Technical Paper
PLCopen Technical Committee 2 – Task Force

Function Blocks for motion control: Part 5 – Homing Procedures

Version 2.0, Published

November 16, 2011

The following paper

Function Blocks for Motion Control – Part 5 – Homing Procedures

is a specification as developed by the PLCopen Task Force Motion Control. As such it is an addition to the PLCopen Task Force Motion Control, Technical Documents Part 1 till 4.

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

Joachim Unfried	B&R
Nils Gotha	Baumüller
Wilfried Plaß	Beckhoff
Djafar Hadiouche	GE Intelligent Platforms
Juergen Hipp	ISG
Harald Buchgeher	Keba
Dr. Johannes Kühn	Lenze
Candido Ferrio	Omron
Josep Lario	Omron
Wolfgang Fien	Schneider Electric
Willi Gagsteiger	SIEMENS A&D
Edwin Schwellinger	3S
Hilmar Panzer	3S
Lutz Augenstein	Stöber
Istvan Ulvros	TetraPak
Eelco van der Wal	PLCopen

Change Status List:

Version	***Date***	***Change comment***
V 0.1	September 24, 2003	Result of the meeting Sept. 22 – 24, 2003 to include homing in a separate document
V 0.2	November 17, 2004	Result of the meeting Sept. 22 – 23, 2004 to merge with proposal Omron
V 0.3	December 7, 2004	Result of the meeting Dec. 6+7, Phoenix Contact
V 0.4	May 04, 2005	Result of the April 2005 meeting. Edited by EvdW
V 0.5	May 17, 2005	New structure: separation between procedures and FBs – EvdW
V 0.6	October 4, 2005	Includes feedback C. Ferrio. Send to the whole MC group for feedback.
V 0.99	November 10, 2005	Released for comments till April 30, 2006
V0.99b	June 15, 2006	Result from May meeting. Total re-editing by EvdW and CFD-OYMC
V 0.99C	August 30, 2006	Internal version for review, done by EvdW
V 0.99D	March 23, 2007	Final version for review
V 0.99E	October 24, 2007	Added multi reference homing procedure
V 0.99F	May 26, 2010	Work on the document restarted as result of V 2.0 of Part 1 & 2 Results of the meeting in Kempten, Germany
V 0.99G	April 13, 2011	As result from the web meeting
V 0.99H	June 10, 2011	Version G as result of the May web meeting
V 0.99I	July 1, 2011	As result from the webmeeting in June. Added input 'ReferenceSignal'
V 0.99J	July 22	As preparation for the webmeeting on July 26
V 0.99K	July 26, 2011	As result of the webmeeting July 26
V 2.0	November 16, 2011	Published

Contents

1 Introduction

This document is a part of a set of specifications as provided by the association PLCopen. These specifications include Part 1 – Basics, which describe the fundamentals of the PLCopen Motion Control specification. Within this part, a homing function block has been defined, as well as the state 'Homing' in the State Diagram. During additional discussions, it became clear that the specified functionality was too limited. For this reason, additional homing procedures and function blocks are specified within this document, based on the work done in Part 1.

1.1. Homing Procedures

The homing functionality was originally seen as a separate sequence in the start up phase of a machine or axis. The sequence for an axis was: power on, homing, and move 'something'. The whole homing procedure itself was hidden to the user. However, from multiple discussions, some users needed more control on the homing functionality itself. For this reason, a set of building blocks have been identified to be able to define dedicated homing procedures. This procedure can be encapsulated in a dedicated homing function block, which can be added to the library and used for this specific procedure.
As examples, several possible homing procedures are described in Chapter 2, which are coupled in Chapter 3 to the homing building blocks. The procedures are:

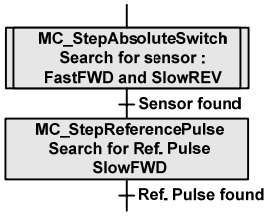
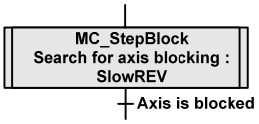
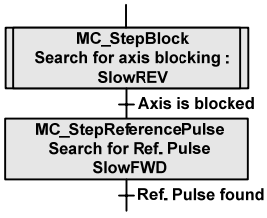
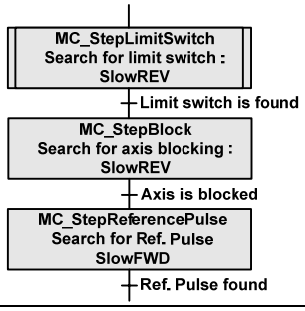
- HomeAbsoluteSwitch - Homing against an absolute switch plus limit switches
- HomeLimitSwitch - Homing against limit switches
- HomeBlock - Homing against hardware parts blocking the movement
- HomeReferencePulse - Homing using encoder reference pulse "Zero Mark"
- HomeReferencePulseSet - Homing using a set of encoder reference pulses "Zero Mark"
- HomeDistanceCoded - Homing using a set of distance coded reference marks
- HomeDirect - Static Homing, forcing a position from a user reference
- HomeAbsolute - Static Homing, forcing a position from an absolute encoder

1.2. Homing Function Blocks

To make these procedures available to the user, a toolkit is defined with additional Function Blocks, FBs. By using one or more connected function blocks one can create complex homing sequences. With these, a user can create its own homing procedure in more detail, and even create own, user specific, homing procedures. Additionally, these function blocks include advanced homing process error reporting, evaluating time, distance and torque limits.

Homing step function blocks.
The following defined FBs match the homing procedures and are issued in or change the state to the 'Homing' state. They do not change the state of the axis after finalizing. This means that it stays in the 'Homing' state.

- MC_StepAbsoluteSwitch - Homing against an absolute switch plus limit switches
- MC_StepLimitSwitch - Homing against limit switches
- MC_StepBlock - Homing against hardware parts blocking movement
- MC_StepReferencePulse - Homing using encoder reference pulse "Zero Mark"
- MC_StepDistanceCoded - Homing using a set of distance coded reference marks

Typical 2 step homing procedure : Physical sensor and reference pulse of encoder	*Simple Axis homing against hardware part blocking movement*	*2 Step homing : Against block, then search for reference pulse*	*Special customer homing procedure : find limit switch, then against block and then search for reference pulse of encoder*
			

Example of constructs of homing procedures

In this way, any combination sequence is possible without the need of predefining hundreds of homing methods. The individual adjustments for torque limit, time limit and distance limit provide control of the sequence error conditions.

Finalizing homing procedure and leaving the homing state.
The following Function Blocks lead to a final position, automatically changing the state from 'Homing' to 'Standstill' (if started in the 'Homing' state):

- MC_HomeDirect - Static 'Homing' forcing a position from a user reference
- MC_HomeAbsolute - Static 'Homing' forcing a position from an absolute encoder
- MC_FinishHoming - Finishes the homing procedure with the possibility to do a relative move to the operational area.

These functions can also be called in other states than 'Homing' making a home procedure possible without having power-stage set in the drive (e.g. by manually moving the axis to the right position).

Procedures for homing-on-the-fly.
In addition, the homing functionality is needed while the machine is operating, i.e. not in the state 'Homing'. This 'homing-on-the-fly' is called passive homing. They have no effect on the State Diagram, like the administrative FBs, and can be called in any movement state. They consist of:

- MC_StepReferenceFlyingSwitch
- MC_StepReferenceFlyingRefPulse
- MC_AbortPassiveHoming

1.3. Length of FB names and ways to shorten them

There are systems that only support a limited number of significant characters in the name. For these rules for shorter names are provided here. These names are still seen as compliant, although have to be mentioned in the certification document.
List of rules to shorten names:

Position	Pos [(1)]
Velocity	Vel
Acceleration	Acc
Absolute	Abs
Direct	Dir
Relative	Rel
Actual	Act
Reference	Ref
Positive	Pos [(1)]
Negative	Neg
Limit	Lim
Flying	Fly
Distance	Dist
Passive	Ps

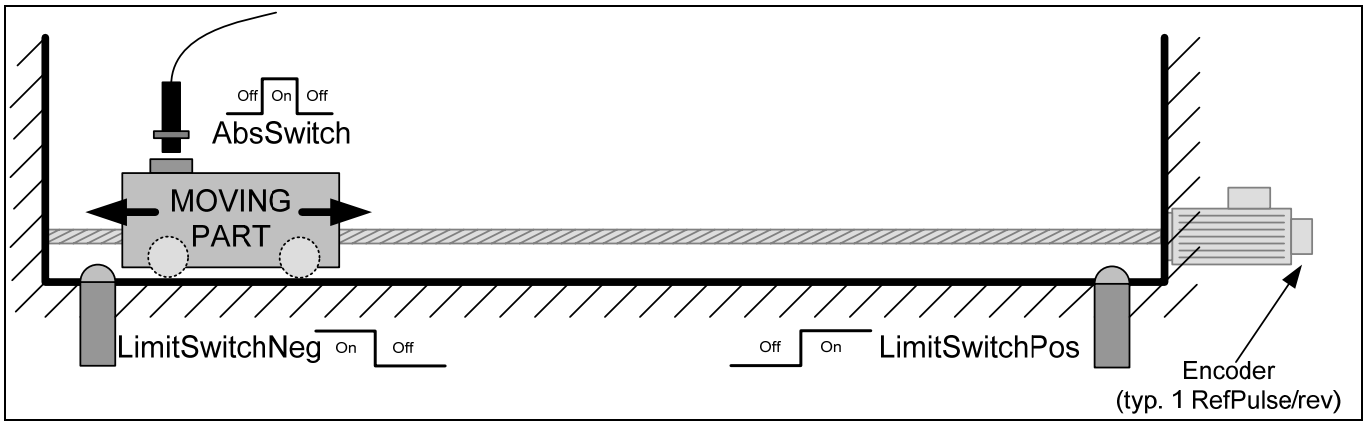
Note [(1)] : it is clear from the context if 'Position' or 'Positive' is meant with 'Pos'

Resulting compliant names:

Original	**Shortened (if applicable)**	**Lenght (Characters)**
MC_StepAbsoluteSwitch	MC_StepAbsSwitch	16
MC_StepLimitSwitch	MC_StepLimSwitch	16
MC_StepBlock	MC_StepBlock	12
MC_StepReferencePulse	MC_StepRefPulse	15
MC_StepDistanceCoded	MC_StepDistCoded	16
MC_HomeDirect	MC_HomeDirect	13
MC_HomeAbsolute	MC_HomeAbs	10
MC_StepReferenceFlyingSwitch	MC_RefFlySwitch	15
MC_StepReferenceFlyingRefPulse	MC_RefFlyPulse	14
MC_AbortPassiveHoming	MC_AbortPsHoming	16

2 Homing Procedures

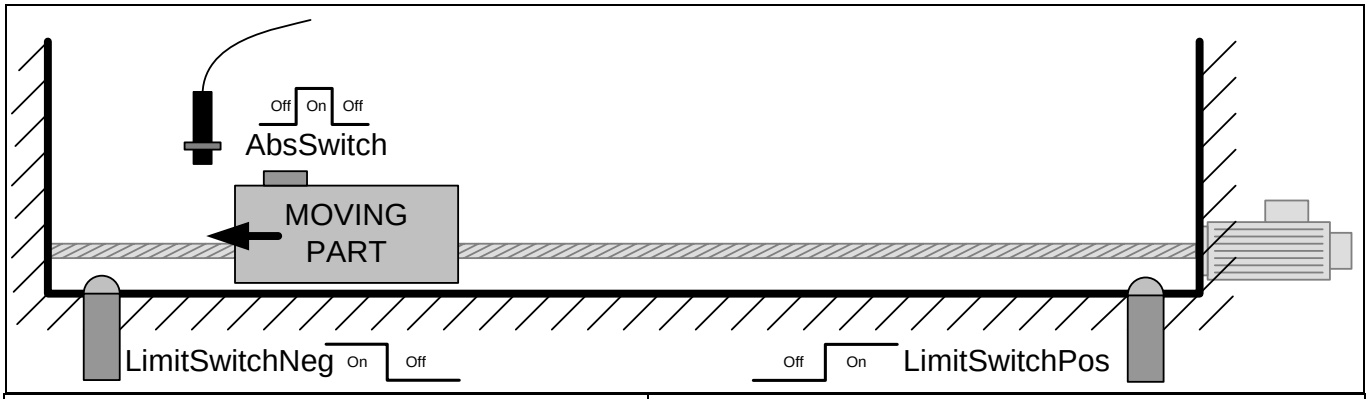
In general a homing procedure couples a position to a specific axis. 'Homing' is depending on the encoders used (absolute or relative), as well as the system used (linear versus circular). Absolute encoders do not need a movement during the homing procedure, since the exact positions can be directly transferred to the system.
For other encoder types, a movement is necessary, since there is no knowledge on the exact position within the system. Basically, this movement is at low speed in some direction till a certain measuring point is reached. A general scenario is given in the picture below.



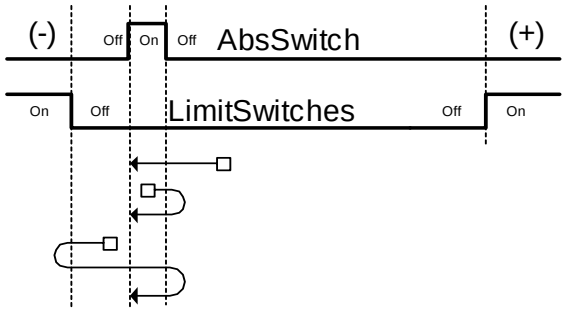
The example above describes a simple sequence consisting of 3 steps:

1. search for a 'signal'
2. search for another 'signal'
3. move the axis to a pre-defined position between the limit switches (operational area).

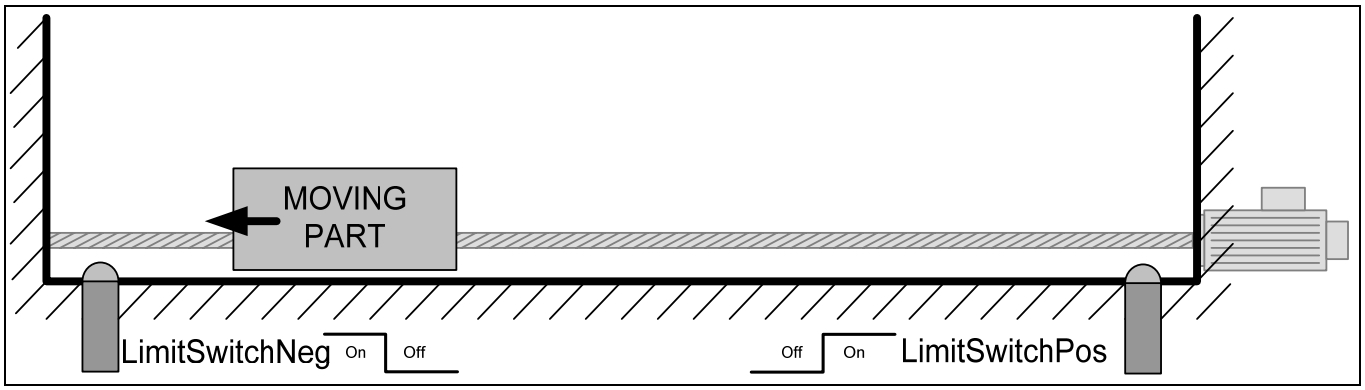
2.1. HomeAbsoluteSwitch

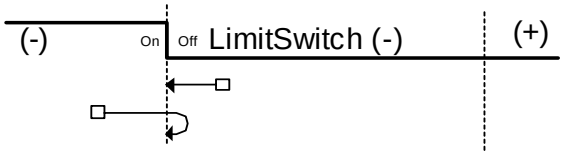


This homing procedure performs a homing function by searching for an absolute positioned external physical switch. (The AbsoluteSwitch has two "Off" (or "On") areas – see example). The exact procedure is hidden for the user.
The procedure itself needs a limited intelligence. In the example on the right side, three different starting positions (indicated by the small squares) lead to three completely different motion sequences, but with identical final result. This behavior is possible with just one FB.

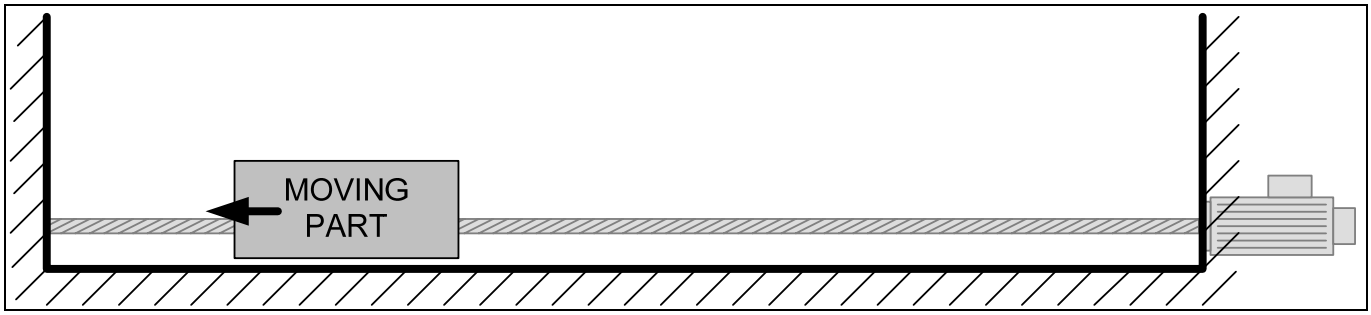


2.2. HomeLimitSwitch



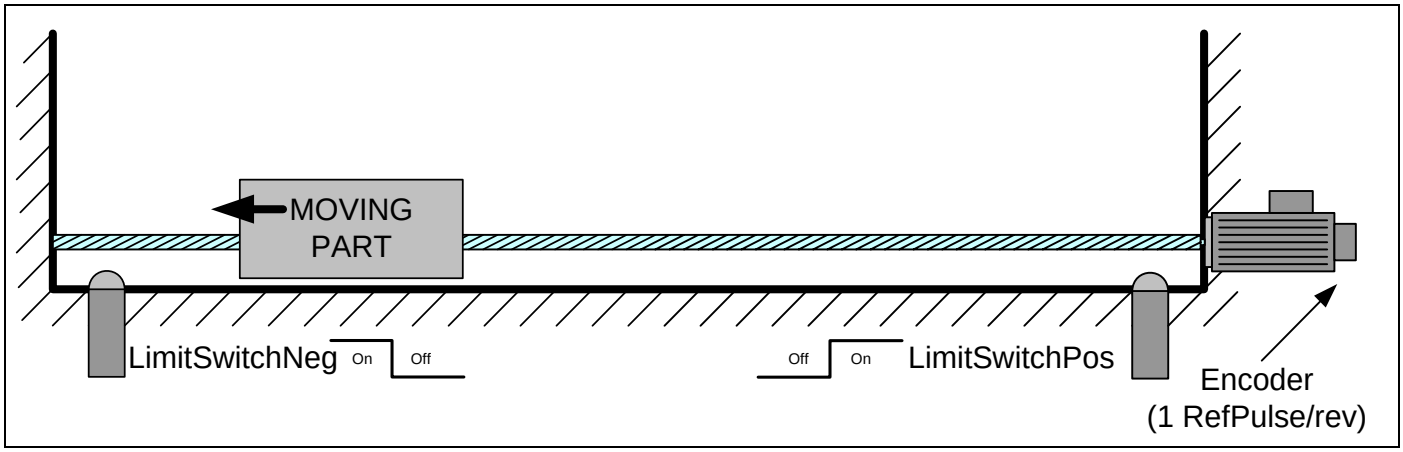
This homing procedure performs a homing function searching for a sensor using only the limit switches. (A 'LimitSwitch' has 1 "Off" (or "On") area). A possible scenario for this is trying to reference against a negative limit switch – see example on the right side. (The small square is the starting point)	

2.3. HomeBlock

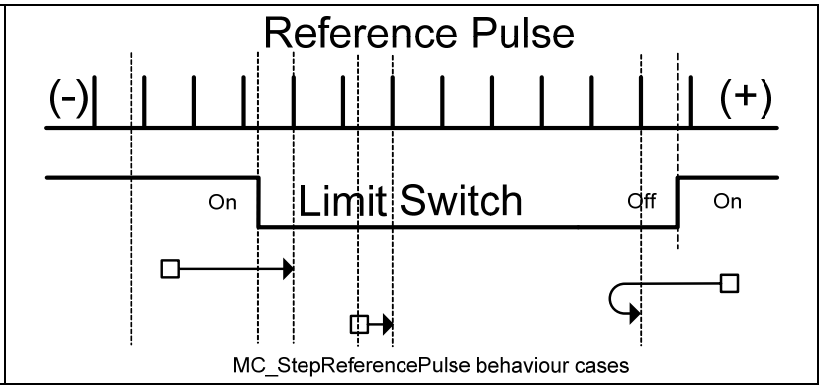


This homing procedure performs homing against a physical object, mechanically blocking the movement. In this mode there is no limit switch or reference Ppulse. Adequate torque limits are required for not damaging mechanics during homing process.

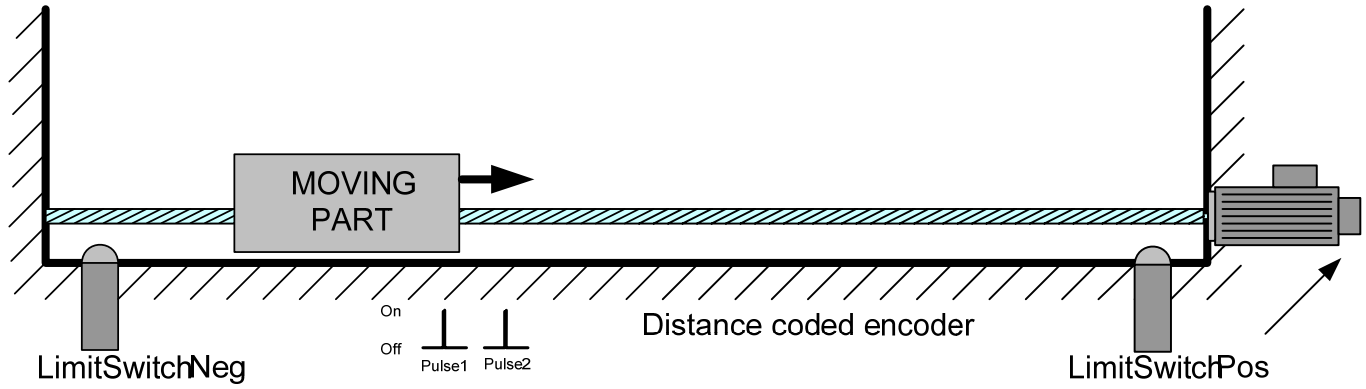
2.4. HomeReferencePulse

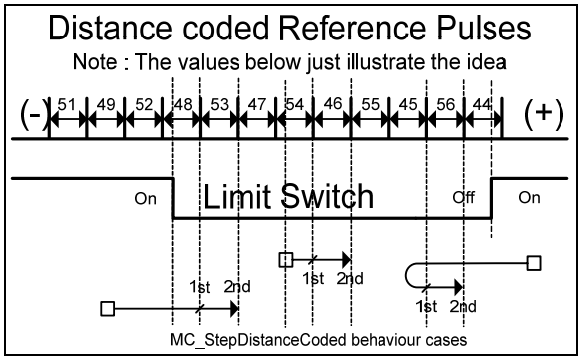


This homing procedure performs homing by searching for a reference pulse (also called marker or zero pulse) from the encoder. The reference pulse appears once per encoder revolution. The advantage in using a reference pulse for homing is the higher accuracy and precision that can be achieved compared to traditional optical, mechanical or magnetic sensors.
A possible scenario is shown on the right.



2.5. HomeDistanceCoded



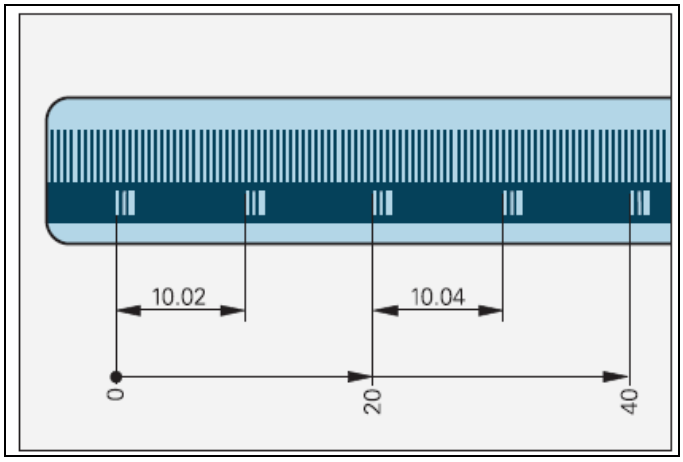


This homing procedure performs homing by searching for a set of reference pulses which appear repeatedly on the distance coded linear encoder length. The advantage in using distance coded reference pulses for homing is the reduced need for traveling distance and reduced time.
A possible scenario is:

- start in a preferred direction with low speed
- find first reference pulse. If block or end switch are found instead stop and retry the full procedure in inverse direction
- find second reference pulse. If block or end switch are found instead stop and retry the full procedure in inverse direction
- calculate absolute position

If the procedure still finds block or end switch conditions when doing the retry abort the process and signal a fault situation.

With the incremental measuring method, the graduation consists of a periodic grating structure. The position information is obtained by counting the individual increments (measuring steps) from some point of origin. Since an absolute reference is required to ascertain positions, the scales or scale tapes are provided with an additional track that bears a reference mark. The absolute position on the scale, established by the reference mark, is gated with exactly one signal period. The reference mark must therefore be scanned to establish an absolute reference or to find the last selected datum.
In some cases this may necessitate machine movement over large lengths of the measuring range. To speed and simplify such "reference runs", many encoders feature distance-coded reference marks—multiple reference marks that are individually spaced according to a mathematical algorithm. The subsequent electronics find the absolute reference after traversing two successive reference marks—only a few millimetres traverse (see table).
With distance-coded reference marks, the absolute reference is calculated by counting the signal periods between two reference marks.



Schematic representation of an incremental graduation with distance-coded reference marks (picture courtesy Heidenhain)

2.6. HomeDirect

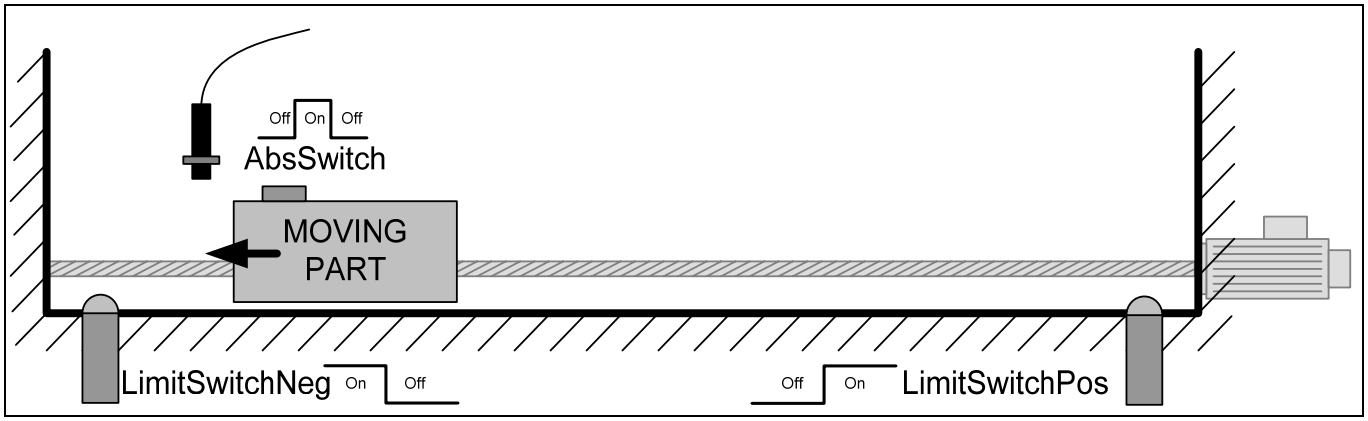
This homing procedure performs a static homing by directly forcing an actual position. No physical motion is performed in this mode. This is equivalent to a MC_SetPosition action, but changes the state from 'Homing' to 'Standstill'.

2.7. HomeAbsolute

This homing procedure performs a static homing function by setting the 'ActualPosition' input to the position of an absolute encoder. No physical motion is performed in this mode. It is equivalent to issuing MC_SetPosition with 'SetPosition' coming from absolute encoder reading, but changes the state from 'Homing' to 'Standstill'.

3 Homing Step Function Blocks

3.1. MC_StepAbsoluteSwitch

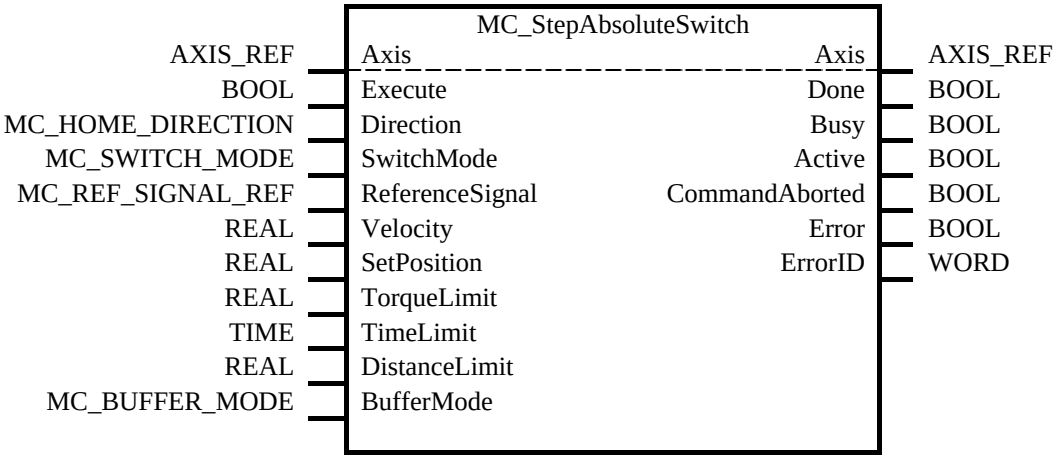


FB-Name		MC_StepAbsoluteSwitch		
This Function Block performs a homing function HomeAbsoluteSwitch by searching for an absolute positioned external physical switch. (An absolute switch has two “Off” (or “On”) areas – see example)				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function step at rising edge
	E	Direction	MC_HOME_DIRECTION	Specifies the direction of the motion (if any): • mcPositiveDirection = Starts in positive direction always • mcNegativeDirection = Starts in negative direction always • mcSwitchPositive = Depends on switch status at ‘Execute’ edge. If switch is “Off”, ‘Direction’ is positive, if “On” it is negative. • mcSwitchNegative = Like previous, but opposite.
	E	SwitchMode	MC_SWITCH_MODE	Sensor condition to finalize StepAbsoluteSwitch in any switch mode: • mcOn = When sensor is “On” • mcOff = When sensor is “Off” • mcRisingEdge = When “Off” to “On” transition in sensor • mcFallingEdge = When “On” to “Off” transition in sensor • mcEdgeSwitchPositive = Edge depends on motion direction • mcEdgeSwitchNegative = Like previous but opposite
	E	ReferenceSignal	MC_REF_SIGNAL_REF	References to the referencing switch with its associated data
	E	Velocity	REAL	Value of the maximum velocity (not necessarily reached) [u/s]
	E	SetPosition	REAL	Value of the position [u] to be set when conditions are met
	E	TorqueLimit	REAL	Maximum torque or force [in t.u.]. 0 means no torque limit
	E	TimeLimit	TIME	If StepAbsSwitch condition is not met in the ‘TimeLimit’, ‘Error’ is issued. 0 means no time limit
	E	DistanceLimit	REAL	If StepAbsSwitch condition is not met within a ‘DistanceLimit’ travel, then an ‘Error’ is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 ‘Aborting versus Buffered modes’
VAR_OUTPUT				
	B	Done	BOOL	StepAbsSwitch conditions are met
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis

	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification

Notes:

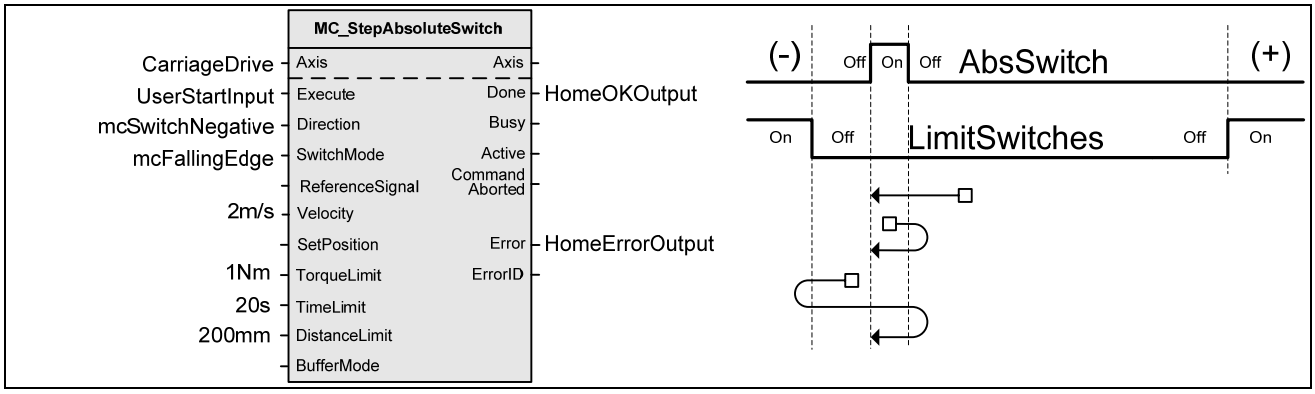
- The real meaning of "On" and "Off" will depend on the switch logic and controller input logic configurations.
- It is called AbsoluteSwitch because it might be active only in a specific area of the moving part limits, and being "Off" in the areas surrounding this switch area.
- Additional limit switches (as shown in the figure) can be applicable for the security of the system (if homing started in a direction that the absolute reference cannot be found) (see hereunder)
- The input 'ReferenceSignal' can link to different signals when used by different FBs on the same axis. The structure can be similar to the MC_TRIGGER_REF for the MC_TouchProbe, and can include an input signal (BOOL) and additional information, like a time stamp or a differential position. The signal input is meant for connecting to a reference switch. This reference switch can be a simple switch or a fast digital IO (with time stamp), or a reference mark of any position sensor, or any other input. The reference signal can include the signal or a pointer to the data.



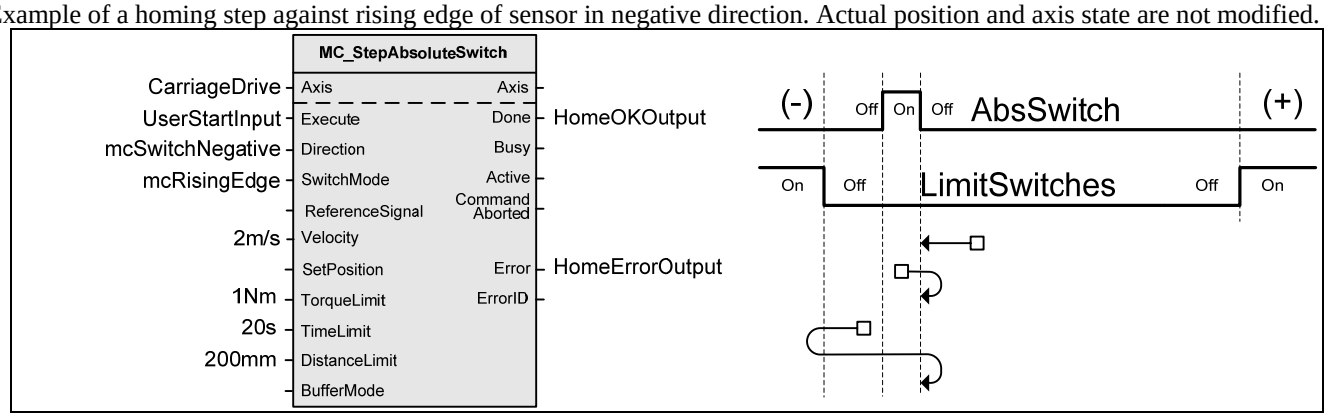
The physical layout has the risk that homing is started in the wrong direction (moving away from the switch). To support such a case, it implements a special behavior when limit switches are found (or the AbsSwitch itself is "On" at 'Execute'):

- Axis State is set to 'Homing' (if not already in that state).
- The 'Homing' is commanded in the most likely direction were the sensor can be found. In this example (-).
- The velocity is defined by the input.
- The torque is limited.
- Both 'TimeLimit' and 'DistanceLimit' cause an error if exceeded.
- If any limit switch is found during 'Homing' (any of them), then a special process is started in the opposite direction, the absolute switch is searched to switch off (or on depending in 'SwitchMode' setting). The edge (passed by), and homing process is restarted in the original direction and with the same conditions. This ensures that the end conditions are always same.
- If the 'SwitchMode' is either mcEdgeSwitchNegative or mcEdgeSwitchPositive, then the special process is also started in opposite direction depending from the switch state at 'Execute'.
- 'SetPosition' input disconnected: the FB doesn't modify the actual position.
- 'SetPosition' input connected: This FB modifies the actual position to the 'SetPosition' value when the homing condition is met.

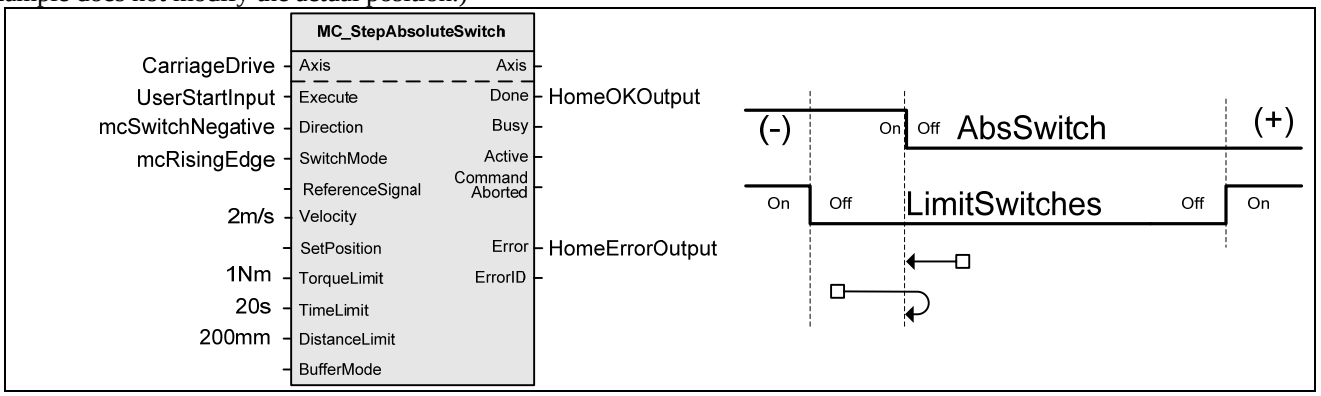
Example of a homing step against falling edge of sensor in negative direction. Actual position and axis state are not modified



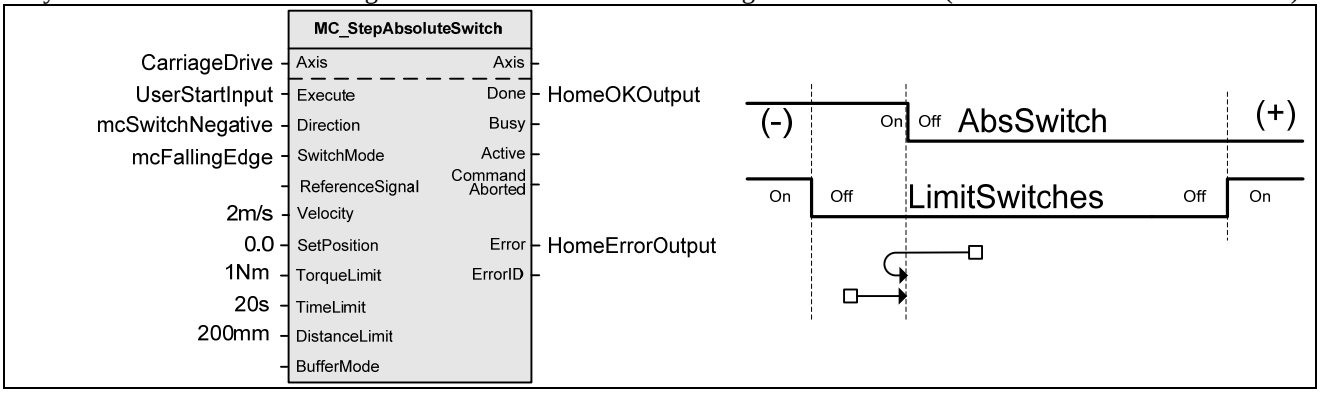
Example of a homing step against rising edge of sensor in negative direction. Actual position and axis state are not modified.



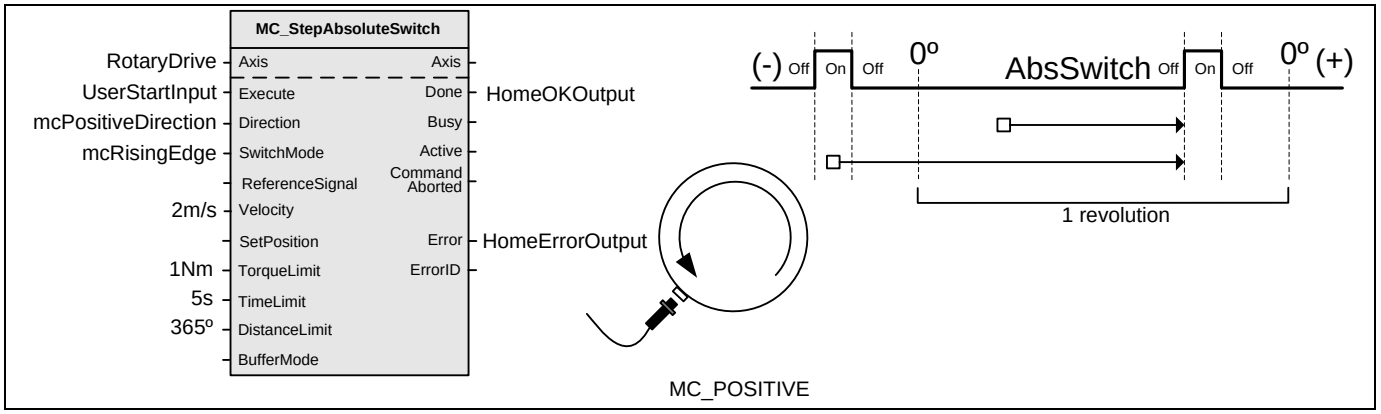
An overlapping switch configuration is also possible. This has same behavior as working on the limit switches. (Note: this example does not modify the actual position.)



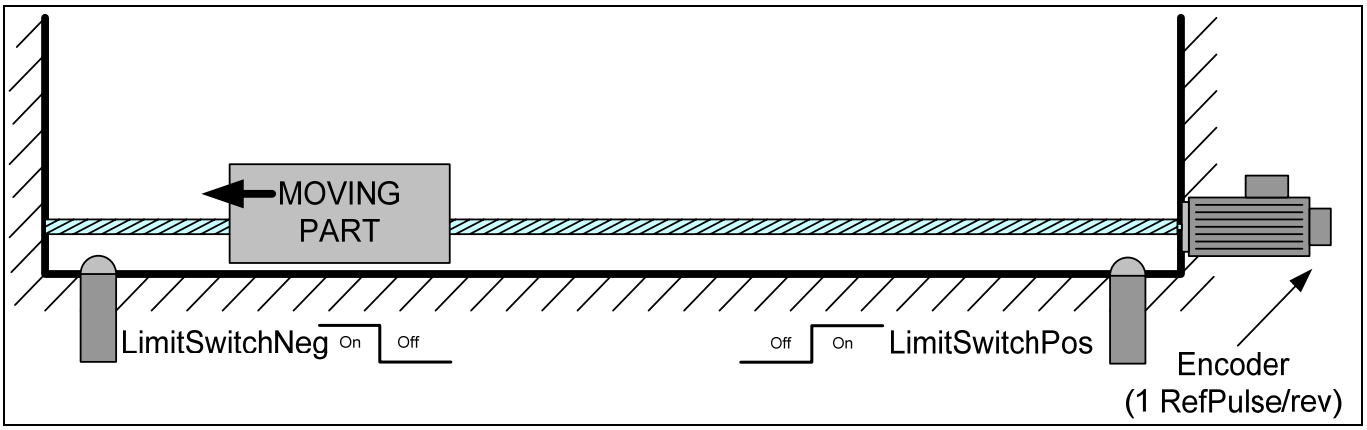
This additional example is a homing step against falling edge this time on the same negative direction. This example does modify the Actual Position at the edge detection and leaves the 'Homing' state of the axis (because of data in 'SetPosition')



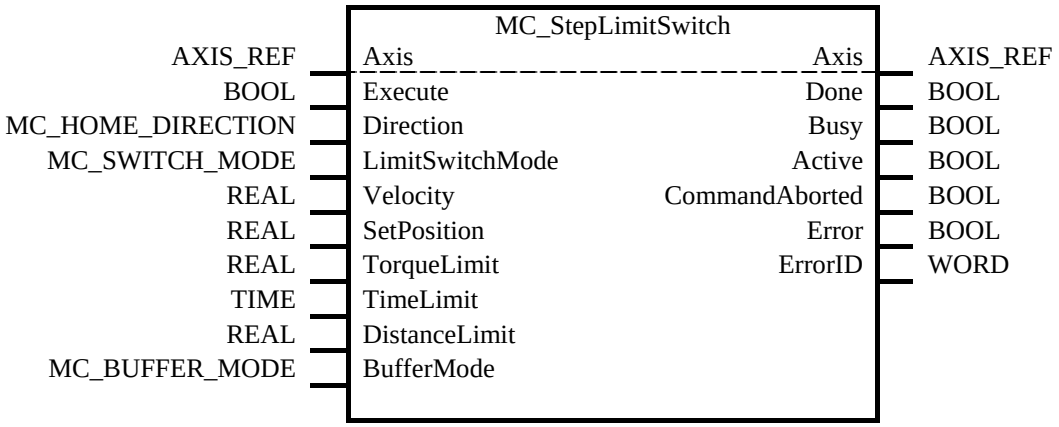
Last example: if the input 'Direction' is set to a fixed direction (mcPositiveDirection or mcNegativeDirection), then the initial switch state is ignored (used for example in rotary axis where only one sense of rotation is allowed). Axis actual position and state are not modified.



3.2. MC_StepLimitSwitch

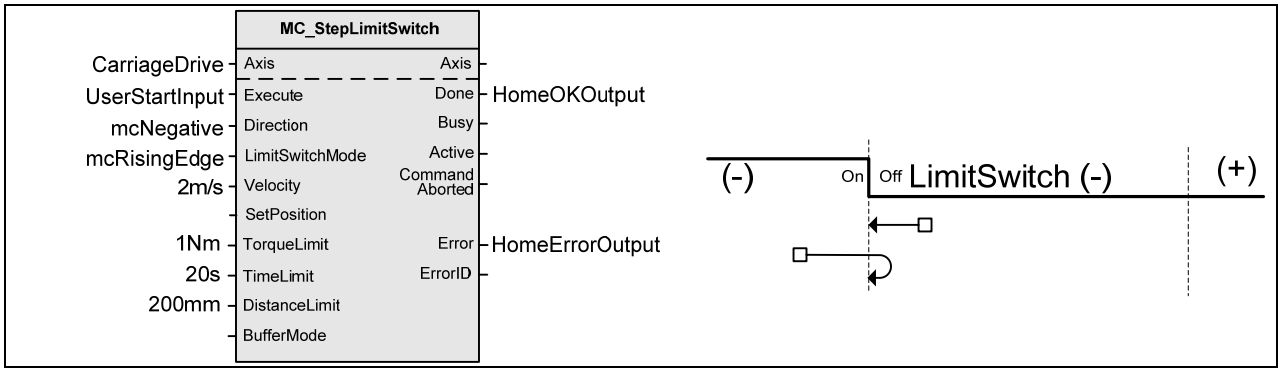


FB-Name		MC_StepLimitSwitch		
This Function Block performs the homing procedure HomeLimitSwitch by searching for a sensor using only the limit switches. (A limit switch has 1 “Off” (or “On”) area)				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function step at rising edge
	E	Direction	MC_HOME_DIRECTION	Specifies the direction of the motion and corresponding limit switch to search for : • mcPositiveDirection = Positive direction searching positive limit switch • mcNegativeDirection = Negative direction searching negative limit switch ‘Direction’ is automatically reversed from limit switch initial state.
	E	LimitSwitchMode	MC_SWITCH_MODE	Sensor condition to finalize StepLimitSwitch : • mcOn = When sensor is “On” • mcOff = When sensor is “Off” • mcRisingEdge = When “Off” to “On” transition in sensor • mcFallingEdge = When “On” to “Off” transition in sensor
	E	Velocity	REAL	Value of the velocity of motion (not necessarily reached) [u/s]
	E	SetPosition	REAL	Value of the position [u] to be set when conditions are met
	E	TorqueLimit	REAL	Maximum torque or force [in t.u.]. 0 means no torque Limit
	E	TimeLimit	TIME	If StepLimitSwitch condition is not met in the ‘TimeLimit’, ‘Error’ is issued. 0 means no time limit
	E	DistanceLimit	REAL	If StepLimitSwitch condition is not met within a ‘DistanceLimit’ travel, then ‘Error’ is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 ‘Aborting versus Buffered modes’
VAR_OUTPUT				
	B	Done	BOOL	StepLimitSwitch conditions are met
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	‘Command’ is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Note: The real meaning of “On” and “Off” will depend on the limit switch logic and controller input logic configurations.				

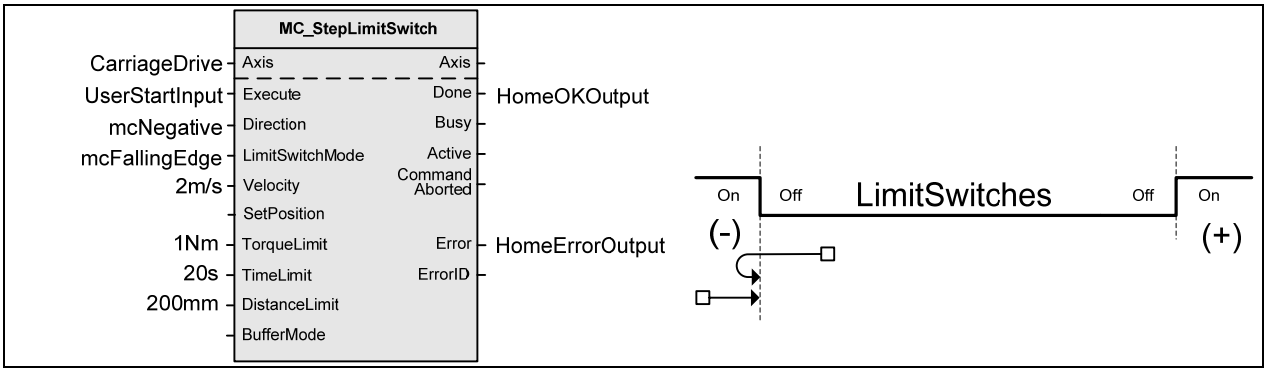


- The axis state is changed to 'Homing' (if not already in this state).
- The 'Homing' is commanded by the user in the desired homing direction at the selected 'Velocity'.
- If a limit switch is found "On" on rising 'Execute', then the process is started in the opposite direction as specified, the search is for the limit switch "Off" (or "On" depending in 'LimitSwitchMode' setting) edge (released), and the process is restarted in the original direction. This ensures that the end conditions are always the same.
- The torque is limited.
- Both 'TimeLimit and 'DistanceLimit' cause an error if exceeded.
- 'SetPosition' input disconnected: This FB doesn't modify the actual position.
- SetPosition' input connected: the FB modifies the actual position to the 'SetPosition' value when the homing condition is met.

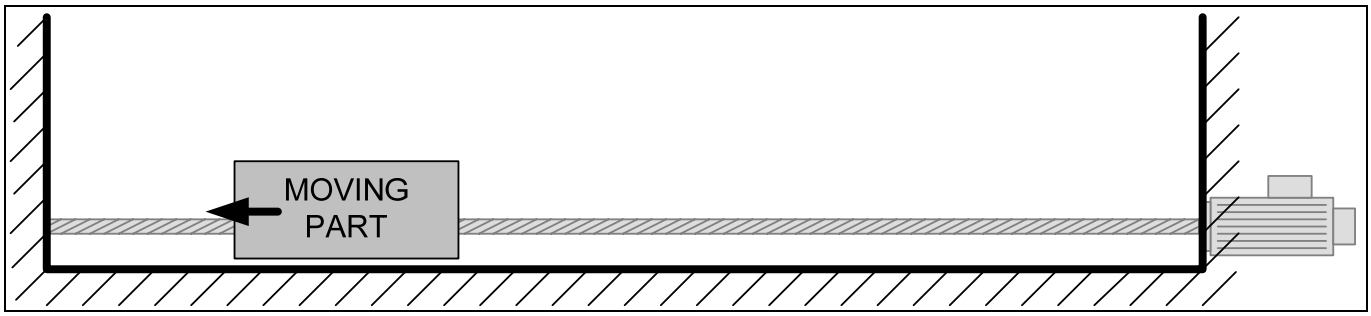
Example with the result in execution from 2 different starting points (squares).
Actual Position is not modified and the axis remains in homing state.



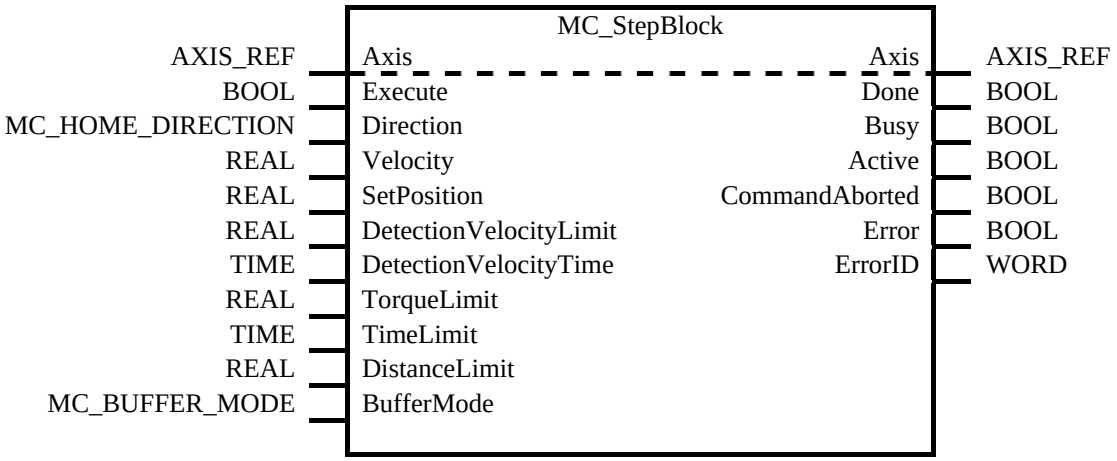
The same conditions, but forcing a different edge detection (mcFallingEdge). Note: the Actual Position is not modified.



3.3. MC_StepBlock

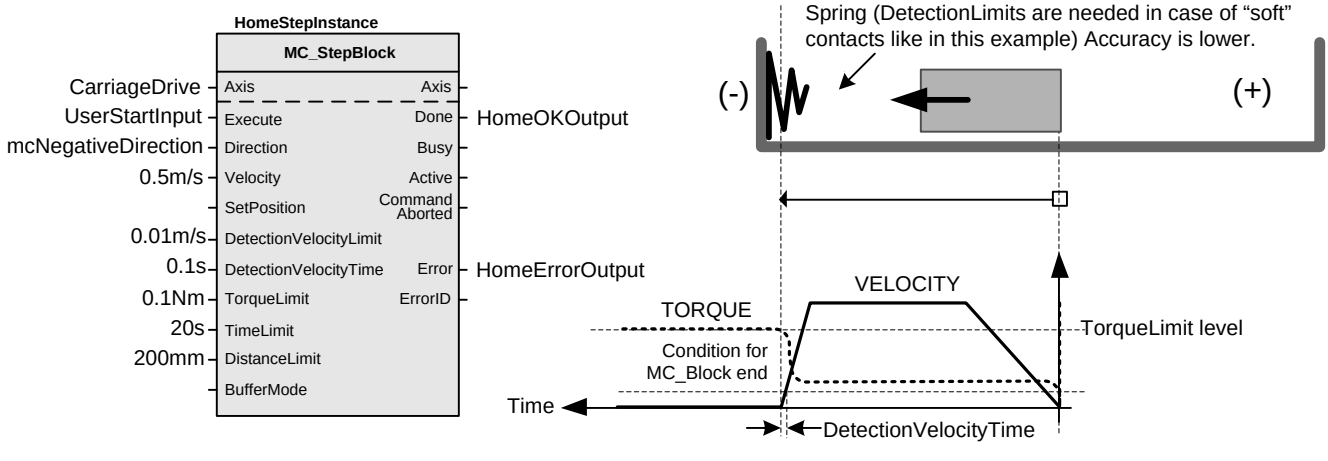


FB-Name		MC_StepBlock		
This Function Block performs the homing procedure HomeBlock against a physical object, mechanically blocking the movement. In this mode there is no limit switch or reference pulse. Adequate torque limits are required for not damaging mechanics during the homing process. The 'StepBlock' condition is that the torque limit is reached and the actual velocity falls below the value of the 'DetectionVelocityLimit' input for at least the 'DetectionVelocityTime'.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function step at rising edge
	E	Direction	MC_HOME_DIRECTION	Specifies the direction of the motion (if any): • mcPositiveDirection = Starts in positive direction always • mcNegativeDirection = Starts in negative direction always
	E	Velocity	REAL	Value of the velocity of motion (not necessarily reached) [u/s]
	E	SetPosition	REAL	Value of the position [u] to be set when conditions are met
	E	DetectionVelocityLimit	REAL	Limit value of the actual velocity if torque limit is reached in [u/s]
	E	DetectionVelocityTime	TIME	Minimum time that the 'TorqueLimit' and 'DetectionVelocityLimit' is reached. 0 means that the first torque and velocity reached is the accepted condition
	E	TorqueLimit	REAL	Maximum torque or force [in t.u.]. It has to be set to a value that doesn't overcome the mechanical blocking. 0 means no torque limit
	E	TimeLimit	TIME	If 'StepBlock' condition is not met in the 'TimeLimit', 'Error' is issued. 0 means no time limit
	E	DistanceLimit	REAL	If 'StepBlock' condition is not met within a 'DistanceLimit' travel, then 'Error' is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 'Aborting versus Buffered modes'
VAR_OUTPUT				
	B	Done	BOOL	StepBlock conditions are met. Torque limit is reached and actual velocity is below 'DetectionVelLimit' for at least the time as set at DetectionTime
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: --				

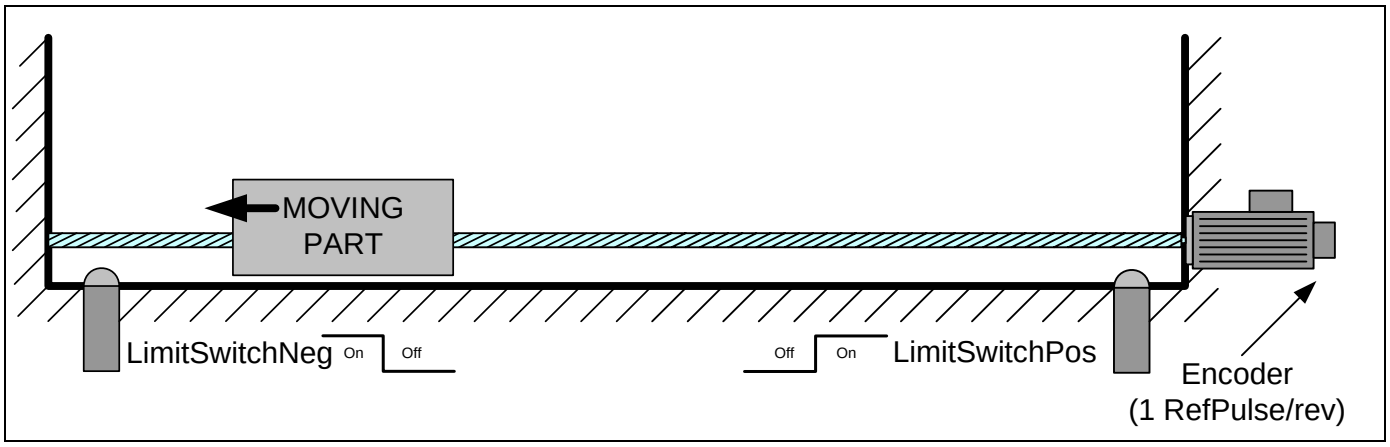


- Axis State is changed to 'Homing' (if not already in homing).
- The homing procedure is commanded by the user in the desired homing direction at the selected velocity
- The torque is limited.
- Both 'TimeLimit and 'DistanceLimit' cause an error if exceeded.
- Process is finished when torque is in limit condition and real velocity is below 'DetectionVelocityLimit' for the amount of 'DetectionVelocityTime'.
- 'SetPosition' input disconnected: This FB doesn't modify the actual position.
- 'SetPosition' input connected: This FB modifies the actual position to the 'SetPosition' value when the homing condition is met.

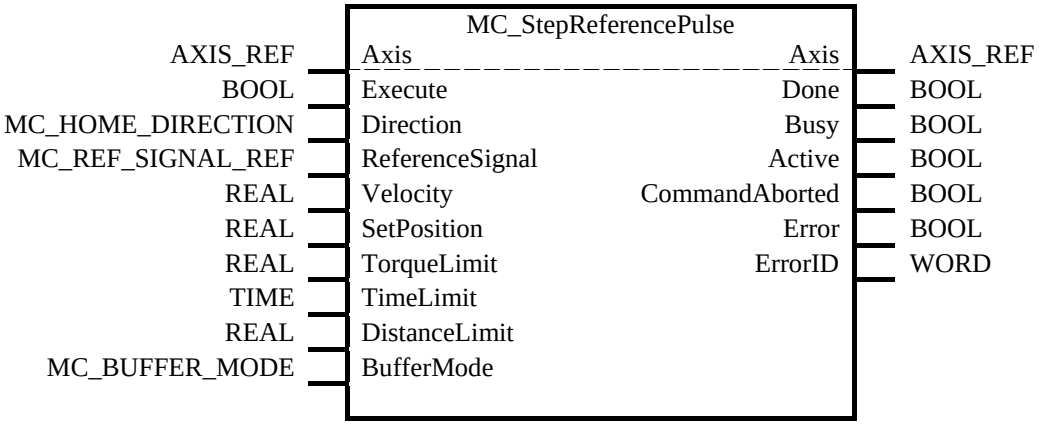
Example:



3.4. MC_StepReferencePulse

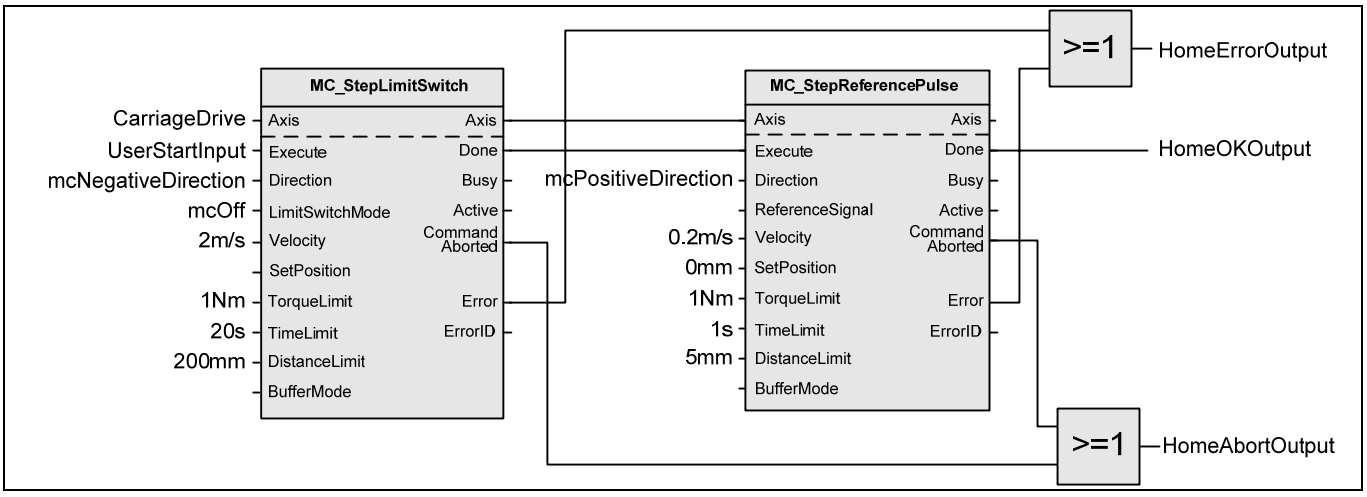


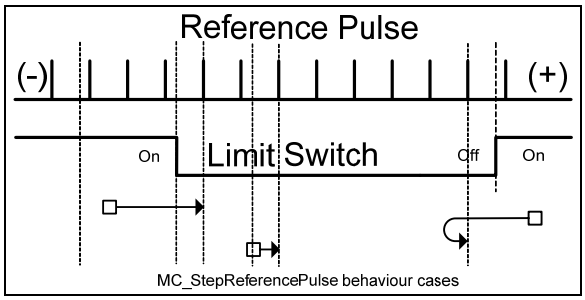
FB-Name		MC_StepReferencePulse		
This Function Block is the main part of the homing procedure HomeReferencePulse by searching for a reference pulse (also called marker or zero pulse) in the encoder. The reference pulse appears once per encoder revolution. The advantage in using a reference pulse for homing is the higher accuracy and precision that can be achieved compared to traditional optical, mechanical or magnetic sensors.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function step at rising edge
	E	Direction	MC_HOME_DIRECTION	Specifies the direction of the motion (if any): • mcPositiveDirection = Starts in positive direction always • mcNegativeDirection = Starts in negative direction always
	E	ReferenceSignal	MC_REF_SIGNAL_REF	References to the referencing switch with its associated data. See also 3.1 StepAbsoluteSwitch
	E	Velocity	REAL	Value of the velocity of motion (not necessarily reached) [u/s]
	E	SetPosition	REAL	Value of the position [u] to be set when conditions are met
	E	TorqueLimit	REAL	Maximum torque or force [in t.u.]. 0 means no torque limit
	E	TimeLimit	TIME	If StepRefPulse condition is not met in the 'TimeLimit', 'Error' is issued. 0 means no time limit
	E	DistanceLimit	REAL	If StepRefPulse condition is not met within a 'DistanceLimit' travel, then 'Error' is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 'Aborting versus Buffered modes'
VAR_OUTPUT				
	B	Done	BOOL	StepRefPulse conditions are met
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: -				



- The axis state is changed to 'Homing' (if not already in that mode).
- The 'Homing' is commanded by the user in the desired homing direction at the programmed velocity.
- At the first occurrence of the reference pulse, the procedure is finished.
- The torque is limited.
- Both 'TimeLimit and 'DistanceLimit' cause an error if exceeded.
- 'SetPosition' input is not connected: This FB doesn't modify the actual position.
- 'SetPosition' input is connected: This FB modifies the actual position to the 'SetPosition' value when the homing condition is met.

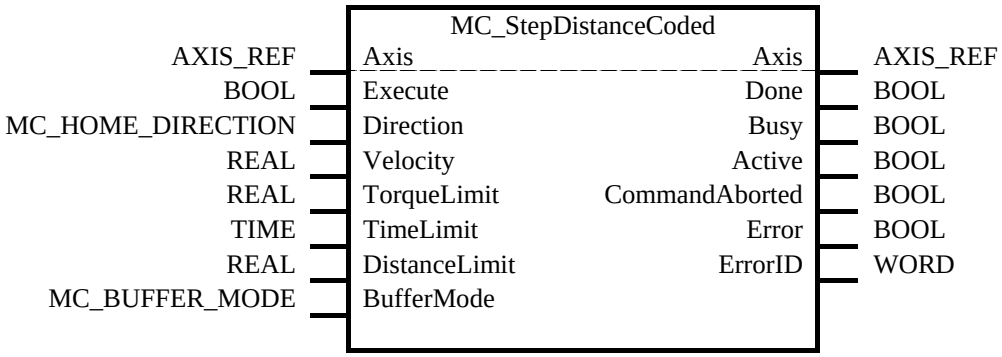
It is common that a first approach is performed against a mechanical sensor at a higher velocity, and after detecting a reference pulse continues at a lower velocity. This is a traditional 2-Step homing (coarse by external switch in reverse and fine by reference pulse in forward). For ease of use both functions could be grouped together in a single FB. The advantage of having separate function blocks is that any combination is possible (MC_StepBlock and after that MC_StepReferencePulse, etc.), stating different velocity and conditions for each step, providing a higher flexibility without increasing the homing FB complexity too much.

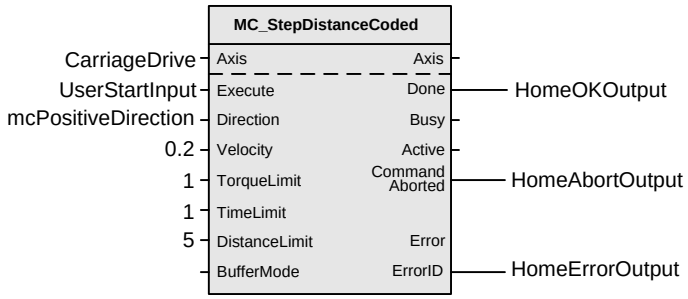


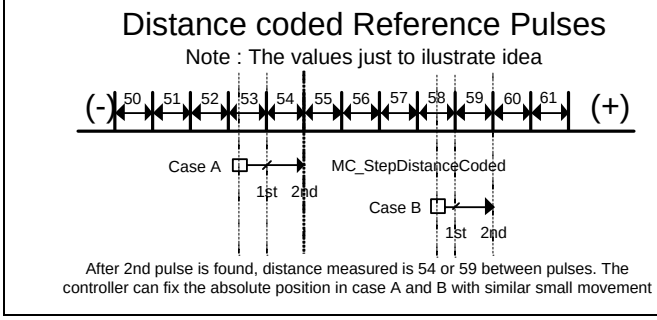


3.5. MC_StepDistanceCoded

FB-Name		MC_StepDistanceCoded		
This Function Block performs the homing procedure by moving in the specified direction until several coded reference marks are found. After calculation it sets the actual position. If there are no coded marks found within a certain time or distance traveled, it generates an error.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function step at rising edge
	E	Direction	MC_HOME_DIRECTION	Specifies the direction of the motion (if any): • mcPositiveDirection = Starts in positive direction always • mcNegativeDirection = Starts in negative direction always
	E	Velocity	REAL	Value of the velocity of motion (not necessarily reached) [u/s]
	E	TorqueLimit	REAL	Maximum torque or force [in t.u.]. 0 means no torque limit
	E	TimeLimit	TIME	If no valid reference marks are found within the 'TimeLimit', an 'Error' is issued. 0 means no time limit
	E	DistanceLimit	REAL	If no valid reference marks are found within a 'DistanceLimit' travel, then an 'Error' is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 'Aborting versus Buffered modes'
VAR_OUTPUT				
	B	Done	BOOL	Position is known and set
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: -				

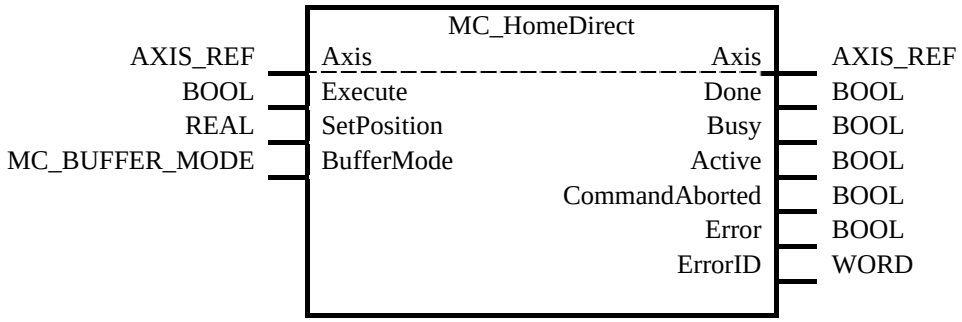






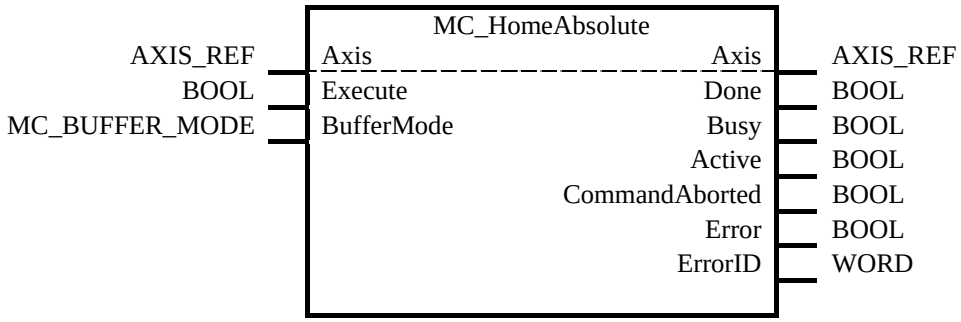
3.6. MC_HomeDirect

FB-Name		MC_HomeDirect		
This Function Block performs the static homing procedure HomeDirect by directly setting the actual position to the 'SetPosition' input value. No physical motion is performed in this mode. This is equivalent to a MC_SetPosition action, but in the 'Homing' state.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function at rising edge
	E	SetPosition	REAL	Value of the absolute position [u] to be set when homing is done
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 'Aborting versus Buffered modes'
VAR_OUTPUT				
	B	Done	BOOL	Position set
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes:				



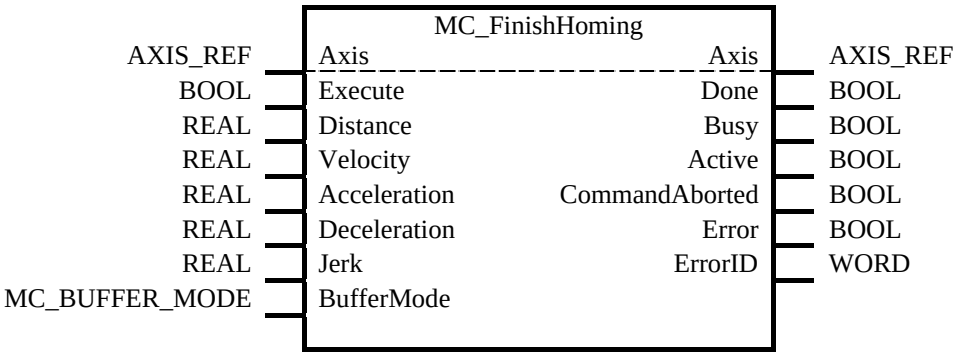
3.7. MC_HomeAbsolute

FB-Name			MC_HomeAbsolute	
This Function Block performs the static homing procedure HomeAbsolute by setting the actual position to the position read from an absolute encoder (via AXIS_REF). No physical motion is performed in this mode. Equivalent to MC_SetPosition with 'SetPosition' coming from absolute encoder reading but in the state 'Homing'.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function at rising edge
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 'Aborting versus Buffered modes'
VAR_OUTPUT				
	B	Done	BOOL	Absolute position set
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes:				

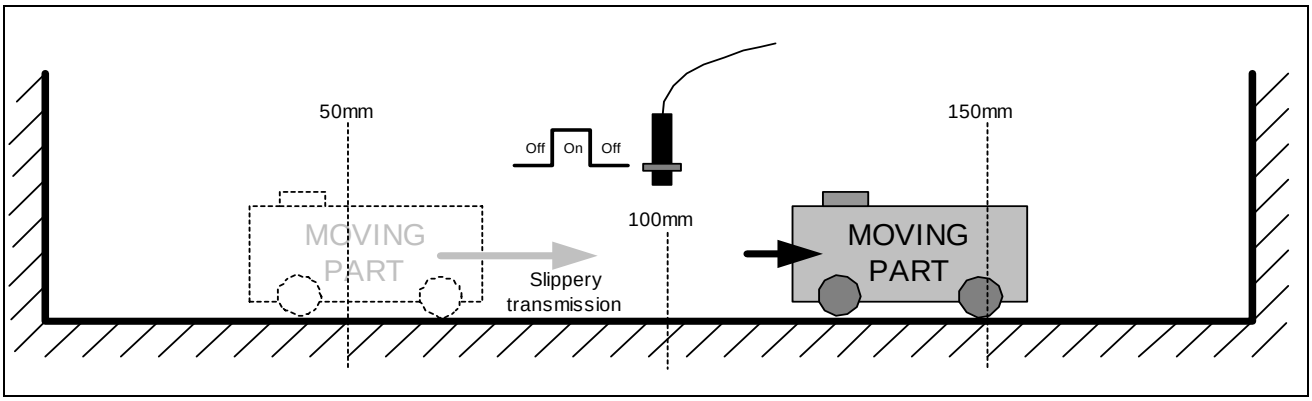


3.8. MC_FinishHoming

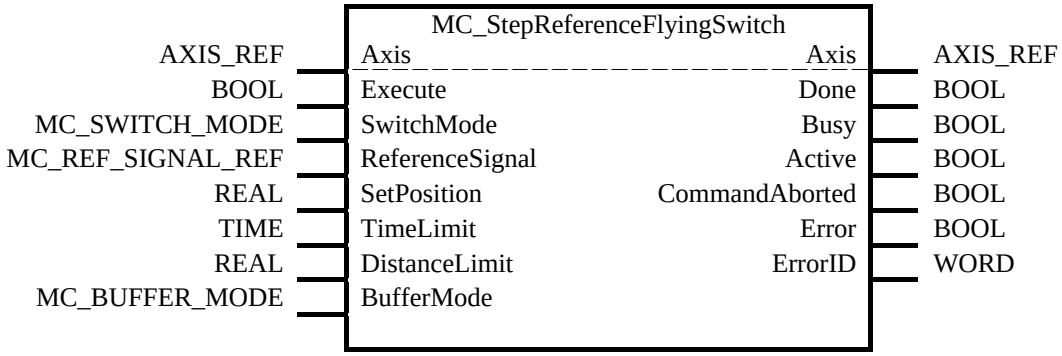
FB-Name		MC_FinishHoming		
This FB transfers an axis from the state 'Homing' to the state 'Standstill' and finalizes the homing procedure. In addition, it can do a relative movement to move the axis in the allowed area (working area) in case the homing steps left the axis on the wrong side of the switch.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Finishes or aborts the homing step function at the rising edge
	B	Distance	REAL	Relative distance for the motion (in technical unit [u])
	E	Velocity	REAL	Value of the maximum velocity (not necessarily reached) [u/s]
	E	Acceleration	REAL	Value of the 'Acceleration' (increasing energy of the motor) [u/s^2]
	E	Deceleration	REAL	Value of the 'Deceleration' (decreasing energy of the motor) [u/s^2]
	E	Jerk	REAL	Value of the 'Jerk' [u/s^3]
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 'Aborting versus Buffered modes'
VAR_OUTPUT				
	B	Done	BOOL	FinishHoming conditions are met
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: --				



3.9. MC_StepReferenceFlyingSwitch



FB-Name			MC_StepReferenceFlyingSwitch	
This function block serves for referencing a position on the fly while a movement is in progress. In this mode, no motion is started or modified; the axis state is not changed.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the homing function step at rising edge
	E	SwitchMode	MC_SWITCH_MODE	Sensor condition to finalize the process : • mcOn = When sensor is “On” • mcOff = When sensor is “Off” • mcRisingEdge = When “Off” to “On” transition in sensor • mcFallingEdge = When “On” to “Off” transition in sensor • mcEdgeSwitchPositive = Edge depends on motion direction • mcEdgeSwitchNegative = Like previous but opposite
	E	ReferenceSignal	MC_REF_SIGNAL_REF	References to the referencing switch with its associated data. See also 3.1 StepAbsoluteSwitch
	E	SetPosition	REAL	Value of the position [u] to be set when conditions are met
	E	TimeLimit	TIME	If StepReferenceFlyingSwitch condition is not met in the ‘TimeLimit’, ‘Error’ is issued. 0 means no time limit
	E	DistanceLimit	REAL	If StepReferenceFlyingSwitch condition is not met within a ‘DistanceLimit’ travel, then ‘Error’ is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 ‘Aborting versus Buffered modes’
VAR_OUTPUT				
	B	Done	BOOL	StepReferenceFlyingSwitch conditions are met
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	‘Command’ is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: MC_StepReferenceFlyingSwitch and MC_StepReferenceFlyingRefPulse are special modes. For this reason the name is ‘Reference’ instead of ‘Homing’. They can be considered as special types of MC_TouchProbe functionality, waiting for the specified switch conditions to happen, in order to reference the axis exactly in that point to a new coordinate set by MC_SetPosition command. This function especially benefits from a direct implementation in a single FB, as a fast reaction from the hardware is required.				

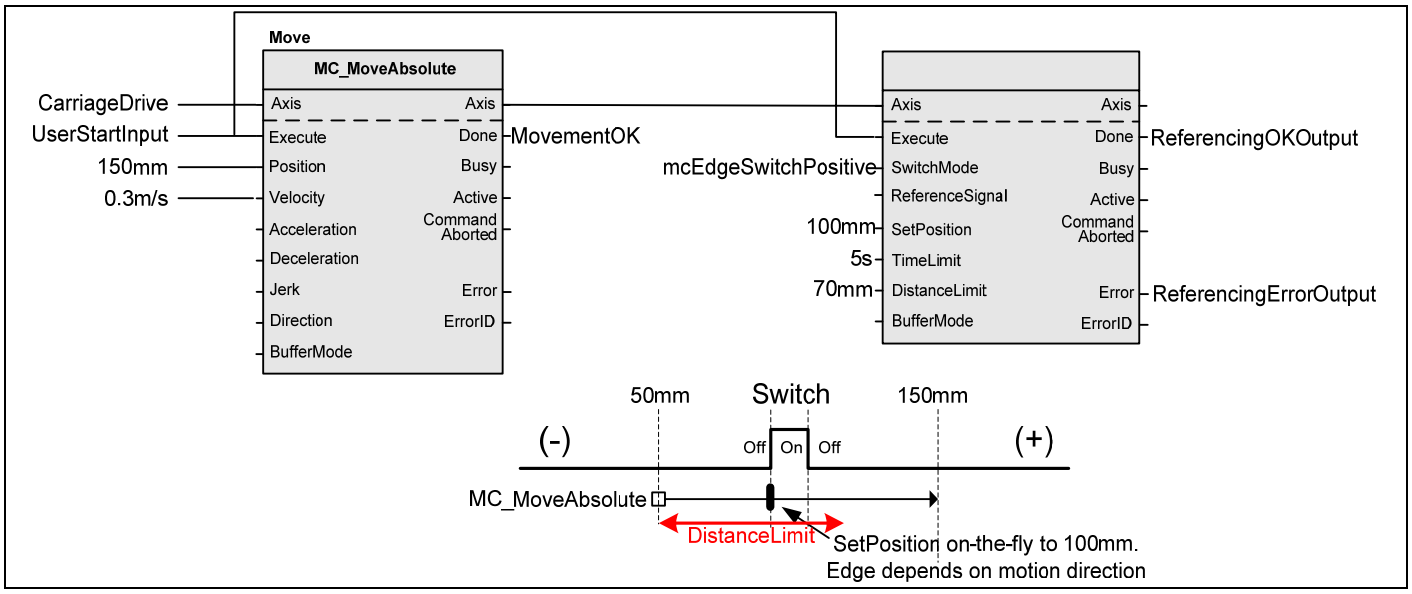


This FB is of use when a flying reference is required in running machines were a separate homing procedure is not allowed, and 'Homing' has to be performed "on the fly". Also applicable for machines were non-rigid transmissions allow for uncontrollable slipping of the real position versus the feedback from the controller (for example, a wheel based or flat belt transmissions, where the risk of slip is high)

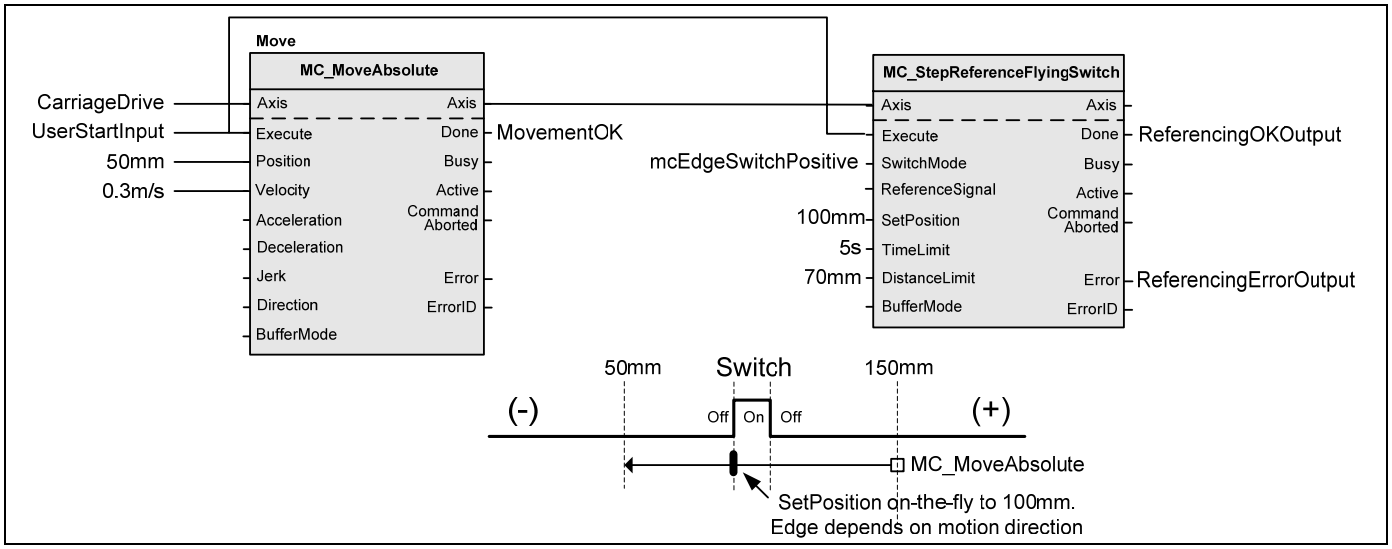
- Actual motion state and command are kept in execution (including other home commands)
- 'TimeLimit' and 'DistanceLimit' can cause an 'Error' if exceeded from the point were the MC_StepReferenceFlyingSwtich command is issued.
- When switch conditions are met, a 'SetPosition' on the fly is performed, without changing absolute targets of movement.
- Being on top of the switch when 'Execute' is commanded is not considered as edge.
- This Function Block does not modify the axis state.

The switch conditions as mcEdgeSwitchPositive and mcEdgeSwitchNegative are useful to catch always the same point as in the example. This mode can cause a strange behavior in some types of movement, like MC_MoveAbsolute, if the real and expected positions are too far away. Typically some kind of windowing prior to the activation of the command is recommended in this case. Then the 'TimeLimit' and 'DistanceLimit' can identify if we have a referencing problem.

Example 1: The reference is done while MC_MoveAbsolute moves from position 50 to 150. The switch target edge at 100mm.



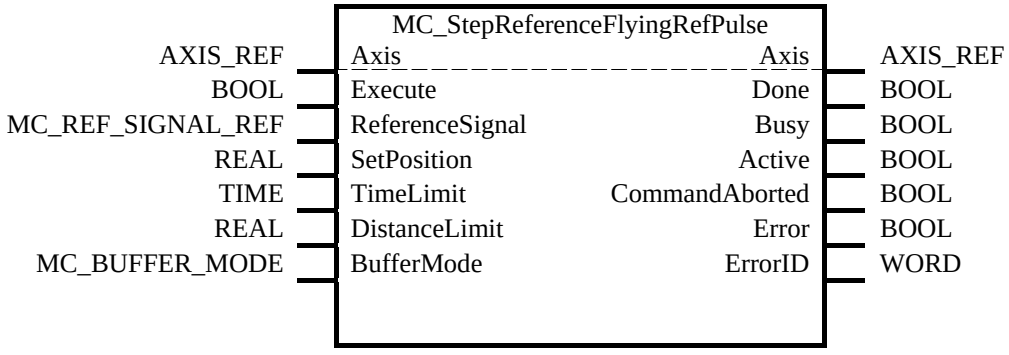
Example 2: The same MC_StepReferenceFlyingSwitch command is used but with a MC_MoveAbsolute from 150 to 50. mcEdgeSwitchPositive helps to set the position always on the same edge of the switch, regardless of the direction of the motion.



3.10. MC_StepReferenceFlyingRefPulse

This FB is similar to MC_StepReferenceFlyingSwitch but for the referencing it uses the reference pulse from an encoder (often called “marker pulse” or “zero pulse”), which is often coupled to a rotary axis, making it suitable for referencing to an angle. This FB does not need the input ‘SwitchMode’ since the pulse conditions are well defined and its narrow size eliminates the need of edge detection.

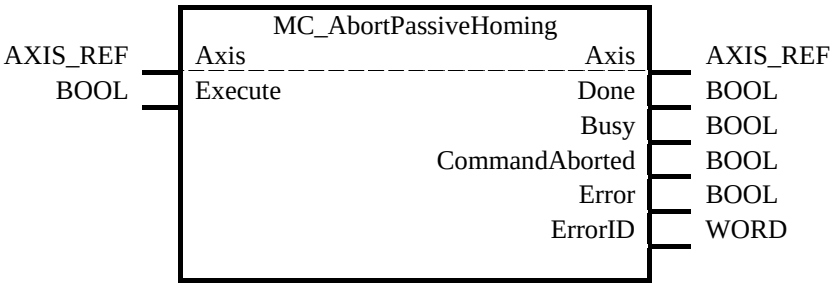
FB-Name		MC_StepReferenceFlyingRefPulse		
This function Block serves for referencing a position on the fly while a movement is in progress. In this mode, no motion is started or modified. The axis state is not changed.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Starts the referencing function at rising edge
	E	ReferenceSignal	MC_REF_SIGNAL_REF	References to the referencing switch with its associated data. See also 3.1 StepAbsoluteSwitch
	E	SetPosition	REAL	Value of the position [u] to be set when conditions are met
	E	TimeLimit	TIME	If ReferenceFlyingRefPulse condition is not met in the ‘TimeLimit’, ‘Error’ is issued. 0 means no time limit
	E	DistanceLimit	REAL	If ReferenceFlyingRefPulse condition is not met within a ‘DistanceLimit’ travel, then ‘Error’ is issued. 0 means no distance limit.
	E	BufferMode	MC_BUFFER_MODE	Defines the chronological sequence of the FB. See Part 1, V 2.0 par. 2.4.2 ‘Aborting versus Buffered modes’
VAR_OUTPUT				
	B	Done	BOOL	ReferenceFlyingRefPulse conditions are met
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	Active	BOOL	Indicates that the FB has control on the axis
	E	CommandAborted	BOOL	‘Command’ is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: MC_StepReferenceFlyingSwitch and MC_StepReferenceFlyingRefPulse are special modes. They can be considered as a kind of MC_TouchProbe functionality, waiting for the specified reference pulse conditions to happen, in order to reference the axis exactly in that point at a new coordinate set by Set Position on-the-fly. These functions especially benefit from an implementation in a single FB, as a fast reaction is required.				



- Actual motion state and motion command are kept in execution (including other home commands)
- Time and Distance Limits can cause an ‘Error’ if exceeded from the point were the MC_StepReference-FlyingRefPulse command is issued.
- When the RefPulse is found, a ‘SetPosition’ on the fly is performed, without changing absolute targets of movement.
- This Function Block does not modify the state diagram.

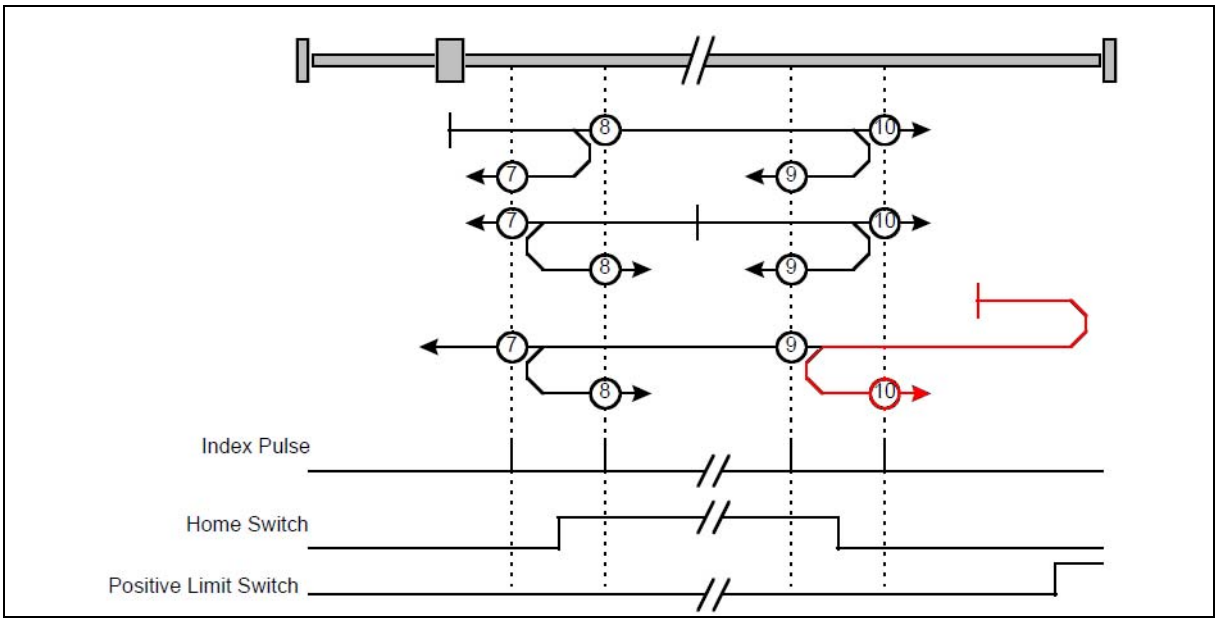
3.11. MC_AbortPassiveHoming

FB-Name			MC_AbortPassiveHoming	
This FB aborts the passive homing mode. It does not perform any movement nor generates a state transition.				
VAR_IN_OUT				
	B	Axis	AXIS_REF	Reference to the axis
VAR_INPUT				
	B	Execute	BOOL	Finishes or aborts the homing function at the rising edge
VAR_OUTPUT				
	B	Done	BOOL	Homing finished or aborted
	E	Busy	BOOL	The FB is not finished and new output values are to be expected
	E	CommandAborted	BOOL	'Command' is aborted by another command
	B	Error	BOOL	Signals that an error has occurred within the Function Block
	E	ErrorID	WORD	Error identification
Notes: --				

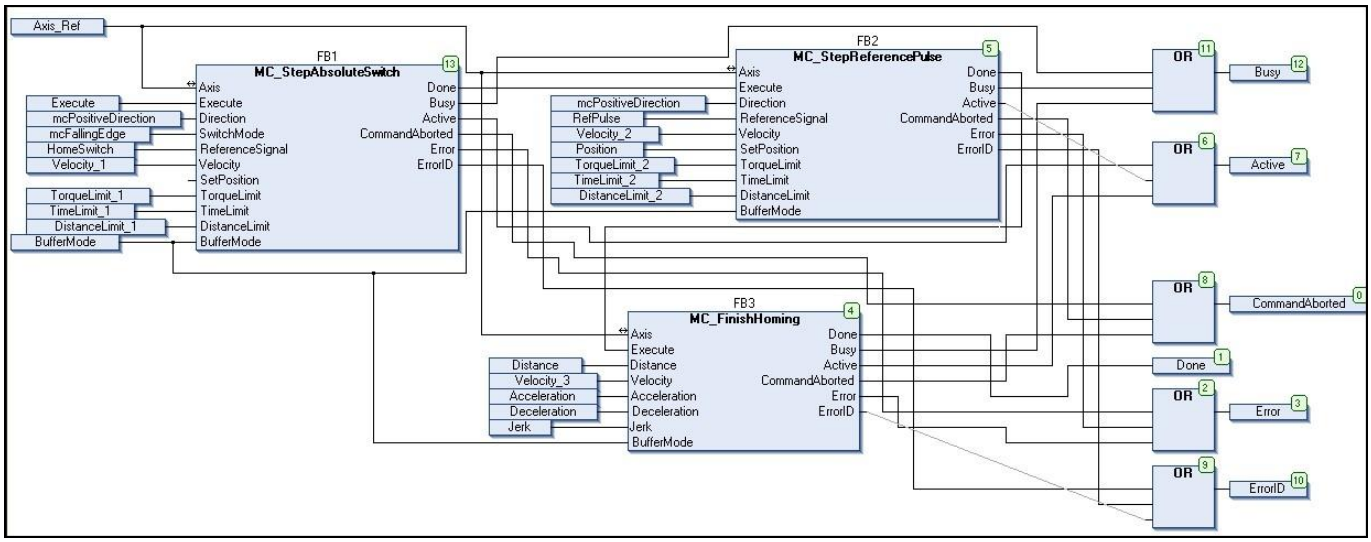


4 Example of creating a dedicated homing function block

In this example a homing procedure on a home switch and an index pulse is used. This method uses a home switch, which is active over only a portion of the travel; in effect the switch has a 'momentary' action as the axis's position sweeps past the switch. There are two approaches: the initial direction movement is to the right (here shown as 7 to 10) or to the left (not shown here), except if the home switch is active at the start of the motion. In that case, the initial direction of motion can be dependent on the edge being sought. The home position shall be at the index pulse on either side of the rising or falling edges of the home switch, as shown in the figure below. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch (shown in red).



Possible solution:
By combining a MC_StepAbsoluteSwitch and a MC_StepReferencePulse one can implement the functionality. After finalizing MC_StepReferencePulse one initiates MC_FinishHoming to position the axis in the usable area between the switches. One can combine the functionality as shown in a user derived homing function block, dedicated to this homing procedure.



5 Compliance Procedure and Compliance List

Listed in this chapter are the requirements for the compliance statement from the supplier of the Motion Control Function Blocks for Part 5 – Homing Procedures. The compliance statement consists of a list of the Function Blocks (see 5.3 Overview of the Homing Procedures and Function Blocks and its paragraphs) and a detailed list of the supported Function Blocks in combination with the applicable inputs and outputs (see 5.4 MC_StepAbsoluteSwitch and further paragraphs). The supplier is required to fill out the tables for the Function Blocks, according to their product, committing their support to the specification.

By submitting these tables to PLCopen, and after approval by PLCopen, the list will be published on the PLCopen website, www.plcopen.org , as well as a short form overview, as specified in see 5.3 Overview of the Homing Procedures and Function Blocks and its paragraphs.

In addition to this approval, the supplier is granted access and usage rights of the PLCopen Motion Control logo, as described in chapter 5.15 The PLCopen Motion Control Logo and Its Usage.

5.1. Statement of Supplier

Supplier name	
Supplier address	
City	
Country	
Telephone	
Fax	
Email address	
Product Name	
Product version	
Release date	

I hereby state that the following tables as filled out and submitted do match our product as well as the accompanying user manual, as stated above.

Name of representation (person):

Date of signature (dd/mm/yyyy):

Signature:

5.2. Supported Data types

Defined datatypes with MC library:	Supported	If not supported, which datatype used
BOOL		
WORD		
REAL		
ENUM		
TIME		

Table 1: Supported datatypes

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

Derived datatypes:	Where used	Supported	Which structure
AXIS_REF	In all FBs		
MC_HOME_DIRECTION (extended)	MC_StepAbsoluteSwitch MC_StepLimitSwitch MC_StepBlock MC_StepReferencePulse MC_StepDistanceCoded		
MC_SWITCH_MODE (extended)	MC_StepAbsoluteSwitch MC_StepLimitSwitch MC_StepReferenceFlyingSwitch		
MC_REF_SIGNAL_REF (extended)	MC_StepAbsoluteSwitch MC_StepReferencePulse MC_StepReferenceFlyingSwitch MC_StepReferenceFlyingRefPulse		
MC_BUFFER_MODE (extended)	MC_StepAbsoluteSwitch MC_StepLimitSwitch MC_StepBlock MC_StepReferencePulse MC_StepDistanceCoded MC_HomeDirect MC_HomeAbsolut MC_FinishHoming MC_StepReferenceFlyingSwitch MC_StepReferenceFlyingRefPulse		

Table 2: Supported derived datatypes

5.3. Overview of the Homing Procedures and Function Blocks

Function Blocks	Supported Yes / No	Comments (<= 48 char.)
MC_StepAbsoluteSwitch		
MC_StepLimitSwitch		
MC_StepBlock		
MC_StepReferencePulse		
MC_StepDistanceCoded		
MC_HomeDirect		
MC_HomeAbsolute		
MC_FinishHoming		
MC_StepReferenceFlyingSwitch		
MC_StepReferenceFlyingRefPulse		
MC_AbortPassiveHoming		

Table 3: Short overview of the Function Blocks

5.4. MC_StepAbsoluteSwitch

If Supported	**MC_StepAbsoluteSwitch**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	Direction		
E	SwitchMode		
E	ReferenceSignal		
E	Velocity		
E	SetPosition		
E	TorqueLimit		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.5. MC_StepLimitSwitch

If Supported	**MC_StepLimitSwitch**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	Direction		
E	LimitSwitchMode		
E	Velocity		
E	SetPosition		
E	TorqueLimit		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.6. MC_StepBlock

If Supported	**MC_StepBlock**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	Direction		
E	Velocity		
E	SetPosition		
E	DetectionVelocityLimit		
E	DetectionVelocityTime		
E	TorqueLimit		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.7. MC_StepReferencePulse

If Supported	**MC_StepReferencePulse**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	Direction		
E	ReferenceSignal		
E	Velocity		
E	SetPosition		
E	TorqueLimit		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.8. MC_StepDistanceCoded

If Supported	**MC_StepDistanceCoded**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	Direction		
E	Velocity		
E	TorqueLimit		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.9. MC_HomeDirect

If Supported	**MC_HomeDirect**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	SetPosition		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.10. MC_HomeAbsolute

If Supported	**MC_HomeAbsolute**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.11. MC_FinishHoming

If Supported	**MC_FinishHoming**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
B	Distance		
E	Velocity		
E	Acceleration		
E	Deceleration		
E	Jerk		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.12. MC_StepReferenceFlyingSwitch

If Supported	**MC_StepReferenceFlyingSwitch**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	SwitchMode		
E	ReferenceSignal		
E	SetPosition		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.13. MC_StepReferenceFlyingRefPulse

If Supported	**MC_StepReferenceFlyingRefPulse**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
E	ReferenceSignal		
E	SetPosition		
E	TimeLimit		
E	DistanceLimit		
E	BufferMode		
VAR_OUTPUT			
B	Done		
E	Busy		
E	Active		
E	CommandAborted		
B	Error		
E	ErrorID		

5.14. MC_AbortPassiveHoming

If Supported	**MC_AbortPassiveHoming**	Sup.Y/N	Comments
VAR_IN_OUT			
B	Axis		
VAR_INPUT			
B	Execute		
VAR_OUTPUT			
B	Done		
E	Busy		
E	CommandAborted		
B	Error		
E	ErrorID		

5.15. The PLCopen Motion Control Logo and Its Usage

For quick identification of compliant products, PLCopen has developed a logo for the motion control Function Blocks:

Figure 1: The PLCopen Motion Control Logo

This motion control logo is owned and trademarked by PLCopen.

In order to use this logo free-of-charge, the relevant company has to fulfill all the following requirements:

1. the company has to be a voting member of PLCopen;
2. the company has to comply with the existing specification, as specified by the PLCopen Task Force Motion Control, and as published by PLCopen, and of which this statement is a part;
3. this compliance application is provided in written form by the company to PLCopen, clearly stating the applicable software package and the supporting elements of all the specified tables, as specified in the document itself;
4. in case of non-fulfillment, which has to be decided by PLCopen, the company will receive a written statement concerning this from PLCopen. The company will have a one month period to either adopt their software package in such a way that it complies, represented by the issuing of a new compliance statement, or remove all reference to the specification, including the use of the logo, from all their specification, be it technical or promotional material;
5. the logo has to be used as is - meaning the full logo. It may be altered in size providing the original scale and color setting is kept.
6. the logo has to be used in the context of Motion Control.