User Manual

Weintek Library

This manual walks through the steps to install Weintek function library, and explains the functions.

V1.0.2

Table of Contents

Other company names, product names, or trademarks in this document are the trademarks or registered trademarks of their respective companies.

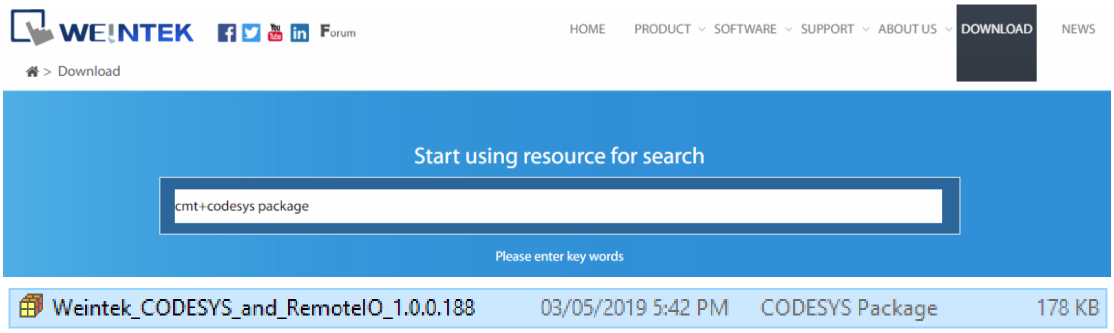
This document is subject to change without notice.

1. Overview

This manual explains some functions in Weintek Library. New functions and function blocks will be added in the future when needed.

2. Installing Weintek Library

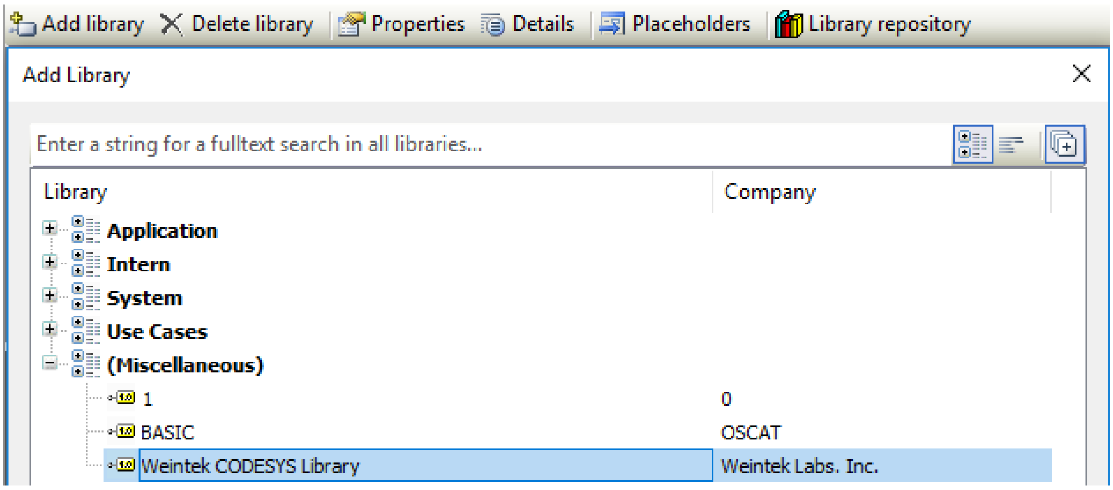
Step 1. Visit Weintek official website and download Weintek_CODESYS_and_RemoteIO package.
Download link:
https://www.weintek.com/globalw/Download/Download.aspx
The version of the installation file should be 1.0.0.188 or later, and Weintek_CODESYS_Library is automatically installed in CODESYS.



cmt+codesys package

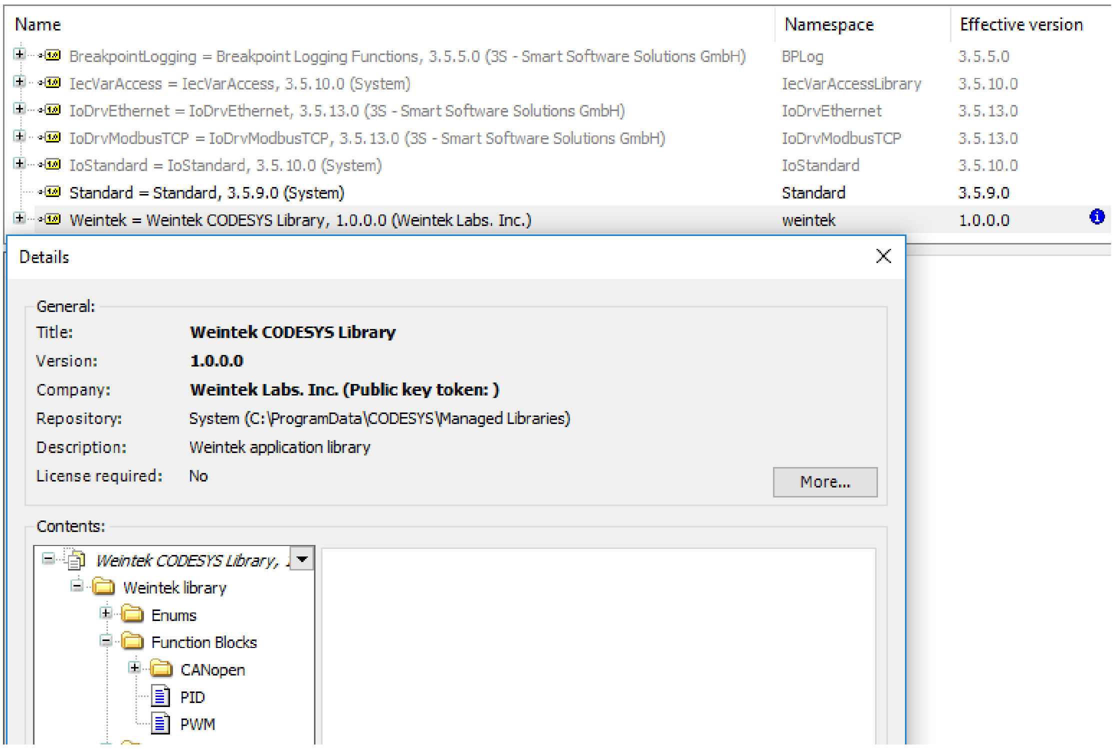
Weintek_CODESYS_and_RemoteIO_1.0.0.188 03/05/2019 5:42 PM CODESYS Package 178 KB

Step 2. Weintek CODESYS Library is added in [Add library] » (Miscellaneous).



Library	Company
Application	
Intern	
System	
Use Cases	
(Miscellaneous)	
1	0
BASIC	OSCAT
Weintek CODESYS Library	Weintek Labs. Inc.

Step 3. Open [Details] and see Function Blocks.



Name	Namespace	Effective version
BreakpointLogging = Breakpoint Logging Functions, 3.5.5.0 (3S - Smart Software Solutions GmbH)	BPLog	3.5.5.0
IecVarAccess = IecVarAccess, 3.5.10.0 (System)	IecVarAccessLibrary	3.5.10.0
IoDrvEthernet = IoDrvEthernet, 3.5.13.0 (3S - Smart Software Solutions GmbH)	IoDrvEthernet	3.5.13.0
IoDrvModbusTCP = IoDrvModbusTCP, 3.5.13.0 (3S - Smart Software Solutions GmbH)	IoDrvModbusTCP	3.5.13.0
IoStandard = IoStandard, 3.5.10.0 (System)	IoStandard	3.5.10.0
Standard = Standard, 3.5.9.0 (System)	Standard	3.5.9.0
Weintek = Weintek CODESYS Library, 1.0.0.0 (Weintek Labs. Inc.)	weintek	1.0.0.0

Details

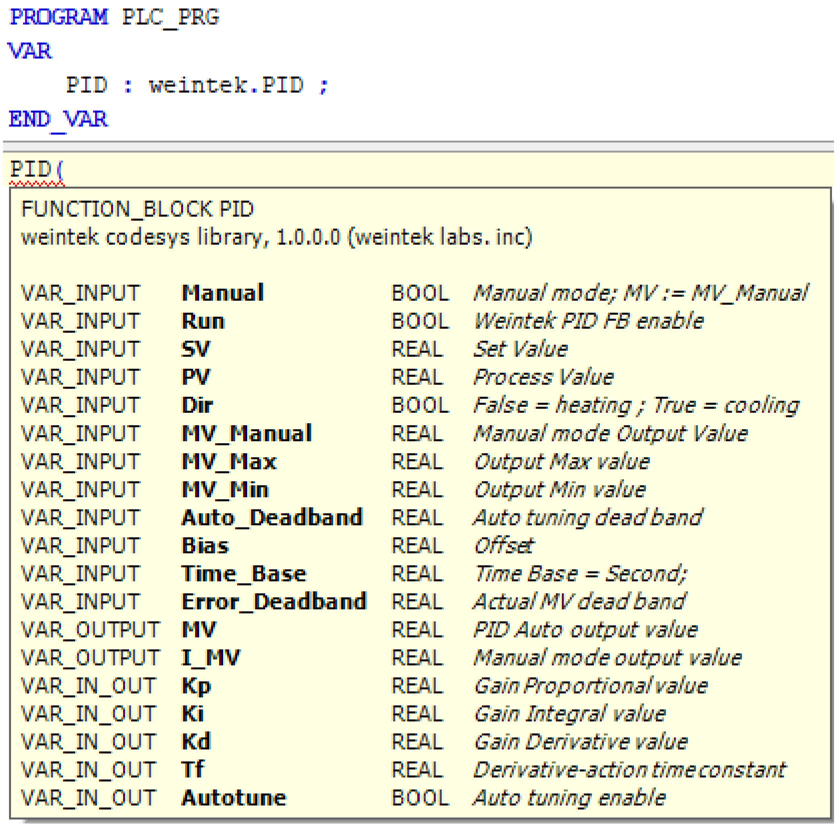
General:

Title:	**Weintek CODESYS Library**
Version:	**1.0.0.0**
Company:	**Weintek Labs. Inc. (Public key token:)**
Repository:	System (C:\ProgramData\CODESYS\Managed Libraries)
Description:	Weintek application library
License required:	No

Contents:

- *Weintek CODESYS Library, 1*
  - Weintek library
    - Enums
    - Function Blocks
      - CANopen
      - PID
      - PWM

Step 4. Function Block declaration in the program.



```
PROGRAM PLC_PRG
VAR
    PID : weintek.PID ;
END_VAR
```

```
PID(
```

```
FUNCTION_BLOCK PID
weintek codesys library, 1.0.0.0 (weintek labs. inc)

VAR_INPUT    Manual          BOOL   Manual mode; MV := MV_Manual
VAR_INPUT    Run             BOOL   Weintek PID FB enable
VAR_INPUT    SV              REAL   Set Value
VAR_INPUT    PV              REAL   Process Value
VAR_INPUT    Dir             BOOL   False = heating ; True = cooling
VAR_INPUT    MV_Manual       REAL   Manual mode Output Value
VAR_INPUT    MV_Max          REAL   Output Max value
VAR_INPUT    MV_Min          REAL   Output Min value
VAR_INPUT    Auto_Deadband   REAL   Auto tuning dead band
VAR_INPUT    Bias            REAL   Offset
VAR_INPUT    Time_Base       REAL   Time Base = Second;
VAR_INPUT    Error_Deadband  REAL   Actual MV dead band
VAR_OUTPUT   MV              REAL   PID Auto output value
VAR_OUTPUT   I_MV            REAL   Manual mode output value
VAR_IN_OUT   Kp              REAL   Gain Proportional value
VAR_IN_OUT   Ki              REAL   Gain Integral value
VAR_IN_OUT   Kd              REAL   Gain Derivative value
VAR_IN_OUT   Tf              REAL   Derivative-action time constant
VAR_IN_OUT   Autotune        BOOL   Auto tuning enable
```

3. PID commands

3.1. PID(FB)

Function: Proportional–Integral–Derivative controller

Implementing PID algorithm will start when “Run” is true.

When “Run” and “AutoTune” are TRUE, the parameters are tuned automatically, and “AutoTune” turns to FALSE after tuning is completed.

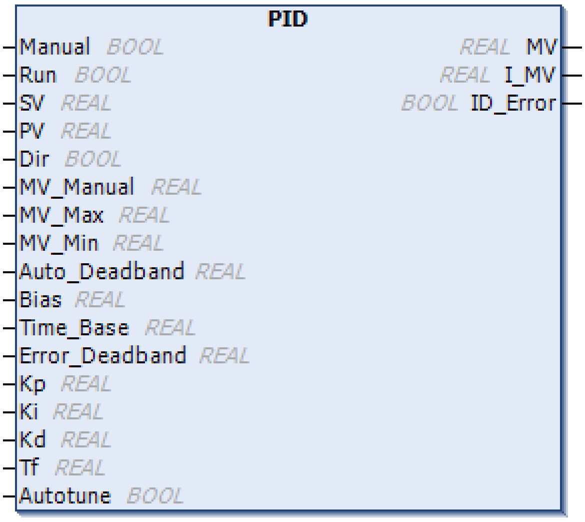
“Dir” is FALSE when SV > PV, “Dir” is TRUE when SV < PV

“Time_Base” cannot be set to 0.

Formula: $\mathrm{MV} = K_p E + K_i \int_0^t E dt + K_d \frac{dE}{dt} + \mathrm{BIAS}$

$$E = \mathrm{SV} - \mathrm{PV}, \text{when Dir} = \mathrm{FALSE}$$

$$E = \mathrm{PV} - \mathrm{SV}, \text{when Dir} = \mathrm{TRUE}$$



Parameters:

Input Parameters	Data Type	Definition	Description

Manual	BOOL	Manual Mode	TRUE= Mout (manual output)，
			FALSE= Implement PID algorithm

Run	BOOL	Enabling Function Block	
SV	REAL	Target Value	
PV	REAL	Present Value	

Dir	BOOL	PID Forward/Reverse Direction	FALSE=heating action/
			TRUE=cooling action

MV Manual	REAL	Manual Output	
MV_Max	REAL	Maximum Output Value	The upper limit of output value.
MV_Min	REAL	Minimum Output Value	The lower limit of output value.
Auto_Deadband	REAL	Auto Tuning Non-aciton Zone	Enabled when SV±Auto_Deadband is in the non-action zone.
BIAS	REAL	Feed Forward Output Value	

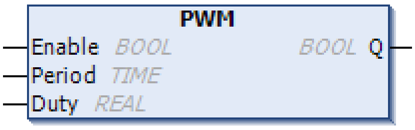
Time_Base	REAL	Sampling Time	Unit= Second(s)
Error_Deadband	REAL	Deadband	Range within which the Output value is counted as zero. SV±Error_Deadband
In-Out Parameters	**Data Type**	**Definition**	**Description**
Tf	REAL	Derivate-action Time Constant	Discrete MV of derivative term: $MV_d = \frac{T_f * MV_{d(last\ cycle)} + K_d * E}{T_f + T_s}$ MV_d= Discrete MV of derivative term T_s= Time_Base
Kp	REAL	Proportional Coefficient	
Ki	REAL	Integral Coefficient	
Kd	REAL	Derivative Coefficient	
Autotune	BOOL	PID Control Mode	The parameters are tuned automatically for the temperature control. PID Controller will start controlling when the tuning of the parameters is completed and is filled in with appropriate parameters (Tf, Kp, Ki, Kd)
Output Parameters	**Data Type**	**Definition**	**Description**
MV	REAL	Manipulated Value	
I_MV	REAL	Accumulated Integral Value	
ID_Error	BOOL	Invalid ID of Device	Using a CODESYS controller that is not a Weintek product to execute PID Function Block may result in error.

*Please find the iR_Application_Oven_Demo project for more information on how to use these functions.

3.2. PWM(FB)

Function: Output PWM signal when “Enable” is TRUE

The time when “Q” is TRUE = “Period” * “Duty”



Parameters:

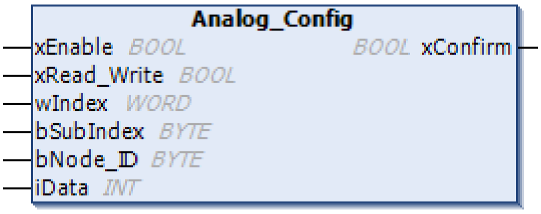
Input Parameters	Data Type	Definition	Description
Enable	BOOL	Enabling the PWM function block	Activation by TRUE
Period	TIME	Time Period	A TRUE then FALSE per cycle
Duty	REAL	Ratio of output duration in a cycle	Range between 0~100%
Output Parameters	**Data Type**	**Definition**	**Description**
Q	BOOL	Output	Output starting at TRUE

4. iR-COP commands

4.1. Analog_Config(FB)

Function: Read/Write an Analog module’s parameters when “xEnable” turns from FALSE to TRUE.

This Function Block can only be used to Read/Write an analog module that is connected to iR-COP.



Parameters:

Input Parameters	Data Type	Definition	Description
xEnable	BOOL	Enable	Triggered by FALSE to TRUE

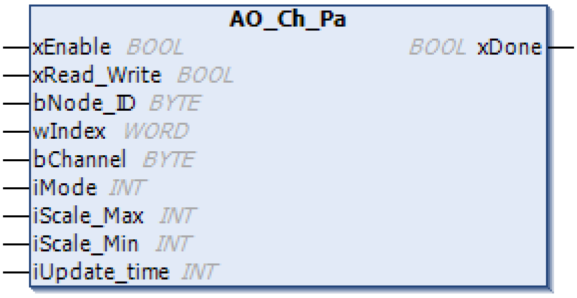
xRead_Write	BOOL	Switch between Read/Write	TRUE=Write
			FALSE=Read

wIndex	WORD	Index	Index of Object Dictionary
bSubIndex	BYTE	Sub-index	Sub Index of Object Dictionary
bNode_ID	BYTE	Node ID	Node ID of iR-COP
Output Parameters	**Data Type**	**Definition**	**Description**
xConfirm	BOOL	Completed	Read/Write operation completed
In-Out Parameters	**Data Type**	**Definition**	**Description**
iData	INT	Data	Data being read or written

4.2. AO_Ch_Pa(FB)

Function: Read/Write an Analog output channel when “xEnable” turns from FALSE to TRUE.

This Function Block can only be used to Read/Write an analog module that is connected to iR-COP. Applicable for: AQ04-VI, AM06-VI.



Input Parameters	Data Type	Definition	Description
xEnable	BOOL	Enable	Triggered by FALSE to TRUE

xRead_Write	BOOL	Switch between Read/Write	TRUE=Write
			FALSE=Read

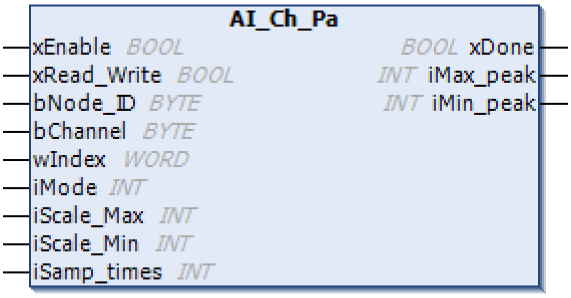
bNode_ID	BYTE	Node ID	Node ID of iR-COP
wIndex	WORD	Index	Index of Object Dictionary
bChannel	BYTE	Channel	Analog Channel no. 0~3
Output Parameters	**Data Type**	**Definition**	**Description**

xDone	BOOL	Completed	Function completed
In-Out Parameters	**Data Type**	**Definition**	**Description**
iMode	INT	Channel Mode	
iScale_Max	INT	Maximum channel scale	
iScale_Min	INT	Minimum channel scale	
iUpdate_Time	INT	Channel update time	

4.3. AI_Ch_Pa(FB)

Function: Read/Write an Analog input channel when “xEnable” turns from FALSE to TRUE.

This Function Block can only be used to Read/Write an analog module that is connected to iR-COP. Applicable for: AI04-VI, AM06-VI.



Parameters:

Input Parameters	Data Type	Definition	Description
xEnable	BOOL	Enable	Triggered by FALSE to TRUE

xRead_Write	BOOL	Switch between Read/Write	TRUE=Write
			FALSE=Read

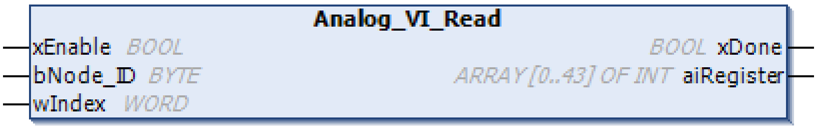
bNode_ID	BYTE	Node ID	Node ID of iR-COP
wIndex	WORD	Index	Index of Object Dictionary
bChannel	BYTE	Channel	Analog Channel no. 0~3
Output Parameters	**Data Type**	**Definition**	**Description**
xDone	BOOL	Completed	Function completed
iMax_Peak	INT	Maximum peak value	
iMin_Peak	INT	Minimum peak value	
In-Out Parameters	**Data Type**	**Definition**	**Description**
iMode	INT	Channel Mode	
iScale_Max	INT	Maximum channel scale	
iScale_Min	INT	Minimum channel scale	
iSamp_Times	INT	Channel Input Filter Frame Size	

4.4. Analog_VI_READ(FB)

Function: Read all analog parameters. when “xEnable” turns from FALSE to TRUE.

This Function Block can only be used to Read/Write an analog module that is connected to iR-COP.

Applicable for: AI04-VI, AQ04-VI, and AM06-VI.



Analog_VI_Read

xEnable BOOL

bNode_ID BYTE

wIndex WORD

BOOL xDone

ARRAY [0..43] OF INT aiRegister

Input Parameters	Data Type	Definition	Description
xEnable	BOOL	Enable	Triggered by FALSE to TRUE
bNode_ID	BYTE	Node ID	Node ID of iR-COP
wIndex	WORD	Index	Index of Object Dictionary
Output Parameters	**Data Type**	**Definition**	**Description**
xDone	BOOL	Completed	Function completed
aiRegister	INT[0..43]	Analog Module Register	Read all the registers of Analog module (44 registers)

5. Motion Control commands

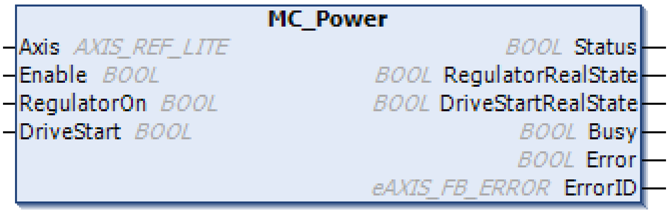
5.1. MC_Power(FB)

Function:

When “Enable”, “RegulatorOn”, “DriveStart” are TRUE, the designated axis enters Standstill state, which means the axis is ready for motion instructions.

When “Enable” and “RegulatorOn” are FALSE, the designated axis enters Disable state, which means the axis is inactive.

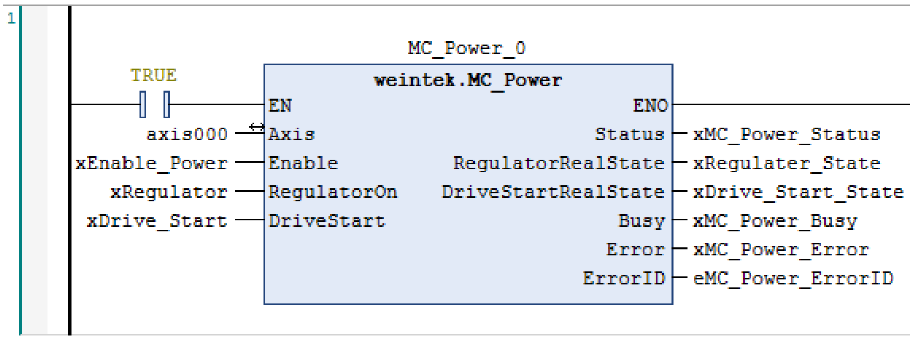
When “DriveStart” is FALSE, the axis quickly stops.



Input Parameters	Data Type	Definition	Description
Enable	BOOL	Enable Function Block	TRUE: Enable Function Block.
RegulatorOn	BOOL	Operate Motion Control system	FALSE: Disable Motion Control system. TRUE: Enable Motion Control system, and get ready for motion instructions.
DriveStart	BOOL	Trigger Quick Stop	FALSE: Enable Quick stop. TRUE: Disable Quick stop.
Output Parameters	**Data Type**	**Definition**	**Description**
Status	BOOL	Function Block Status	TRUE: The designated axis enters Standstill state.
RegulatorRealState	BOOL	Motion Control Status	FALSE: Motion Control system is not enabled. TRUE: Motion Control system is enabled and is ready for motion instructions.
DriveStartRealState	BOOL	Quick Stop Status	FALSE: Quick stop is enabled. TRUE: Quick stop is disabled.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

```
// MC_Power function block
MC_Power_1(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>    xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);
```

5.2. MC_MoveVelocity(FB)

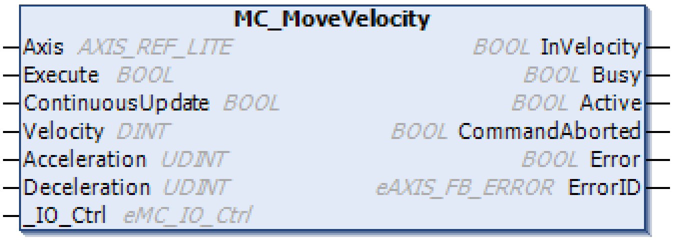
Function:

Perform velocity control by specifying the target velocity for the designated axis.

Velocity control is executed when “Execute” turns from FALSE to TRUE.

Positive velocity = positive direction, negative velocity = negative direction.

MC_MoveVelocity function block can be used for JOG, for more information, please see Demo Project: DEM19004E_iR_Application_JOG_Demo_20190906



MC_MoveVelocity

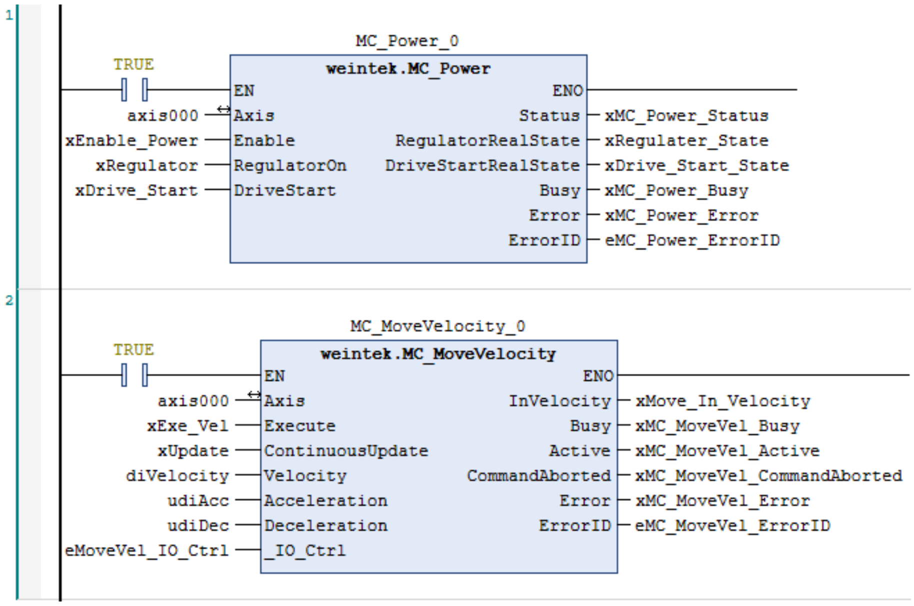
Axis AXIS_REF_LITE	BOOL InVelocity
Execute BOOL	BOOL Busy
ContinuousUpdate BOOL	BOOL Active
Velocity DINT	BOOL CommandAborted
Acceleration UDINT	BOOL Error
Deceleration UDINT	eAXIS_FB_ERROR ErrorID
_IO_Ctrl eMC_IO_Ctrl	

Input Parameters	**Data Type**	**Definition**	**Description**
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
ContinuousUpdate	BOOL	Continuously updates the velocity during motion	TRUE= The target velocity, acceleration rate and deceleration rate can be changed when the axis is operating.
Velocity	DINT	Target Velocity	Specify the target velocity. The velocity unit is user-defined unit divided by s (seconds).
Acceleration	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s^2 (second square).
Deceleration	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s^2 (second square).
IO_Ctrl	eMC_IO_Ctrl	I/O Control	None: Not used. I0~I2: Trigger execution using digital input.
Output Parameters	**Data Type**	**Definition**	**Description**
InVelocity	BOOL	Target Velocity Reached	TRUE: Target Velocity is reached.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**

Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

```
// MC_Power function block
MC_Power_1(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>    xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);

// MC_Velocity function block
MC_MoveVelocity_0(
        Axis:= Axis000,
        Execute:= xExe_Vel,
        ContinuousUpdate:= xUpdate,
        Velocity:= diVelocity,
        Acceleration:= udiAcc,
        Deceleration:= udiDec,
        _IO_Ctrl:= eMoveVel_IO_Ctrl,
        InVelocity=> xMove_In_Velocity,
        Busy=> xMC_MoveVel_Busy,
        Active=> xMC_MoveVel_Active,
        CommandAborted=> xMC_MoveVel_CommandAborted,
        Error=> xMC_MoveVel_Error,
        ErrorID=> eMC_MoveVel_ErrorID);
```

5.3. MC_MoveAbsolute(FB)

Function:

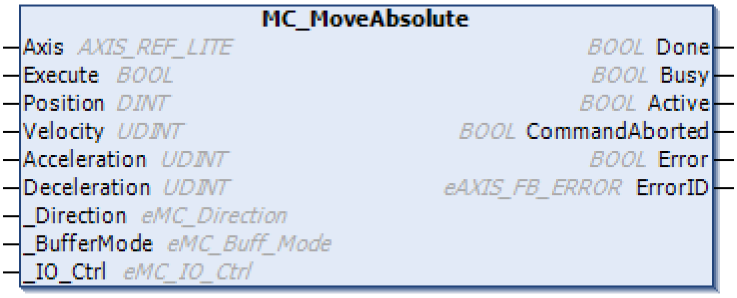
Moves the axis to a specified absolute target position.

Positioning control is executed when “Execute” turns from FALSE to TRUE.

Positioning control can be performed regardless of homing.

The configurable parameters include: Velocity, Acceleration Rate, Deceleration Rate, Buffer Mode, and IO Control.

Direction can be specified for a rotating axis.

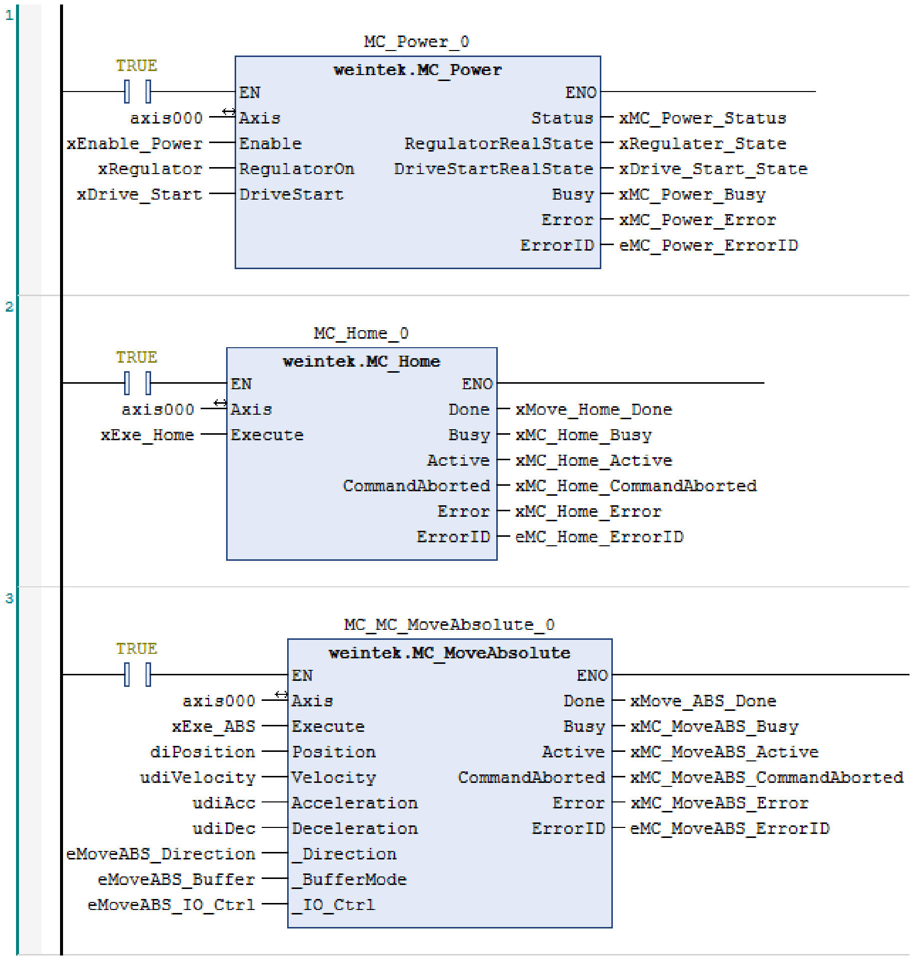


Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Position	DINT	Target Position	Specify the absolute target position.
Velocity	UDINT	Target Velocity	Specify the target velocity. The velocity unit is user-defined unit divided by s (seconds).
Acceleration	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s^2 (second square).
Deceleration	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s^2 (second square).
_Direction	eMC_Direction	Direction in which the axis rotates.	Positive: positive direction. ShortestWay: shortest way. Negative: negative direction. Current: Follow the last direction. (This setting is available only for a rotating axis.)
_BufferMode	eMT_Buff_Mode	Chronological sequence of the Function Block	Aborting: Not used. Buffered: Continuously executes the next instruction after the ongoing motion is completed. BlendingPrev: Continuously executes the next instruction after the ongoing motion is completed with blending previous velocity.

_IO_Ctrl	eMC_IO_Ctrl	I/O Control	None: Not used. I0~I2: Trigger execution using digital input. O0~O2: Output after the ongoing motion is completed.
Output Parameters	**Data Type**	**Definition**	**Description**
Done	BOOL	Function Block is done	True: Target position reached.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

```
// MC_Power function block
MC_Power_0(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>     xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);
// MC_Home function block
MC_Home_0(
        Axis:= Axis000,
        Execute:= xExe_Home,
        Done=> xMove_Home_Done,
        Busy=> xMC_Home_Busy,
        Active=> xMC_Home_Active,
        CommandAborted=> xMC_Home_CommandAborted,
        Error=> xMC_Home_Error,
        ErrorID=> eMC_Home_ErrorID);
// MC_MoveAbsolute function block
MC_MC_MoveAbsolute_0(
        Axis:= Axis000,
        Execute:= xExe_ABS,
        Position:= diPosition,
        Velocity:= udiVelocity,
        Acceleration:= udiAcc,
        Deceleration:= udiDec,
        _Direction:= eMoveABS_Direction,
        _BufferMode:= eMoveABS_Buffer,
        _IO_Ctrl:= eMoveABS_IO_Ctrl,
        Done=> xMove_ABS_Done,
        Busy=>     xMC_MoveABS_Busy,
        Active=> xMC_MoveABS_Active,
        CommandAborted=> xMC_MoveABS_CommandAborted,
        Error=> xMC_MoveABS_Error,
        ErrorID=> eMC_MoveABS_ErrorID);
```

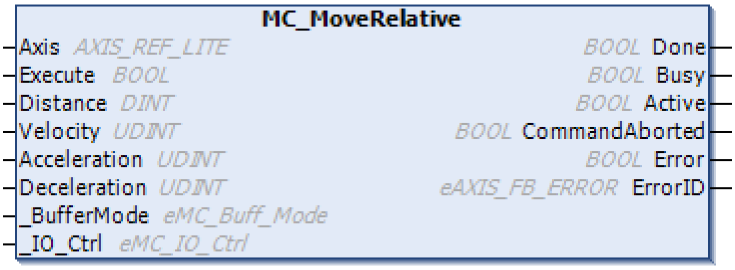
5.4. MC_VelocityRelative(FB)

Function:

Performs positioning for a specified travel distance from the current position.

Positioning control is executed when “Execute” turns from FALSE to TRUE.

The configurable parameters include: Velocity, Acceleration Rate, Deceleration Rate, Buffer Mode, and IO Control.



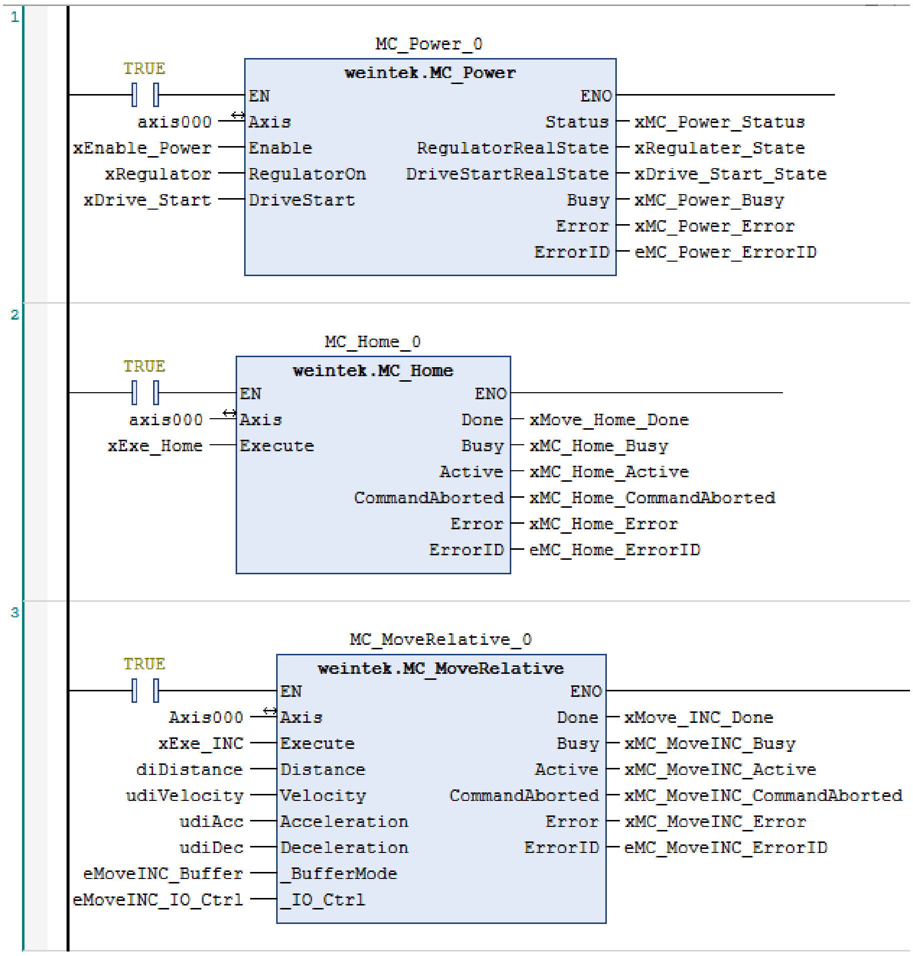
```
                          MC_MoveRelative
Axis  AXIS_REF_LITE                                   BOOL  Done
Execute  BOOL                                         BOOL  Busy
Distance  DINT                                        BOOL  Active
Velocity  UDINT                             BOOL  CommandAborted
Acceleration  UDINT                                   BOOL  Error
Deceleration  UDINT                        eAXIS_FB_ERROR  ErrorID
_BufferMode  eMC_Buff_Mode
_IO_Ctrl  eMC_IO_Ctrl
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Distance	DINT	Travel Distance from the Current Position.	Specify the travel distance from the current position. The unit is user-defined unit.
Velocity	UDINT	Target Velocity	Specify the target velocity. The velocity unit is user-defined unit divided by s (seconds).
Acceleration	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s^2 (second square).
Deceleration	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s^2 (second square).
_BufferMode	eMT_Buff_Mode	Chronological sequence of the Function Block	Aborting: Not used. Buffered: Continuously executes the next instruction after the ongoing motion is completed. BlendingPrev: Continuously executes the next instruction after the ongoing motion is completed with blending previous velocity.
_IO_Ctrl	eMC_IO_Ctrl	I/O Control	None: Not used. I0~I2: Trigger execution using digital input. O0~O2: Output after the ongoing motion is completed.
Output Parameters	**Data Type**	**Definition**	**Description**
Done	BOOL	Function Block is done	True: Target position reached.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.

Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

```
// MC_Power function block
MC_Power_0(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>    xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);
// MC_Power function block
MC_Home_0(
        Axis:= Axis000,
        Execute:= xExe_Home,
        Done=> xMove_Home_Done,
        Busy=> xMC_Home_Busy,
        Active=> xMC_Home_Active,
        CommandAborted=> xMC_Home_CommandAborted,
        Error=> xMC_Home_Error,
        ErrorID=> eMC_Home_ErrorID);
// MC_Relative function block
MC_MoveRelative_0(
        Axis:= Axis000,
        Execute:= xExe_INC,
        Distance:= diDistance,
        Velocity:= udiVelocity,
        Acceleration:= udiAcc,
        Deceleration:= udiDec,
        _BufferMode:= eMoveINC_Buffer,
        _IO_Ctrl:= eMoveINC_IO_Ctrl,
        Done=> xMove_INC_Done,
        Busy=> xMC_MoveINC_Busy,
        Active=> xMC_MoveINC_Active,
        CommandAborted=> xMC_MoveINC_CommandAborted,
        Error=> xMC_MoveINC_Error,
        ErrorID=> eMC_MoveINC_ErrorID );
```

5.5. MC_Home(FB)

Function:

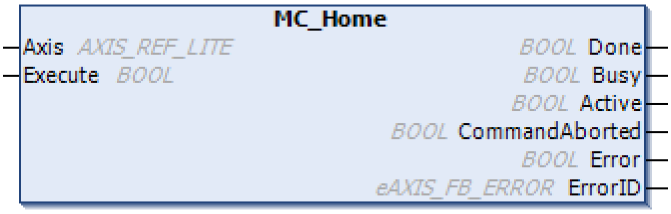
Performs homing when “Execute” turns from FALSE to TRUE.

Object Dictionary 6098: Homing method.

37 homing methods are provided, which can be selected by using [Add SDOs]

A homing method (1~37) can be selected by using [Add SDOs] to write value in CODESYS.

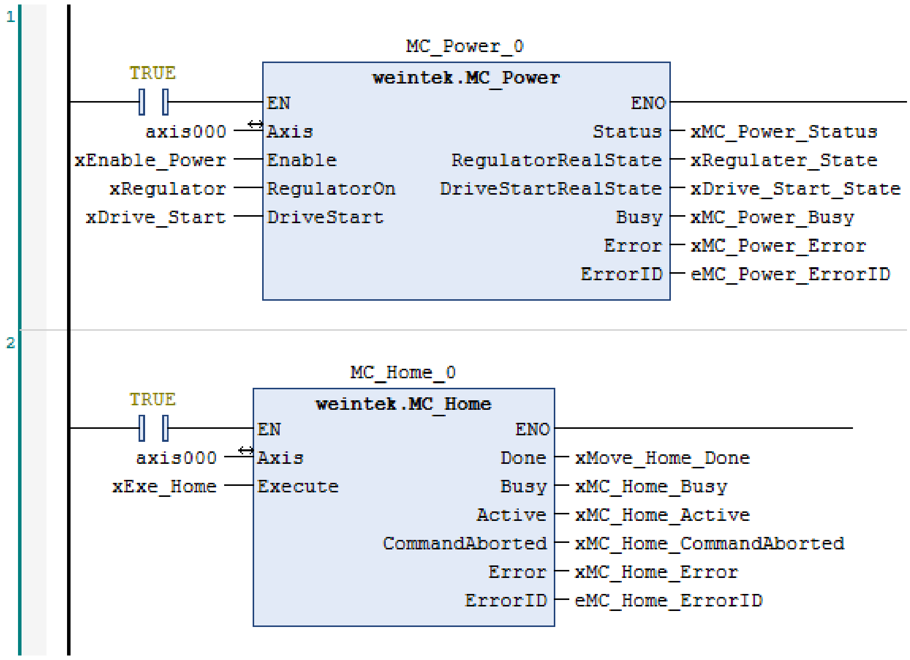
For more information on homing methods, please see Appendix B in this manual.



Input Parameters	**Data Type**	**Definition**	**Description**
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Output Parameters	**Data Type**	**Definition**	**Description**
Done	BOOL	Function Block is done	True: Homing is completed.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

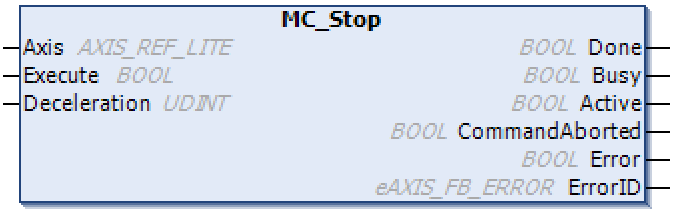
```
// MC_Power function block
MC_Power_0(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>    xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);
// MC_Power function block
MC_Home_0(
        Axis:= Axis000,
        Execute:= xExe_Home,
        Done=> xMove_Home_Done,
        Busy=> xMC_Home_Busy,
        Active=> xMC_Home_Active,
        CommandAborted=> xMC_Home_CommandAborted,
        Error=> xMC_Home_Error,
        ErrorID=> eMC_Home_ErrorID);
```

5.6. MC_STOP(FB)

Function:

Decelerates the axis to stop (velocity = 0) when “Execute” turns from FALSE to TRUE.

Instructions can only be given after the axis stops.



MC_Stop

Axis *AXIS_REF_LITE*

Execute *BOOL*

Deceleration *UDINT*

BOOL Done

BOOL Busy

BOOL Active

BOOL CommandAborted

BOOL Error

eAXIS_FB_ERROR ErrorID

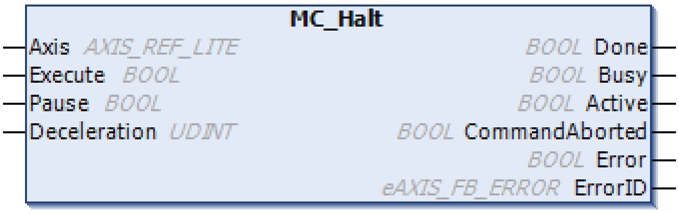
Input Parameters	**Data Type**	**Definition**	**Description**
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Deceleration	UDINT	Deceleration Rate	Specify the deceleration rate, The unit is user-defined divided by s^2 (second square).
Output Parameters	**Data Type**	**Definition**	**Description**
Done	BOOL	Function Block is done	True: Velocity = 0 and Execute = FALSE
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

5.7. MC_Halt(FB)

Function:

Decelerates the axis to stop (velocity = 0) when “Execute” turns from FALSE to TRUE. Instructions can be given during deceleration.

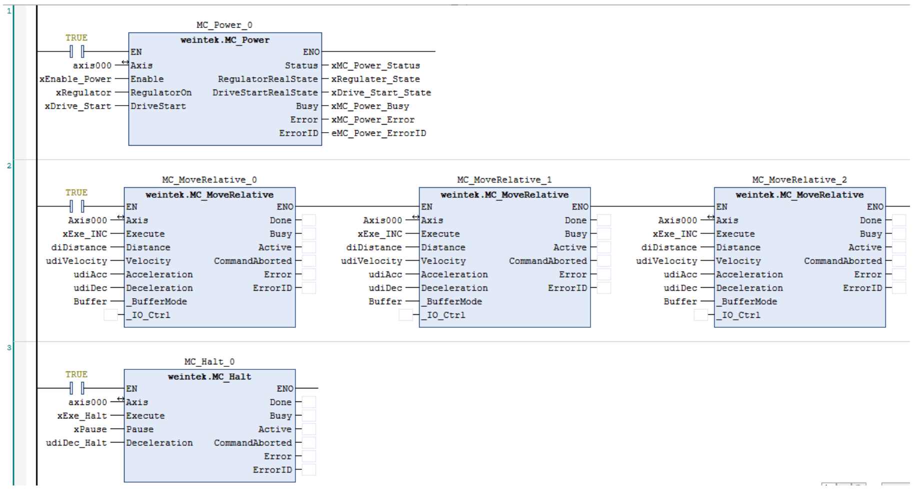
MC_Halt is executed when “Pause” is TRUE. The paused motion will be continued after “Execute” returns to FAUSE.



Input Parameters	**Data Type**	**Definition**	**Description**
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Pause	BOOL	Pause	TRUE: Execute MC_Halt to pause current motion. The buffered motion will not be interrupted.
Deceleration	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s^2 (second square).
Output Parameters	**Data Type**	**Definition**	**Description**
Done	BOOL	Function Block is done	True: Velocity = 0 and Execute = FALSE
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

```
// MC_Power function block
MC_Power_0(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>     xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);
// Buffer mode continuous positioning function block
MC_MoveRelative_0(
        Axis:= Axis000,
        Execute:= xExe_INC,
        Distance:= diDistance,
        Velocity:= udiVelocity,
        Acceleration:= udiAcc,
        Deceleration:= udiDec,
        _BufferMode:= Buffer,
        _IO_Ctrl:= ,
        Done=> ,
        Busy=> ,
        Active=> ,
        CommandAborted=> ,
	Error=> ,
	ErrorID=> );
MC_MoveRelative_1(
	Axis:= Axis000,
	Execute:= xExe_INC,
	Distance:= diDistance,
	Velocity:= udiVelocity,
	Acceleration:= udiAcc,
	Deceleration:= udiDec,
	_BufferMode:= Buffer,
	_IO_Ctrl:= ,
	Done=> ,
	Busy=> ,
	Active=> ,
	CommandAborted=> ,
	Error=> ,
	ErrorID=> );
MC_MoveRelative_2(
	Axis:= Axis000,
	Execute:= xExe_INC,
	Distance:= diDistance,
	Velocity:= udiVelocity,
	Acceleration:= udiAcc,
	Deceleration:= udiDec,
	_BufferMode:= Buffer,
	_IO_Ctrl:= ,
	Done=> ,
	Busy=> ,
	Active=> ,
	CommandAborted=> ,
	Error=> ,
	ErrorID=> );
// Buffer mode continuous positioning function block
//	MC_Halt function block
MC_Halt_0(
	Axis:= Axis000,
	Execute:= xExe_Halt,
Pause:= xPause ,
Deceleration:= udiDec_Halt,
Done=> ,
Busy=> ,
Active=> ,
CommandAborted=> ,
Error=> ,
ErrorID=> );
```

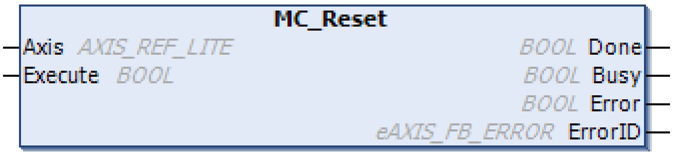
5.8. MC_Reset(FB)

Function:

Resets the errors when the axis turns into Errorstop state due to error.

This Function Block is executed when “Execute” turns from FALSE to TRUE.

When driver error occurs, please clear the driver error before executing MC_Reset.



Input Parameters	**Data Type**	**Definition**	**Description**
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Output Parameters	**Data Type**	**Definition**	**Description**
Done	BOOL	Function Block is done	True: Reset completed.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

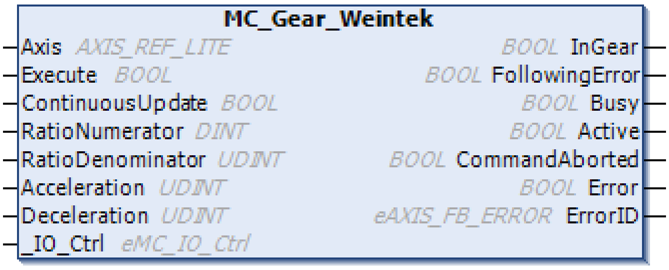
5.9. MC_Gear_Weintek(FB)

Function: Converts the input pulse from MPG (Manual Pulse Generator) into output pulse.

This Function Block is executed when “Execute” turns from FALSE to TRUE.

ContinuousUpdate: When this is TRUE, ratio can be updated continuously when the axis is in motion.

Since pulse input and output use different axes (master axis and slave axis), Pulse Input Method 5501h must be configured. Please set Bit 4 to 1 (external encoder) to use MPG.



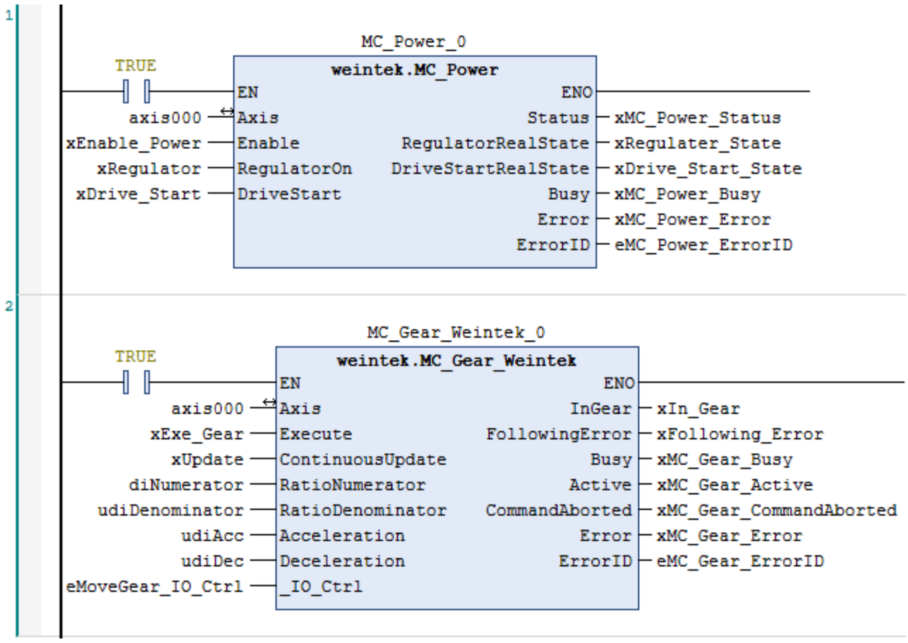
Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
ContinuousUpdate	BOOL	Continuously updates the velocity during motion	TRUE= The target velocity, acceleration rate and deceleration rate can be changed when the axis is operating.
RatioNumerator	REAL	Ratio Numerator	Output Pulse = Input Pulse $*\frac{RatioNumerator}{RatioDenominator}$
RatioDenominator	REAL	Ratio Denominator	
Acceleration	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s^2 (second square).
Deceleration	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s^2 (second square).
IO_Ctrl	eMC_IO_Ctrl	I/O Control	None: Not used. I0~I2: Trigger execution using digital input.

Output Parameters	Data Type	Definition	Description
InGear	BOOL	Target Pulse Reached.	TRUE: Output Pulse = Input Pulse
FollowingError	BOOL	Following Error	TRUE: Output Pulse ≠ Input Pulse, and the difference is greater than tolerable elapsed time.
Busy	BOOL	Function Block Status	TRUE: Function Block has been executed.

Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is aborted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	BOOL	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	**Data Type**	**Definition**	**Description**
Axis	AXIS_REF_LITE	Axis Reference	Reference to the axis.

Programming:

LD:



ST:

```
// MC_Power function block
MC_Power_0(
        Axis:= Axis000,
        Enable:= xEnable_Power,
        RegulatorOn:= xRegulator,
        DriveStart:= xDrive_Start,
        Status=> xMC_Power_Status,
        RegulatorRealState=> xRegulater_State,
        DriveStartRealState=> xDrive_Start_State,
        Busy=>xMC_Power_Busy ,
        Error=>    xMC_Power_Error,
        ErrorID=> eMC_Power_ErrorID);
// MC_Gear function block
MC_Gear_Weintek_0(
        Axis:= Axis000,
        Execute:= xExe_Gear,
        ContinuousUpdate:= xUpdate,
        RatioNumerator:= diNumerator,
        RatioDenominator:= udiDenominator,
        Acceleration:= udiAcc,
        Deceleration:= udiDec,
        _IO_Ctrl:= eMoveGear_IO_Ctrl,
        InGear=> xIn_Gear,
        FollowingError=> xFollowing_Error,
        Busy=> xMC_Gear_Busy,
        Active=> xMC_Gear_Active,
        CommandAborted=> xMC_Gear_CommandAborted,
        Error=> xMC_Gear_Error,
        ErrorID=> eMC_Gear_ErrorID);
```

Appendix A. FB Error Code

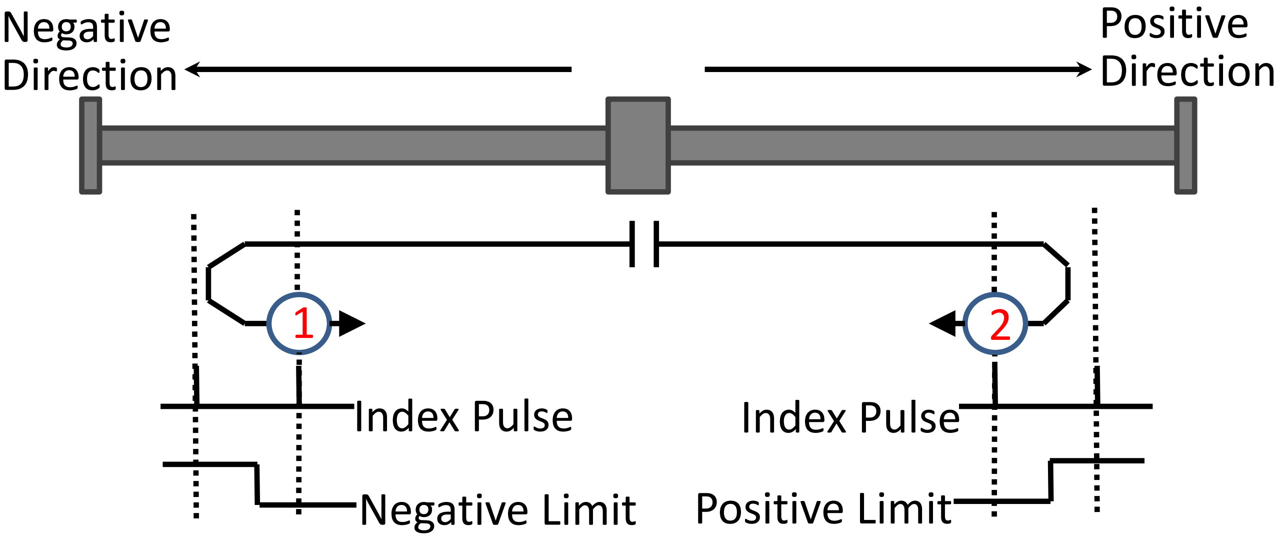
Error Code	State	Description	Error Handling
0	NO_ERROR	No Error	
1	AXIS_NOT_READY	The axis is not ready for operation.	After resolving other errors, enable MC_Power, wait until the Status turns to True, and then restart.
2	AXIS_BUFFER_FULL	Positioning Buffer is full.	Please modify the program to avoid buffering too many positioning controls, and use MC_Reset to clear the error.
3	AXIS_MOTION_ERROR	A motion error occurs.	Please see chapter 4.3 in iR-PU01-P User manual.
4	AXIS_HOMING_ERROR	A homing error occurs.	Please check the homing related settings.
5	AXIS_TRANSITION_ERROR	Incorrect transition of motion mode.	Please modify the program to avoid associating Homing with other motions, and avoid associating Positioning Buffer with non-positioning motions. Please clear the error using MC_Reset.
6	FB_RUNTIME_ERROR	Function Block runtime error.	The Function Block used is not supported by the CODESYS device, please use Weintek's CODESYS controller.

When an error occurs on iR-PU01-P, the diagnostic value is output to the ErrorCode in the AXIS_REF_LITE. Please see Chapter 4 Error Handling in iR-PU01-P User Manual.

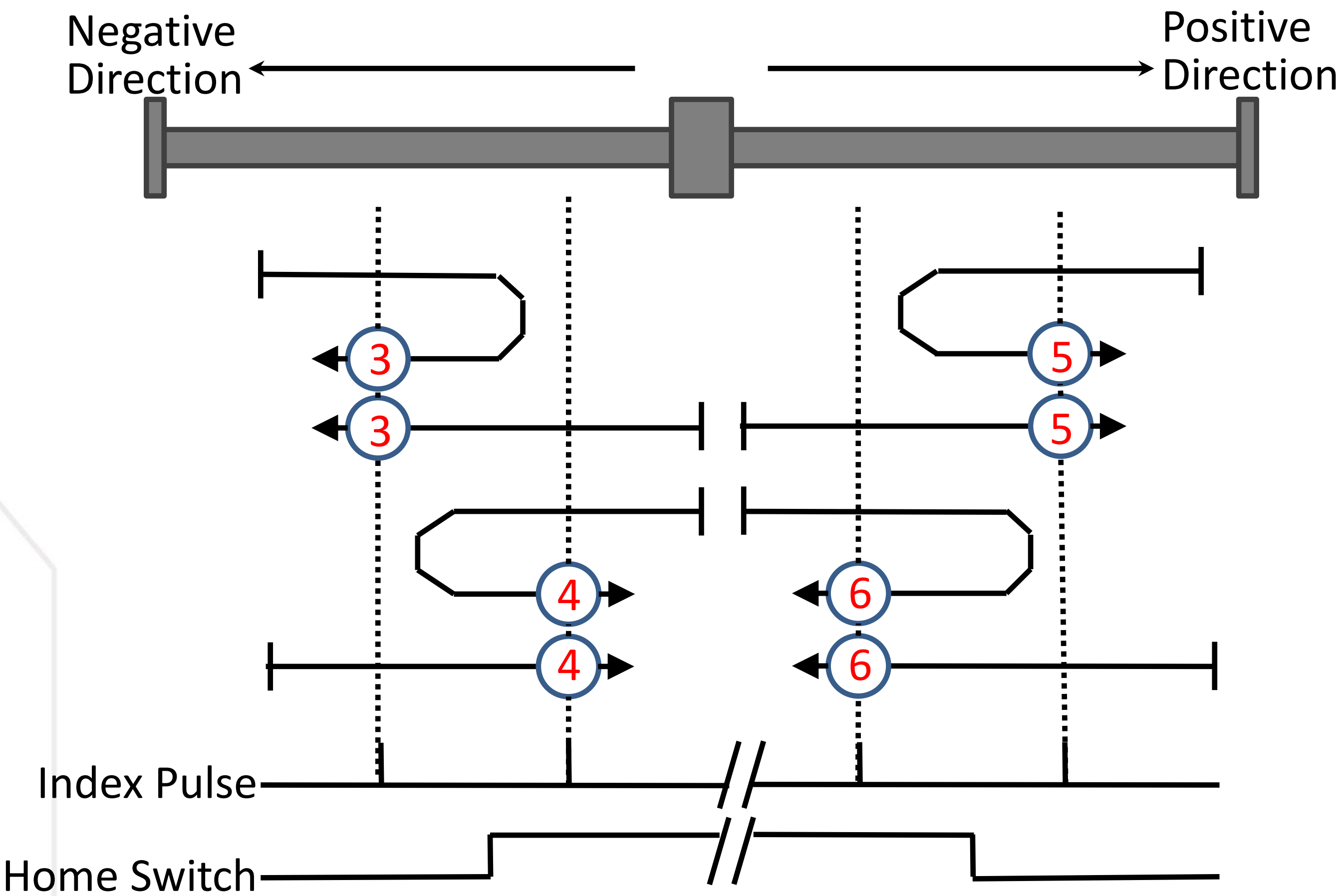
Appendix B. Homing Methods

Method 1~2: Homing on negative / positive limit switch and index pulse.

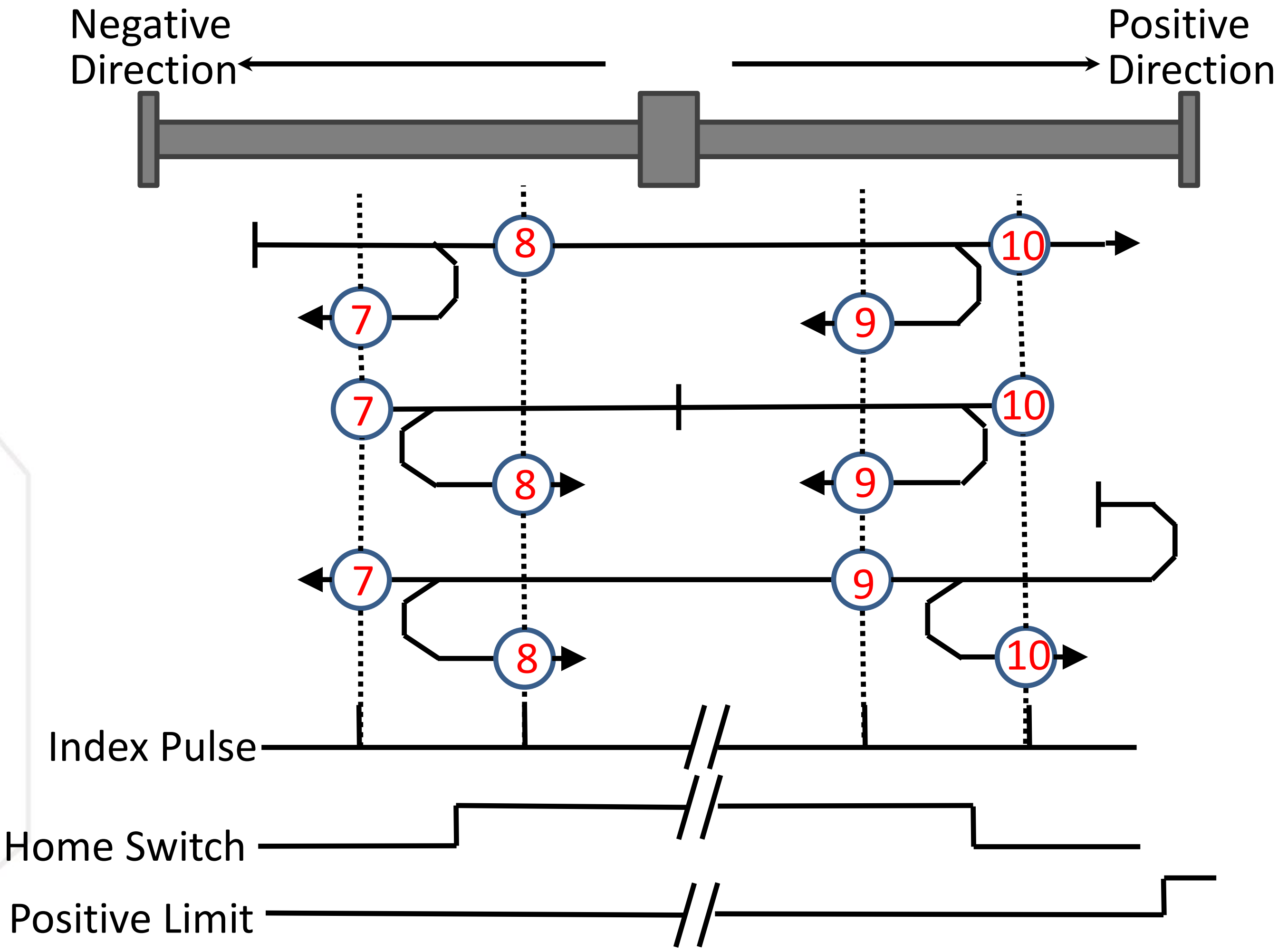
The direction of movement is moving toward the limit switch, and then turning to an opposite direction at the first index pulse.



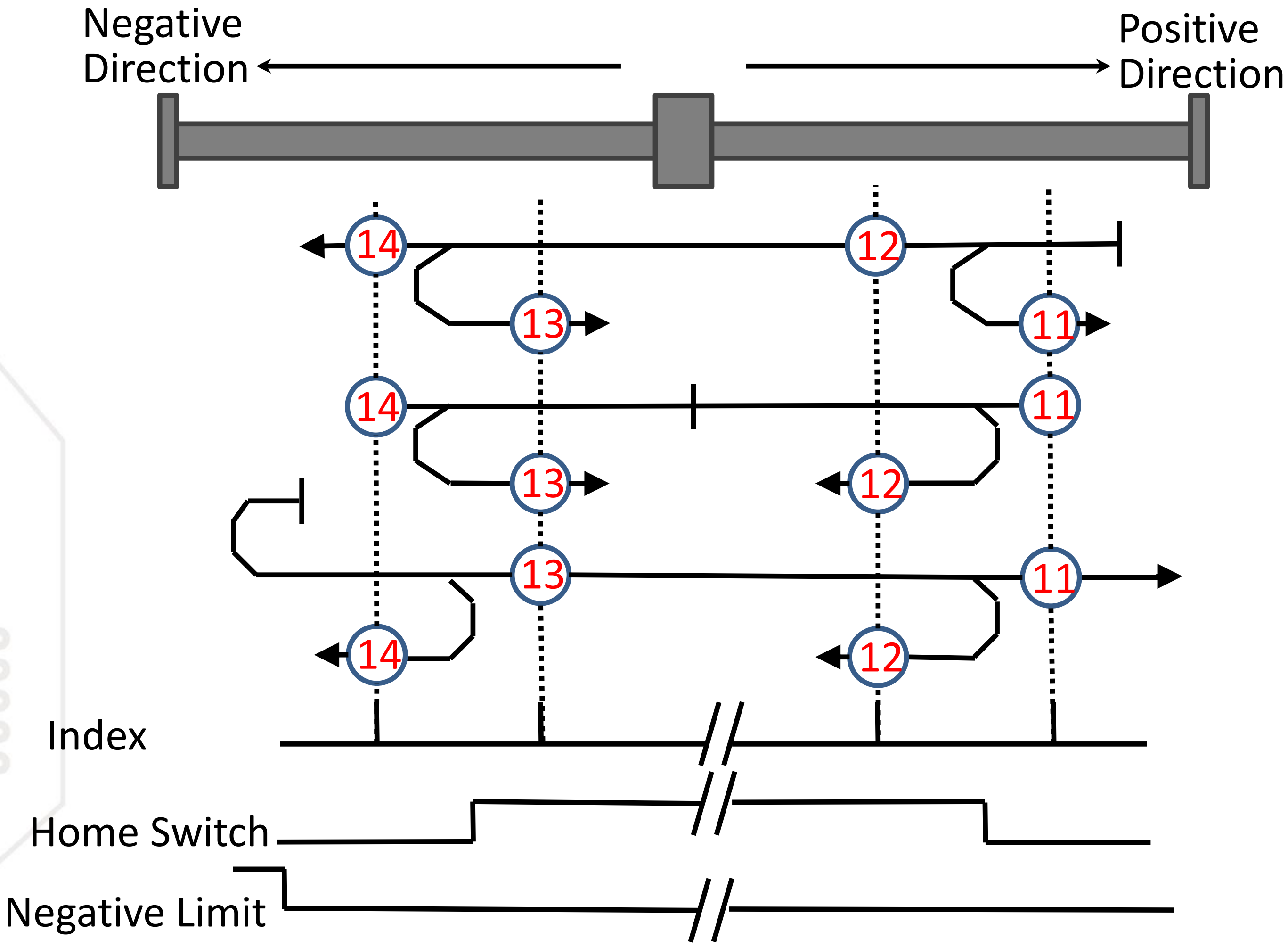
Method 3~6: Homing on positive / negative home switch and index pulse.
The direction of movement is dependent on the state of home switch. The home position shall be at the index pulse to either to the left or right of the point where the home switch changes state.



Method 7~10: Homing on home switch and index pulse – positive initial motion
The initial direction of movement shall be to the right if the home switch is active at the start of the motion. In this case, the initial direction of motion shall 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. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.

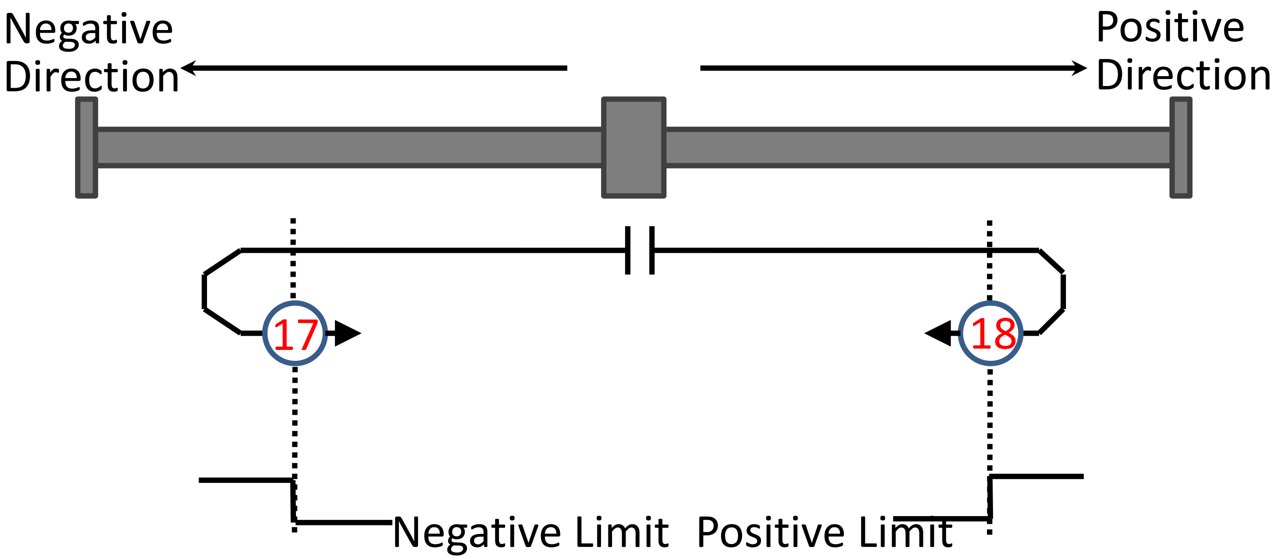


Method 11~14: Homing on home switch and index pulse – negative initial motion The initial direction of movement shall be to the left if the home switch is active at the start of the motion. In this case, the initial direction of motion shall be dependent on the edge being sought. The home position shall be at the index pulse on either side of the rising or falling edges of the home switch. If the initial direction of movement leads away from the home switch, the drive shall reverse on encountering the relevant limit switch.



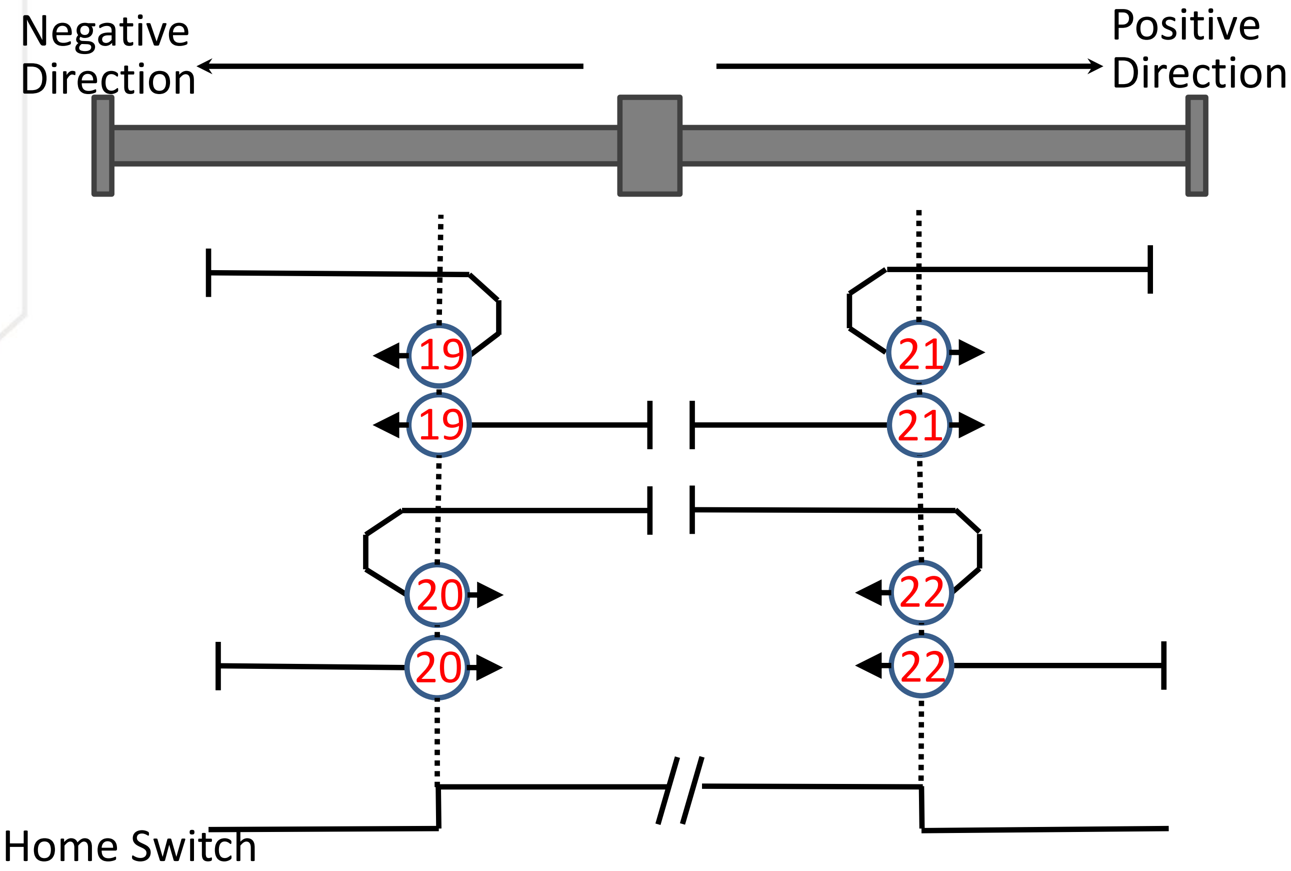
Method 17~18: Homing on negative / positive limit switch

Similar to Method 1~2 except the home position is not dependent on the index pulse.



Method 19~22: Homing on positive / negative home switch

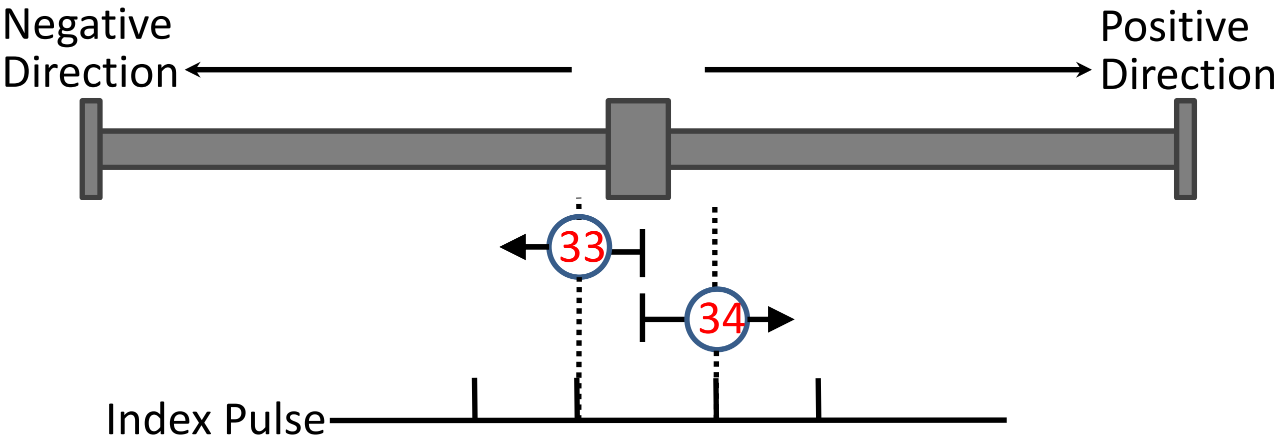
Similar to Method 3~6 except the home position is not dependent on the index pulse.



Method 23~26: Homing on home switch – positive initial motion
Similar to Method 7~10 except the home position is not dependent on the index pulse.

Method 27~30: Homing on home switch – negative initial motion
Similar to Method 11~14 except the home position is not dependent on the index pulse.

Method 33~34: Homing on index pulse
The home position shall be at the index pulse found in the selected direction as shown below.



Method 37 (Default): Homing on current position
In this method, the position sensor information shall be taken to be the home position.
Position actual value = Home offset.