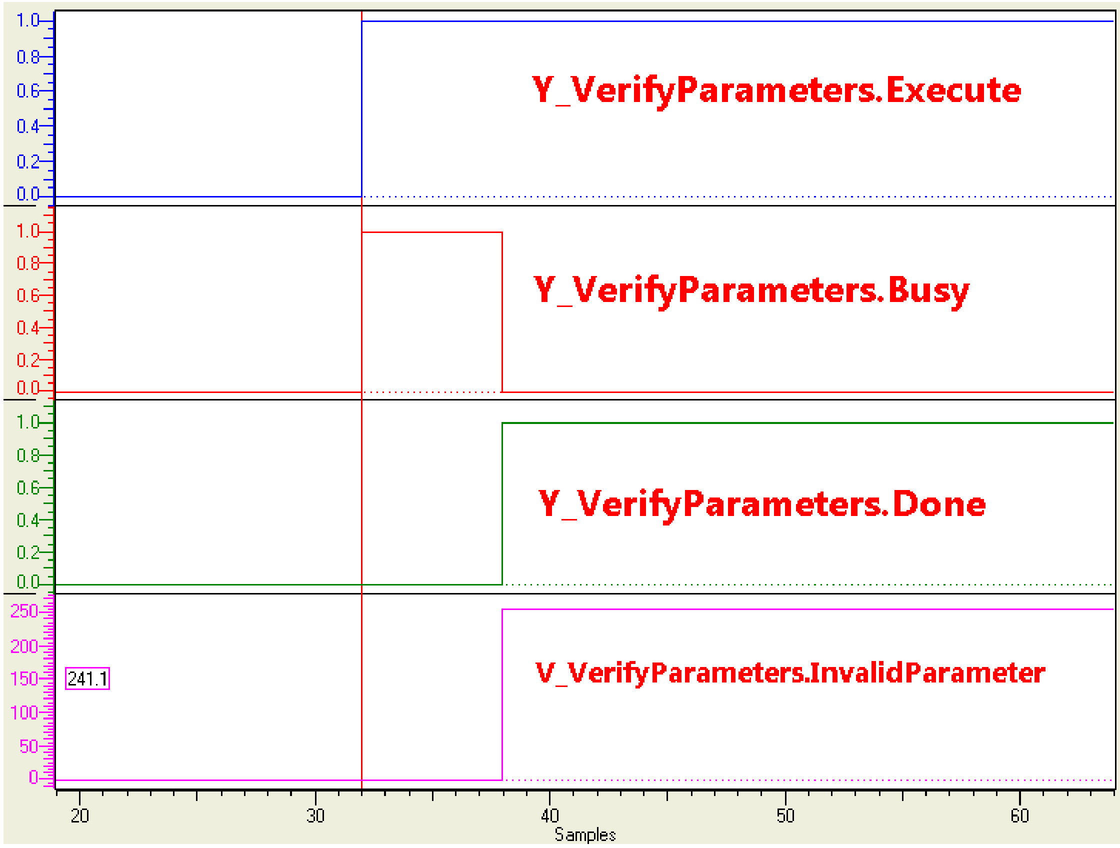
Error Description

ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis
4896	Drive parameter filename does not exist.
4897	The drive's model number or type does not match the parameter file.
57620	The structure size does not match.

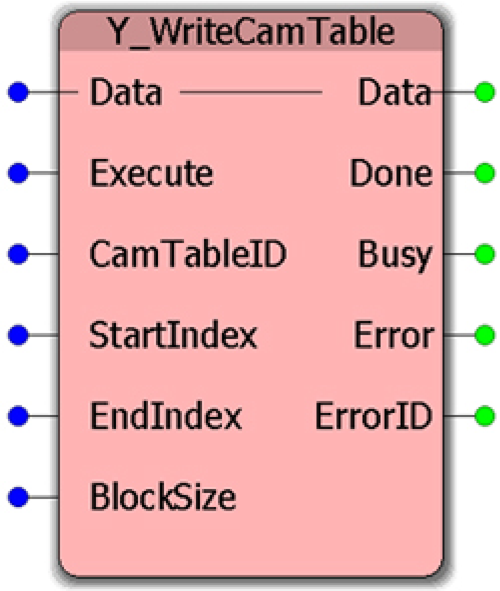
Example

Using Y_VerifyParameters to compare the servopack parameter lists saved in the controller flash and in the servopack. In the example shown below, there is discrepancy in parameter Pn100 (256 in decimal) where the value stored in the controller is 600 but the parameter in the servopack is 650.




Y_VerifyParameters.Execute
Y_VerifyParameters.Busy
Y_VerifyParameters.Done
V_VerifyParameters.InvalidParameter
241.1
Samples

Y_WriteCamTable



This Function Block copies cam data from the application program memory into the motion memory.

Parameters

Parameter		Data Type	Description	
VAR_IN_OUT				
V	Data	Y_MS_CAM_STRUCT	Cam data structure	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
B	CamTableID	UINT	A reference to the cam memory in the motion engine.	UINT#0
V	StartIndex	UDINT	Index into cam table in bytes (as used with Y_Cam_Struct)	UDINT#0
V	EndIndex	UDINT	Index into cam table in bytes (as used with Y_Cam_Struct). 0 is interpreted as	UDINT#0

			the maximum index.	
V	BlockSize	UDINT	Size of cam data in bytes copied per application task rate (if BlockSize is unconnected, then the full amount).	UDINT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low..	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

• This function block requires that a cam file was previously loaded with Y_CamFileSelect or Y_CamStructSelect.

• When writing the cam table, this function block shall not exceed the EndIndex, the cam table size, or the number of elements in Data.

• If EndIndex=0, then it defaults to the cam table size.

• Each scan, the function block copies a portion of data from the application program memory to the motion memory area. The BlockSize input specifies the number of data pairs to transfer per scan. If BlockSize is 0, then the entire table is copied in one PLC scan. If the table is large and the task time is small, a watchdog error may result.

• Y_MS_CAM_STRUCT is any 'ANY' input, but the motion kernel memory checks that it starts with a valid Y_CAM_HEADER.

• Refer to the Internally Created Cam Data diagram in the Cam Data Management section.

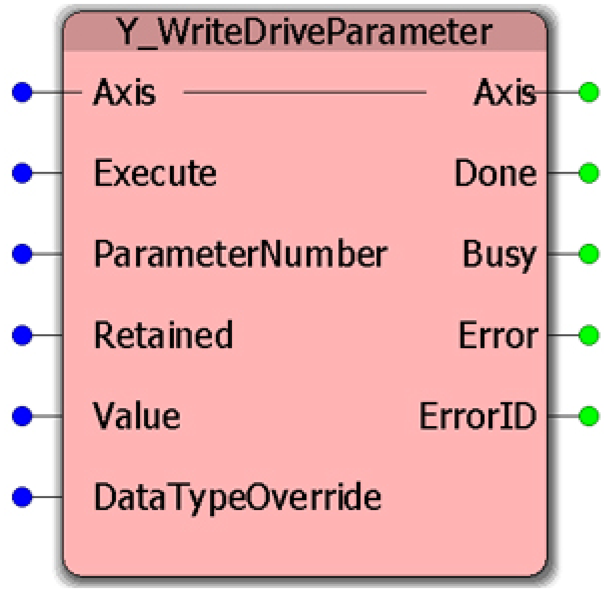
Error Description

ErrorID	Meaning
0	No Error
4378	The function block is not applicable for the external axis specified
4381	Motion aborted due to axis alarm.
4387	Already copying cam data (If Execute transition to TRUE while Busy = TRUE)
4633	Table size results in misaligned data. Refer to the help section “Internally Created Cam Data.” A cam table will have a multiple of 16 bytes if created correctly.
4635	Table type is not supported
4636	Invalid start index.
4637	Invalid end index
4885	Invalid header for the cam file. You must first populate the TableType and DataSize in the Y_MS_CAM_STRUCT before executing the function.
4887	CamTableID does not refer to a valid cam table.

Example

See Example for Y_ReadCamTable

Y_WriteDriveParameter



This Function Block writes the specified parameter to the drive or amplifier of the specified axis. To transfer a copy of all parameters from the controller to the amplifier, use Y_WriteParameters.

Parameters

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
V	ParameterNumber	UINT	Number of the Parameter in the drive. Note that the parameter numbers for the Sigma amplifiers are displayed in hex in all	UINT#0

			documentation. For consistency, the ParameterNumber can be entered in hex as shown in the example below.	
V	Retained	BOOL	If set to TRUE, the parameter is written to RAM and FLASH	FALSE
V	Value	DINT	The drive parameter value	DINT#0
V	DataTypeOverride	INT	Enumeration with the following values: 0 = default (i.e., fetched from the parameter XML file.); 1 = UINT; 2 = UDINT; 3 = INT; 4 = DINT.	INT#0
VAR_OUTPUT				
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.	
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.	
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.	
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.	

Notes

Use the links below to access the ServoPack manuals on www.yaskawa.com to view the list of parameters.

- Sigma II with NS115: SIEPC71080001, see Appendix B 1
- Sigma III: YEA-SIA-S800-11, see section 11.2.2
- Sigma-5 with rotary motor: SIEPS8000046, see Section 10.1.2
- Sigma-5 with linear motor: SIEPS8000048, see Section 9.1.2

In most cases, the drive parameters are 16 bit values and the DataType override is not necessary.

The parameter size (2 or 4 bytes) and sign is fetched from the default parameter XML files. If the parameter is not found in this file, the a “NoDefaultParameterInfo” error will occur

- If the Retained input is TRUE, the change persists across drive power cycles.
- If the user wishes to set an unsigned number greater than 268435455, the user must first use the function UDINT_TO_DINT.
- DataTypeOverride is an enumeration with the following values:

  0 = default (i.e., fetched from the parameter XML file.)

  1 = UINT

  2 = UDINT

  3 = INT

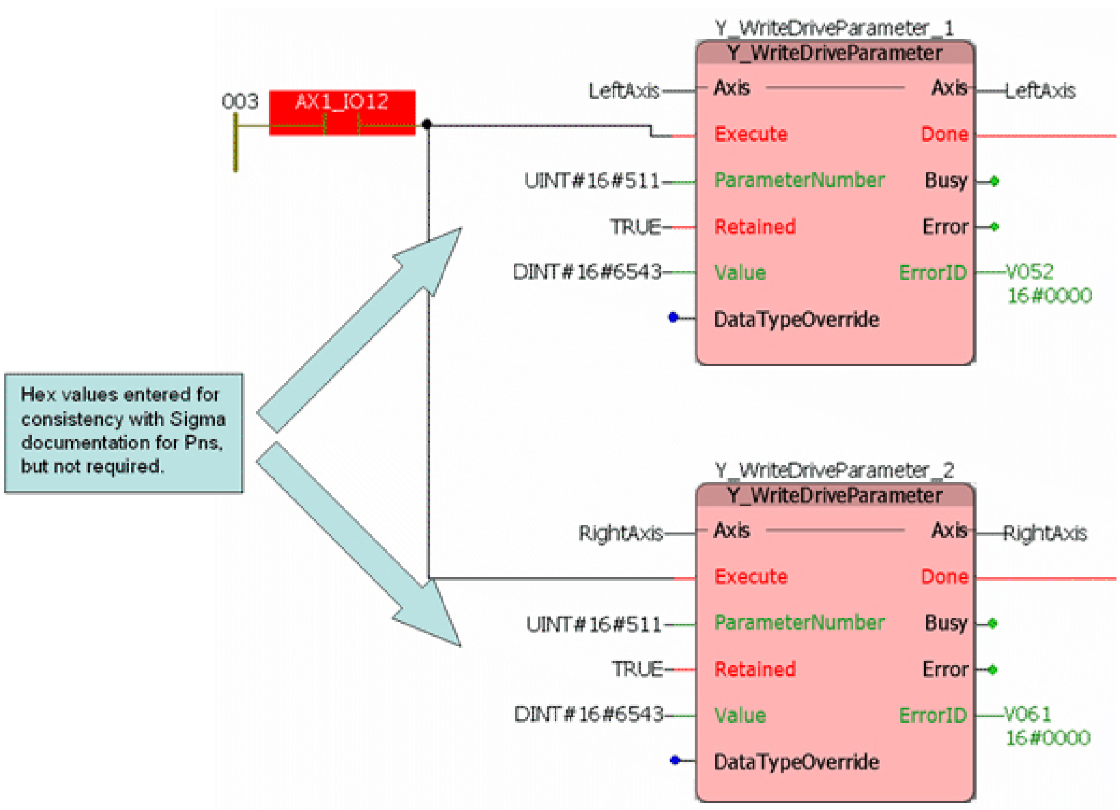
  4 = DINT

Error description

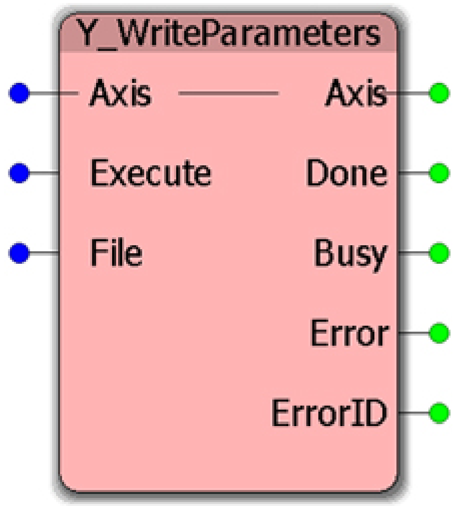
ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4401	The controller cannot communicate with the axis. It may be disconnected from the network.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis

	number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4892	Default drive parameter info is not available for this parameter.
4897	The drive's model number or type does not match the parameter file.
57620	The structure size does not match.

Example



Y_WriteParameters



This Function Block writes all parameters to the drive as stored in the controller via the MotionWorks IEC Configuration.

Parameters

Parameter		Data type	Description	Default
VAR_IN_OUT				
B	Axis	AXIS_REF	Logical axis reference. This value can be located on the Configuration tab in the Hardware Configuration (logical axis number).	
VAR_INPUT				**Default**
B	Execute	BOOL	Upon the rising edge, all other function block inputs are read and the function is initiated. To modify an input, change the value and re-trigger the execute input.	FALSE
V	File	STRING	Optional, but if specified it is relative to /flash/user/driveParam/ on the controller. If the file name is not specified, then it defaults to "AXIS#DrivePn.xml", which is written to the controller when pressing Save from the Hardware Configuration.	See Description

VAR_OUTPUT			
B	Done	BOOL	Set high when the commanded action has been completed successfully. If another block takes control before the action is completed, the Done output will not be set. This output is reset when execute goes low.
B	Busy	BOOL	Set high upon the rising edge of the 'Execute' or 'Enable' input, and reset if Done, CommandAborted, or Error is true.
B	Error	BOOL	Set high if error has occurred during the execution of the function block. This output is cleared when 'Execute' or 'Enable' goes low.
E	ErrorID	UINT	If error is true, this output provides the Error ID. This output is reset when 'Execute' or 'Enable' goes low.

Notes

This function is useful if a drive is replaced in the field, as the application program can reconfigure the drive for use without additional software.

Related Function Blocks

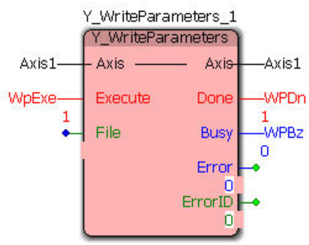
Y_VerifyParameters: compares the current parameters in the drive with the parameter file stored in the controller via the MotionWorks IEC Hardware Configuration.

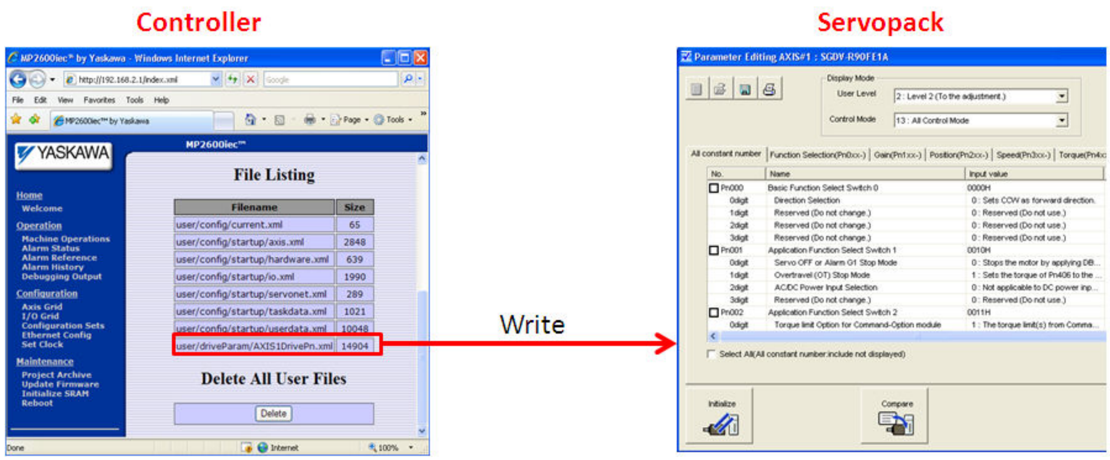
Error Description

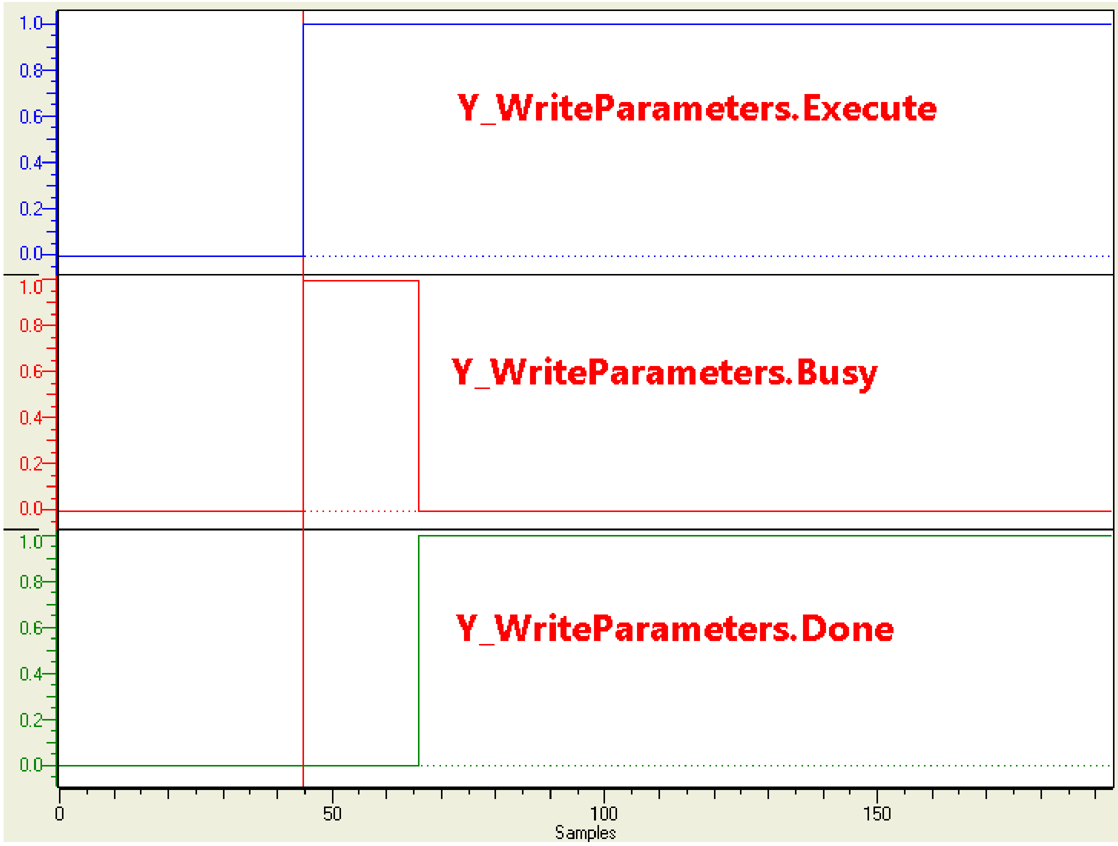
ErrorID	Meaning
0	No Error
4391	The function block can not be used with a virtual axis.
4625	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4648	The parameter number does not exist for the specified axis
4896	Drive parameter filename does not exist.

57620	The structure size does not match.

Example






Y_WriteParameters.Execute
Y_WriteParameters.Busy
Y_WriteParameters.Done
Samples

Controller AlarmID List

The following is a list of alarm codes that are reported in the Hardware Configuration's Controller Alarms tab or via the Y_ReadAlarm function block.

	Hex Code		**Description**
	ErrorClass	**AxisErrorID**	**ErrorClass+AxisErrorID output from MC_ReadAxisError**
	AlarmID		**AlarmID output from Y_ReadAlarm**
motionKernel	1201	0103	An alarm task queue was full when a new alarm was posted. This indicates that the task is being starved of execution time or that the system is generating many alarms simultaneously.
app	1401	0005	The script environment ran out of memory. This is a serious condition because it may prevent further errors from being handled correctly.
app	1401	0006	An error occurred while running the standard error handler for a general script error. This is a serious condition because it indicates the standard error handler is malfunctioning.
app	1401	0007	This error should never occur and is included only for completeness. It indicates that an unknown and potentially fatal problem has occurred within the script engine.
app	1401	000A	The script task failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
app	1401	000B	The command line task failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
app	1403	0002	The task responsible for publishing events to a remote client failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
app	1403	0003	The task responsible for replying to remote clients failed to stop cleanly, which may result in unreleased system

			resources. Error recovery requires the controller be reset.
app	1403	0004	The task responsible starting and stopping connections to remote clients failed to stop cleanly, which may result in unreleased system resources. Error recovery requires the controller be reset.
app	1407	0001	The file system on which the configuration file directory resides could not be read and may be unmounted or corrupted. The system has booted in a minimal configuration mode, and most functionality is limited. If possible, the file system should be recovered or reformatted and new config files uploaded if applicable.
app	1407	0103	The watchdog timer expired.
app	1407	0108	A CPU exception occurred.
app	1407	0109	The firmware files on the controller do not match the expected checksums.
app	1407	010A	The manufacturing procedure failed. The controller probably could not fetch the current time from the network.
app	140A	0009	Network reset detected multiple Axes connected to the same servo network node.
app	140A	000A	Network reset detected multiple I/O connected to the same network node.
app	140A	0015	Controller memory was corrupted during network reset resulting in a lost logical Axis data structure.
app	140A	0016	Controller memory was corrupted during network reset resulting in a lost logical I/O data structure.
app	140A	0018	An Abort input specified in the configuration could not be found. The abort condition is considered permanently asserted. No motion is possible until the I/O configuration can be matched to the abort inputs (restart required).
app	140A	0021	Too many events were posted from the system ISR. The motion scan and servo net loop have been shut down.
app	140C	1035	The manufacturing data on the controller is invalid. The controller needs to be returned to Yaskawa for reprogramming.
Mechatrolink	2301	0001	The drive returned an invalid watch dog code indicating a possible dropped communication packet.
Mechatrolink	2301	0002	The drive failed to return confirmation of last aux command within the default timeout period.
Mechatrolink	2301	0003	An unrecoverable error occurred during auto configuration. As a result, one or more drives are excluded from the servo network.

Mechatrolink	2301	0004	Overriding the auto configured axes parameters failed. As a result, one or more drives are excluded from the servo network.
Mechatrolink	2301	0005	Two or more nodes have the same ID. As a result, all servo network communication has been suspended.
Mechatrolink	2301	0006	The controller must be the root node on the servo network. All servo network communication has been suspended
Mechatrolink	2301	0007	The servo network communication device failed to initialize. Servo network communication is not possible.
Mechatrolink	2301	0008	An error occurred sending command to a node during initialization. The node may not support the configured communications rate. Communication with this node has been prohibited, but communication with other nodes may be possible.
Mechatrolink	2301	000E	The drive does not return response packet.
Mechatrolink	2301	000F	Bus reset generation that controller is not demanding.
Mechatrolink	2301	0010	It receives response with the same channel at the same Iso cycle.
Mechatrolink	2301	0011	The ID in the response packet is not same to ID of AxisNode.
Mechatrolink	2301	0012	The data length in the response packet is not same to value of CSR register(SEND_DSP_DATA_LENGTH) of drive.
Mechatrolink	2301	0013	The packet type in the response packet is not same S-DSP.
Mechatrolink	2301	0014	Invalid cycle time has passed with configuration file 'servonet.xml'. As a result, all servo network communication has been suspended.
Mechatrolink	2301	0015	Node is not found on 1394 network.
Mechatrolink	2301	0016	Invalid node.
Mechatrolink	2301	0017	Error matching node IDs.
DPRAM	2309	0001	Invalid watch dog code from drive
DPRAM	2309	0002	Aux command confirmation failure
DPRAM	2309	0003	Auto configuration failed
DPRAM	2309	0004	Overriding auto configuration failed
DPRAM	2309	0005	Invalid cyclic check sum from drive
DPRAM	2309	0006	Invalid watch dog from drive
DPRAM	2309	0007	Control mode is not supported
DPRAM	2309	0008	Communication with a node failed during servo network startup

motionKernel	3103	0100	Controller SRAM battery is low
motionKernel	3103	0101	The file system failed the integral consistency check. **Remedy**: Power up the controller in supervisory mode using the SUP switch. Clear the alarm. Turn off the SUP switch. Power cycle the controller.
motionKernel	3201	0001	The motion kernel didn't request to enable axis. But, the axis is enabled.
motionKernel	3201	0002	The motion kernel didn't request to disable axis. But, the axis is disabled.
motionKernel	3201	0004	The encoder position stored in SRAM could not be validated. The value has been reset.
motionKernel	3201	0005	Main bus power was disconnected while the axis was enabled. Main power must be restored and this alarm cleared before motion can continue.
motionKernel	3201	0101	Configuration error: multiple alarm tasks with duplicate priority.
motionKernel	3201	0102	Configuration error: Alarm task not configured. Using default priority and name.
motionKernel	3202	0001	Axis Coordinate System: The command position was outside the allowable range for the axis in the positive direction (positive overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	3202	0002	Axis Coordinate System: The command position was outside the allowable range for the axis in the negative direction (negative overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	3202	0003	Axis Coordinate System: The command speed was greater than the allowable range for the axis in the positive direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0004	Axis Coordinate System: The command speed was greater than the allowable range for the axis in the negative direction (overspeed). The axis may not be moved again

			until the alarm condition is cleared.
motionKernel	3202	0005	Axis Coordinate System: The command acceleration was greater than the allowable range for the axis in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0006	Axis Coordinate System: The command acceleration was greater than the allowable range for the axis in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0007	Axis Coordinate System: The command torque was greater than the allowable range for the axis in the positive direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0008	Axis Coordinate System: The command torque was greater than the allowable range for the axis in the negative direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0011	Joint Coordinate System: The command position was outside the allowable range for the axis in the positive direction (positive overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	3202	0012	Joint Coordinate System: The command position was outside the allowable range for the axis in the negative direction (negative overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	3202	0013	Joint Coordinate System: The command speed was greater than the allowable range for the axis in the positive direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0014	Joint Coordinate System: The command speed was greater than the allowable range for the axis in the negative direction (overspeed). The axis may not be moved again until the alarm condition is cleared.

motionKernel	3202	0015	Joint Coordinate System: The command acceleration was greater than the allowable range for the axis in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0016	Joint Coordinate System: The command acceleration was greater than the allowable range for the axis in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0017	Joint Coordinate System: The command torque was greater than the allowable range for the axis in the positive direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0018	Joint Coordinate System: The command torque was greater than the allowable range for the axis in the negative direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0021	World Coordinate System: The command position was outside the allowable range for the axis in the positive direction (positive overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	3202	0022	World Coordinate System: The command position was outside the allowable range for the axis in the negative direction (negative overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	3202	0023	World Coordinate System: The command speed was greater than the allowable range for the axis in the positive direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0024	World Coordinate System: The command speed was greater than the allowable range for the axis in the negative direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0025	World Coordinate System: The command acceleration was

			greater than the allowable range for the axis in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0026	World Coordinate System: The command acceleration was greater than the allowable range for the axis in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0027	World Coordinate System: The command torque was greater than the allowable range for the axis in the positive direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0028	World Coordinate System: The command torque was greater than the allowable range for the axis in the negative direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0031	The move specified would exceed the software position limits in the positive direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0032	The move specified would exceed the software position limits in the negative direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0033	The move specified would exceed the software speed limits in the positive direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0034	The move specified would exceed the software speed limits in the negative direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0035	The move specified would exceed the software acceleration limits in the positive direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0036	The move specified would exceed the software acceleration limits in the negative direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0037	The move specified would exceed the software torque limits in the positive direction and was rejected before being started. The group may be moved again immediately if desired.

motionKernel	3202	0038	The move specified would exceed the software torque limits in the negative direction and was rejected before being started. The group may be moved again immediately if desired.
motionKernel	3202	0039	The predictive soft limit encountered a segment that doesn't support the predicted stopping point.
motionKernel	3202	0041	Cam and Contour tables must have a header indicating the number of rows and columns and a feed forward velocity flag. Comma separated data values following the header.
motionKernel	3202	0042	In CamTables, the first (master) column must be either increasing or decreasing.
motionKernel	3202	0043	In ContourTables, the first (time) column must start at zero and be increasing.
motionKernel	3202	0044	The master position was outside the range of the CamTable, which automatically stopped the cam motion.
motionKernel	3202	0045	One or more slave axes could not attain the target position and velocity within the user specified time limit for the Cam or Gear motion.
motionKernel	3202	0046	One or more slave axes could not attain the target position and velocity within the user specified distance limit for the Cam or Gear motion.
motionKernel	3202	0051	Axis enable failed. This problem is usually a result of communication problems with the servo drive.
motionKernel	3202	0052	Runtime computation detected an invalid motion parameter. This alarm ID can occur if a discrete move has to be completed but the commanded deceleration for that move is not sufficient. For example if a MoveAbsolute aborts another move and the axis has to stop at a position that will come up in a couple of scans, but the deceleration input on the MoveAbsolute is not high enough to make the desired profile, this alarm will occur.
motionKernel	3202	0061	The axis Positive Overtravel (P-OT) limit has been exceeded. Motion is prevented in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0062	The axis Negative Overtravel (N-OT) limit has been exceeded. Motion is prevented in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	3202	0100	The inverse kinematics computation detected a world position that can not be reached.

motionKernel	3202	0101	The inverse kinematics computation detected that the elbow 'handedness' (orientation) does not match the configuration. The 'handedness' must be fixed by commanding the individual axes or manually moving the robot.
motionKernel	3202	0102	The robot XY position intruded into the configured dead zone area near the origin.
Mechatrolink	3301	0009	Some motor properties, such as encoder resolution, maximum speed, and maximum torque, could not be determined for the attached motor. The serial encoder may be malfunctioning, incorrectly programmed, or unplugged.
Mechatrolink	3301	000B	Setting of Pn002, digits 3 and 4, disables torque limit and/or velocity limit in velocity and/or torque control modes. Set Pn002 = xx11 to initialize. Saving in the Hardware Configuration will automatically set Pn002.
Mechatrolink	3301	000D	The servo network does not support this motion control mode.
Mechatrolink	3301	0018	The command position specified an instantaneous jump too large relative to the current position. Sigma-5 amplifiers give an A.94b warning and ignore subsequent position commands for any absolute position reference greater than 2,097,152 encoder pulses (2 revolutions of a 20-bit encoder). The controller watches for deviation between command position and actual motor position greater than 1,966,080 encoder pulses and issues an alarm. This is at 1.875 revolutions of a 20-bit motor little bit of margin. Sigma-II/III drives have a lower maximum following error limit of 1,048,576 encoder pulses. The position error limit on the Servopack (Pn520) should not be set greater than 1.875 rev = 1,966,080.
Mechatrolink	3301	0019	Setting of Pn002 digit 4 specifies torque feed-forward, but the SERVOPACK model does not support torque FF in position mode.
Mechatrolink	3302	00E4	The setting of the MECHATROLINK-II transmission cycle is out of the allowable range.
Mechatrolink	3304	0000	The base code for io alarms. The io's alarm value is bitwise OR'd in with this base value.
DPRAM	3309	0009	An error occurred sending command to a servo
DPRAM	3309	000A	The drive has an alarm
DPRAM	3309	000B	The data buffer for reading drive parameters via the messaging interface was too small

DPRAM	3309	1000	Error code prefix for data link errors
DPRAM	3309	100F	Servo check sum error for data link
DPRAM	3309	1010	Invalid function code for data link
DPRAM	3309	1040	Option card computed an invalid check sum
DPRAM	3309	1041	Invalid data size from the option card
DPRAM	3309	2000	Error code prefix for message errors
DPRAM	3309	2001	Unsupported message function code
DPRAM	3309	20A0	Controller option card detected bad CRC
DPRAM	3309	3000	Error code prefix for data link errors
Mechatrolink	3312	0000	The base code for inverter alarms. The inverter's alarm value is bitwise OR'd in with this base value.
Mechatrolink	3312	0001	reserved
Mechatrolink	3312	0002	reserved
Mechatrolink	3312	0003	reserved
Mechatrolink	3312	0004	reserved
Mechatrolink	3312	0005	reserved
Mechatrolink	3312	0006	reserved
Mechatrolink	3312	0007	reserved
Mechatrolink	3312	0008	reserved
Mechatrolink	3312	0009	reserved
Mechatrolink	3312	000A	reserved
Mechatrolink	3312	000B	reserved
Mechatrolink	3312	000C	reserved
Mechatrolink	3312	000D	reserved
Mechatrolink	3312	000E	reserved
Mechatrolink	3312	000F	reserved
Mechatrolink	3312	0010	reserved
Mechatrolink	3312	0011	reserved
Mechatrolink	3312	0012	reserved
Mechatrolink	3312	0013	reserved
Mechatrolink	3312	0014	reserved
Mechatrolink	3312	0015	reserved
Mechatrolink	3312	0016	reserved
Mechatrolink	3312	0018	reserved
Mechatrolink	3312	0019	reserved
Mechatrolink	3312	001A	reserved
Mechatrolink	3312	001B	reserved

Mechatrolink	3312	001C	reserved
Mechatrolink	3312	001D	reserved
Mechatrolink	3312	001E	reserved
Mechatrolink	3312	001F	reserved
Mechatrolink	3312	0020	reserved
Mechatrolink	3312	0021	reserved
Mechatrolink	3312	0025	reserved
Mechatrolink	3312	0026	reserved
Mechatrolink	3312	0027	reserved
Mechatrolink	3312	0028	reserved
Mechatrolink	3312	0029	reserved
Mechatrolink	3312	002A	reserved
Mechatrolink	3312	002B	reserved
Mechatrolink	3312	002C	reserved
Mechatrolink	3312	002D	reserved
Mechatrolink	3312	002E	reserved
Mechatrolink	3312	002F	reserved
Mechatrolink	3312	0031	reserved
Mechatrolink	3312	0083	reserved
Mechatrolink	3312	0084	reserved
Mechatrolink	3312	0085	reserved
Mechatrolink	3312	0086	reserved
Mechatrolink	3312	0087	reserved
Mechatrolink	3312	0088	reserved
Mechatrolink	3312	0089	reserved
Mechatrolink	3312	008A	reserved
Mechatrolink	3312	008B	reserved
Mechatrolink	3312	0091	reserved
Mechatrolink	3312	0092	reserved
Mechatrolink	3312	0093	reserved
Mechatrolink	3312	0094	reserved
Mechatrolink	3312	00E6	reserved
Mechatrolink	3312	00EC	Power reset required.
Mechatrolink	3312	00ED	(Access not possible 10 consecutive times). Power reset required.
Mechatrolink	3312	00EE	(1s elapsed). Power reset required.
app	3401	0001	The user script encountered an alarm, suspending its

			operation.
app	3401	0002	Script syntax errors are detected before the script is actually executed, during the pre-compile phase. The syntax must be corrected before the script can be run successfully.
app	3401	0003	Script runtime errors can be caused by a variety of incorrect script routines. The most common error is an attempt to use a 'nil' object where it should not be used.
app	3401	0004	The system could not find the file specified.
app	3401	0011	A data value argument provided to the API function was out of the expected range.
app	3401	0012	An argument provided to the API function was not the expected type.
app	3401	0013	An object argument provided to the API function was not the expected object type.
app	3401	0014	A scalar value was provided where a vector was expected, or a vector value was provided where a scalar was expected.
app	3401	0015	The script attempted to write to a read-only variable.
app	3401	0016	Use of that API function is not permitted with the current conditions and/or arguments.
app	3401	0017	The number of data values provided did not match the expected number of axes.
app	3401	0018	CamTable must have a header indicating the number of rows and columns and a feed forward velocity flag. Comma separated data values follows the header. The first (master) column must be either increasing or decreasing.
app	3401	0019	ContourTables must have a header indicating the number of rows and columns and a feed forward velocity flag. Comma separated data values follow the header. In ContourTables, the first (time) column must start at zero and be increasing.
app	3401	001A	It is prohibited to start a torque (or velocity) move when any moves other than torque moves (or velocity moves) are currently in progress or queued.
app	3401	00ED	'LastMove' events should be detected when a move completes normally or is aborted. However, the controller detected a situation in which the move finished but the event did not occur. Please submit an SCR.
app	3406	0001	A web server login user was assigned to a group which did not exist. The system is unaffected, but that user will have

			limited (default) access.
app	3406	0002	The default login group for the web server was assigned to a group which did not exist. Access control has been disabled, because a minimal amount of access is required in order to log in. The configuration file should be fixed before continuing.
app	3406	0003	The web server configuration specified access control should be enabled, but did not specify at least one path to control access to. Access control has been disabled. The configuration file should be fixed before continuing.
app	3407	0002	The base directory for configuration files was missing and has been created automatically. The system has booted in a minimal configuration mode, and most functionality is limited. Please upload a new complete configuration file set.
app	3407	0003	A required default configuration file was missing. A minimal configuration for the corresponding component has been loaded, and some functionality may be limited.
app	3407	0004	A required default configuration file was incorrectly formatted. A minimal configuration for the corresponding component has been loaded, and some functionality may be disabled.
app	3407	0005	A configuration file specified by the user configuration file set was incorrectly formatted. The corresponding default configuration file is being used instead.
app	3407	0006	The file describing which configuration set to use was corrupted. The default configuration set is being used.
app	3407	0007	An error occurred while writing a config file. The file system may be full or damaged.
app	3407	0101	The configured RAM disk on the controller was unable to be created.
app	3407	0102	Detected an unsupported option card inserted in the controller.
app	3407	0104	Data in the controller SRAM did not match the expected value. It should be treated as corrupted until it is re-initialized.
app	3407	0106	The SRAM battery backup power failed. SRAM data should be treated as corrupted until it is re-initialized.
app	3407	0107	The controller's time-of-day clock detected a voltage decrease in the backup battery. The current time and date is likely to be incorrect. This alarm can be cleared, but will recur when the controller is powered ON until the time

			and day is reset and the battery is replaced.
app	3407	0204	Unable to set configured network default gateway
app	3409	0001	The servo network axis node for the axis specified in the configuration file was not found.
app	3409	0002	Axis enable failed. This problem is usually a result of communication problems with the drive. It may occur after a drive was disconnected from the network. In this case, use Y_ResetMechatrolink to establish communication with the drives once again.
app	3409	0003	Axis group motion activation failed. Some axes in the group are currently under control of another group, or motion has been blocked by the user.
app	3409	0004	The motion segment could not be added to the motion queue because it is already queued.
app	3409	0005	Moves are prohibited when any of the group's axes are disabled, have an alarm, or are in violation of their soft limits.
app	340A	0001	The source for the logical input was not found, the configured input will not be available.
app	340A	0002	The source for the logical output was not found, the configured output will not be available.
app	340A	0003	Two or more axis in the configuration file had the same axis ID.
app	340A	0004	The servo network axis node for the axis specified in the configuration file was not found.
app	340A	0005	The axis group specified in the configuration file could not be created because either one or more of its axes are invalid or the group name is already being used.
app	340A	0006	The type of AtTargetAgent specified in the configuration file is unknown. This is because AtTargetAgent could not be created.
app	340A	0007	The number of constraints for axis group soft limit must be the same as the number of axes in the axis group.
app	340A	0008	The axis group doesn't have the configured frame.
app	340A	000B	A continuous-wrap range for an axis causes its position to automatically wrap around between two user-specified numbers. Generally these numbers evaluate to full revolutions of the encoder but other ranges are permitted. However, all ranges specified in user units must map exactly to an integral number of encoder pulses. This alarm indicates that the mapping from user units to encoder ticks was inexact. Use more precise numbers to

			describe the range or choose a different range that evaluates to an integral number of encoder pulses. When this alarm occurs at startup or servo-net reset, it indicates that the axis has not been connected to an axis node and cannot be servoed on. Otherwise, this alarm indicates that the specified continuous-wrap range was not put into effect.
app	340A	000D	Two or more logical outputs specified in the I/O configuration file use the same physical bit. This can cause writes to not correctly generate value-change events on logical outputs for the shared bits. The configuration file should be fixed.
app	340A	000E	One or more of the data parameters in the axis configuration file were out-of-range or otherwise incorrectly specified for the axis. The axis was not created and is not available.
app	340A	0010	After servo network reset, the Axis failed to reconnect to the servo network. The drive might have been removed from the network, the node ID of the drive might have changed or there might be a communication problem.
app	340A	0012	After servo network reset, the network I/O failed to reconnect to the servo network. The network I/O module might have been removed from the network, the node ID of the network I/O module might have changed or there might be a network communication problem.
app	340A	0013	After servo network reset, a new axis node was discovered. This axis node is not associated with any existing axes and will not be available. To make this node available, update the configuration and power cycle the controller.
app	340A	0014	After servo network reset, a new I/O node was discovered. This I/O node is not associated with any existing I/O and will not be available. To make this node available, update the configuration and power cycle the controller.
app	340A	0017	One or more of the axis data or configuration parameters were inconsistent or incompatible with the axis node specified. The axis was created but was not connected to the servo node.
app	340A	001B	Two or more LogicalInput have the same ID. The configuration file should be fixed.
app	340A	001C	Two or more LogicalOutput have the same ID. The configuration file should be fixed.

app	340A	001D	Two or more AnalogInput have the same ID. The configuration file should be fixed.
app	340A	001E	Two or more AnalogOutput have the same ID. The configuration file should be fixed.
app	340A	001F	Analog I/O configuration is missing the 'hardwareConfig' element, and configuration could not be resolved by the physical hardware. The configuration file should be fixed by adding this element to the analog I/O element.
app	340A	0020	One or more axes failed to respond to a servo-off command during a system I/O initiated abort. This is normally the result of communication problems with the drive, which also causes an automatic servo-off.
app	340A	0022	Reset of a servo node failed.
app	340A	0023	The axis position may not be valid because the persistent axis data was corrupted. SRAM should be reinitialized and the axis should be homed.
app	340C	0000	All PLCopen error codes are in the range from 0x0000 to 0x0fff.
app	340C	0001	Time limit exceeded.
app	340C	0002	Distance limit exceeded.
app	340C	0003	Torque limit exceeded.
app	340C	0100	Modbus TCP I/O Driver Error on Server because of invalid address range
app	340C	0101	MBTCP Client I/O driver, MBTCP Connection config is missing input member
app	340C	0102	I/O memory area is not aligned to the correct byte to accommodate reading and writing.
app	340C	0103	Watchdog Error
app	340C	0104	Reserved
app	340C	0106	Reserved
app	340C	0107	Reserved
app	340C	0108	Reserved
app	340C	0109	Reserved
app	340C	010A	Not enough memory on PLC for POU during insertion. Project size must be reduced.
app	340C	010B	Internal PLC Error in memory management. This error can occur if an older project was loaded on the controller which was compiled to use lees of the controllers total memory space. By using the "Resource" Dialog box, perform "Delete On target," for the bootproject, and then download the application code again.

app	340C	010C	Internal PLC Error: POU invalid
app	340C	010D	Internal PLC Error: Unknown POU type
app	340C	010E	Cannot insert a POU because there is no project.
app	340C	010F	Internal PLC Error: Cannot insert a POU because it does not belong to the project.
app	340C	0110	Internal PLC Error: Cannot insert a POU.
app	340C	0111	Internal PLC Error: Invalid POU type
app	340C	0112	Internal PLC Error: Memory reorganization not possible; PLC stopped.
app	340C	0113	Internal PLC Error: SPG defined more than once.
app	340C	0114	Internal PLC Error: Memory error for initialized data of POU.
app	340C	0115	Internal PLC Error: Retain CRC failed. Possible reasons: (1) actual project does not have any retain data, (2) actual project is 'old style' without retain CRC (3) PLC isn't in STOP mode
app	340C	0116	Internal PLC Error: FB defined more than once.
app	340C	0117	Internal PLC Error: Not all POU sent.
app	340C	0118	Internal PLC Error: No program memory defined.
app	340C	0119	Internal PLC Error: Invalid FB number.
app	340C	011A	Internal PLC Error: Invalid PG number.
app	340C	011B	Internal PLC Error: Invalid SPG number.
app	340C	011C	POU uses more than 80 percent of POU memory.
app	340C	011D	Project uses more than 80 percent of program memory.
app	340C	011E	Internal PLC Error: Invalid function or function block.
app	340C	011F	Internal PLC Error: Invalid firmware function or function block.
app	340C	0120	Internal PLC Error: Invalid program.
app	340C	0121	Internal PLC Error: Invalid change of mode.
app	340C	0122	Internal PLC Error: Unknown system mode! PLC stopped!
app	340C	0123	Stack overflow. Increase stack size.
app	340C	0124	System error in module. Check debugging output via controller's web interface.
app	340C	0125	System error in module. Check debugging output via controller's web interface.
app	340C	0126	Internal PLC Error: Error during indirect variable access.
app	340C	0127	PLC CPU overload.
app	340C	0128	Internal PLC Error: Breakpoint unexpected.
app	340C	0129	Internal PLC Error: Error in data configuration.

app	340C	012A	Internal PLC Error: Error in retain data configuration.
app	340C	012B	Internal PLC Error: Floating point error.
app	340C	012C	Internal PLC Error: Fatal error.
app	340C	012D	Output string is too short.
app	340C	012E	Input string is too short.
app	340C	012F	Invalid input parameter 'p' or 'l' (position or length).
app	340C	0130	String is identical to the output string.
app	340C	0131	Invalid string comparison.
app	340C	0132	Invalid data type for string conversion.
app	340C	0133	Error in format string.
app	340C	0134	Error during string conversion.
app	340C	0135	Error in I/O configuration.
app	340C	0136	Initializing I/O driver failed.
app	340C	0137	Board not instantiated.
app	340C	0138	Board number not allowed.
app	340C	0139	Input Group doesn't fit.
app	340C	013A	Output Group doesn't fit.
app	340C	013B	Board not found.
app	340C	013C	Error reading inputs.
app	340C	013D	Error writing outputs.
app	340C	013E	Error creating I/O semaphore.
app	340C	013F	Invalid memory size.
app	340C	0140	Invalid I/O memory address.
app	340C	0141	Internal PLC Error: PG defined more than once.
app	340C	0142	POU exceeds 64K module size during insertion. POU size must be reduced.
app	340C	0143	Internal PLC Error: Error in task configuration.
app	340C	0143	Unknown I/O Driver.
app	340C	0200	Common causes of invalid configuration include duplicate t2o/o2t assembly instances or invalid client connection parameters.
app	340C	0202	Unable to connect to the EtherNet/IP remote server. Common causes include: invalid remote server address, invalid gateway, invalid subnet mask, or the Ethernet network is not correctly configured.
app	340C	0203	There is no route to the EtherNet/IP server. Common causes include: invalid remote server address, invalid gateway, invalid subnet mask, or the Ethernet network is not correctly configured.

app	340C	0204	Unable to reach the network for the EtherNet/IP server. Common causes include: invalid remote server address, invalid gateway, invalid subnet mask, or the Ethernet network is not correctly configured.
app	340C	0205	Remote server rejected connection attempt. The remote server may not be listening for connections or there may be a firewall preventing the connection.
app	340C	0206	The Ethernet/IP client ran out of connection slot resources. Reduce the number of concurrent client connections.
app	340C	0302	Unable to connect to the Modbus TCP slave. Common causes include: invalid Modbus TCP slave address, invalid gateway, invalid subnet mask, or the Ethernet network is not correctly configured.
app	340C	0303	There is no route to the Modbus TCP slave. Common causes include: invalid Modbus TCP slave address, invalid gateway, invalid subnet mask, or the Ethernet network is not correctly configured.
app	340C	0304	Unable to reach the network for the Modbus TCP slave. Common causes include: invalid Modbus TCP slave address, invalid gateway, invalid subnet mask, or the Ethernet network is not correctly configured.
app	340C	0305	Modbus TCP slave rejected connection attempt. The Modbus TCP slave may not be listening for connections or there may be a firewall preventing the connection.
app	340C	0306	The Modbus TCP master ran out of connection slot resources. Reduce the number of concurrent slave connections.
app	340C	1020	The controller battery voltage has dropped, indicating it has failed or is about to fail. While the controller is powered on, the battery should be replaced as soon as possible or a prolonged power-down state will cause various static data to be lost.
app	340C	1028	The driver parameter specified in the axis configuration caused an exception
app	340C	1029	The driver parameter did not match the axis configuration
app	340C	1030	The configured axis count exceeded the allowable limit.
app	340C	1031	The axis count exceeded the allowable limit due to an auto-detected axis.
app	340C	1033	Using an incompatible version of the PLCopenPlus firmware function block library may result in controller instability. Consequently, the PLC application will not be

			allowed to run. Please change either the controller's firmware or the firmware function block library.
app	340C	1110	All motion error codes are in the range from 0x1111 to 0x111f.
app	340C	1111	The move could not be buffered because the motion queue for that axis is full.
app	340C	1112	The move could not be started because motion is prohibited.
app	340C	1113	The servo drive failed to enable or disable.
app	340C	1114	Drive parameter read/write did not complete.
app	340C	1115	Drive parameter read/write failed
app	340C	1116	Torque move prohibited while non-torque moves queued or in progress.
app	340C	1117	CamOut called while not camming.
app	340C	1118	The master slave relationship can not be modified because the master axis has not been set yet.
app	340C	1119	CamFileSelect can not open a second cam table while the first cam table is still being opened.
app	340C	111A	The function block can not command an external axis.
app	340C	111B	The homing sequence is already in progress.
app	340C	111C	MC_SetPosition can not be called while the axis is moving.
app	340C	111D	Motion aborted due to axis alarm.
app	340C	111E	MC_SetPosition can not set the position to be outside the configured wrap range.
app	340C	111F	Can not transition to homing state; must be in StandStill state first.
app	340C	1120	Clear alarms is already in progress.
app	340C	1121	Axis reset is already in progress.
app	340C	1122	Mechatrolink reset is already in progress.
app	340C	1123	CamStructSelect cannot tansfer a second cam structure while the first cam structure is being transferred.
app	340C	1124	CamTableRead cannot be read a second cam structure while the first cam structure is being read.
app	340C	1125	CamTableWrite cannot write a second cam structure while the first cam structure is being written.
app	340C	1126	MC_SetPosition cannot be called while either the master or slave axis is caming.
app	340C	1127	The function block can not be used with a virtual axis.
app	340C	1128	The function block can not be used with an inverter axis.

app	340C	1129	Y_VerifyParmeters and Y_WriteParameters can not be called a second time while the first one is in progress.
app	340C	1210	All error codes for structures are in the range from 0x1211 to 0x121f.
app	340C	1211	Axis ID does not correspond to an axis.
app	340C	1212	The master slave relationship is not defined.
app	340C	1213	The input reference does not correspond to a real input
app	340C	1214	The output reference does not correspond to a real output.
app	340C	1215	The input/output number does not correspond to a real input or output bit.
app	340C	1216	Trigger reference is not valid.
app	340C	1217	The cam switch structure is not valid.
app	340C	1218	The track structure is not valid.
app	340C	1219	Table size results in misaligned data.
app	340C	121A	Buffer size results in misaligned data.
app	340C	121B	Table type is not supported.
app	340C	121C	Invalid start index.
app	340C	121D	Invalid end index.
app	340C	1220	All error codes for invalid enumeration values are in the range from 0x1221 to 0x122f.
app	340C	1221	'BufferMode' does not correspond to a valid enumeration value.
app	340C	1222	'Direction' does not correspond to a valid enumeration value.
app	340C	1223	'StartMode' does not correspond to a valid enumeration value.
app	340C	1224	'ShiftMode' does not correspond to a valid enumeration value.
app	340C	1225	'OffsetMode' does not correspond to a valid enumeration value.
app	340C	1226	'Mode' does not correspond to a valid enumeration value.
app	340C	1227	'SynchMode' does not correspond to a valid enumeration value.
app	340C	1228	'Parameter' does not correspond to a valid enumeration value.
app	340C	1229	'AdjustMode' does not correspond to a valid enumeration value.
app	340C	122A	'RampIn' does not correspond to a valid enumeration value.

app	340C	122B	'ControlMode' does not correspond to a valid enumeration value.
app	340C	1230	All error codes for range errors are from 0x1221 to 0x122f.
app	340C	1231	Distance parameter is less than zero.
app	340C	1232	Velocity parameter is less than or equal to zero.
app	340C	1233	Acceleration is less than or equal to zero.
app	340C	1234	Deceleration is less than or equal to zero.
app	340C	1235	Torque is less than or equal to zero.
app	340C	1236	Time is less than or equal to zero
app	340C	1237	Specified time was less than zero.
app	340C	1238	Specified scale was less than or equal to zero.
app	340C	1239	Velocity is negative.
app	340C	123A	Denominator is zero.
app	340C	123B	Jerk is less than or equal to zero.
app	340C	123C	TorqueRamp is less than or equal to zero.
app	340C	123D	Engage position is outside the table domain.
app	340C	123E	Negative engage width.
app	340C	123F	Disengage position is outside the table domain.
app	340C	1240	Negative disengage width.
app	340C	1241	StartPosition is outside of master's range.
app	340C	1242	EndPosition is outside of master's range.
app	340C	1310	All error codes for invalid input data range from 0x1211 to 0x121f.
app	340C	1311	The specified Pn does not exist.
app	340C	1312	The mask does not correspond to valid tracks.
app	340C	1313	The profile must start with relative time equal to zero, and the time must be increasing.
app	340C	1314	The specified cam file does not exist.
app	340C	1315	Invalid header for the cam file. Cam tables must have a header indicating the number of rows, number of columns and a feed forward velocity flag
app	340C	1316	The first (master) column must be either increasing or decreasing.
app	340C	1317	Cam table reference does not refer to a valid cam table.
app	340C	1318	The engage phase exceeded the time limit. Slave axis could not attain the target position and velocity within the user specified time limit.
app	340C	1319	The engage phase exceeded the distance limit. Slave axis

			could not attain the target position and velocity within the user specified master distance.
app	340C	131A	Invalid width input. Width is an enumeration type with the following allowable values 'WIDTH_8'=0, 'WIDTH_16'=1, and 'WIDTH_32'=2.
app	340C	131B	The slave axis can not be the same as the master axis.
app	340C	131C	Default drive parameter info is not available for this parameter.
app	340C	131D	Invalid external axis.
app	340C	131E	Invalid virtual axis.
app	340C	131F	File extension is not recognized or missing.
app	340C	1320	Cound not find the axis parameter file.
app	340C	2110	All log error codes are in the range from 0x2111 to 0x211f.
app	340C	2111	Adding log items or setting up log is not possible because the data log is already set up.
app	340C	2112	Starting or stopping logging is not possible because the data log is not set up.
app	340C	2113	Invalid handle for user log item.
app	340C	2114	Data log can not be created because too many data logs are in use.
app	340C	2115	Invalid handle for data log.
app	340C	2116	A user log item can only support eight inputs for each type.
app	340C	2117	Saving the log failed.
app	340C	B114	Failed to send clear alarms command.
app	340C	B115	Failed to reset Mechatrolink.
app	340C	B116	Mechatrolink reset is prohibited while axes are moving.
app	340C	B117	Failed to initialize abs encoder.
app	340C	E110	All error codes for ProConOS errors range from 0xE111 to 0xE11f.
app	340C	E111	Instance object is NULL.
app	340C	E112	The instance data is NULL.
app	340C	E113	The structure pointer check sum is invalid.
app	340C	E114	The structure size does not match.
app	340C	EDED	This function block was implemented in a later firmware version. If you would like to use this function block, then the controller must be udpated.
app	340C	F110	All error codes for kernel errors range from 0xF111 to

			0xF11f.
app	340C	F111	An internal assertion in the motion kernel failed indicating the controller is not in a stable state. This error should be reported to Yaskawa Electric America.
user	3501	0000	A user script task posted an alarm directly.
motionKernel	4202	0001	The command position will soon reach the allowable range for the axis in the positive direction (positive overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	4202	0002	The command position will soon reach the allowable range for the axis in the negative direction (negative overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	4202	0003	The command speed will soon reach the allowable range for the axis in the positive direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0004	The command speed will soon reach the allowable range for the axis in the negative direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0005	The command acceleration will soon reach the allowable range for the axis in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0006	The command acceleration will soon reach the allowable range for the axis in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0007	The command torque will soon reach the allowable range for the axis in the positive direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0008	The command torque will soon reach the allowable range for the axis in the negative direction (overtorque). The

			axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0011	The command position will soon reach the allowable range for the axis in the positive direction (positive overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	4202	0012	The command position will soon reach the allowable range for the axis in the negative direction (negative overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	4202	0013	The command speed will soon reach the allowable range for the axis in the positive direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0014	The command speed will soon reach the allowable range for the axis in the negative direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0015	The command acceleration will soon reach the allowable range for the axis in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0016	The command acceleration will soon reach the allowable range for the axis in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0017	The command torque will soon reach the allowable range for the axis in the positive direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0018	The command torque will soon reach the allowable range for the axis in the negative direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0021	The command position will soon reach the allowable range

			for the axis in the positive direction (positive overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	4202	0022	The command position will soon reach the allowable range for the axis in the negative direction (negative overtravel). The axis may not be moved again until the alarm condition is cleared. After the alarm is cleared, it is permissible to execute a move which brings the axis back toward the allowed region, even though the axis is probably still outside the allowed region. Any move which pulls the axis further away from the allowed region will re-trigger the alarm.
motionKernel	4202	0023	The command speed will soon reach the allowable range for the axis in the positive direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0024	The command speed will soon reach the allowable range for the axis in the negative direction (overspeed). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0025	The command acceleration will soon reach the allowable range for the axis in the positive direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0026	The command acceleration will soon reach the allowable range for the axis in the negative direction. The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0027	The command torque will soon reach the allowable range for the axis in the positive direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
motionKernel	4202	0028	The command torque will soon reach the allowable range for the axis in the negative direction (overtorque). The axis may not be moved again until the alarm condition is cleared.
Mechatrolink	4301	000A	The SERVOPACK model type was unable to be determined. This can indicate that some parameters may be incorrect.
Mechatrolink	4301	000C	The controller was unable to send the drive command because servo network resources were allocated to

			motion. Brake on, brake off, absolute encoder initialization and alarm clear can only be sent when not moving.
Mechatrolink	4301	001C	The Mechatrolink.xml file specified duplicate configuration structures for a node. The first match was used, subsequent matches were ignored.
Mechatrolink	4301	001D	The Mechatrolink.xml file specified duplicate default configuration structures for a node type. The first default structure was used, subsequent structures were ignored.
Mechatrolink	4301	001E	A node was detected on the mechatrolink network, but it is not supported by the software.
Mechatrolink	4301	001F	The Mechatrolink comm board inverter control reference/run control is not enabled. Change the settings in parameters b1-01 and b1-02 to '3' to select PCB reference/run source.
Mechatrolink	4301	0020	The drive returned an invalid watch dog code indicating a possible dropped communication packet.
Mechatrolink	4302	0000	The base code for Sigma-II drive warnings. The drive's warning value is bitwise OR'd in with this base value.
Mechatrolink	4302	0091	This warning occurs before the overload alarms (A.710 or A.720) occur. If the warning is ignored and operation continues, an overload alarm may occur.
Mechatrolink	4302	0092	This warning occurs before the regenerative overload alarm (A.32) occurs. If the warning is ignored and operation continues, a regenerative overload alarm may occur.
Mechatrolink	4302	0093	This warning occurs when the absolute encoder battery voltage is lowered. Continuing the operation in this status may cause an alarm.
Mechatrolink	4302	0094	A value outside the setting range was set using MECHATROLINK-II communications.
Mechatrolink	4302	0095	A command not supported in the product specifications was sent, OR the command reception conditions were not met.
Mechatrolink	4302	0096	A communications error occurred (once).
Mechatrolink	4303	0000	The base code for Sigma-III drive warnings. The drive's warning value is bitwise OR'd in with this base value.
Mechatrolink	4303	0900	Position error pulse exceeded the parameter settings (Pn520 x Pn51E/100).
Mechatrolink	4303	0901	When the servo turned ON, the position error pulses exceeded the parameter setting (Pn526 x Pn528/100).
Mechatrolink	4303	0910	This warning occurs before the overload alarms (A.710 or A.720) occur. If the warning is ignored and operation

			continues, an overload alarm may occur.
Mechatrolink	4303	0911	Abnormal vibration at the motor speed was detected. The detection level is the same as A.520. Set whether to output an alarm or warning by "Vibration Detection Switch" of Pn310.
Mechatrolink	4303	0920	This warning occurs before the regenerative overload alarm (A.320) occurs. If the warning is ignored and operation continues, a regenerative overload alarm may occur.
Mechatrolink	4303	0930	This warning occurs when the absolute encoder battery voltage is lowered. Continuing the operation in this status may cause an alarm.
Mechatrolink	4303	0941	The change of the parameters can be validated only after turning the power ON from OFF.
Mechatrolink	4303	094A	Incorrect command parameter number was set.
Mechatrolink	4303	094B	Command input data is out of range.
Mechatrolink	4303	094C	Calculation error was detected.
Mechatrolink	4303	094D	Data size does not match.
Mechatrolink	4303	095A	Command was sent though command sending condition was not satisfied.
Mechatrolink	4303	095B	Unsupported command was sent.
Mechatrolink	4303	095C	Command condition is not satisfied for parameter settings.
Mechatrolink	4303	095D	Command, especially latch command, interferes.
Mechatrolink	4303	095E	Subcommand and main command interfere.
Mechatrolink	4303	0960	Communications error occurred during MECHATROLINK communications.
Mechatrolink	4304	0000	The base code for io warnings. The io's warning value is bitwise OR'd in with this base value.
DPRAM	4309	1000	Error code prefix for data link errors
DPRAM	4309	1011	Invalid register
DPRAM	4309	1012	Value exceeded data limit
DPRAM	4309	1013	Data math error
DPRAM	4309	1014	Register number and data size do not agree
DPRAM	4309	1015	Invalid data size
DPRAM	4309	1030	Servo and option card accessed data link channel at the same time
DPRAM	4309	10FF	Unknown data link error
DPRAM	4309	2000	Error code prefix for message errors
DPRAM	4309	2002	Invalid register

DPRAM	4309	2003	Message size and data quantity do no match
DPRAM	4309	2030	Invalid register
DPRAM	4309	2031	Register access not allowed
DPRAM	4309	2032	Setting value is out of range
DPRAM	4309	2033	Messaging accessed only part of a register group or spanned register groups
DPRAM	4309	2034	Message command could not be processed because pre-conditions have not been met
DPRAM	4309	2035	Command processing is not possible due to conflict
DPRAM	4309	20A1	Controller option card received an empty message response
Mechatrolink	4312	0000	The base code for inverter warnings. The inverter's warning value is bitwise OR'd in with this base value.
Mechatrolink	4312	0001	Reserved
Mechatrolink	4312	0002	Reserved
Mechatrolink	4312	0003	Reserved
Mechatrolink	4312	0004	Reserved
Mechatrolink	4312	0005	Reserved
Mechatrolink	4312	0006	Reserved
Mechatrolink	4312	0007	Reserved
Mechatrolink	4312	0008	Reserved
Mechatrolink	4312	0009	Reserved
Mechatrolink	4312	000A	Reserved
Mechatrolink	4312	000B	Reserved
Mechatrolink	4312	000C	Reserved
Mechatrolink	4312	000D	Reserved
Mechatrolink	4312	000E	Reserved
Mechatrolink	4312	0010	Reserved
Mechatrolink	4312	0011	Reserved
Mechatrolink	4312	0012	Reserved
Mechatrolink	4312	0013	Reserved
Mechatrolink	4312	0014	Reserved
Mechatrolink	4312	0017	Reserved
Mechatrolink	4312	0018	Reserved
Mechatrolink	4312	001A	Reserved
Mechatrolink	4312	001B	Reserved
Mechatrolink	4312	001C	Reserved
Mechatrolink	4312	001D	Reserved

Mechatrolink	4312	001E	Reserved
Mechatrolink	4312	001F	Reserved
Mechatrolink	4312	0022	Reserved
Mechatrolink	4312	0023	Reserved
Mechatrolink	4312	0024	Reserved
Mechatrolink	4312	0025	Reserved
Mechatrolink	4312	0026	Reserved
Mechatrolink	4312	0094	Reserved
Mechatrolink	4312	0095	Reserved
Mechatrolink	4312	0096	Reserved
Mechatrolink	4312	00E5	Reserved
app	4401	0008	Each call to groupAxes() must be matched by a corresponding call to ungroupAxes(). If a script exits without such a matching call (thus leaving an 'orphaned' group behind), this warning is issued. Clearing the warning also ungroups the orphaned group.
app	4401	0009	The debug stack trace was longer than expected. It may be clipped.
app	4403	0001	The event queue for the remote client was full, and an event was dropped. This is generally caused either by exceeding the network bandwidth or exceeding the general system processing power (starving the connection). When an event is dropped in this manner, the connection is terminated.
app	4403	0005	An RMI connection was attempted by an external client and rejected due to the concurrent connection limit.
app	4407	0001	The configuration file directory is read-only or resides on a read-only file system. Attempts to update the configuration or create directories will fail.
app	4407	0002	An attempt was made to write to a read-only configuration file. The write failed.
app	4407	0105	There was an indication that the SRAM battery backup power may have failed temporarily. SRAM data may have been compromised.
app	4408	0001	The alarm history was configured to use NVRAM storage, but either the available NVRAM was not sufficient to contain the configured buffer size, or the configured buffer size was not large enough to contain the configured number of records. The alarm history will contain fewer records than configured.
app	4408	0002	The alarm history was configured to use NVRAM storage

			and the data was found to be corrupted. The alarm history has been lost. NOTE: this alarm also occurs if the configured size of the alarm history has been changed.
app	440A	000C	The position and torque scales specified in the configuration file have different signs. As a result, a positive acceleration results in a negative torque, and position limits are opposite in sign as the torque limits.
app	440A	000F	The axis was temporarily disconnected from the servo network during reset. During this time, the feedback data is not valid and the axis cannot be moved.
app	440A	0011	The network I/O was temporarily disconnected from the servo network during reset. During this time, any network I/O state change will be unobservable to the controller.
app	440A	0019	The system was rebooted by the user.
app	440A	001A	The system failed to shut down gracefully during a reboot, although the reboot did occur. This does not necessarily indicate that the software is damaged.
app	440B	0001	The controller is running out of memory. Memory should be freed as soon as possible. Try closing connections to the controller or stopping scripts.
app	440B	0003	The largest free memory block is approaching the critical level. Memory should be freed as soon as possible. Try closing connections to the controller or stopping scripts.
app	440C	0105	Reserved
app	440C	1032	The configuration file version is not compatible with the fimware version. Please use the configuration tool to update the configuration files to match the the firmware version.
app	440C	1034	Some function blocks are not supported by the controller firmware. If these function blocks are used in the PLC application, then their ErrorID will always equal 60909. If these function blocks are needed, then please upgrade the controller's firmware.
app	4501	0000	A user script task posted a warning directly.

Function Block ErrorID List

ErrorID	Name	Description
0	NoError	No Error
1	TimeLimitExceeded	Time limit exceeded.
2	DistanceLimitExceeded	Distance limit exceeded.
3	TorqueLimitExceeded	Torque limit exceeded.
Motion State Errors		
4368	MotionError	General motion error
4369	MotionQueueFull	The move could not be buffered because the axis motion queue is full. 16 moves is the maximum which can be buffered.
4370	MotionProhibited	The move could not be started because motion is prohibited. MC_Stop.Execute might be held high, preventing motion. If MC_Stop has control of the axis, no other function block can override the "Stopping" state. Other blocks that try to cause motion while MC_Stop has control of the axis will generate this error. Also verify that the limit switches are not active by checking the Global Variables for the axis.Also, a motion block may be attempting to abort an MC_TorqueControl move.
4371	EnabledFailed	The servo drive failed to enable or disable. Check the amplifier wiring for L1 / L2 / L3. The amplifier could be e-stopped or have an alarm.
4372	Drive parameter read/write did not complete.	Drive parameter read/write did not complete.
4373	Drive parameter read/write failed.	Drive parameter read/write failed
4374	TorqueMoveProhibited	Torque move prohibited while non-torque moves queued or in progress.
4375	NoCamForCamOut	CamOut called while not camming.
4376	MasterNotSet	The master slave relationship can not be modified because the master axis has not been set yet.
4377	CamTableSelectInProgress	File reading already in progress
4378	InvalidAxis	The function block is not applicable for the external axis specified
4379	HomingSequenceInProgress	A homing sequence is already in progress.
4380	SetPositionWhileMoving	MC_SetPosition can not be executed while the axis is

		moving.
4381	AxisAlarm	Motion aborted due to axis alarm.
4382	SetPositionRangeError	When the axis is in rotary mode, and the MC_SetPosition tries to set a position that is equal to or greater than the MachineCycle, this error is generated, and the position is not set.
4383	HomingFailed	Axis must be commanded at standstill when homing is attempted.Refer to the Motion State Diagram and MC_ReadStatus
4384	ClearAlarmsInProgress	Clear alarms already in progress
4385	AxisResetInProgress	Axis reset is already in progress.
4386	MechatrolinkResetInProgress	MECHATROLINK reset is already in progress.
4387	CamStructSelectInProgress	Already copying cam data (If Execute transition to TRUE while Busy = TRUE)
4388	ReadCamTableInProgress	CamTableRead can not write a second cam structure while the first cam structure is being written.
4389	WriteCamTableInProgress	CamTableWrite can not write a second cam structure while the first cam structure is being written.
4390	SetPositionProhibited	Position cannot be defined while the axis is the cam master of other axes.
4391	VirtualAxisNotAllowed	The function block can not be used with a virtual axis.
4392	InverterAxisNotAllowed	The function block can not be used with an inverter axis.
4393	ParameterFileInProgress	Y_VerifyParameters and Y_WriteParameters can not be called a second time while the first one is in progress.
4394	UnableToAddPositionMonitor	More than 10 Y_CamIn, Y_CamOut, or MC_GearInPos function blocks for a given axis are active at the same time. Most likely the application program is not coded correctly, and the Execute input is being fired too frequently.
4395	InvalidPositionMonitor	Window parameters are outside of the cams Machine Cycle. (0 to Prm1502, the last master position in the active cam table.)
4396	AxisLatchFunctionInUse	Axis latch function already in use.
4397	FailedToMoveAwayFromOT	Over travel limit still ON after attempting to move away from it.
4398	CamShiftNotPossibleWithinRange	The cam shift is not possible with EndPosition and current master position.This error occurs if the shift is greater than the distance to the end of the window. For example: shift = 90, window [180,360],

		and the master position = 300 when Y_CamShift.Execute=TRUE.
4399	NoDrivePower	The L1 / L2 / L3 power inputs on the drive may not be supplied with power, possibly due to an E-Stop condition.
4400	HardwareBaseBlock	The safety input (HBB on the CN8 connector) is preventing the drive from enabling.
4401	AxisUnavailable	The controller cannot communicate with the axis. It may be disconnected from the network.
4402	ExternalAxisRequired	The scan compensation delay parameter 1305 is only valid for external encoders.
4403	HighSpeedOutputNotSupported	The High Speed Output functionality is only available on external encoders.
4404	NotGearing	Can not execute MC_GearOut because axis is not in gear
4405	CamOutCanceled	Y_CamOut was aborted.
4406	UnsupportedContinuousLatch	Continuous Latch Mode is only supported on Sigma V servopacks.
4407	InternalBufferOverflow	Continuous latch buffer exhausted
4408	PatternOutofRange	Invalid pattern size or count
4409	PrmWriteInProgress	Parameter write already in progress.
4410	ReadOnlyParameter	Parameter is read-only.
4411	Parameter read already in progress.	The function block cannot be re-executed while it is in progress
4412	Parameter not supported for this axis.	Parameter not supported for this axis.
4413	StepperAxisNotAllowed	The Stepper axis does not support the mode of motion commanded
4414	MECHATROLINKCommsLost	MECHATROLINK Communications to the drive was disrupted. Execute MC_Reset to restore the connection.
4415	Reboot is already in progress	Reboot already in progress
Invalid Structure Values		
4624	InvalidStructureValue	RESERVED
4625	InvalidAxisID	Axis ID does not correspond to an axis configured on the system. Verify the value of AxisNum matches a logical axis number in the configuration. Tip: Make sure AXIS_REF is properly declared as a VAR or VAR_GLOBAL in all relevant POUs.
4626	InvalidMasterSlave	The master slave relationship is defined. A slave cannot be a master to another axis.

4627	InvalidInput	The input reference does not correspond to real input
4628	InvalidOutput	The output reference does not correspond to a real output
4629	InvalidIONumber	The input/output number does not correspond to a real input or output bit
4630	InvalidTrigger	Trigger or pattern reference is not valid
4631	InvalidCamSwitch	The cam switch structure in not valid
4632	InvalidCamSwitch	The track structure is not valid
4633	InvalidTableSize	Table size results in misaligned data. Refer to the help section "Internally Created Cam Data." A cam table will have a multiple of 16 bytes if created correctly.
4634	InvalidBufferSize	Buffer size results in misaligned data
4635	UnsupportedTableType	Table type is not supported
4636	InvalidStartIndex	Invalid start index.
4637	InvalidEndIndex	Invalid end index
4638	BufferOverrun	User Buffer Full.
Invalid Enumeration Type Errors		
4640	InvalidEnumerationType	RESERVED
4641	InvalidBufferMode	Buffer mode does not correspond to a valid enumeration value.
4642	InvalidDirection	Direction does not correspond to a valid enumeration value.
4643	InvalidStartMode	Start mode does not correspond to a valid enumeration value.
4644	InvalidShiftMode	Invalid shift mode.
4645	InvalidOffsetMode	Offset mode does not correspond to a valid enumeration value.
4646	InvalidMode	Mode does not correspond to a valid enumeration value.
4647	InvalidSynchMode	The synch mode does not correspond to a valid enumeration value.
4648	InvalidParameter	The parameter number does not exist for the specified axis
4649	InvalidAdjustMode	Invalid adjust mode
4650	InvalidRampInType	'RampIn' does not correspond to a valid enumeration value.
4651	InvalidControlMode	'ControlMode' does not correspond to a valid enumeration value.
4652	InvalidEndMode	Y_CamOut only supports "AtPosition"

Input Range Check Errors		
4656	RangeError	RESERVED
4657	NonPositiveDistance	Distance parameter is less than or equal to zero.
4658	NonPositiveVelocity	Velocity parameter is less than or equal to zero.
4659	NonPositiveAcceleration	Acceleration is less than or equal to zero.
4660	NonPositiveDeceleration	Deceleration is less than or equal to zero.
4661	NonPositiveTorque	Torque is less than or equal to zero.
4662	NonPositiveTime	Time is less than or equal to zero
4663	NegativeTime	Specified time was less than zero.
4664	NonPositiveScale	Specified scale was less than or equal to zero.
4665	NegativeVelocity	Velocity parameter is negative.
4666	ZeroDenominator	Denominator is zero.
4667	NonPositiveJerk	Jerk is less than or equal to zero.
4668	NonPositiveTorqueRamp	Torque Ramp is less than or equal to zero.
4669	InvalidEngagePosition	Engage position is outside the cam table domain.
4670	InvalidEngageWindow	Engage window is less than zero.
4671	InvalidDisengagePosition	Disengage position is outside the cam table domain.
4672	NegativeDisengageWidth	Negative Disengage Window
4673	InvalidStartPosition	StartPosition is outside of master's range.
4674	InvalidEndPosition	EndPosition is outside of master's range.
4675	InvalidFilterTC	Axis filter time constant out of range, or an attempt to change the value was made while the axis was enabled. (The axis must disabled to change the moving average time constant.)
4676	InvalidScanCompensationDelay	The time value must be within 0 to 10 MECHATROLINK cycles.
4677	InvalidArraySize	Array size is too large
4678	InvalidBufferArrayIndex	Buffer array index out of range
4679	InvalidDateOrTime	Invalid date or time values entered.
Invalid Input Data Errors		
4880	InvalidInputData	RESERVED
4881	InvalidPn	The specified Pn does not exist.
4882	InvalidTrackMask	The mask does not correspond to valid tracks.
4883	InvalidProfile	The profile must start with relative time equal to zero, and the time must be increasing.
4884	UnknownCamFile	The specified cam file does not exist.
4885	InvalidCamFileHeader	Invalid header for the cam file. You must first populate the TableType and DataSize in the

		Y_MS_CAM_STRUCT before executing the function.
4886	InvalidCamTableFormat	The first (master) column must be either increasing or decreasing. If the master data is incremental, even the very first point cannot be zero.
4887	InvalidCamRef	CamTableID does not refer to a valid cam table.
4888	ExceededSynchTime	The engage phase exceeded the time limit. Slave axis could not attain the target position and velocity within the user specified time limit.
4889	ExceededSynchDistance	The engage phase exceeded the distance limit. Slave axis could not attain the target position and velocity within the user specified master distance.
4890	InvalidWidth	Invalid width input. Width is an enumeration type with the following allowable values 'WIDTH_8'=0, 'WIDTH_16'=1, and 'WIDTH_32'=2.
4891	IdenticalMasterSlave	The slave axis can not be the same as the master axis.
4892	NoDefaultParameterInfo	Default drive parameter info is not available for this parameter.
4893	InvalidExternalAxis	The specified external axis may not be used. A physical axis is required
4894	InvalidVirtualAxis	The specified virtual axis may not be used with this function block.
4895	MissingOrUnknownFileExtension	Missing or unknown file extension
4896	FilenameDoesNotExist	Drive parameter filename does not exist.
4897	ParameterFileMismatch	The drive's model number or type does not match the parameter file.
4898	NoAxisFilter	No filter configured for axis.
4899	PosCompNotFound	Axis position compensation file not found.
4900	InvalidPosCompFormat	Invalid axis position compensation file format.
4901	PosCompAxisEnabled	Cannot enable/disable axis position compensation while servo on.
4902	InvalidCompensationRange	Invalid compensation table wrap range.
Logging Errors		
8464	LogError	RESERVED
8465	DataLogAlreadySetup	The data log is already setup.
8466	DataLogNotSetup	The data log is not setup.
8467	InvalidUserLogItemHandle	Invalid handle for user log item.
8468	TooManyDataLogsInUse	Data log can not be created because too many data logs are in use.
8469	InvalidDataLogHandle	Invalid handle for data log.

8470	ExceededMaxTypeCount	A user log item can only support eight inputs for each type.
8471	SaveLogFailed	Saving the log failed.
Communication Errors		
8705	MaxUserFDLimit	Exceeded maximum number of open user IO devices
8706	InvalidUserFDHandle	Invalid user IO device handle
8707	InvalidMasterSlave	Invalid IP address
8708	SocketCreateFailed	Socket initialization failed
8709	AddressInUse	The address is already in use on the local network
8710	AddressUnavailable	Address Unavailable
8711	AcceptFailed	Accept Connection Failed
8712	BindFailed	Bind Failed
8713	InvalidSocketType	Invalid Socket Type
8714	InvalidBindAddress	Invalid Bind Address
8715	ConnectSocketFailed	Connecting to the socket failed
8716	AddressUnreachable	The remote IP address is unreachable. Check the default gateway
8717	SocketAlreadyConnected	The socket is already connected
8718	ConnectionRefused	The connection was refused by the remote device
8719	SocketNotConnected	No TCP socket is connected
8720	DeviceOptionError	Device Option Error
8721	DeviceReadFailed	Reading from the comm device has failed
8722	DeviceWriteFailed	Device Writing has failed
8723	InvalidDeviceIOBuffer	Invalid Device IO Buffer
8724	InvalidDeviceOptionID	Invalid Device Option ID
8725	InvalidDeviceOptionValue	An option specified for the device is invalid
8726	InvalidSerialPort	The serial port specified is invalid
8727	SerialPortOpenFailed	The serial port specified could not be opened
Axis Errors		
40960	ServoPackAlarms	RESERVED
45332	ClearAlarmsFailed	Sending clear alarms command to servo drive failed.
45333	MechatrolinkResetFailed	
45334	MechatrolinkResetProhibited	Function cannot be utilized if there is a servo enabled or in motion on the network.
45335	AbsoluteEncoderInitializationFailed	Failed to initialize absolute encoder.
45336	DownloadInProgress	Function block could not be executed because a program download was in progress.
44337	Rebooting the controller is prohibited	Rebooting the controller is prohibited while an axis is

	while an axis is enabled.	enabled.
Operating System Error		
57616	ProConOSError	RESERVED
57617	NullInstanceObject	Instance object is NULL.
57618	NullInstanceData	The instance data is NULL.
57619	InvalidStructureCheckSum	The structure pointer check sum is invalid.
57620	InvalidStructureSize	The structure size does not match. This error may occur because data passed to an 'Axis' input on a PLCopen function block is not an AXIS_REF. If you have included a data element into a user structure which includes an AXIS_REF, be sure that the input to the function block is entered correctly.
57872	EclrErrorPrefix	RESERVED
57873	InvalidStructureSize	The structure size does not match.
57874	NullArgument	Argument data is NULL. The EngageData input must be connected.
Kernel Error		
60909	FunctionBlockNotSupported	Some function blocks are not supported by the controller firmware. If these function blocks are used in the PLC application, then their ErrorID will equal 60909. Upgrade the controller's firmware to eliminate this problem.
61712	KernelError	RESERVED
61713	InternalMotionKernelError	An internal motion kernel command failed. This error could be caused by inserting multiple MC_Power function blocks in the program for the same axis. Only one MC_Power function per axis is required. (Do not include more than one.)

Please refer to the following manuals for details regarding servo amplifier errors:

- Sigma II with NS115: SIEPC71080001, see section 9.3
- Sigma III: YEA-SIA-S800-11, see section 10.1.4
- Sigma-5 with rotary motor: SIEPS8000043, see Section 6.1
- Sigma-5 with linear motor: SIEPS8000044, see Section 6.1

Axis Parameter List

The following tables contain controller-side axis parameters which can be read or written using the function blocks MC_ReadParameter, MC_ReadBoolParameter, MC_WriteParameter, MC_WriteBoolParameter, and Y_ReadStringParameter. This is a comprehensive list that contains parameters that may not be applicable for all types of axes. For each parameter the following information is available:

Name	Prm	DataType	R/W	Default	Comments
ActualPosition	1000	LREAL	R	N/A	Feedback position in user units.
ActualPositionCyclic	1005	LREAL	R	N/A	Requires firmware version 1.0.6 or greater.
ActualPositionNonCyclic	1006	LREAL	R	N/A	Requires firmware version 1.0.6 or greater.
ActualTorque	1004	LREAL	R	N/A	Feedback torque in percentage of rated torque.
ActualVelocity	1001	LREAL	R	N/A	Feedback velocity in user units per second.
AmplifierModel	1819	STRING	R	N/A	Amplifier model number.
AxisType	1810	LREAL	R	N/A	Returns a code corresponding to the type of axis configured. 0=Servo, 1=VFD, 2=External Encoder, 3=Virtual, 4=Stepper
BufferedMotionBlocks	1600	LREAL	R	N/A	The number of motion blocks buffered in the motion queue. This value will increase when a motion block is executed with any of the non aborting types and decrement as each buffered block has control of the motion.
CamMasterCycle	1512	LREAL	R	1.0	If the axis is currently linked to a master axis for camming, this parameter indicates the cam master cycle as determined by the Cam Table currently in use. This parameter is not valid until Y_CamIn.Execute occurs. If Y_CamIn is executed while camming is already in

					operation, this parameter will not update until Y_CamIn.InSync is TRUE.
CamMasterPosition	1500	LREAL	R	N/A	This is a copy of the master position plus scan compensation added. See the Camming Block Diagram.
CamMasterScale	1510	LREAL	R	N/A	See the Camming Block Diagram.
CamMasterShift	1511	LREAL	R	N/A	This value holds the cumulative value of all previous Y_CamShifts. To reset this parameter to zero, execute Y_CamShift with Y_CamShift.PhaseShift equal and opposite of the value in this parameter. See the Camming Block Diagram.
CamMasterShiftedCyclic	1502	LREAL	R	N/A	This is the master position that is fed into the cam lookup function. It includes all adjustments made from the initial master position, such as scan compensation, it is modularized by the master data in the cam table, and any shift applied via Y_CamShift. See the Camming Block Diagram.
CamMasterShiftedPosition	1501	LREAL	R	N/A	See the Camming Block Diagram.
CamOffset	1531	LREAL	R	N/A	See the Camming Block Diagram.
CamOffsetRemaining	1533	LREAL	R	N/A	If a Y_SlaveOffset is in progress, this is the remaining amount of offset to be added to the absolute CamOffset (Parameter 1531), otherwise this value is zero.
CamScale	1530	LREAL	R	100.0	This a multiplication factor applied to the slave data. See the Camming Block Diagram.
CamScaleRemaining	1534	LREAL	R	N/A	If a Y_CamScale is in progress, this is the remaining amount of Scaling to be added to the absolute CamScale (Parameter 1530), otherwise this value is zero.
CamShiftRemaining	1513	LREAL	R	N/A	If a Y_CamShift is in progress, this is the remaining amount of phase shift to be added to the CamMasterShift (Parameter 1511), otherwise this value is zero.
CamState	1540	LREAL	R	N/A	See CamState in the Camming Overview

					section of this manual. 0 = Not Engaged, 1 = Waiting to Engage, 2 = Engaging, 3 = Engaged, 4 = Waiting to Disengage, 5 = Disengaging
CamTableCumulative Output	1521	LREAL	R	N/A	Initialized to 0 when the cam first engages and represents the total commanded slave distance traveled.
CamTableIDEngaged	1541	LREAL	R	0	Indicates the cam table currently in use by the motion engine. This number becomes valid when the CamState changes from 0 to 1. If a cam is already engaged (CamState = 3), this number becomes valid when the new table becomes engaged.
CamTableOutput	1520	LREAL	R	N/A	This is the table value selected as the synchronized position based on the master position. See the Camming Block Diagram.
CommandedAcceleration	1012	LREAL	R	N/A	Commanded acceleration in user units /second^2.
CommandedAcceleration Filtered	1022	LREAL	R	N/A	Commanded acceleration in user units /second^2 (Post S-curve filter.)
CommandedPosition	1010	LREAL	R	N/A	Commanded position in user units.
CommandedPositionCyclic	1015	LREAL	R	N/A	If axis is set to rotary type, this value reports the position from 0 to MachineCycle.
CommandedPosition NonCyclic	1016	LREAL	R	N/A	Reports the unmodularized commanded position regardless of whether the axis is configured as rotary or linear.
CommandedPosition NonCyclicFiltered	1020	LREAL	R	N/A	Commanded Position sent to the servopack (Post S-curve filter.) Refer to the Command Filtering (MP2300Siec/MP2310iec) and Command Filtering (MP2600iec) block diagrams for details
CommandedPosition SubFilter	1311	LREAL	R	N/A	Configures the servo amplifier to interpolate intermediate points in the motion profile between MECHATROLINK updates from the controller. This provides for a smoother motion profile. Settings are as follows: 0 = No interpolation; 1 = Exponential interpolation; 2 = Moving

					average filter.
CommandedTorque	1014	LREAL	R	N/A	Commanded torque in percentage of rated torque. Valid only when commanding a torque using MC_TorqueControl.
CommandedTorqueFiltered	1024	LREAL	R	N/A	Commanded torque in percentage of rated torque. Valid only when commanding a torque using MC_TorqueControl.
CommandedVelocity	1011	LREAL	R	N/A	Commanded velocity in user units / second.
CommandedVelocityFiltered	1021	LREAL	R	N/A	Commanded velocity in user units / second. (Post S-curve filter.)
ControllerFeedForward Enable	1310	BOOL	R/W	TRUE	Alternative of servo amplifier's Pn109 parameter. User can enable this gain in either the controller of servo amplifier. Both settings are not recommended simultaneously.
ExternalRawPositionCyclic	1007	LREAL	R	N/A	Applicable for External Encoder set in rotary mode only. Refer to the external encoder block diagram for details (Requires FW version 1.2.3 or higher)
ExternalRawPositionNon Cyclic	1008	LREAL	R	N/A	Applicable for External Encoder only. Refer to the external encoder block diagram for details. (Requires FW version 1.2.3 or higher)
ExternalVelocityUnfiltered	1009	LREAL	R	N/A	Instantaneous external encoder velocity. Refer to the external encoder block diagram for details. (Requires FW version 1.2.3 or higher)
FilterMovingAverage	1301	LREAL	R/W		This value represents the S-Curve time constant. The units are seconds, and the range is 0.0 to 5.0 (Zero exclusive). This parameter should only be written if the axis is in the "Standstill" state. Use MC_ReadStatus to verify.
FilterMovingAverageEnable	1300	BOOL	R/W		Apply S-Curve filter.
HighSpeedOutputEnable	1050	BOOL	R/W	FALSE	Set TRUE to arm or toggle to re-arm the external encoder high speed output.
HighSpeedOutputPosition	1052	LREAL	R/W	0.0	Set this value before the high speed output function is enabled.

HighSpeedOutputPositionNonCyclic	1053	LREAL	R/W	0.0	This is the user unit equivalent of the raw 32 bit encoder value set in the LIO hardware for high speed output compare.
HighSpeedOutputStatus	1051	BOOL	R	N/A	Status bit indicates when the hardware sets the high speed output and remains set until the function is disabled.
LatchPositionCyclic	1030	LREAL	R	N/A	Reports the modularized latch position. This value is only valid if the axis is configured as rotary.
LatchPositionNonCyclic	1031	LREAL	R/W	N/A	Reports the unmodularized latch position regardless of whether the axis is configured as rotary or linear.
LimitAccelEnable	1222	BOOL	R/W	TRUE	Enable the acceleration limit function.
LimitAccelNegative	1220	LREAL	R/W	-1.797693E+308	Negative acceleration limit
LimitAccelPositive	1221	LREAL	R/W	1.797693E+308	Positive acceleration limit
LimitDecelEnable	1232	BOOL	R/W	TRUE	Enable the deceleration limit function.
LimitDecelNegative	1230	LREAL	R/W	-1.797693E+308	Negative deceleration limit
LimitDecelPositive	1231	LREAL	R/W	1.797693E+308	Positive deceleration limit
LimitPositionEnable	1202	BOOL	R/W	TRUE	Enable the position limit function.
LimitPositionNegative	1200	LREAL	R/W	-1.797693E+308	Negative position limit
LimitPositionPositive	1201	LREAL	R/W	1.797693E+308	Positive position limit
LimitTorqueDefault	1400	LREAL	R/W	100.0	Default torque limit for blocks with a torque limit input
LimitVelocityEnable	1212	BOOL	R/W	TRUE	Enable the velocity limit function.
LimitVelocityNegative	1210	LREAL	R/W	-1.797693E+308	Negative velocity limit
LimitVelocityPositive	1211	LREAL	R/W	1.797693E+308	Positive velocity limit

LoadType	1807	BOOL	R	N/A	0=Linear, 1=Rotary; as set in the Hardware Configuration
MachineCycle	1833	LREAL	R	N/A	If the LoadType is set for Rotary operation, this is the rollover position. If the load type is set for Linear, this value has no meaning. Firmware Version 1.2.2 is required.
MechatrolinkCompensation	1307	BOOL	R	TRUE	Only applicable for camming and gearing modes and for MECHATROLINK axes. This value determines if scan compensation is calculated to account for the network delay when sending commanded positions to the amplifier. Its purpose is to eliminate master / slave phase lag due to the time required to send the position data to the amplifier. Firmware Version 1.2.3 is required. Refer to the Camming Block Diagram.
MotorModel	1823	STRING	R	N/A	Motor model number. Requires FW 2.0. Use Y_ReadStringParameter in YMotion firmware library
NodeStatus	1330	BOOL	R	N/A	Indicates if the node (drive) is powered up and the MECHATROLINK cable is properly connected. TRUE = Node is communicating.
OptionMonitor	1312	LREAL	R	N/A	Returns the value of the servo amplifier's Un Monitor as selected by Pn825 according to the MECHATROLINK Communication Manual SIEPS80000054, section 5.7.3. For example, to read the RMS torque output, set Pn825 to UINT#16#19. On MP2300iec controllers, firmware version 1.2.2 is required. On the MP2600iec, firmware version 2.1.0 is required.
PositionCompensated CommandPosition	1020	LREAL	R	N/A	Commanded position output from the position compensation function
PositionCompensation Enable	1308	BOOL	R/W	FALSE	Enables/disables position compensation mode

PositionError	1130	LREAL	R	N/A	Position Error, following error, or difference between commanded and actual position in user units
PositionScalePerTick	1830	LREAL	R	N/A	User units per encoder count
PositionScalePerRev	1831	LREAL	R	N/A	User units per motor revolution
ScanCompensation	1305	LREAL	W	2 scans	For external encoders only. This value provides scan compensation to ensure the master and slave remain synchronized even at high speeds. Units are in seconds. The default of 2 scans was predetermined by Yaskawa and should not need adjustment in most cases. The maximum compensation is 10 MECHATROLINK scans. For example, if the MECHATROLINK update is 2 ms, then parameter 1305 can range from 0 to 0.020000 seconds). Firmware Version 1.2.2 is required. Refer to the Camming Block Diagram.
VelocityFilter	1306	LREAL	R/W	0.0	Provides a moving average filter for the feedback velocity over a specified time period. Units are in seconds. Note that the time value will be rounded to the nearest number of MECHATROLINK or DPR scans. For example, if the MECHATROLINK is set for 2.0 ms, and the VelocityFilter is set to 0.010, then the velocity will be averaged over 5 samples. The maximum filter time is 0.100 seconds.

High Speed Output

Firmware Version 1.2.2 is required for high speed output support. The option card LIO-01 (DO_01), LIO-02 (DO_01), LIO-06 (DO_07), and the MP2600iec (DO_07) have the capability to set an output at the hardware level within 13 μs based on a position compare value.

Notes

• The output remains ON from the position specified by HighSpeedOutputPosition until HighSpeedOutputEnable is set FALSE.

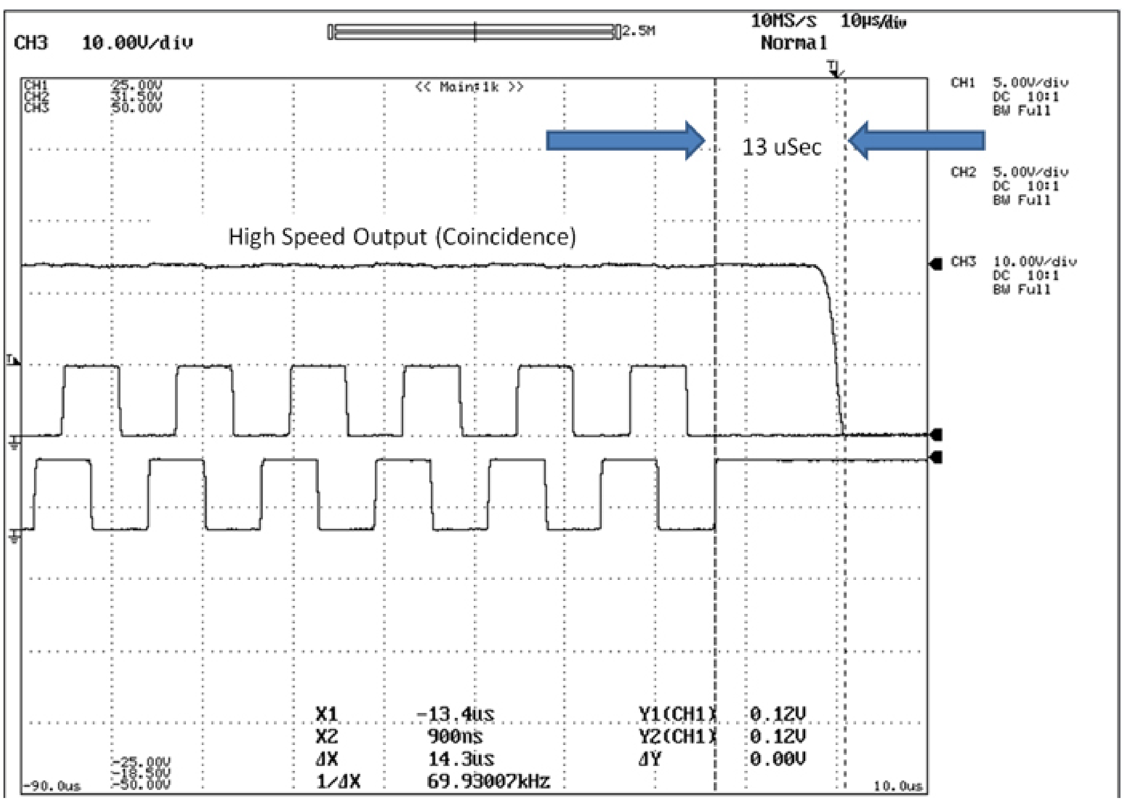
• If the axis is rotary type, then value must be within the MachineCycle. If the value must be set outside of the MachineCycle range, use HighSpeedOutputPositionNonCyclic. The latter of HighSpeedOutputPosition or HighSpeedOutputPositionNonCyclic being set by the application program will be used as the high speed output position. This allows the capability for rotary axis applications to set a value which may be several machine cycles away.

• While this function is enabled, the application program cannot control the associated output directly, nor monitor its state by referring to its global variable. Use the HighSpeedOutputStatus parameter instead to monitor its state.

High Speed Output Quick Reference

Device	Output Number	Pin Number	Software Default Name
LIO-01	DO-01	A14	M□□_DO_01
LIO-02	DO-01	A14	M□□_DO_01
LIO-06	DO-07	49	M□□_DO_07
MP2600	DO-07	44, 49	MO1_DO_01

Timing Diagram



Camming

Camming Introduction

At its core, an electronic cam is simply a list of master and slave positions that describe the synchronized relationship of two axes. For a given master position, the slave is commanded at the corresponding position in the table. Surrounding this core are many functional elements, including methods to load cam data, configuration for the type of data, engage & disengage methods, on-the-fly adjustments, and the possibility to switch cam tables on the fly.

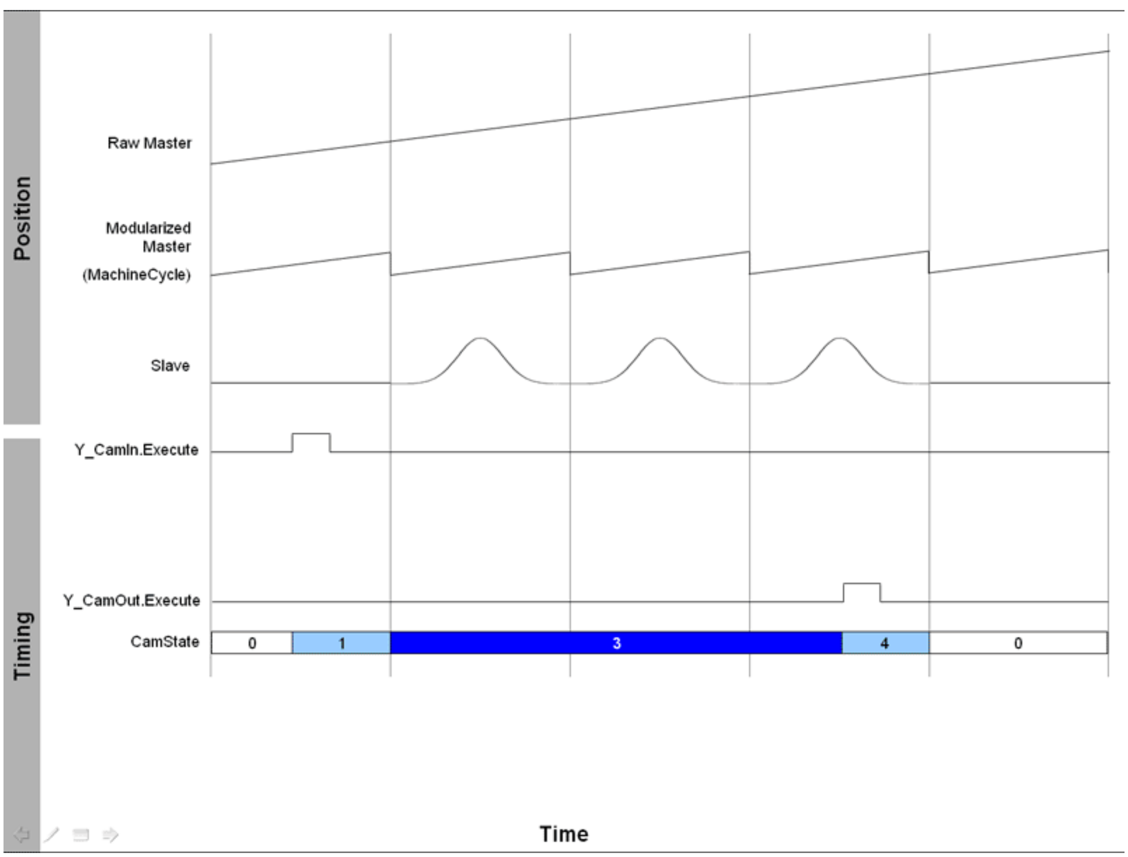
CamState

Similar to the Motion State Diagram for general motion, the camming mode has a CamState, parameter 1540. This value indicates the slave's current mode of operation, and is very useful for debugging and program logic flow. Possible values are:

CamState	Meaning
0 = Not Engaged	Axis is not involved in a cam operation
1 = Waiting to Engage	Y_CamIn has been executed, but the slave is not yet following the master because it has not passed into the engage window.
2 = Engaging	The very short time the master is within the window and the slave is moving to the very first commanded cam position.
3 = Engaged	The slave's commanded position is dictated by the cam function as the master moves through the

	data points.
4 = Waiting to Disengage	Y_CamOut has been executed, but the slave is still following the master because it has not traveled to the disengage window.
5 = Disengaging	The very short time the master is within the window and the slave is moving to the very last commanded cam position.

States 2 & 5 are special cases which may only become active if the window is set very large for engaging or disengaging, or if the slave axis is faulted an cannot achieve the first cam point or final position. These states are only active when the master is in the window. The following graphic details the behavior of the Cam mode.



Cam Masters

An external encoder connected to an LIO card, virtual master or Mechatrolink motion axis can be a cam master. The master is selected by connecting it's AXIS_REF to the Master input on Y_CamIn, Y_CamShift, Y_CamScale, or Y_SlaveOffset.

Master Cycle

The master cycle of the cam is typically identical to the MachineCycle of the master, although this is not required as some applications benefit from the ability to operate a cam cycle over multiple cycles of the master axis. The slave's cam master cycle is available on the output of Y_CamFileSelect or a cam slave's parameter 1512.

It is essential to understand that the cam function operates on a master cycle which is determined by the "Last Master Position" in the cam table. This cycle may be different than the MachineCycle set in the Hardware Configuration. In this way, two positions for the master can be realized.

Sewing Machine Application Example:

Consider a reciprocating needle axis which is the cam master, and must have the ability to stop at top dead center. The cam motion for the machine creates stitches and may require any number of cycles of the needle axis to complete one stitch pattern.

Hardware Configuration: Needle axis Machine Cycle = 360.0. Cam table's last master position may be any number of required needle cycles, or 360 * Stitches. If the cam is executed in non periodic mode (Y_CamIn.Periodic=FALSE, the cam slave will follow the master for the entire length of master travel as defined in the Cam table, which will be 360 * Stitches.

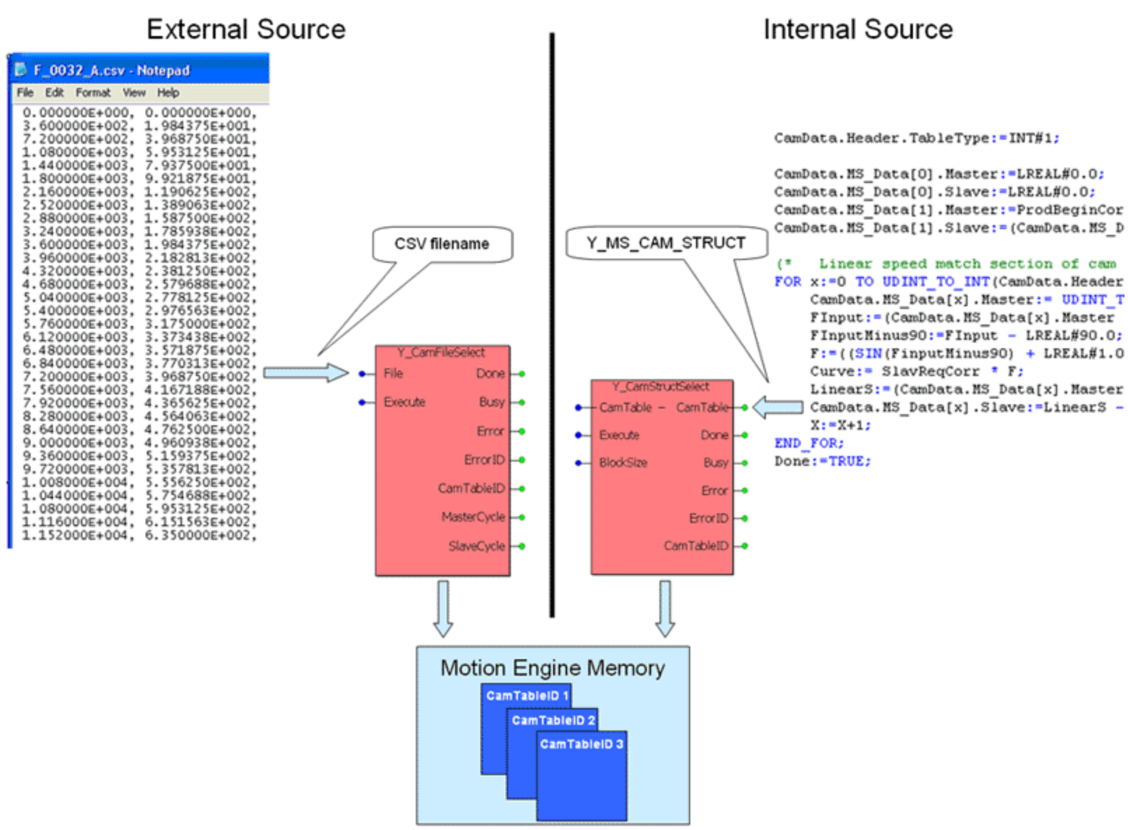
Camming Function Blocks

MPiec camming functionality consists of 10 function blocks:

Cam Data Management	Cam Engagement	On The Fly Adjustments	Cam Data Transfer
Y_CamFileSelect	Y_CamIn	Y_CamShift	Y_ReadCamTable
Y_CamStructSelect	Y_CamOut	Y_CamScale	Y_WriteCamTable
Y_ReleaseCamTable		Y_SlaveOffset	

Creating a Cam Table

There are two basic methods of creating cam files, externally and internally. The cam data must be loaded into the motion engine before it can be used. When cam data is loaded with either the Y_CamFileSelect or Y_CamStructSelect function blocks, a CamTableID is returned, which will be used by other camming blocks to reference the cam table. Many cam tables may be loaded into the Motion Memory. There is no specific limit on the number of files that can be loaded, the limit is available memory based on each table size. The following graphic provides an explanation of these to file loading methods.



Externally Created Cam Data

A cam table can start as an excel workbook, or within Yaskawa's Cam Tool Software, or other cam generation software. It must be converted to a CSV file for transfer into the MPiec motion engine memory. The CSV may contain integer or floating point data in the same position units as defined for each of the axes in the application via the configuration software.

Tips when using Cam Tool:

1) On the Set Style screen, select "No Unit" for the Phase & Position (Master & Slave.) This makes it possible enter data in the same units as already specified with the Configuration software. None of the Cam Tool data will be converted to other units as suggested in the lower portion of the Set Style screen, so no other information on the Set Style screen must be entered, unless you want Cam Tool to show motor performance characteristics at various points in the curve.

2) Save the data as a CDT or CDD file, which will preserve the original cam data, such as curve type, for future edits. Once the file is saved in a native Cam Tool format, then also save the file as a CSV.

NOTE: CSV files must not be Unicode format. If using excel or other software, check for this setting. Excel has a few CSV settings, select MS-DOS CSV output.verification.

Transferring Cam Files to an MPiec Controller

Cam files and other user data files can be stored on the controller. There are several ways to load the files on the controller, including:

1. MotionWorks IEC Resource Dialog window. (MP2300Siec and MP2310iec only)

2. Hardware Configuration Utilities Menu

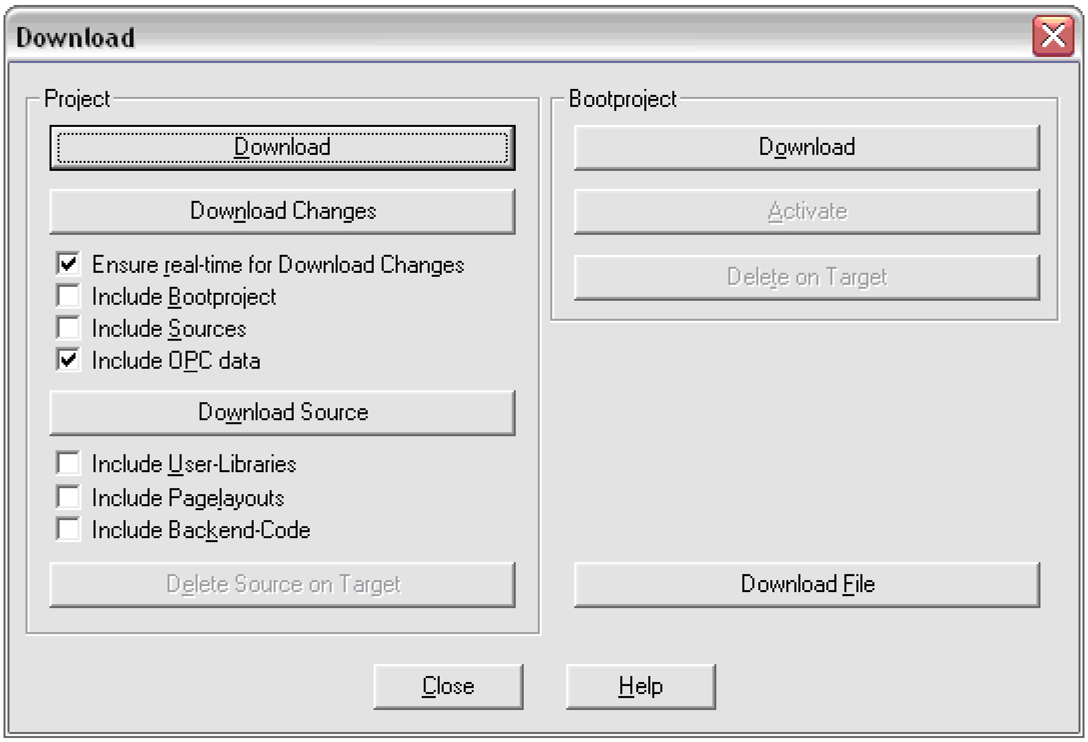
3. Web Server Project Archive Page (firmware 2.2.1 or higher is required for added functionality of adding user data files individually.)

4. Write a PC application to send files to the controller using the HTTP POST method.

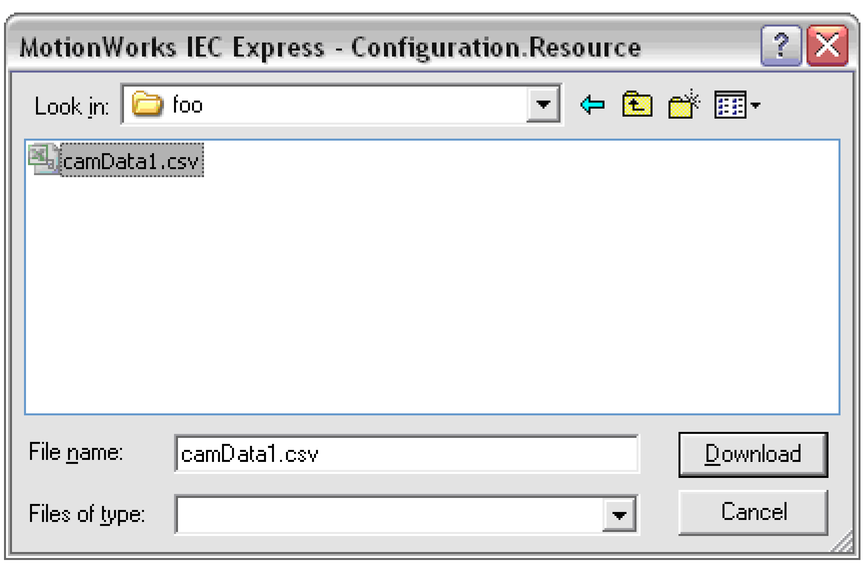
Downloading with MotionWorks IEC

In MotionWorks IEC, launch the “Project Control Dialog” shown below.

Press the “Download” button to launch the “Download” dialog as shown below.



Press the “Download File” button.



Browse to the desired directory, select the cam file, and press the "Download" button. This places the file /flash/procon/any directory on the controller. This is the default directory for the Y_CamFileSelect function block, so any cam file downloaded with this procedure can be selected simply by using the file name in a string without referencing the directory.

Using C#

Cam files can be downloaded to the controller programmatically using a C# API that sends a file via an HTTP POST. While programming within C#, the key steps are:

- 1. Create a Yaskawa.IEC61131.RMI.Modules.Controller object.
- 2. Connect to the controller using Controller.Connect()
- 3. Access Yaskawa.IEC61131.RMI.Modules.IConfigUtility via the Controller object
- 4. Call IConfigUtility.UploadFileToUserDirectory()

MotionWorks IEC Programming After Files Have Been Sent

The Y_CamFileSelect function block can access files in the /flash/user/data/cam and /ramdisk/user/data/cam directories, but the directory "data/cam/" or "tempData/cam" must be added to the filename string. For example, if Y_CamFileSelect.Filename = "data/cam/profile1.csv", then Y_CamFileSelect loads /flash/user/data/cam/profile1.csv.

User File Storage

User data files can be stored in FLASH or SDRAM. Cam files are copied to SDRAM for use by the motion engine via the Y_CamFileSelect function block.

Configuring FileName Input for Y_CamFileSelect

The table below summarizes where files are placed on the controller based on download method and how to access the file via the Y_CamFileSelect function block.

Download Method		C# "directory " Argument	Path prepended to Y_CamFileSelect.FileName	Location On Controller
MotionWorks IEC Resource Dialog Window		n/a	(none)	/flash/procon/any/
MotionWorks IEC Hardware Configuration		n/a	data/cam	/flash/user/data/cam/
C#	FLASH	data/cam	data/cam	/flash/user/data/cam/
	RAMDisk	tempData/cam	tempData/cam	/ramdisk/user/data/cam/

Y_CamFileSelect.Filename Examples

From MotionWorks IEC Resource Dialog Window

- File: Profile1.csv
- File path on controller: /flash/procon/any/Profile1.csv
- Y_CamFileSelect.Filename='Profile1.csv'

From MotionWorks IEC Hardware Configuration Online Utilities Menu

- File: Profile1.csv
- File path on controller: /flash/user/data/cam/Profile1.csv

- Y_CamFileSelect.Filename='data/cam/Profile1.csv'

C# upload to flash

- File: Profile1.csv
- Directory argument for UploadFileToUserDirectory() = “data/cam”
- File path on controller: /flash/user/data/cam/Profile1.csv
- Y_CamFileSelect.Filename='data/cam/Profile1.csv'

C# upload to ramdisk

- File: Profile1.csv
- Directory argument for UploadFileToUserDirectory() = “tempData/cam”
- File path on controller: /ramdisk/user/data/cam/Profile1.csv
- Y_CamFileSelect.Filename='tempData/cam/Profile1.csv'

Internally Created Cam Data

Cam tables can be calculated within the application. The DataType called Y_MS_CAM_STRUCT specified in the MotionBlockTypes DataType worksheet must be used in conjunction with the Y_CamStructSelect function block. The structure contains two headers to accommodate future cam file formats.

Notice that the DataSize value shown at the right has a value of 2880. This is the actual size of the cam table in bytes. Since each element (Master or Slave position value) is an LREAL, each cam point occupies 16 bytes. This means that the cam table shown below contains 2880/16, or 180 pairs. Also note that the value of DataSize must be less than or equal to the hard coded array size defined in the DataType definition for MS_Array_Type. The default size may be changed to accommodate larger cam tables if desired.

Y_MS_CAM_STRUCT			
Header [6 bytes]			
TableType		INT	
Reserved1		UINT	
DataSize		UDINT	
MS_Header [8 bytes]			
MasterIncremental		BOOL	
SlaveIncremental		BOOL	
Reserved1		UINT	
Reserved2		UINT	
Reserved3		UINT	
MS_Data [DataSize bytes]			
Master[0]	LREAL	Slave[0]	LREAL
Master[1]	LREAL	Slave[1]	LREAL
Master[2]	LREAL	Slave[2]	LREAL
Master[3]	LREAL	Slave[3]	LREAL
Master[4]	LREAL	Slave[4]	LREAL
Master[5]	LREAL	Slave[5]	LREAL
Master[6]	LREAL	Slave[6]	LREAL
Master[7]	LREAL	Slave[7]	LREAL
Master[8]	LREAL	Slave[8]	LREAL
Master[9]	LREAL	Slave[9]	LREAL
Master[10]	LREAL	Slave[10]	LREAL

Variable	Value	Type
MyCam		Y_MS_CAM_STRUCT
Header		Y_CAM_HEADER
TableType	0	INT
Reserved1	0	UINT
DataSize	2880	UDINT
MS_Header		Y_MS_HEADER
SlaveIncremental	FALSE	BOOL
MasterIncremental	FALSE	BOOL
Reserved1	0	UINT
Reserved2	0	UINT
Reserved3	0	UINT
MS_Data		MS_Array_Type
[0]		Y_MS_PAIR
Master	0.0000000	LREAL
Slave	0.0000000	LREAL
[1]		Y_MS_PAIR
Master	0.0000000	LREAL
Slave	0.0000000	LREAL
[2]		Y_MS_PAIR
Master	0.0000000	LREAL
Slave	0.0000000	LREAL
[3]		Y_MS_PAIR
Master	0.0000000	LREAL
Slave	0.0000000	LREAL
[4]		Y_MS_PAIR
Master	0.0000000	LREAL
Slave	0.0000000	LREAL

The following is an example of a structured text program that calculates a modified sine superimposed on a straight line.

```
RTRIG_Execute(CLK:=Execute);
IF RTRIG_Execute.Q THEN
    IF CamData.Header.DataSize<=UDINT#0 THEN
        Error:=TRUE;
        ErrorID:=UINT#8484;
        RETURN;
    END_IF;
    IF ProductLength<=LREAL#0.0 THEN
        Error:=TRUE;
        ErrorID:=UINT#8485;
        RETURN;
    END_IF;
    IF SlaveCycle<=LREAL#0.0 THEN
        Error:=TRUE;
        ErrorID:=UINT#8486;
        RETURN;
    END_IF;

    SlaveDia:=SlaveCycle/Pi;
    ProdBeginCorr:=ProductLength/LREAL#3.0;                     (*  Product Begin Correction   *)
    ProdEndCorr:=ProductLength;                                 (*  Product End of Correction (Product Length)  *)
    ProdCorrDist:=ProdEndCorr-ProdBeginCorr;                    (*  Product Start Correction  *)
    SlaveStartCorr:=(LREAL#1.0-((ProdEndCorr-ProdBeginCorr)/ProdEndCorr)) * SlaveCycle;    (*   Slave start correction
    SlaveEndCorr:=SlaveCycle;
    SlaveCorrDur:=SlaveEndCorr - SlaveStartCorr;                (*   Slave Correction Duration   *)
    SlavReqCorr:=ProdEndCorr/(SlaveCycle) * SlaveEndCorr - SlaveStartCorr;
    CamData.Header.TableType:=INT#1;                            (*  Master / Slave Table type *)
    (*   Linear speed match section of cam  *)
    CamData.MS_Data[0].Master:=LREAL#0.0;                       (*  First Master data point  *)
    CamData.MS_Data[0].Slave:=LREAL#0.0;                        (*  First Slave data point  *)
    CamData.MS_Data[1].Master:=ProdBeginCorr;
    CamData.MS_Data[1].Slave:=(CamData.MS_Data[1].Master / SlaveCycle) * SlaveCycle;
    (*   Tangent match section (modified sine superimposed on a straight line)  *)
    FOR x:=0 TO 359 DO
        CamData.MS_Data[x].Master:= UDINT_TO_LREAL( INT_TO_UDINT(x) / UDINT#360) * ProductLength;
        FInput:=(CamData.MS_Data[x].Master - ProdBeginCorr) * LREAL#180.0 / ProdCorrDist;
        FInputMinus90:=FInput - LREAL#90.0;
        F:=((SIN(FinputMinus90) + LREAL#1.0 ) / LREAL#2.0);
        Curve:= SlavReqCorr * F;
        LinearS:=(CamData.MS_Data[x].Master / (SlaveDia * Pi)) * SlaveCycle;
        CamData.MS_Data[x].Slave:=LinearS - Curve;
        X:=X+1;
    END_FOR;
    Done:=TRUE;
END_IF;
IF Execute=FALSE THEN
    Done:=FALSE;
    Error:=FALSE;
    ErrorID:=UINT#0;
END_IF;
```

Cam Table Types

The MPiec supports cam tables containing either relative or absolute data. The data is treated as absolute by default. If the data is incremental, the CSV file must contain the identifiers MasterRelative=TRUE, MasterRelative=FALSE in the first line. The file can contain incremental data for only the master or slave as necessary and by only including the proper identifier. The following is an example of a CSV that has incremental data for both the master and the slave. Each master value represents one full rotation of 360 degrees over which the slave moves the incremental amount of its user units as shown. When the master is in-between values in the table, the controller interpolates to find the appropriate position for the slave.

F_0200_R.csv - Notepad

File Edit Format View Help

```
MasterIncremental=TRUE,SlaveIncremental=TRUE
360,0.3460208
360,1.0380622
360,1.730104
360,2.422145
360,3.114187
360,3.806231
360,4.49827
360,5.19031
360,5.70934
360,5.19031
360,4.49827
360,3.80623
360,3.11419
360,2.42214
360,1.73011
360,1.03806
240,0.34602
360,0.13301
360,0.9095
360,2.35758
360,4.28172
360,6.42202
360,8.48945
360,10.20476
360,11.33631
360,11.73125
360,11.3364
360,10.2047
360,8.4895
360,6.422
360,4.2817
360,2.3576
360,0.9095
240,0.133
360,2.9412
360,2.9412
360,2.9411
```

On-The-Fly Adjustments

There are three types of on the fly adjustments that can be performed. Shift, Offset, and Scale. These adjustments are shown in the camming block diagram in this section. If the application has multiple slaves, note that adjustments are made only to individual slaves, as each slave has its own copy of the master data. For example, a CamShift for slave #1, will not affect Slave #2 unless the same function with the same shift amount is executed for Slave #2.

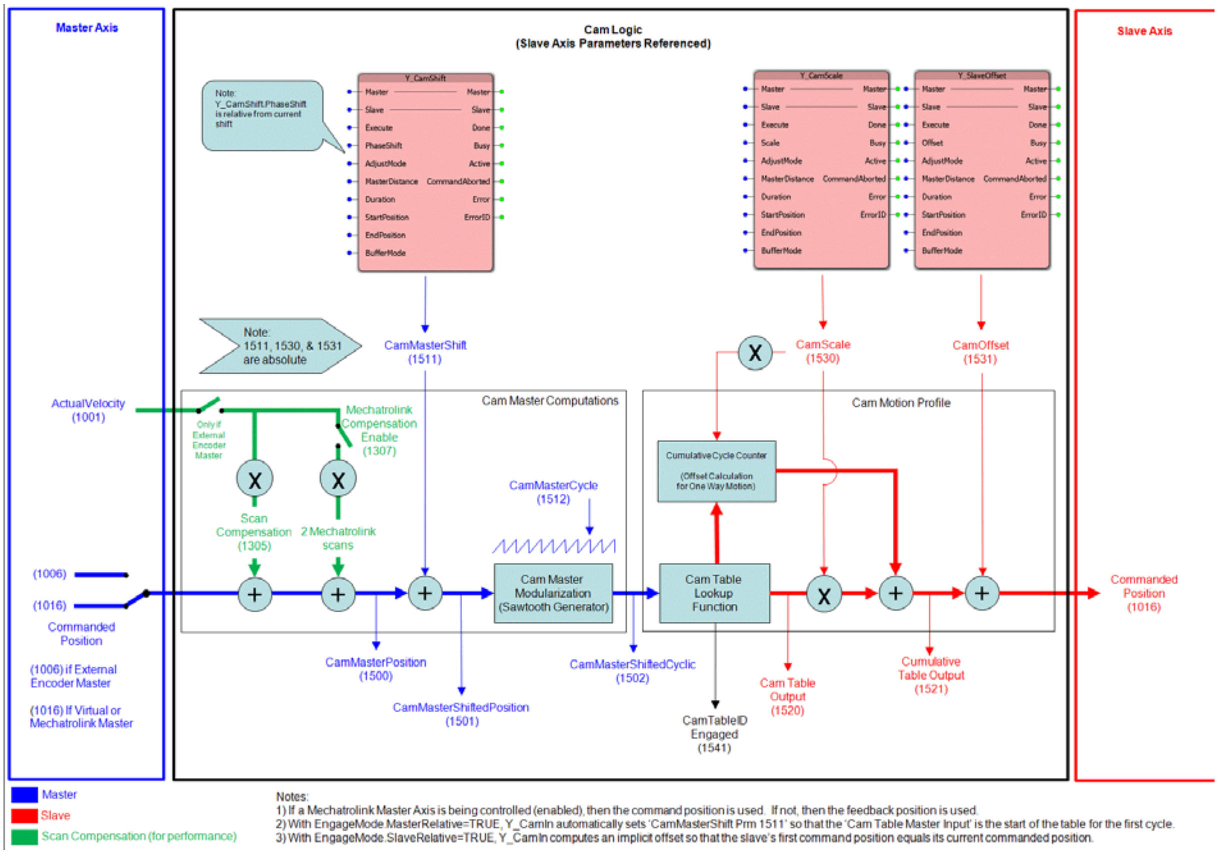
All adjustments can be made in three ways, based on the data configured in the Adjustmode input of the function.

- Over a relative change in position of the master
- Over time
- To start and complete between two specified master positions

If the master is outside the range when the block executes, the adjustment will wait until the master crosses into the range. If the master is already within the range when the block executes, the adjustment will stat immediately.

For all cases, the correction is governed by a modified sine progression from 0 to 100% of the correction. This provides a smooth for the slave..

Camming Block Diagram



Notes:

1) If Master Axis is being controlled, then the command position is used. If not, then the feedback position is used.

2) With EngageMode.MasterRelative=TRUE, Y_CamIn automatically sets 'CamMasterShift' so that the 'Cam Table Master Input' is the start of the table for the first cycle.

3) With EngageMode.SlaveRelative=TRUE, Y_CamIn computes an implicit offset so that the slave's command position starts at its current commanded position.

Engage / Disengage Window

The optimal Engage or Disengage Window requires knowing the fastest master speed at which the machine is expected to operate. The window must be large enough to ensure that the motion engine, executing at the MECHATROLINK update rate can see the master position within the window during one scan. Yaskawa recommends setting the window to the distance the master can travel in two scans.

1. Convert maximum expected master speed into the same units as configured for the master axis.
2. Determine the MECHATROLINK update rate or Dual Port RAM update interval (MP2600iec.) by looking in the Hardware Configuration.
3. Convert to find the distance the master can travel in one MECHATROLINK update interval.
4. Double the scan rate for safety margin.

Maximum expected speed: 500 ft/min.

Convert to configured user units of ft/sec: 500 / 60 = 8.333 ft/sec.

MECHATROLINK update rate = 2 mSec.

Invert update rate to find number of scans per second: 1000 / 2 = 500.

Maximum distance travelled per scan: 8.333 / 500 = 0.016666 feet.

Double for safety margin: 0.03333.

Setting a larger window is not recommended. At extreme levels approaching 50% of the cam master cycle or greater, the slave will seem to casually engage at any master position, and exhibit a jump in position when synchronizing with the master.

Cam Transitions Matrix

This chart describes the effect on various cam parameters and situations when given conditions occur.

Green indicates the event has no effect, the prevailing value listed along the top of the chart will be unaffected.

Yellow indicates that the event described at the left of the chart will impact the parameter listed at the top in some way.

* The master/slave relationship is defined the first time a Y_CamIn, Y_CamShift, Y_CamScale, or Y_SlaveOffset block executes, where first time is defined as being in the Null state when any of these function blocks execute. If a relationship is currently defined, then it is checked, and if inconsistent with the initial definition, the block produces an error (4633, Invalid master slave combination).

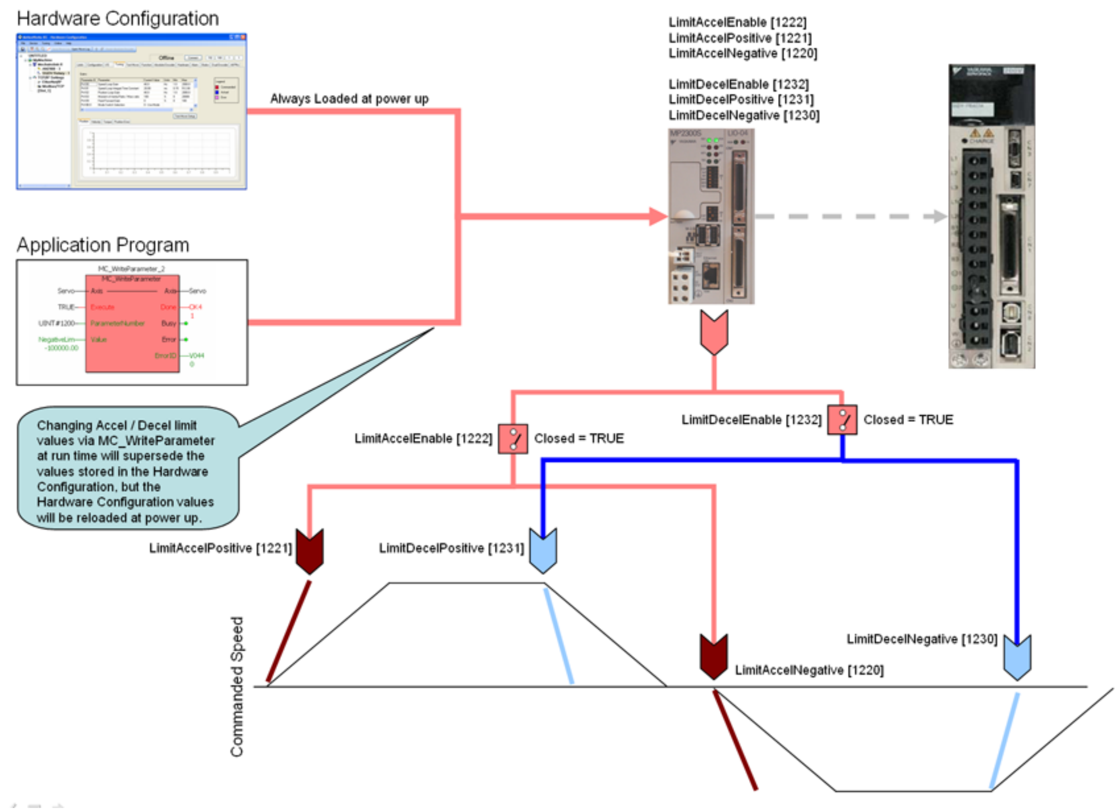
** Master/slave relationship is reset, i.e. the slave has no cam master.

Name	CamScale	CamOffset	CamShift	CamState	Master/Slave Pair
Parameter #	1530	1531	1511	1540	(Internal)
Event					
Cold Start	100	0	0	0	Null
Warm Start	100	0	0	0	Null
Power Up	100	0	0	0	Null
Hot Start	100	0	0	0	Null
Y_CamIn.Execute	Retained	Retained	Retained	Changes from 0 to 1	Defined or checked *
Y_CamIn.InSync	Retained	Retained	Retained	Changes from 1 to 3	Retained
Y_CamOut.Execute	Retained	Retained	Retained	Changes from 3 to 4	Retained
Y_CamOut.Done	Retained	Retained	Retained	Changes from 4 to 0	Retained
Y_CamShift.Execute	Retained	Retained	Starts change to new relative shift value	Retained	Defined or checked
Y_CamScale.Execute	Starts change	Retained	Retained	Retained	Defined or checked

	to new absolute scale value				
Y_SlaveOffset.Execute	Retained	Starts change to new absolute offset value	Retained	Retained	Defined or checked
(Master) MC_SetPosition	Retained	Retained	0	Retained	Retained
(Slave) MC_SetPosition	Retained	0	Retained	Retained	Retained
(Master) MC_Stop	Retained	Retained	Retained	Retained	Retained
(Slave) MC_Stop	100	0	0	0	Reset (Null) **
(Slave) MC_Reset	Retained	Retained	Retained	Retained	Retained
(Slave) MC_Power.Enable = FALSE	Retained	Retained	Retained	0	Retained
NextBlock.Active (Aborting)	Retained	Retained	Retained	0	Retained
NextBlock.Active (Buffered)	Retained	Retained	Retained	Retained	Retained

Motion Details

Acceleration / Deceleration Limits



Accel / Decel Limits

• The software acceleration & deceleration limits are managed by the MPiec controller.

• When an acceleration or deceleration limit is exceeded, a controller alarm will be generated, obtainable via the MC_ReadAxisError function block, or the web server.

• The controller alarm will be 16#3202 0005 if the positive position limit is exceeded and 16#3202 0006 if the negative position limit is exceeded.

Acceleration Limits

• Acceleration is defined as increasing velocity away from zero.

• The parameters are called LimitAccelPositive and LimitAccelNegative, with values of UINT#1221 and UINT#1220 respectively. Use the MC_WriteParameter function block for these and all controller side parameters. Acceleration limit parameters are in user units / sec2.

• To disable the acceleration limit, set LimitAccelEnable, parameter 1222 to zero.

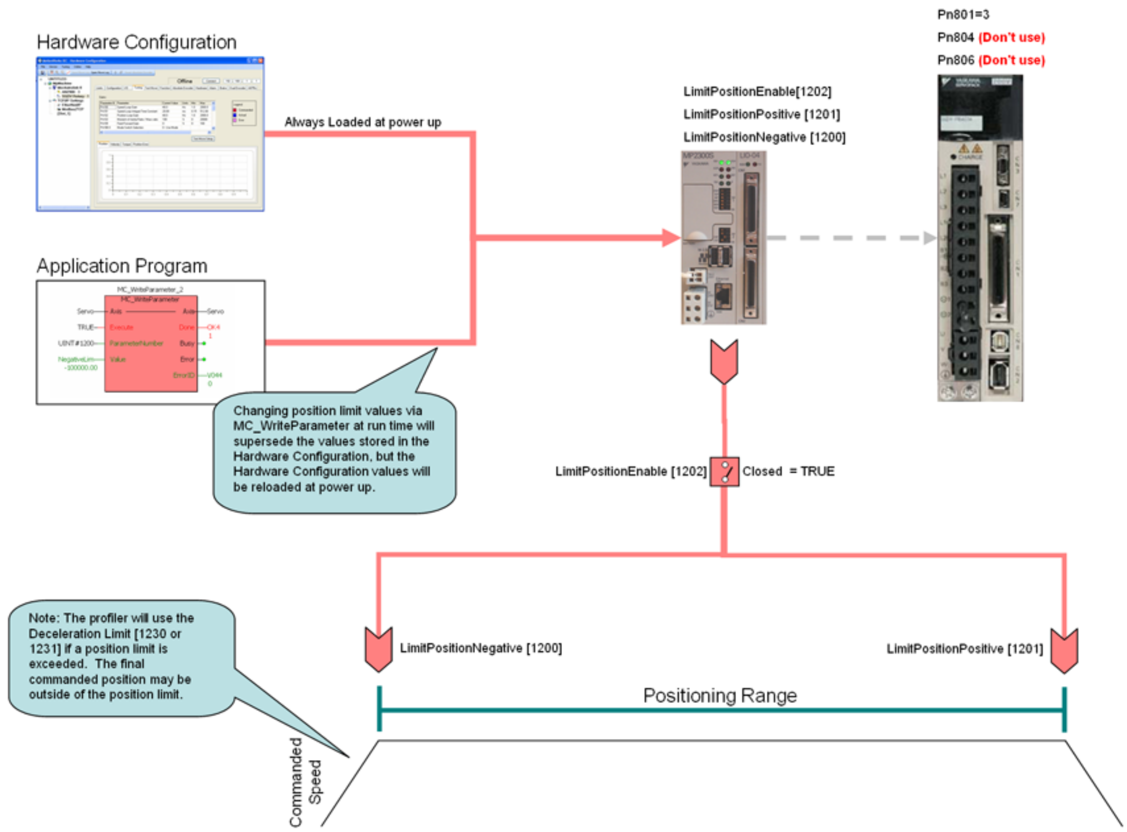
Deceleration Limits

• Deceleration is defined by decreasing velocity towards zero.

• The parameters are called LimitDecelPositive and LimitDecelNegative, with values of UINT#1231 and UINT#1230 respectively. Use the MC_WriteParameter function block for these and all controller side parameters. Deceleration limit parameters are in user units / sec2.

• To disable the deceleration limit, set LimitDecelEnable, parameter 1232 to zero.

Position Limits



• The software position limits are managed by the MPiec controller. The parameters are called LimitPositionPositive and LimitPositionNegative, with values of UINT#1201 and UINT#1200 respectively. Use the MC_WriteParameter function block for these and all controller side parameters. Position limit parameters are in user units.

• When a position limit is exceeded, a controller alarm will be generated, obtainable via the MC_ReadAxisError function block, or the web server.

• The controller alarm will be 16#3202 0001 if the positive position limit is exceeded and 16#3202 0002 if the negative position limit is exceeded.

• To disable the position limits, set LimitPositionEnable, parameter 1202 to zero.

• LimitPositionPositive must be greater than LimitPositionNegative.

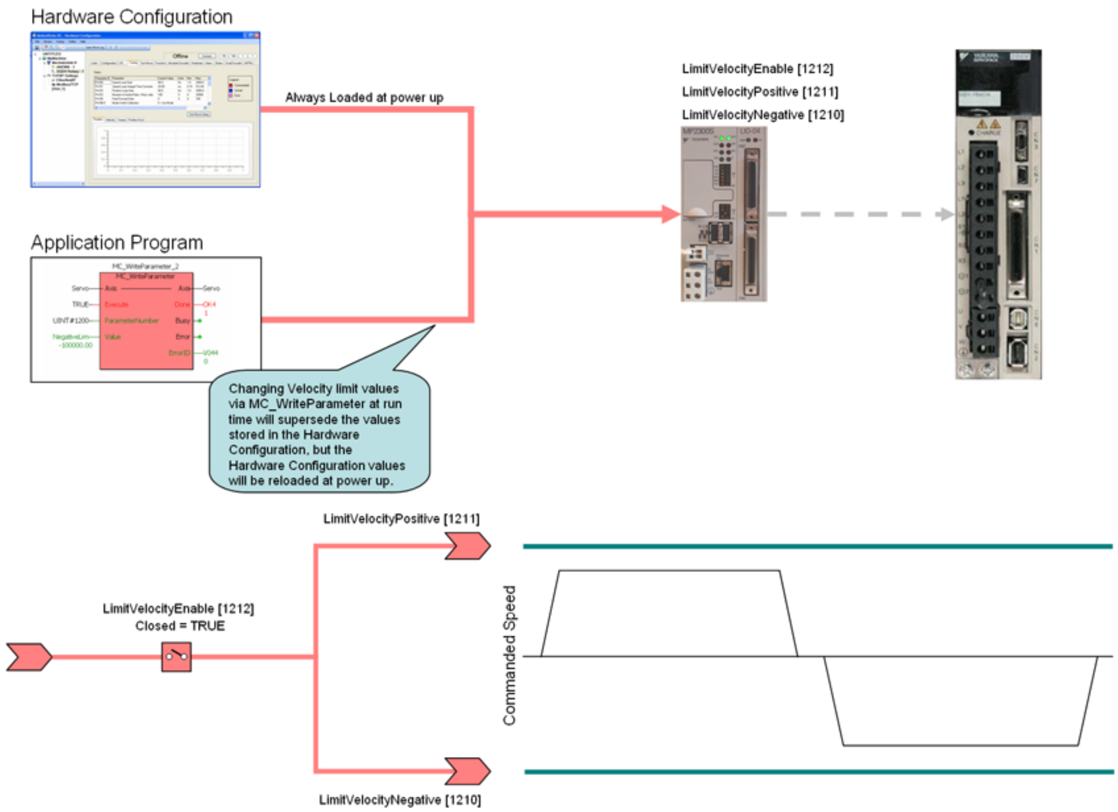
• LimitPositionNegative must be lower than LimitPositionPositive.

Notes:

• The position limit parameters in the Sigma amplifiers should not be used when controlled by an MP2000iec. Sigma parameter Pn801 is forced to a value of 3 (disabled in both directions) when the Save function is invoked from the Hardware Configuration software and Pn804 and Pn806 are not used.

• However, if a user changes Pn801 in SigmaWin to enable the software limits in the servopack, the servopack limits will supersede the controller position limits.

Velocity Limits



• The software velocity limits are managed by the MPiec controller. The parameters are called LimitVelocityPositive and LimitVelocityNegative, with values of UINT#1211 and UINT#1210 respectively. Use the MC_WriteParameter function block for these and all controller side parameters. Velocity limit parameters are in user units / sec.

• When a velocity limit is exceeded, a controller alarm will be generated, obtainable via the MC_ReadAxisError function block, or the web server.

• The controller alarm will be 16#3202 0003 if the positive velocity limit is exceeded and 16#3202 0004 if the negative velocity limit is exceeded.

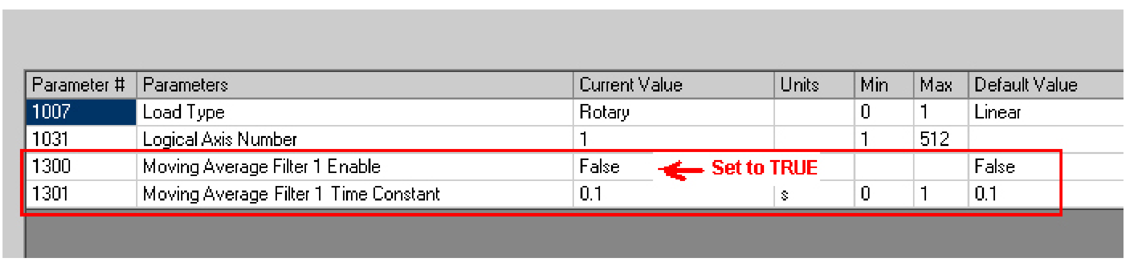
• To disable the velocity limits, set LimitVelocityEnable, parameter 1212 to zero.

• LimitVelocityPositive must be zero or greater.

• LimitVelocityNegative must be zero or lower.

Moving Average Filter (S-Curve)

A moving average filter is available for discrete motion profiles in firmware version 1.1.2.5 and higher. To use the moving average filter, it must first be enabled in the Hardware Configuration on the axis configuration tab as shown below. The filter time constant can also be specified there, or via MC_WriteParameter.



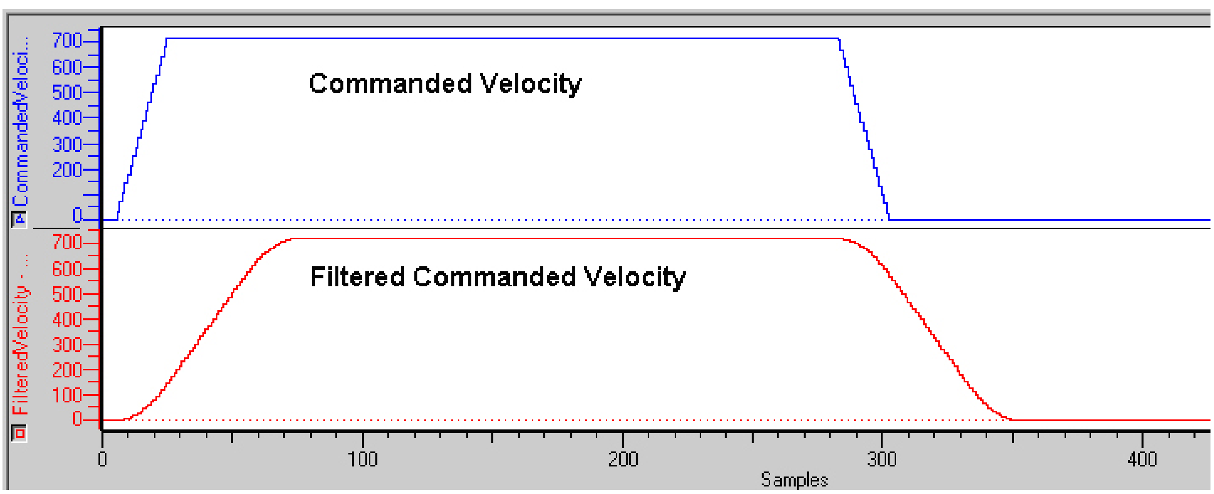
Parameter #	Parameters	Current Value	Units	Min	Max	Default Value
1007	Load Type	Rotary		0	1	Linear
1031	Logical Axis Number	1		1	512	
1300	Moving Average Filter 1 Enable	False Set to TRUE				False
1301	Moving Average Filter 1 Time Constant	0.1	s	0	1	0.1

Once the moving average filter is enabled in the Hardware Configuration, it can be changed in the application program using MC_WriteBoolParameter and read using MC_ReadBoolParameter. Use MC_ReadParameter and MC_WriteParameter to set or read the Moving Average Time Constant (parameter 1301.) The range of the Moving Average Filter Time constant is 0.0 to 5.0 seconds (0 excluded).

The moving average filter limits are [0,5). (0 excluded).

If the Moving average filter time constant is set to 0.1 seconds and the MECHATROLINK/DPRAM update rate is set to 2 ms, the moving average will provide a filter of 50 data points (50 data points in 100 ms)

An example of a move profile with the Moving Average Filter applied is shown below. The logic analyzer plot below shows the effect of a 0.1 s moving average filter set up. The logic analyzer is synchronized with a 2 ms application task. It can be seen that the filtered profile (in red, which is the actual velocity profile in this case) settles to steady state 50 scans after the commanded profile. Since each scan is 2 ms in time, 50 scans show 100ms in time. Thus it can be seen that a 0.1s moving average filter set up generates an s curve velocity profile where the filtered velocity will lag the commanded trapezoidal profile by 0.1 seconds.



Commanded velocity in blue (Parameter 1011). Post S - curve commanded velocity in red (Parameter 1021)

Note:

1) For rotary mode applications using the Moving Average Filter, use firmware 1.2.1 or greater.

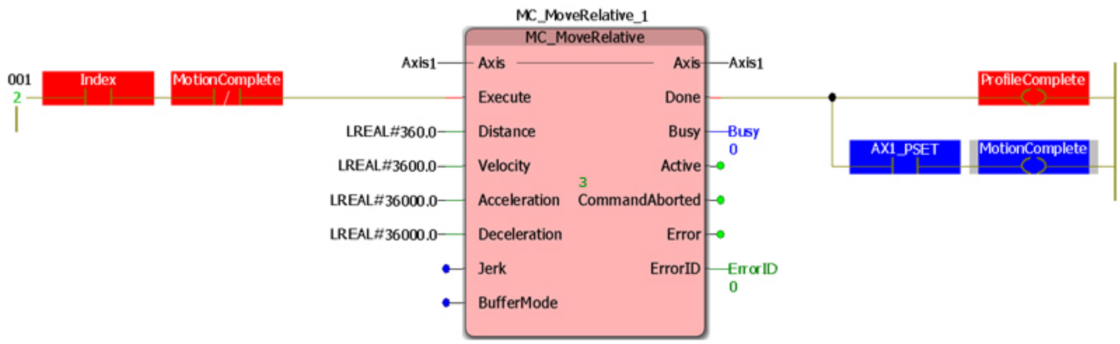
2) The Moving Average Filter Time Constant parameter (1301) can be changed only after the current moving average time constant has been processed by the motion profiler. For example, if parameter 1301 is set to 2.0 s using MC_WriteParameter at time instant 't', the next change to 1301 can be made only after time instant 't+2.0'. If an attempt to write it sooner is made, MC_WriteParameter will output an errorID 4675.

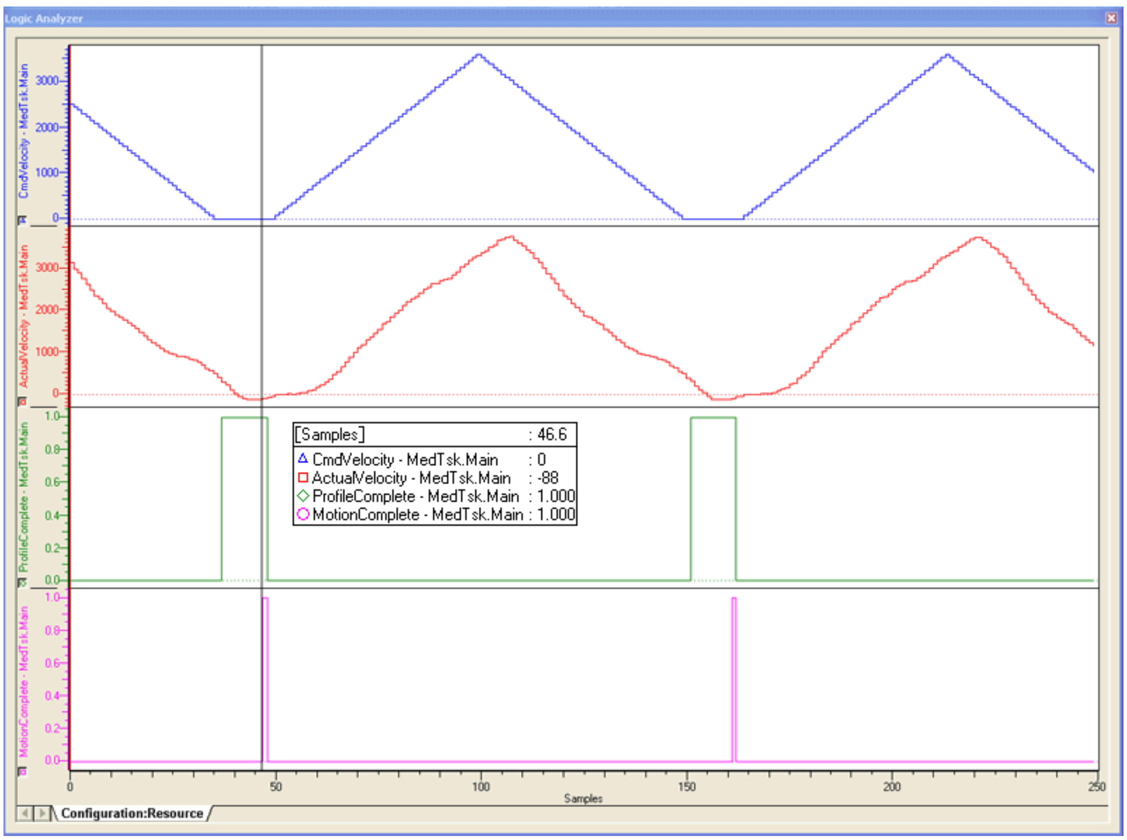
Determining When Motion is Complete

The Done output on MC_MoveRelative, MC_MoveAbsolute, MC_Stop, MC_StepLimitSwitch, and MC_StepRefPulse, indicates that the controllers motion profiler has completed the calculations for the move as specified by the function block inputs. The axis may physically be in motion and still settling on the final commanded position based on tuning parameters, load inertia, friction, and other factors. To determine when the actual motion has reached the commanded position, use the P_SET Global variable provided by the Hardware Configuration.

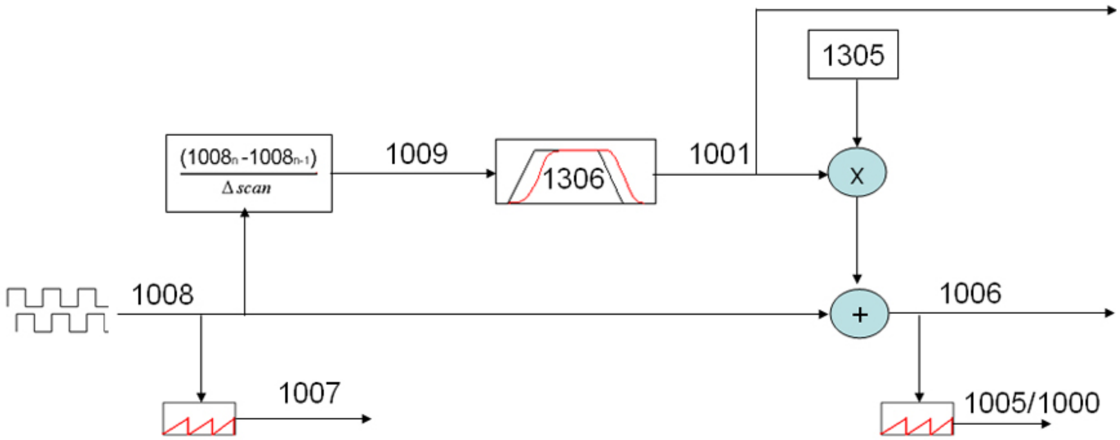
<SGDV Rotary> - Sigma-V Rotary Servo Amplifier - 1:3 (* Modify Variable Names, Not Group Name. *)				
AX3_SI1_POT	BOOL	VAR_GLOBAL	POT, default on pin #7, configurable by Pn50A.3	%IX53376.0
AX3_SI2_NOT	BOOL	VAR_GLOBAL	NOT, default on pin #8, configurable by Pn50B.0	%IX53376.1
AX3_SI3_DEC	BOOL	VAR_GLOBAL	DEC, default on pin #9, configurable by Pn511.0	%IX53376.2
AX3_SI4_EXT1	BOOL	VAR_GLOBAL	EXT1, default on pin #10, configurable by Pn511.1	%IX53376.6
AX3_SI5_EXT2	BOOL	VAR_GLOBAL	EXT2, default on pin #11, configurable by Pn511.2	%IX53376.7
AX3_SI6_EXT3	BOOL	VAR_GLOBAL	EXT3, default on pin #12, configurable by Pn511.3	%IX53377.0
AX3_BRK	BOOL	VAR_GLOBAL	Brake Output Status	%IX53377.1
AX3_HBB	BOOL	VAR_GLOBAL	HBB, Stop Signal Input	%IX53377.2
AX3_SI0_IO12	BOOL	VAR_GLOBAL	Configurable by Pn81E.0, default is unallocated	%IX53377.4
AX3_SI1_IO13	BOOL	VAR_GLOBAL	Configurable by Pn81E.1, default is unallocated	%IX53377.5
AX3_SI2_IO14	BOOL	VAR_GLOBAL	Configurable by Pn81E.2, default is unallocated	%IX53377.6
AX3_SI3_IO15	BOOL	VAR_GLOBAL	Configurable by Pn81E.3, default is unallocated	%IX53377.7
AX3_ALM	BOOL	VAR_GLOBAL	Alarm On Drive	%IX53380.0
AX3_WARNG	BOOL	VAR_GLOBAL	Warning On Drive	%IX53380.1
AX3_SVON	BOOL	VAR_GLOBAL	Servo On	%IX53380.3
AX3_PON	BOOL	VAR_GLOBAL	Main Circuit Power On	%IX53380.4
AX3_PSET	BOOL	VAR_GLOBAL	Positioning Completed	%IX53380.7
AX3_SO1	BOOL	VAR_GLOBAL	SO1, pins 1 and 2, configurable by Pn82E, Pn50E, Pn50F, P...	%QX53376.0
AX3_SO2	BOOL	VAR_GLOBAL	SO2, pins 23 and 24, configurable by Pn82E, Pn50E, Pn50F,...	%QX53376.1
AX3_SO3	BOOL	VAR_GLOBAL	SO3, pins 25 and 26, configurable by Pn82E, Pn50E, Pn50F,...	%QX53376.2

Note that P_SET will be ON any time the actual position is within a certain distance of the commanded position, as specified by Servopack Pn 522, so it is recommended to use P_Set in conjunction with the Done out of a motion function block. See the following graphics for examples.


MC_MoveRelative_1
MC_MoveRelative
001
2
Index
MotionComplete
Axis1
Axis
Execute
Distance
Velocity
Acceleration
Deceleration
Jerk
BufferMode
LREAL#360.0
LREAL#3600.0
LREAL#36000.0
LREAL#36000.0
Axis
Done
Busy
Active
CommandAborted
Error
ErrorID
3
Axis1
Busy
0
ErrorID
0
ProfileComplete
AX1_PSET
MotionComplete

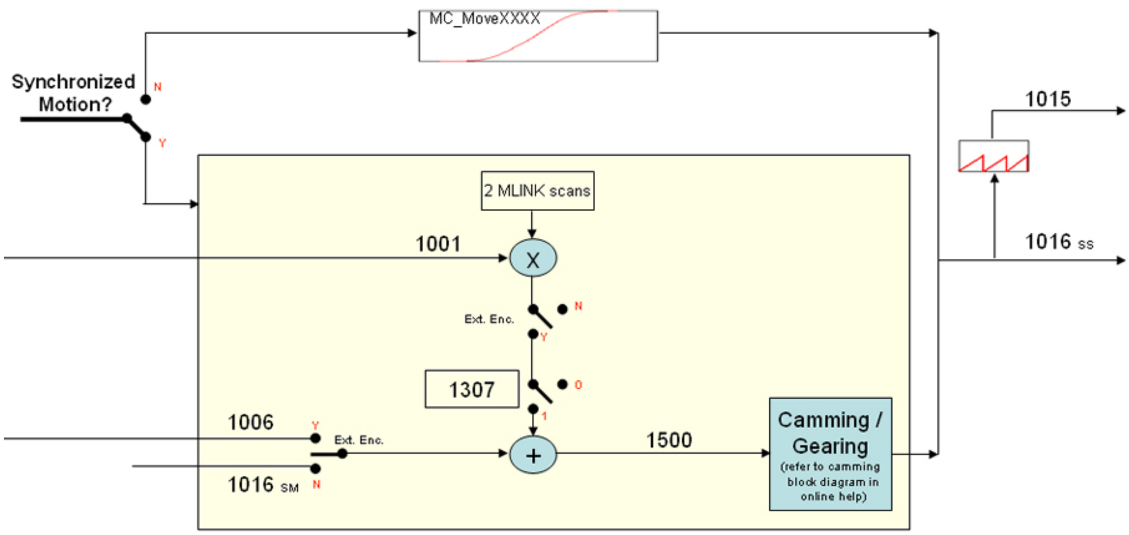

Logic Analyzer
CmdVelocity - MedTsk.Main
ActualVelocity - MedTsk.Main
ProfileComplete - MedTsk.Main
MotionComplete - MedTsk.Main
[Samples] : 46.6
CmdVelocity - MedTsk.Main : 0
ActualVelocity - MedTsk.Main : -88
ProfileComplete - MedTsk.Main : 1.000
MotionComplete - MedTsk.Main : 1.000
Samples
Configuration:Resource

External Encoder Block Diagram



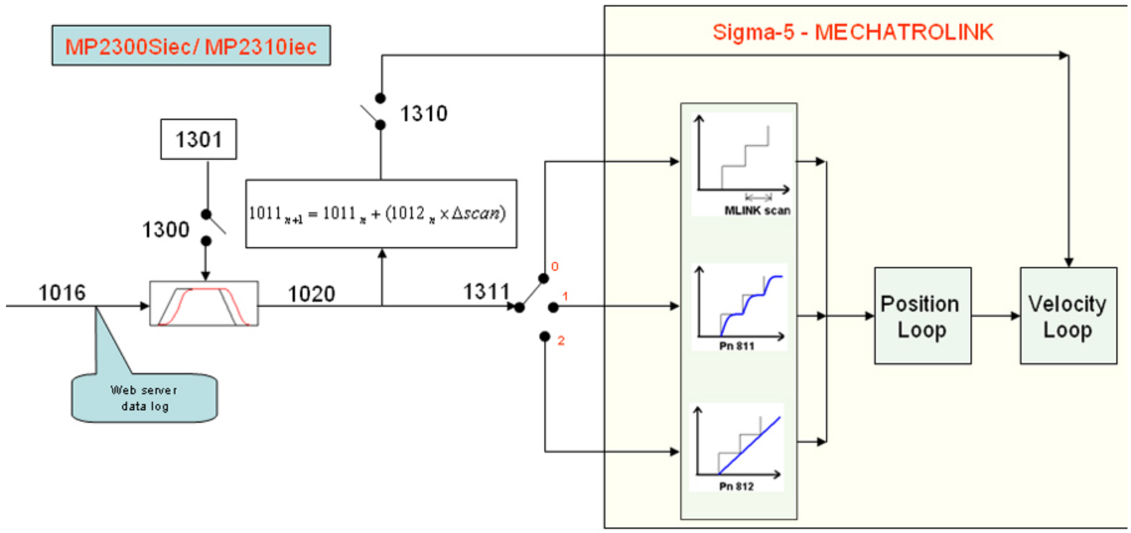
Controller Parameter	Description	Data Type	Default	Units
1001	Actual Velocity (Post S-Curve Filter)	LREAL	N/A	User units/s
1005	Actual Position Cyclic	LREAL	N/A	User units/s
1006	Actual Position Non-Cyclic	LREAL	N/A	User units/s
1007	External Raw Position Cyclic	LREAL	N/A	User units/s
1008	External Raw Position Non-Cyclic	LREAL	N/A	User units/s
1009	External Velocity Unfiltered	LREAL	N/A	User units/s
1305	Scan Compensation	LREAL	2 MLINK Scans	s
1306	Velocity Filter	LREAL	0.0	s

Commanded Position Output



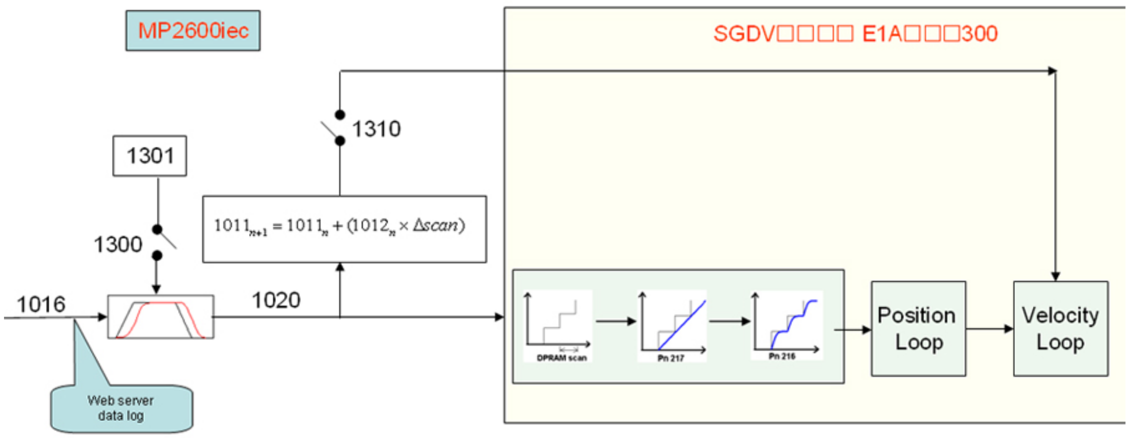
Controller Parameter	Description	Data Type	Default	Units
1001	Actual Velocity (Post S-Curve Filter)	LREAL	N/A	User units/s
1006	Actual Position Non-Cyclic	LREAL	N/A	User units
1015	Commanded Position Cyclic	LREAL	N/A	User units
1016 SM	Commanded Position Cyclic (Servo Master)	LREAL	N/A	User units
1016 SS	Commanded Position Cyclic (Servo Slave)	LREAL	N/A	User units
1307	MECHATROLINK Compensation	BOOL	TRUE	
1500	Cam Master Position	LREAL	N/A	User units

Command Filtering (MPiec with MECHATROLINK interface)



Controller Parameter	Description	Data Type	Default	Units
1011	Commanded Velocity	LREAL	N/A	User units/s
1012	Commanded Acceleration	LREAL	N/A	User units/s^2
1016	Commanded Position Non Cyclic	LREAL	N/A	User units
1020	Commanded Position Non Cyclic (Post S-Curve Filter)	LREAL	N/A	User units
1300	Filter Moving Average Enable	BOOL	FALSE	N/A
1301	Filter Moving Average	LREAL	0.1	s
1310	Controller Feed Forward Enable	BOOL	TRUE	N/A
1311	Commanded Position Sub-Filter	LREAL	0	N/A

Command Filtering (MP2600iec - Dual Port RAM interface)



Controller Parameter	Description	Data Type	Default	Units
1011	Commanded Velocity	LREAL	N/A	User units/s
1012	Commanded Acceleration	LREAL	N/A	User units/s^2
1016	Commanded Position Non Cyclic	LREAL	N/A	User units
1020	Commanded Position Non Cyclic (Post S-Curve Filter)	LREAL	N/A	User units
1300	Filter Moving Average Enable	BOOL	FALSE	N/A
1301	Filter Moving Average	LREAL	0.1	s
1310	Controller Feed Forward Enable	BOOL	TRUE	N/A