

User Manual

Weintek Library

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

UM018017E_20210125

Table of Contents

1. Overview.....	1
2. Installing Weintek Library	1
3. PID commands.....	3
3.1. PID(FB).....	3
3.2. PWM(FB)	4
4. iR-COP commands	5
4.1. Analog_Config(FB).....	5
4.2. AO_Ch_Pa(FB)	5
4.3. AI_Ch_Pa(FB).....	6
4.4. Analog_VI_READ(FB).....	6
5. Motion Control commands.....	8
5.1. Overview.....	8
5.2. MC_Power(FB).....	8
5.3. MC_MoveVelocity(FB).....	11
5.4. MC_MoveAbsolute(FB)	14
5.5. MC_VelocityRelative(FB)	18
5.6. MC_Home(FB)	21
5.7. MC_STOP(FB).....	23
5.8. MC_Halt(FB)	24
5.9. MC_Reset(FB).....	28
5.10. MC_Gear_Weintek(FB).....	29
5.11. MC_Cam_Weintek(FB)	31
5.12. MC_TorqueControl(FB).....	34
6. Writing Parameters into iR-ETN to configure iR-PU01-P	36
6.1. ETN_PU Function Block	36
6.2. Accessing iR-PU01-P	36
6.3. ETN_PU_SDO(FB)	36
6.4. ETN_PU_Pulse_Method(FB).....	37
6.5. ETN_PU_Pulse_Out_Unit(FB).....	38
6.6. ETN_PU_Max_Setting(FB)	39
6.7. ETN_PU_Motion_Config(FB)	40
6.8. ETN_PU_DI_Setting(FB).....	41
6.9. ETN_PU_DI_Filter(FB).....	42
6.10. ETN_PU_DO_Setting(FB).....	43
6.11. ETN_PU_DO_Abort_Option(FB).....	44
6.12. ETN_PU_Home_setting(FB).....	45
6.13. ETN_PU_AddPosition_Unit(FB).....	46

6.14.	ETN_PU_Motion_DIO_Setting(FB)	47
6.15.	ETN_PU_PWM_Setting(FB)	48
6.16.	ETN_PU_Axis_Setting	49
7.	PU_PWM(FB)	50
7.1.	Overview	50
7.2.	PU_Frequency_Transfer_PWM(FUN)	50
7.3.	PWM_Output_COP(FB)	50
7.4.	PU_PWM_Output_ECAT(FB)	51
8.	VAR_Access	52
8.1.	Overview	52
8.2.	Read_Symbol	52
8.3.	Write_Symbol	52
9.	Weintek_iBus_Library	54
9.1.	Overview	54
9.2.	iBus_Info_Read	55
9.3.	iBus_PU_Read	56
9.4.	iBus_PU_Write	57
9.5.	iBus_Reg_Read	58
9.6.	iBus_Reg_Write	59
Appendix A.	Motion Control FB Error Code	61
Appendix B.	Homing Methods	62
Appendix C.	Enum	68
Appendix D.	iBus FB Error Code	70

CODESYS® is a trademark of 3S-Smart Software Solutions GmbH.

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.

Copyright© 2019 Weintek Labs., INC. All rights reserved.

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.

The screenshot shows the Weintek website search results for 'cmt+codesys package'. The search results table is as follows:

File Name	Date/Time	File Type	Size
Weintek_CODESYS_and_RemoteIO_1.0.0.188	03/05/2019 5:42 PM	CODESYS Package	178 KB

Below the search results, the 'Add Library' dialog box is shown. It contains a search bar and a tree view of libraries. The 'Weintek CODESYS Library' is selected under the '(Miscellaneous)' category.

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\CODSYS\Managed Libraries)

Description: Weintek application library

License required: No More...

Contents:

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

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.

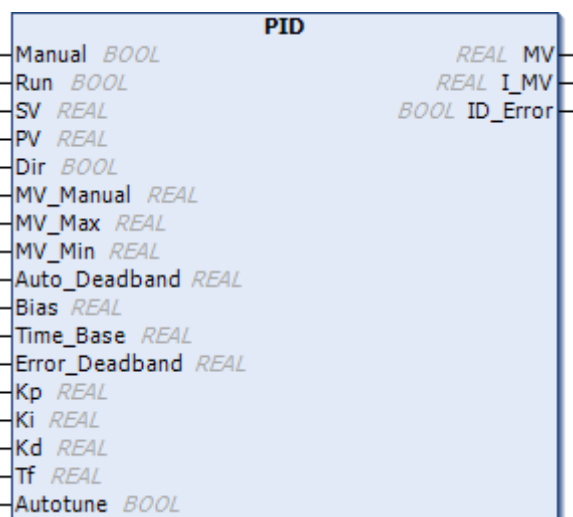
“Dir” is FALSE when SV > PV, “Dir” is TRUE when SV < PV

“Time_Base” cannot be set to 0.

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

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

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



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 $SV \pm \text{Auto_Deadband}$ is in the non-action zone.
BIAS	REAL	Feed Forward Output Value	

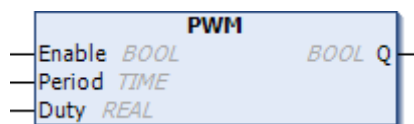
Time_Base	REAL	Sampling Time	Unit= Second(s)
Error_Deadband	REAL	Deadband	Range within which the Output value is counted as zero. SV±Error_Deadband
In-Out Parameters	Data Type	Definition	Description
Tf	REAL	Derivate-action Time Constant	Discrete MV of derivative term: $MV_d = \frac{T_f * MV_d(last\ cycle) + K_d * E}{T_f + T_s}$ 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:

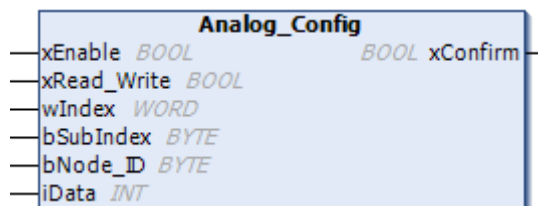
Input Parameters	Data Type	Definition	Description
Enable	BOOL	Enabling the PWM function block	Activation by TRUE
Period	TIME	Time Period	A TRUE then FALSE per cycle
Duty	REAL	Ratio of output duration in a cycle	Range between 0~100%
Output Parameters	Data Type	Definition	Description
Q	BOOL	Output	Output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



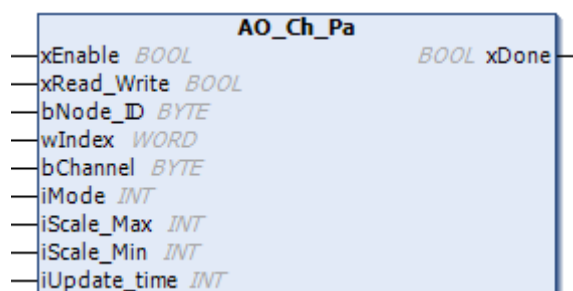
Parameters:

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.



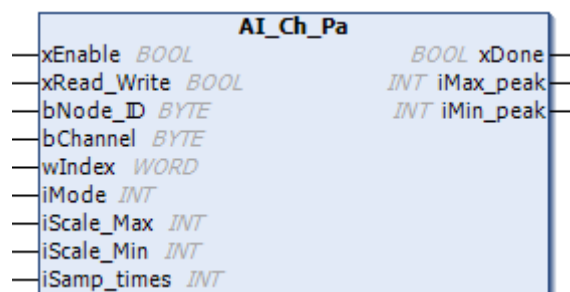
Input Parameters	Data Type	Definition	Description
xEnable	BOOL	Enable	Triggered by FALSE to TRUE
xRead_Write	BOOL	Switch between Read/Write	TRUE=Write FALSE=Read
bNode_ID	BYTE	Node ID	Node ID of iR-COP
wIndex	WORD	Index	Index of Object Dictionary
bChannel	BYTE	Channel	Analog Channel no. 0~3
Output Parameters	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:

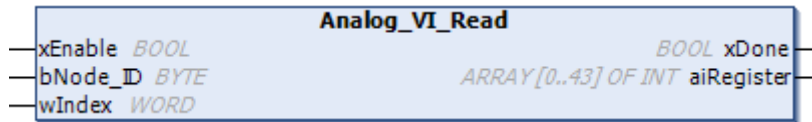
Input Parameters	Data Type	Definition	Description
xEnable	BOOL	Enable	Triggered by FALSE to TRUE
xRead_Write	BOOL	Switch between Read/Write	TRUE=Write FALSE=Read
bNode_ID	BYTE	Node ID	Node ID of iR-COP
wIndex	WORD	Index	Index of Object Dictionary
bChannel	BYTE	Channel	Analog Channel no. 0~3
Output Parameters	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.



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. Overview

The interface and the functions in the motion control system are designed based on PLCopen standard, which is both intuitive and logical. Thanks to PLCopen, the learning curve is smooth when switching between different brands of motion control devices, as long as the device is PLCopen compliant. The function blocks for axis motion control are designed according to CANopen CiA402. Function blocks not only can control Weintek iR-PU01-P module but also support profile position, profile velocity and homing modes based on CiA402. These modes are supported by most of the CANopen and EtherCAT motor drivers, taking full advantage of the benefits of a distributed control system.

Execute and Enable are two input variables that can start function block execution. Execute starts function block execution when it changes to TRUE (Edge) while Enable continues function block execution when it is TRUE and stops execution when it is FALSE (Level). The rest of input variables can be effectively updated when:

A	Busy = TRUE during execution of function block, variables keep on being updated.
B	Busy = FALSE and Execute FALSE->TRUE at the rising edge, variables are updated once.
C	Execute FALSE->TRUE at the rising edge and ContinuousUpdate = TRUE, variables keep on being updated.

Busy and Active are output variables that indicate the execution status of function block. Done and In*** indicate that function block operation ends when the commanded condition is reached.

Function blocks with “_Weintek” in the name can only be used for Weintek iR-PU01-P module. Function block names begin with “_” (e.g. _IO_Ctrl) can only be used with Weintek iR-PU01-P module.

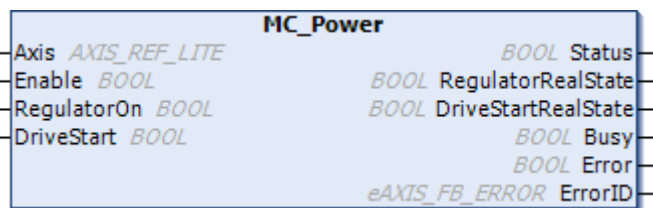
5.2. MC_Power(FB)

Function:

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

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

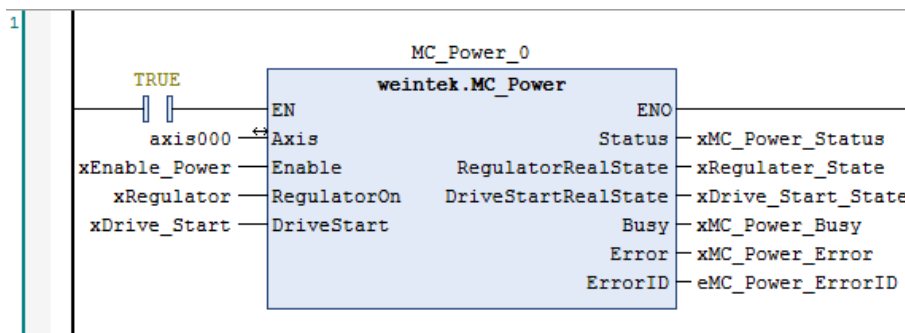
When “DriveStart” is FALSE, the axis quickly stops.



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

DriveStartRealState=> xDrive_Start_State,

Busy=>xMC_Power_Busy ,

Error=> xMC_Power_Error,

ErrorID=> eMC_Power_ErrorID);

5.3. MC_MoveVelocity(FB)

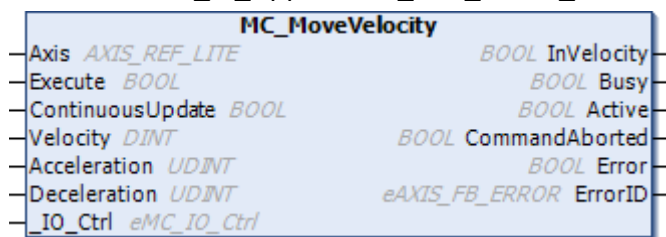
Function:

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

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

Positive velocity = positive direction, negative velocity = negative direction, 0 = decelerates to stop.

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



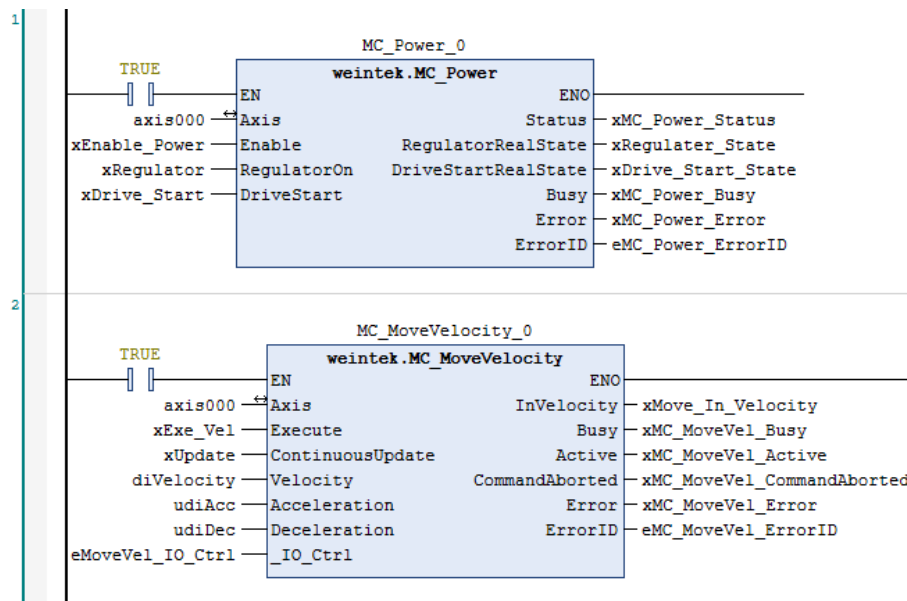
Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
ContinuousUpdate(B)	BOOL	Continuously updates the velocity during motion	TRUE= The target velocity, acceleration rate and deceleration rate can be changed when the axis is operating.
Velocity(C)	DINT	Target Velocity	Specify the target velocity. The velocity unit is user-defined unit divided by s (seconds).
Acceleration(C*)	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s ² (second square).
Deceleration(C*)	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s ² (second square).
IO_Ctrl(B)	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 interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more

Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	Reference to the axis.

*The acceleration rate and deceleration rate cannot be changed alone; they are changed according to the specified target velocity.

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=> xRegulator_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);
```


5.4. MC_MoveAbsolute(FB)

Function:

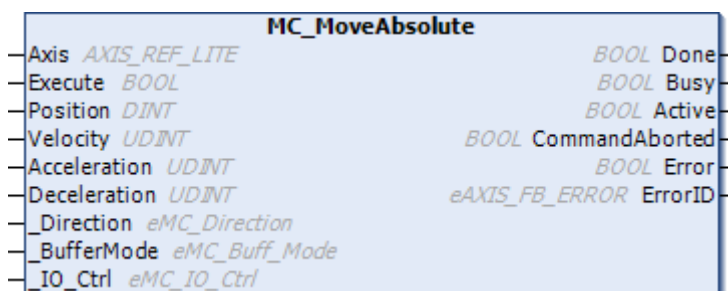
Moves the axis to a specified absolute target position.

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.

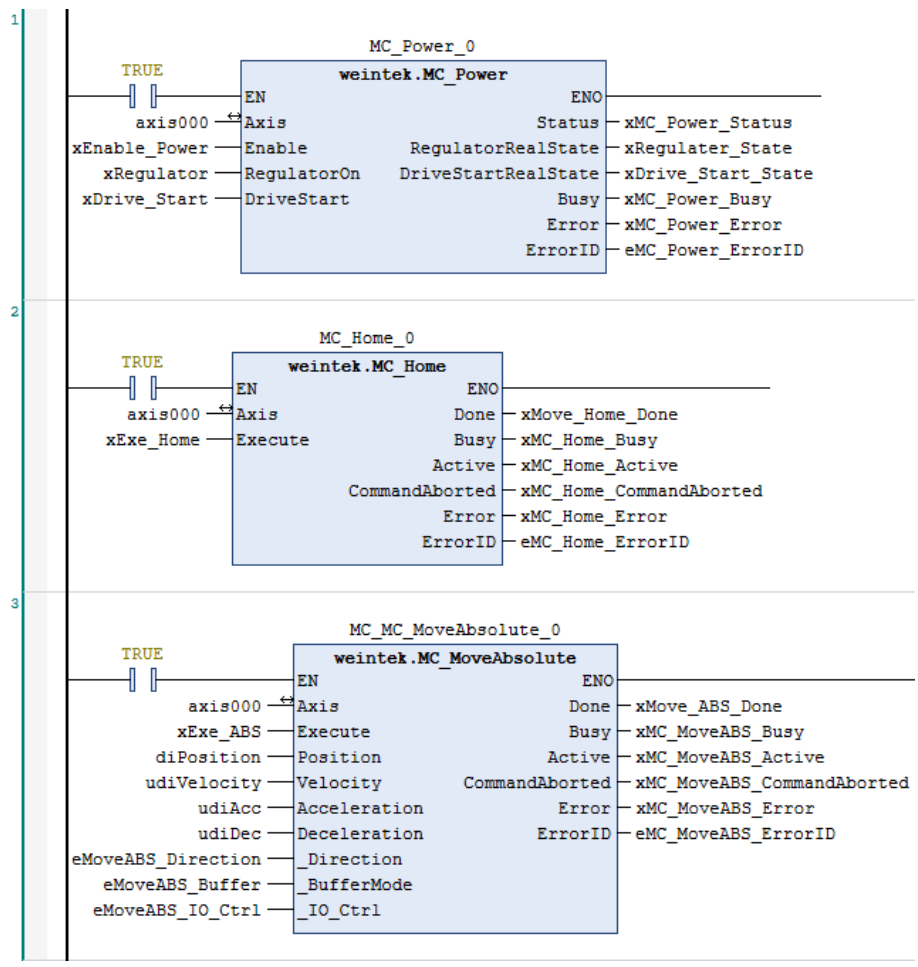


Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Position(B)	DINT	Target Position	Specify the absolute target position.
Velocity(B)	UDINT	Target Velocity	Specify the target velocity. The velocity unit is user-defined unit divided by s (seconds).
Acceleration(B)	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s ² (second square).
Deceleration(B)	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s ² (second square).
_Direction(B)	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(B)	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(B)	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. I_O: Specify Input and Output at a time.
Output Parameters	Data Type	Definition	Description
Done	BOOL	Function Block is done	True: Target position reached.
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	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=> xRegulator_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);
```

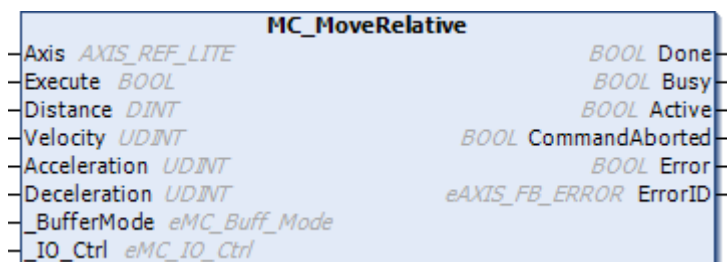
5.5. MC_VelocityRelative(FB)

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.

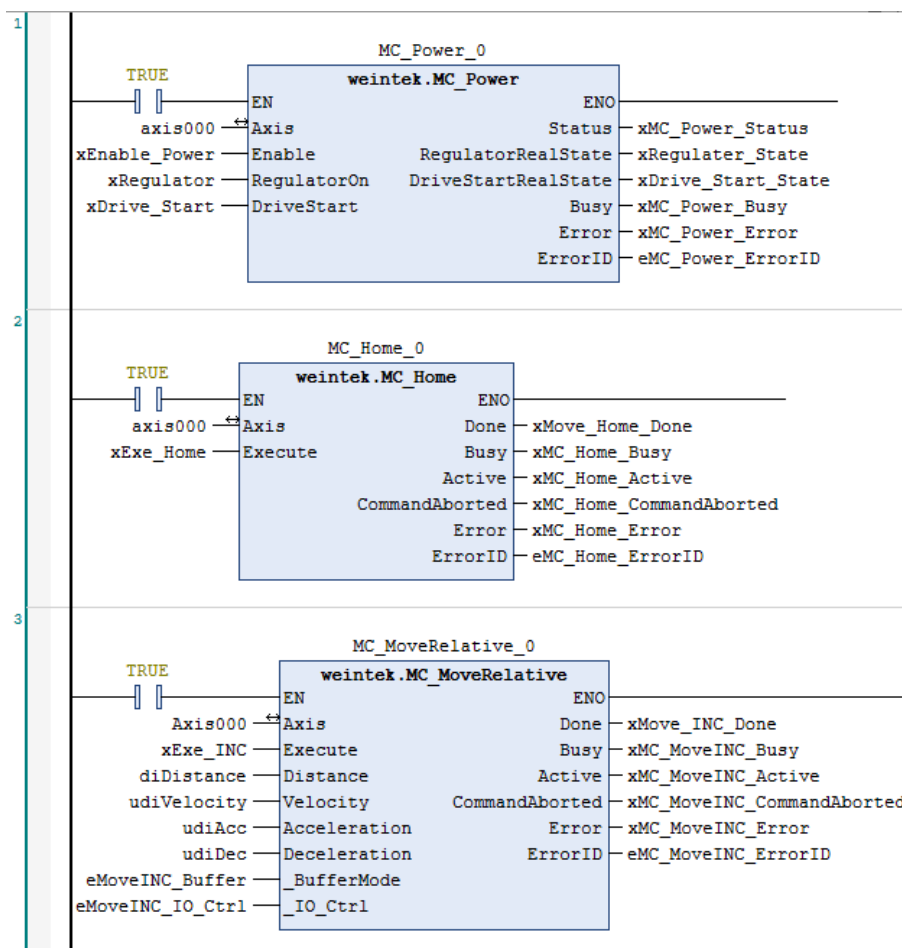


Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Distance(B)	DINT	Travel Distance from the Current Position.	Specify the travel distance from the current position. The unit is user-defined unit.
Velocity(B)	UDINT	Target Velocity	Specify the target velocity. The velocity unit is user-defined unit divided by s (seconds).
Acceleration(B)	UDINT	Acceleration Rate	Specify the acceleration rate, The acceleration unit is user-defined unit divided by s ² (second square).
Deceleration(B)	UDINT	Deceleration Rate	Specify the deceleration rate, The deceleration unit is user-defined unit divided by s ² (second square).
_BufferMode(B)	eMT_Buff_Mode	Chronological sequence of the Function Block	Aborting: Stops executing current instruction and performs positioning. . 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(B)	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 is being executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	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=> xRegulator_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 );
```

5.6. MC_Home(FB)

Function:

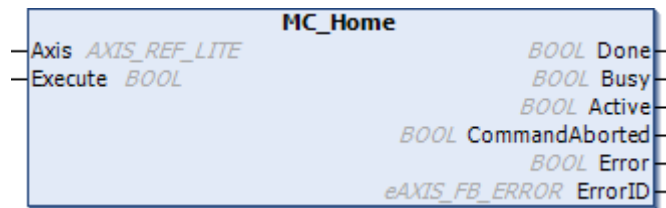
Performs homing when “Execute” turns from FALSE to TRUE.

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.

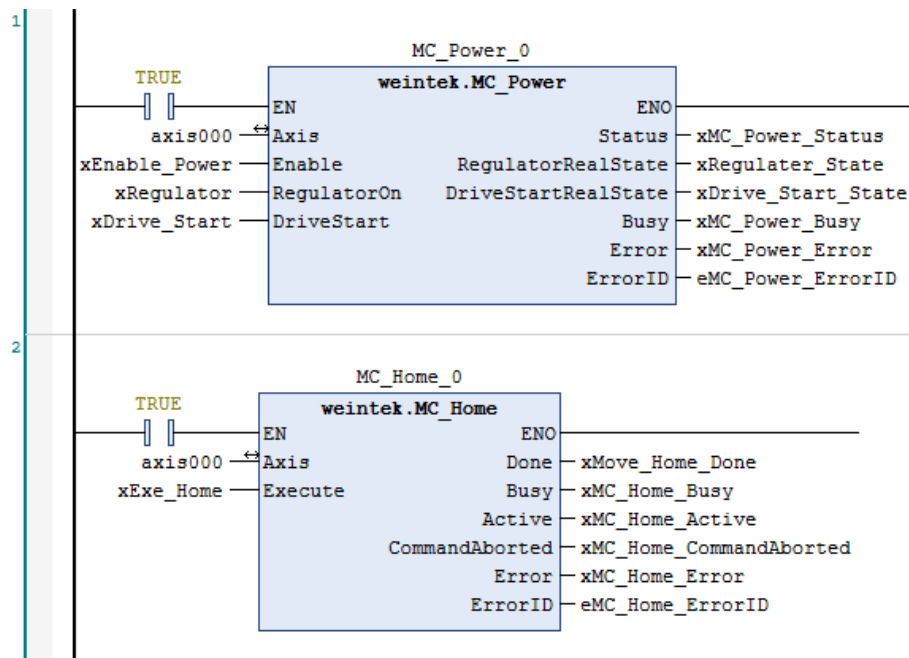
For more information on homing methods, please see Appendix B in this manual.



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 is being executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	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=> xRegulator_State,
    DriveStartRealState=> xDrive_Start_State,
    Busy=>xMC_Power_Busy ,
    Error=> xMC_Power_Error,
    ErrorID=> eMC_Power_ErrorID);
```

```
// MC_Power function block
```

```
MC_Home_0(
```

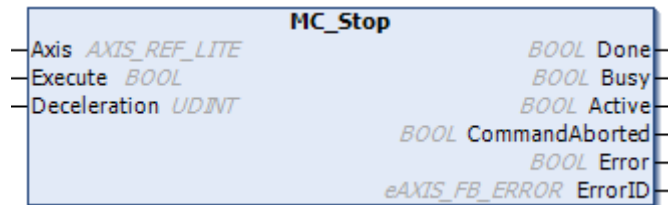
```
    Axis:= Axis000,
    Execute:= xExe_Home,
    Done=> xMove_Home_Done,
    Busy=> xMC_Home_Busy,
    Active=> xMC_Home_Active,
    CommandAborted=> xMC_Home_CommandAborted,
    Error=> xMC_Home_Error,
    ErrorID=> eMC_Home_ErrorID);
```

5.7. MC_STOP(FB)

Function:

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

Instructions can only be given after the axis stops.



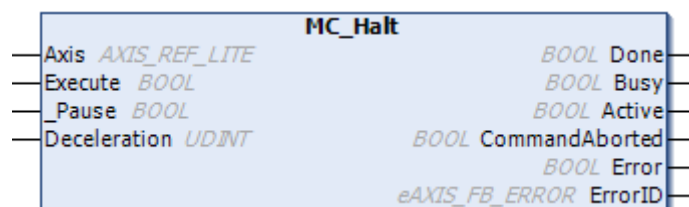
Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
Deceleration(B)	UDINT	Deceleration Rate	Specify the deceleration rate, The unit is user-defined divided by s ² (second square).
Output Parameters	Data Type	Definition	Description
Done	BOOL	Function Block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

5.8. MC_Halt(FB)

Function:

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

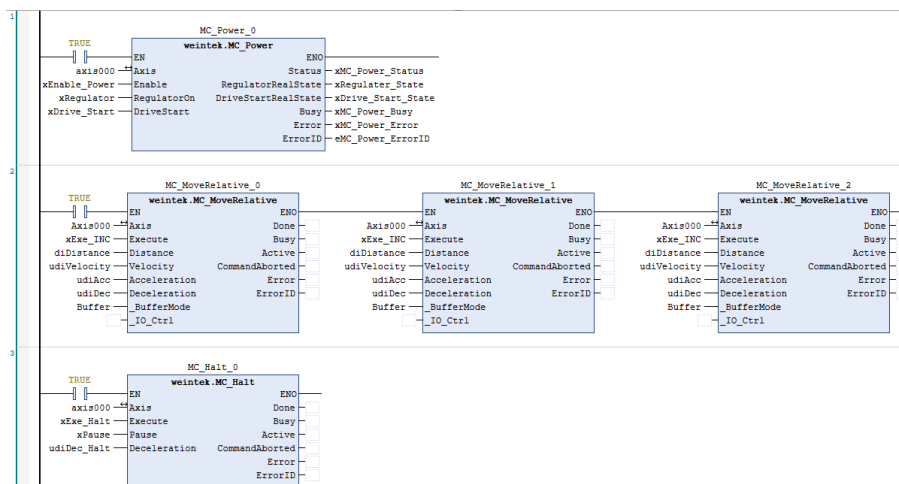
MC_Halt is executed when “Pause” is TRUE. This can only pause positioning control, and positioning control will continue after “Execute” returns to FALSE.



Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
_Pause(B)	BOOL	Pause	TRUE: Execute MC_Halt to pause positioning control. The buffered motion will not be interrupted.
Deceleration(B)	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 is being executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	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=> xRegulator_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=> );
```

5.9. 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 and it cannot be cleared.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	Reference to the axis.

5.10.MC_Gear_Weintek(FB)

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

This function block is only applicable for iR-PU01-P.

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 (main axis encoder) to use Electronic Gear or MPG.

The input pulse unit from the main axis is converted using a conversion ratio to calculate the output pulse unit.

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

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

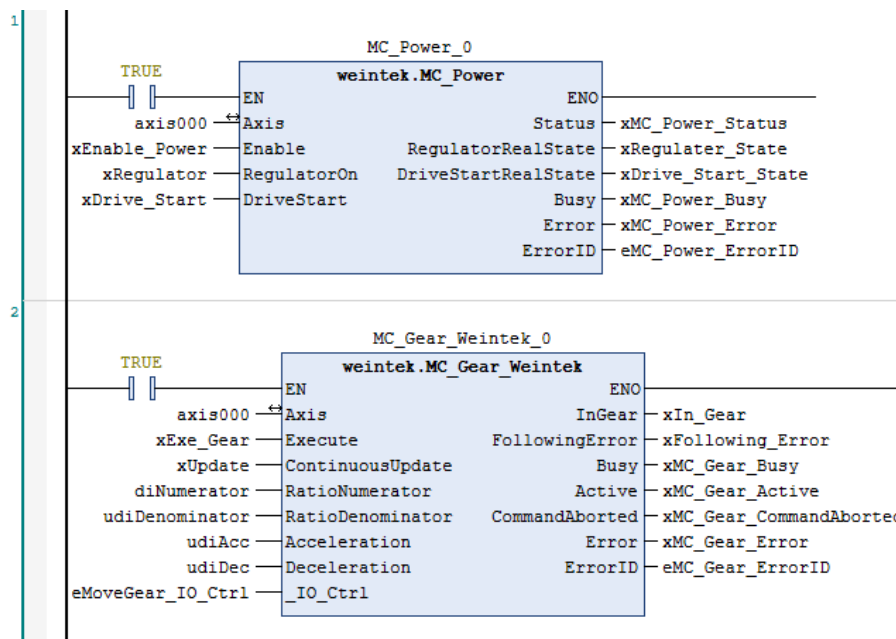


Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
ContinuousUpdate(B)	BOOL	Continuously updates the velocity during motion	TRUE= The target velocity, acceleration rate and deceleration rate can be changed when the axis is operating.
RatioNumerator(C)	REAL	Ratio Numerator	Output Pulse Unit = Input Pulse Unit * $\frac{RatioNumerator}{RatioDenominator}$
RatioDenominator(C)	REAL	Ratio Denominator	
Acceleration(C)	UDINT	Acceleration Rate	Specify the acceleration rate before reaching InGear, The acceleration unit is user-defined unit divided by s ² (second square).
Deceleration(C)	UDINT	Deceleration Rate	Specify the deceleration rate before reaching InGear, The deceleration unit is user-defined unit divided by s ² (second square).
IO_Ctrl(B)	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	TRUE: Output Pulse = Input

		Reached.	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 is being executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	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);

```

5.11.MC_Cam_Weintek(FB)

Function: Synchronizes slave axis position (pulse output) with main axis position (pulse input from iR-PU01-P) according to a Cam Table defined by user.

This function block is only applicable for iR-PU01-P.

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 (main axis encoder) to use Electronic Cam.

The input pulse from the main axis is converted according to the Cam table to calculate the output pulse.

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

The number in CamTableID is the number of the Cam table currently used. (no. 0~2)

MC_Cam_Weintek	
-Axis	AXIS_REF_LITE
-Execute	BOOL
-MasterScaling	UDINT
-SlaveScaling	UDINT
-CamTableID	eMC_CAM_TABLEID
-_IO_Ctrl	eMC_IO_Ctrl
	UINT InCamTableID
	BOOL InCam
	BOOL InSync
	BOOL Busy
	BOOL Active
	BOOL CommandAborted
	BOOL Error
	eAXIS_FB_ERROR ErrorID
	BOOL EndOfProfile

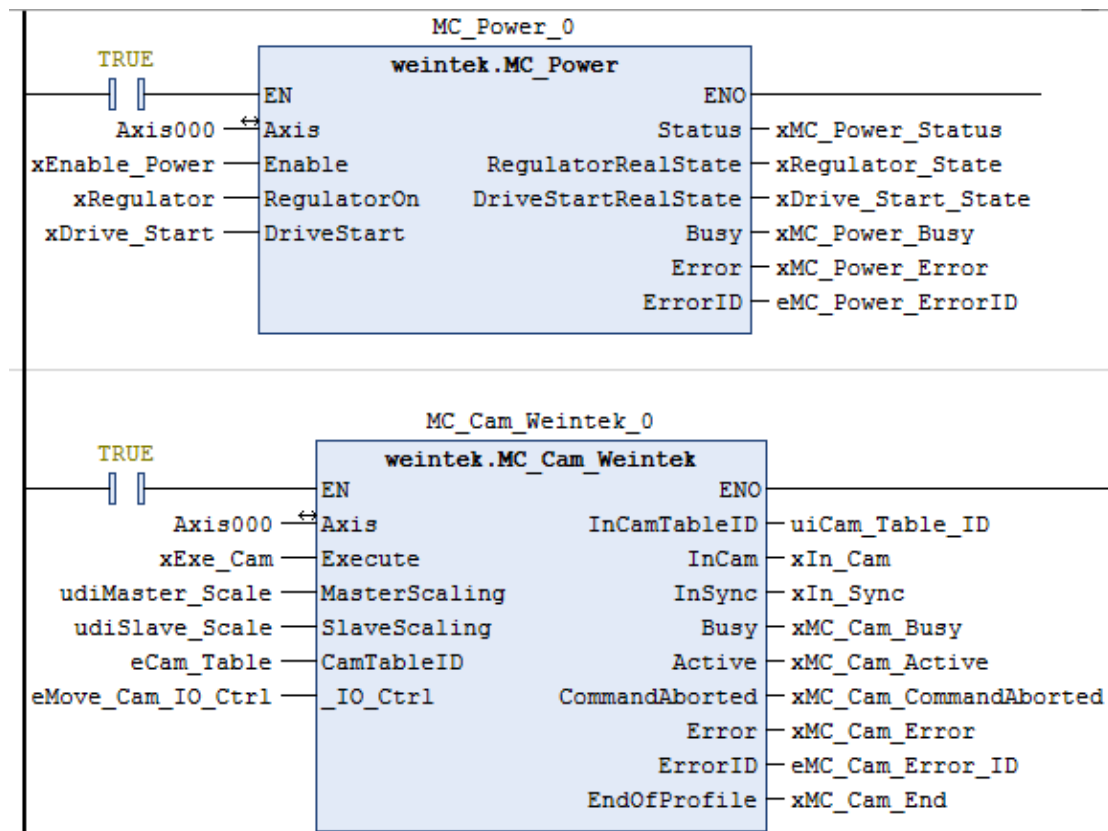
Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
MasterScaling(B)	UDINT	Scale for Master Axis	Specifies the scale for extending or contracting the phase of master axis in Cam Table. The unit is 1/1000. Default value: 1000
SlaveScaling(B)	UDINT	Scale for Slave Axis	Specifies the scale for extending or contracting the phase of slave axis in Cam Table. The unit is 1/1000. Default value: 1000
CamTableID(B*)	eMC_CAM_TABLEID	No. of Cam Table	Specifies the Cam table to be used for engaging the axis by its number. The table can be changed at the beginning of the next cam cycle.
_IO_Ctrl(B)	eMC_IO_Ctrl	I/O Control	None: Not used. I0~I2: Trigger execution using digital input.
Output Parameters	Data Type	Definition	Description
InCamTableID	eMC_CAM_TABLEID	No. of Cam Table for engaging the axes.	Displays the number of the Cam Table being used.
InCam	BOOL	Engage Status	TRUE: Synchronizing the master and slave axis positions according to the definition in Cam Table.
InSync	BOOL	Synchronization Status	TRUE: Slave axis has reached the position of the main axis in the Cam Table.
EndOfProfile	BOOL	Flag of the End of Motion Profile	Indicates the end point in the Cam Table (the start and end point are determined by the direction in which main axis moves). TRUE: A single PLC Task Cycle, this may be the end of a cycle or the beginning of next cycle.
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Active	BOOL	Motion Control	TRUE: Function Block has

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

*EndOfProfile checks whether the CamTableID has changed. If it has changed, the StartMode and Transition Direction will be changed accordingly. MasterAbsolute and SlaveAbsolute will use the parameters previously set, and the end of the current Cam Table will continue to the beginning of the next Cam Table.

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=> xRegulator_State,
DriveStartRealState=> xDrive_Start_State,
Busy=> xMC_Power_Busy,
Error=> xMC_Power_Error,
ErrorID=> eMC_Power_ErrorID);
//MC_Cam_weintek function block
MC_Cam_Weintek_0(
Axis:= Axis000,
Execute:= xExe_Cam,
MasterScaling:= udiMaster_Scale,
SlaveScaling:= udiSlave_Scale,
CamTableID:= eCam_Table,
_IO_Ctrl:= eMove_Cam_IO_Ctrl,
InCamTableID=> uiCam_Table_ID,
InCam=> xIn_Cam,
InSync=> xIn_Sync,
Busy=> xMC_Cam_Busy,
Active=> xMC_Cam_Active,
CommandAborted=> xMC_Cam_CommandAborted,
Error=> xMC_Cam_Error,
ErrorID=> eMC_Cam_Error_ID,
EndOfProfile=>xMC_Cam_End );

```

5.12. MC_TorqueControl(FB)

Function: Continuously exerts a torque or force of the specified magnitude which is approached using a defined ramp, and then sets InTorque output if the commanded torque level is reached.

This Function Block is applicable for drivers that support torque control.

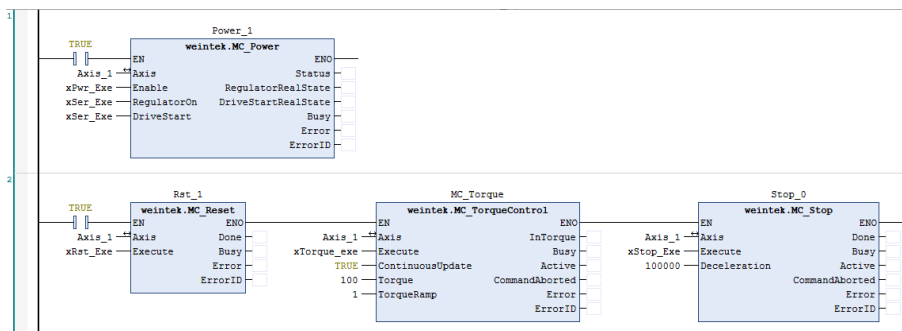
This Function Block is executed when “Execute” turns from FALSE to TRUE.

MC_TorqueControl		
Axis	AXIS_REF_LITE	BOOL InTorque
Execute	BOOL	BOOL Busy
ContinuousUpdate	BOOL	BOOL Active
Torque	INT	BOOL CommandAborted
TorqueRamp	UDINT	BOOL Error
		eAXIS_FB_ERROR ErrorID

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute	Triggered by FALSE to TRUE.
ContinuousUpdate	BOOL	Continuously updates the torque during motion	TRUE= The target torque level can be changed when the axis is operating.
Torque	INT	Value of the Torque	
TorqueRamp	UDINT	Torque Ramp	The maximum time derivative of the set value of the torque.
Output Parameters	Data Type	Definition	Description
InTorque	BOOL	Torque Level Reached	TRUE: The commanded torque level is reached.
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Active	BOOL	Motion Control Status	TRUE: Function Block has control on the axis.
CommandAborted	BOOL	Command Aborted	TRUE: Command is interrupted by other Function Block or event.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	eAXIS_FB_ERROR	Error Code	Error identification, please see Appendix A for more information.
Input / Output Parameters	Data Type	Definition	Description
Axis	AXIS_REF_LITE	Axis Variable Instance	Reference to the axis.

Programming:

LD:



6. Writing Parameters into iR-ETN to configure iR-PU01-P

6.1. ETN_PU Function Block

When iR-ETN is connected to multiple iR-PU01-P modules, it can access data in one object address of one iR-PU01-P at a time. This chapter explains how to write multiple motion control parameters at a time using the ETN_PU function block.

6.2. Accessing iR-PU01-P

Please see index, sub-index, and length in the object dictionary in iR-PU01-P User Manual.

Please see the following table for more information on how to reads or writes iR-PU01-P's parameters over ModbusTCP/IP.

R/W	Address (Hex)	Description				
Write Object	0xFFFF0	Index				
	0xFFFF1	Sub-index (High Byte) Length (Low Byte)				
	0xFFFF2	Hi Byte	0x56		WORD	DWORD
		Lo Byte	0x78	BYTE		
	0xFFFF3	Hi Byte	0x12			
		Lo Byte	0x34			
iR-ETN sequentially writes data into 0xFFFF0~0xFFFF3. Data will be written to iR-PU01-P when iR-ETN writes data into 0xFFFF3.						
Read Object	0xFFFF4	Index				
	0xFFFF5	Sub-index (High Byte) Length (Low Byte)				
	0xFFFF6	Hi Byte	0x56		WORD	DWORD
		Lo Byte	0x78	BYTE		
	0xFFFF7	Hi Byte	0x12			
		Lo Byte	0x34			
Step1: iR-ETN sequentially writes data into 0xFFFF4~0xFFFF5. iR-ETN will start reading iR-PU01-P object when writing data into 0xFFFF5, and the data will be placed in 0xFFFF6~0xFFFF7. Step 2: Read data of 0xFFFF6~0xFFFF7 Object.						

6.3. ETN_PU_SDO(FB)

Function:

Reads or writes an iR-PU01-P parameter.

Please see index, sub-index, and length in the object dictionary in iR-PU01-P User

Manual.

```

ETN_PU_SDO
Read BOOL BOOL Busy
Write BOOL BOOL Done
Index WORD BOOL Error
Sub_Index BYTE
Length BYTE
Modbus_Slave ModbusTCPSlave
Data Modbus_Data
    
```

Input Parameters	Data Type	Definition	Description
Read	BOOL	Execute Read Operation	Triggered by FALSE to TRUE.
Write	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Index	WORD	Address of iR-PU01-P Parameter	Index of Object Dictionary.
Sub_Index	BYTE		Sub Index of Object Dictionary.
Length	BYTE	Data Length to Read / Write	Data length unit = byte
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)
Data	Modbus_Data	Read / Write Data	

6.4. ETN_PU_Pulse_Method(FB)

Function: Write input / output pulse method to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_Pulse_Method
Execute BOOL BOOL Busy
Axis USINT BOOL Done
Pulse_Input_Method USINT BOOL Error
Pulse_Output_Method USINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Pulse_Input_Method	USINT	The Method to Input Pulse	Please see object dictionary in

Pulse_Output_Method	USINT	The Method to Output Pulse	iR-PU01-P User Manual. Index=5501h & 5511h
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.5. ETN_PU_Pulse_Out_Unit(FB)

Function: Writes pulse output unit to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_Pulse_Out_Unit
Execute BOOL BOOL Busy
Axis USINT BOOL Done
Encoder_Increments UDINT BOOL Error
Motor_Revolution UDINT
Motor_Shaft_Revolution UDINT
Driving_Shaft_Revolution UDINT
Feed UDINT
Shaft_Revolution UDINT
Modbus_Slave ModbusTCPSlave
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Encoder_Increments	UDINT	Encoder Increments	Please see object dictionary in iR-PU01-P User Manual. Index=608Fh & 6091h & 6092h
Motor_Revolution	UDINT	Motor Revolution	
Motor_Shaft_Revolution	UDINT	Motor Shaft Revolution	
Driving_Shaft_Revolution	UDINT	Driving Shaft Revolution	
Feed	UDINT	Feed	
Shaft_Revolution	UDINT	Shaft Revolution	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block	TRUE: Function

		Completed	Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.6. ETN_PU_Max_Setting(FB)

Function: Write the maximum allowable value to the designated iR-PU01-P module.
(Axis 0~3)

```

ETN_PU_Max_Setting
Execute BOOL                               BOOL Busy
Axis USINT                                  BOOL Done
Max_Profile_Velocity UDINT                 BOOL Error
Max_Motor_Speed UDINT
Max_Acceleration UDINT
Max_Deceleration UDINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Max_Profile_Velocity	UDINT	Max. Allowable Profile Velocity	Please see object dictionary in iR-PU01-P User Manual. Index = 607Fh & 6080h & 60C5h & 60C6h
Max_Motor_Speed	UDINT	Max. Allowable Motor Speed	
Max_Acceleration	UDINT	Max. Allowable Acceleration Rate	
Max_Deceleration	UDINT	Max. Allowable Deceleration Rate	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.7. ETN_PU_Motion_Config(FB)

Function: Write the configured motion control parameter to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_Motion_Config
Execute BOOL
Axis USINT
Max_Position_Range_Limit DINT
Min_Position_Soft_Limit DINT
Max_Position_Soft_Limit DINT
Quick_Stop_Deceleration UDINT
Profile_Jerk UDINT
Additional_Position_Modulo_Range_1st DINT
Additional_Position_Modulo_Range_2nd DINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Max_Position_Range_Limit	DINT	Max. Position Range Limit	Please see object dictionary in iR-PU01-P User Manual. Index = 607Bh & 607Dh & 6085h & 60A4h & 5528h
Min_Position_Soft_Limit	DINT	Min. Position Soft Limit	
Max_Position_Soft_Limit	DINT	Max. Position Soft Limit	
Quick_Stop_Deceleration	UDINT	Quick Stop Deceleration	
Profile_Jerk	UDINT	Jerk	
Additional_Position_Modulo_Range_1st	DINT	Max. Position Range of 1 st Encoder	
Additional_Position_Modulo_Range_2nd	DINT	Max. Position Range of 2 nd Encoder	
Output Parameters	Data Type	Definition	
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.8. ETN_PU_DI_Setting(FB)

Function: Write digital input setting to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_DI_Setting
Execute BOOL
Axis USINT
Digital_Input_Polarity UDINT
DI_0_Function USINT
DI_1_Function USINT
DI_2_Function USINT
DI_3_Function USINT
DI_A_Function USINT
DI_B_Function USINT
DI_Z_Function USINT
Modbus_Slave ModbusTCPSlave
BOOL Busy
BOOL Done
BOOL Error
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Digital_Input_Polarity	UDINT	Digital Input Polarity	Please see object dictionary in iR-PU01-P User Manual. Index = 5502h & 5503h
DI_0_Function	USINT	DI-0 Function	
DI_1_Function	USINT	DI-1 Function	
DI_2_Function	USINT	DI-2 Function	
DI_3_Function	USINT	DI-3 Function	
DI_A_Function	USINT	DI-A Function	
DI_B_Function	USINT	DI-B Function	
DI_Z_Function	USINT	DI-Z Function	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.9. ETN_PU_DI_Filter(FB)

Function: Write digital input filter setting to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_DI_Filter
Execute BOOL
Axis USINT
DI_0_Filter USINT
DI_1_Filter USINT
DI_2_Filter USINT
DI_3_Filter USINT
DI_A_Filter USINT
DI_B_Filter USINT
DI_Z_Filter USINT
Modbus_Slave ModbusTCPSlave
BOOL Busy
BOOL Done
BOOL Error
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
DI_0_Filter	USINT	DI-0 Filter	Please see object dictionary in iR-PU01-P User Manual. Index = 5504h
DI_1_Filter	USINT	DI-1 Filter	
DI_2_Filter	USINT	DI-2 Filter	
DI_3_Filter	USINT	DI-3 Filter	
DI_A_Filter	USINT	DI-A Filter	
DI_B_Filter	USINT	DI-B Filter	
DI_Z_Filter	USINT	DI-Z Filter	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.10. ETN_PU_DO_Setting(FB)

Function: Write digital output function setting to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_DO_Setting
Execute BOOL BOOL Busy
Axis USINT BOOL Done
Digital_Output_Polarity UDINT BOOL Error
DO_0_Function USINT
DO_1_Function USINT
DO_2_Function USINT
DO_3_Function USINT
DO_PA_Function USINT
DO_PB_Function USINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Digital_Output_Polarity	UDINT	Digital Output Polarity	Please see object dictionary in iR-PU01-P User Manual. Index = 5512h & 5513h
DO_0_Function	USINT	DO-0 Function	
DO_1_Function	USINT	DO-1 Function	
DO_2_Function	USINT	DO-2 Function	
DO_3_Function	USINT	DO-3 Function	
DO_PA_Function	USINT	DO-PA Function	
DO_PB_Function	USINT	DO-PB Function	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.11. ETN_PU_DO_Abort_Option(FB)

Function: Write digital output abortion setting to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_DO_Abort_Option
Execute BOOL
Axis USINT
DO_0_Abort_Option USINT
DO_1_Abort_Option USINT
DO_2_Abort_Option USINT
DO_3_Abort_Option USINT
DO_PA_Abort_Option USINT
DO_PB_Abort_Option USINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
DO_0_Abort_Option	USINT	DO-0 Abortion Option	Please see object dictionary in iR-PU01-P User Manual. Index = 5514h
DO_1_Abort_Option	USINT	DO-1 Abortion Option	
DO_2_Abort_Option	USINT	DO-2 Abortion Option	
DO_3_Abort_Option	USINT	DO-3 Abortion Option	
DO_A_Abort_Option	USINT	DO-A Abortion Option	
DO_B_Abort_Option	USINT	DO-B Abortion Option	
Output Parameters	Data Type	Definition	
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.12. ETN_PU_Home_setting(FB)

Function: Write the Homing setting to the designated iR-PU01-P. (Axis 0~3)

ETN_PU_Home_Setting	
Execute	<i>BOOL</i> <i>BOOL</i> Busy
Axis	<i>USINT</i> <i>BOOL</i> Done
Homing_Method	<i>SINT</i> <i>BOOL</i> Error
Speed_Search_Switch	<i>UDINT</i>
Speed_Search_Zero	<i>UDINT</i>
Home_Offset	<i>DINT</i>
Homing_Acceleration	<i>UDINT</i>
Additional_Home_Offset_1st	<i>DINT</i>
Additional_Home_Offset_2nd	<i>DINT</i>
Modbus_Slave	<i>ModbusTCPSlave</i>

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Homing_Method	USINT	Homing Method	Please see object dictionary in iR-PU01-P User Manual. Index = 6098h & 6099h & 607Ch & 609Ah & 5529h
Speed_Search_Switch	USINT	Homing Speed - Slow	
Speed_Search_Zero	USINT	Homing Speed - Fast	
Home_Offset	USINT	Home Offset	
Homing_Acceleration	USINT	Homing Acceleration Rate	
Additional_Home_Offset_1st	USINT	Home Offset of 1 st Encoder	
Additional_Home_Offset_2nd	USINT	Home Offset of 2 nd Encoder	
Output Parameters	Data Type	Definition	
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.13. ETN_PU_AddPosition_Unit(FB)

Function: Write the add position unit setting to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_AddPosition_Unit
Execute BOOL
Axis USINT
Add_Position_1st_Encoder_Increments UDINT
Add_Position_1st_Motor_Revolution UDINT
Add_Position_1st_Motor_Shaft_Revolution UDINT
Add_Position_1st_Driving_Shaft_Revolution UDINT
Add_Position_1st_Feed UDINT
Add_Position_1st_Shaft_Revolution UDINT
Add_Position_2nd_Encoder_Increments UDINT
Add_Position_2nd_Motor_Revolution UDINT
Add_Position_2nd_Motor_Shaft_Revolution UDINT
Add_Position_2nd_Driving_Shaft_Revolution UDINT
Add_Position_2nd_Feed UDINT
Add_Position_2nd_Shaft_Revolution UDINT
Modbus_Slave ModbusTCPSlave
    
```

BOOL Busy
BOOL Done
BOOL Error

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Add_Position_1st_Encoder_Increments	UDINT	1st Encoder Increments	Please see object dictionary in iR-PU01-P User Manual. Index = 60E6h & 60EBh & 60E8h & 60EDh & 60E9h & 60EEh
Add_Position_1st_Motor_Revolution	UDINT	Motor Revolution of 1 st Encoder	
Add_Position_1st_Motor_Shaft_Revolution	UDINT	Motor Shaft Revolution of 1 st Encoder	
Add_Position_1st_Driving_Shaft_Revolution	UDINT	Driving Shaft Revolution of 1 st Encoder	
Add_Position_1st_Feed	UDINT	Feed of 1 st Encoder	
Add_Position_1st_Shaft_Revolution	UDINT	Shaft Revolution of 1 st Encoder	
Add_Position_2nd_Encoder_Increments	UDINT	2 nd Encoder Increments	
Add_Position_2nd_Motor_Revolution	UDINT	Motor Revolution of 2 nd Encoder	
Add_Position_2nd_Motor_Shaft_Revolution	UDINT	Motor Shaft Revolution of 2 nd Encoder	
Add_Position_2nd_Driving_Shaft_Revolution	UDINT	Driving Shaft Revolution of 2 nd Encoder	
Add_Position_2nd_Feed	UDINT	Feed of 2 nd Encoder	
Add_Position_2nd_Shaft_Revolution	UDINT	Shaft Revolution of 2 nd Encoder	
Output Parameters	Data Type	Definition	
Busy	BOOL	Function Block	TRUE: Function

		Status	Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.14. ETN_PU_Motion_DIO_Setting(FB)

Function: Write the setting of digital input /output during motion to the designated iR-PU01-P. (Axis 0~3)

```

ETN_PU_Motion_DIO_Setting
Execute BOOL
Axis USINT
Motion_Output_Setting_0 UDINT
Motion_Output_Setting_1 UDINT
Motion_Output_Setting_2 UDINT
Motion_Trigger_Setting_0 UINT
Motion_Trigger_Setting_1 UINT
Motion_Trigger_Setting_2 UINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Motion_Output_Setting_0	UDINT	Digital Output Setting in Motion - 0.	Please see object dictionary in iR-PU01-P User Manual. Index = 558Fh & 559Fh
Motion_Output_Setting_1	UDINT	Digital Output Setting in Motion - 1.	
Motion_Output_Setting_2	UDINT	Digital Output Setting in Motion - 2.	
Motion_Input_Setting_0	UINT	Digital Input Setting in Motion - 0.	
Motion_Input_Setting_1	UINT	Digital Input Setting in Motion - 1.	
Motion_Input_Setting_2	UINT	Digital Input Setting in Motion - 2.	

Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.15. ETN_PU_PWM_Setting(FB)

Function: Write PWM setting to the designated iR-PU01-P. (Axis 0~3)

```

Execute BOOL
Axis USINT
D0_PWM_Setting UDINT
D1_PB_PWM_Setting UDINT
Modbus_Slave ModbusTCPSlave
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
D0_PWM_Setting	UDINT	DO-0 Output PWM Setting	Please see object dictionary in iR-PU01-P User Manual. Index = 551Ah
D1_PB_PWM_Setting	UDINT	DO-1 & PB Output PWM Setting	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

6.16. ETN_PU_Axis_Setting

Function: Write the setting for other axes to the designated iR-PU01-P. (Axis 0~3)

ETN_PU_Axis_Setting		
Execute	<i>BOOL</i>	<i>BOOL</i> Busy
Axis	<i>USINT</i>	<i>BOOL</i> Done
Cycle	<i>UDINT</i>	<i>BOOL</i> Error
Bias_Velocity	<i>UDINT</i>	
Backlash_Compensation	<i>UINT</i>	
Modbus_Slave	<i>ModbusTCPSlave</i>	

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ETN. Axis : 0~3
Cycle	UDINT	Motion Scan Cycle	Please see object dictionary in iR-PU01-P User Manual. Index = 5520h & 5521h
Bias_Velocity	UDINT	Bias Velocity	
Backlash_Compensation	UINT	Backlash Compensation	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
Input / Output Parameters	Data Type	Definition	Description
Modbus_Slave	ModbusTCPSlave	Designate ModbusTCPSlave Device	Designate the ModbusTCPSlave device (iR-ETN)

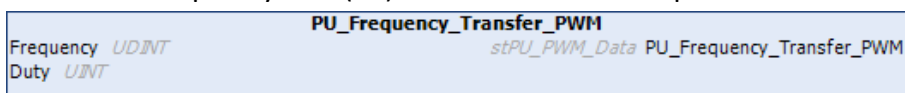
7. PU_PWM(FB)

7.1. Overview

This function block is for users to dynamically change iR-PU01-P’s PWM parameters. (For more information, please see the Object Dictionary in iR-PU01-P User Manual and find PWM Output Setting.)

7.2. PU_Frequency_Transfer_PWM(FUN)

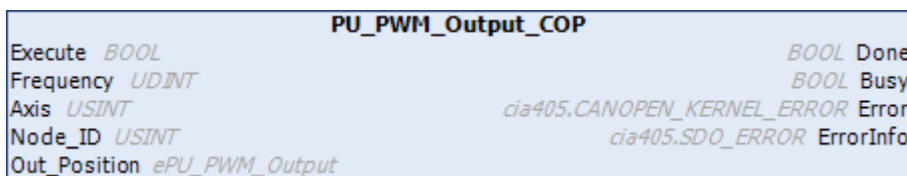
Function: Convert frequency unit (Hz) to iR-PU01-P’s PWM parameter.



Input Parameters	Data Type	Definition	Description
Frequency	USINT	Frequency	The unit is Hz.
Duty	UINT	Duty Cycle	The range is 0%~100%.
Output Parameters	Data Type	Definition	Description
PU_Frequency_Transfer_PWM	stPU_PWM_Data	Set iR-PU01-P PWM data.	Convert frequency unit to iR-PU01-P’s PWM parameter.

7.3. PWM_Output_COP(FB)

Function: Convert frequency unit (Hz) to iR-PU01-P’s PWM parameter and write the data to iR-PU01-P. (Exclusive to iR-COP)



Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Frequency	UDINT	Frequency	The unit is Hz.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-COP. Axis : 0~3
Node_ID	USINT	iR-COP Node ID	
Out_Position	ePU_PWM_Output	Designate an output position.	The output position can be DO-0, DO-1, or PB.
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes

			writing.
Error	CANOPEN_KERNEL_ERROR	CANopen Error Code	See CiA405 function block error code.
ErrorInfo	SDO_ERROR	SDO Error Code	See CiA301 error code.

7.4. PU_PWM_Output_ECATOR(FB)

Function: Convert frequency unit (Hz) to iR-PU01-P's PWM parameter and write the data to iR-PU01-P. (Exclusive to iR-ECAT)

```

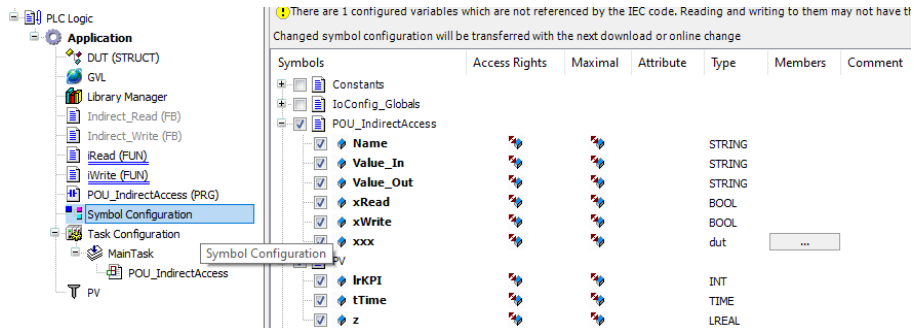
PU_PWM_Output_ECATOR
Execute BOOL BOOL Done
Frequency UDINT BOOL Busy
Axis USINT BOOL Error
Device UINT ETC_CO_ERROR ErrorInfo
Out_Position ePU_PWM_Output
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	Triggered by FALSE to TRUE.
Frequency	UDINT	Frequency	The unit is Hz.
Axis	USINT	Designate iR-PU01-P Module	Designate an iR-PU01-P connected to iR-ECAT. Axis : 0~3
Device	UINT	EtherCAT address	iR-ECAT's EtherCAT address.
Out_Position	ePU_PWM_Output	Designate an output position.	The output position can be DO-0, DO-1, or PB.
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes writing.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorInfo	ETC_CO_ERROR	Error Code	Enumeration ETC_CO_ERR in IODrvEtherCAT library

8. VAR_Access

8.1. Overview

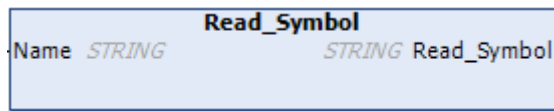
VAR_ACCESS can access tag values after creating the tags in Symbol Configuration.



VAR_ACCESS can only access IEC datatype of tags including: BOOL, BYTE, WORD, DWORD, LWORD, SINT, INT, DINT, LINT, USINT, UINT, UDINT, ULINT, REAL, LREAL, STRING, WSTRING, TIME, DATE

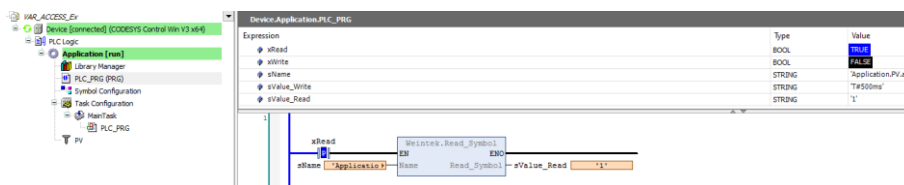
8.2. Read_Symbol

Function: Get tag value by entering tag name.



Input Parameters	Data Type	Definition	Description
Name	STRING	Tag name	
Output Parameters	Data Type	Definition	Description
Read_Symbol	STRING	Returned value	

Demonstration:



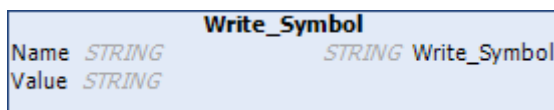
The tag name must be a full name. In the demonstration above, tag “a” under PV is read; therefore, the full name is:

Root Directory(Application).Sub Directory(PV).Tag(a)

The full name in this demonstration is: Application.PV.a

8.3. Write_Symbol

Function: Write the tag by entering its name and value.



Input Parameters	Data Type	Definition	Description
Name	STRING	Tag name	
Value	STRING	Tag value	

Demonstration:

Expression	Type	Value
!Read	BOOL	FALSE
!Write	BOOL	TRUE
sName	STRING	Application.PV.b
sValue_Write	STRING	T#500ms
sValue_Read	STRING	'1'

The tag name must be a full name. In the demonstration above, tag “b” under PV is written; therefore, the full name is:

Root Directory(Application).Sub Directory(PV).Tag(b)

The full name in this demonstration is: Application.PV.b

Expression	Type	Value
a	INT	1
b	TIME	T#500ms

9. Weintek_iBus_Library

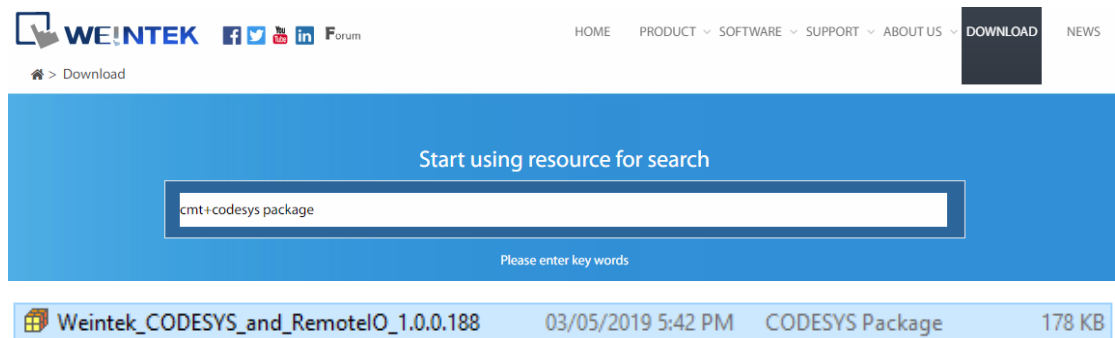
9.1. Overview

Weintek_iBus_Library function block is available exclusively for cMT-CTRL01. This function block can read / write the parameters of the iR modules connected to cMT-CTRL01.

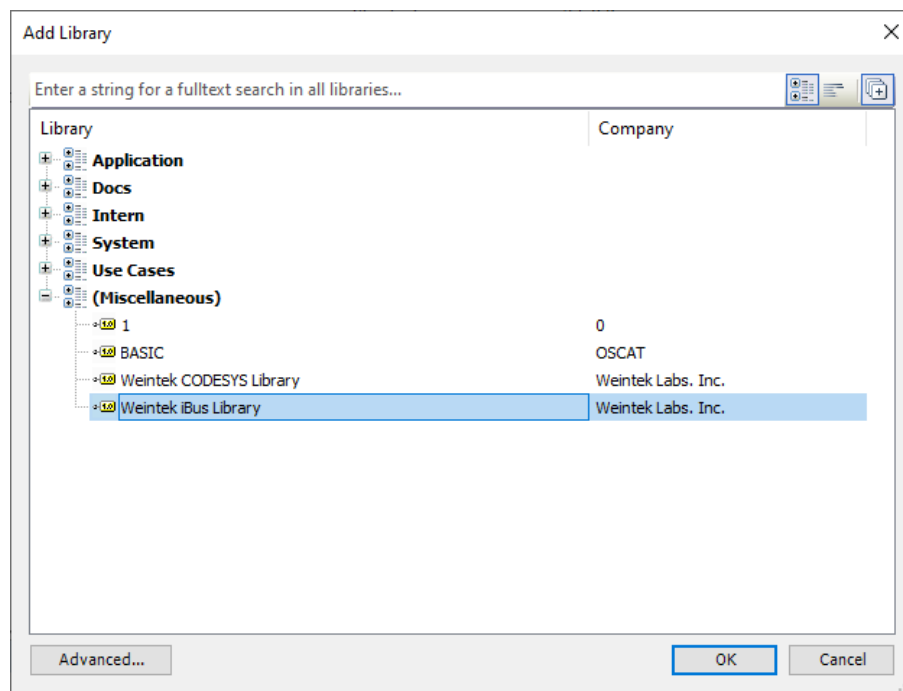
Step 1. Open the download page on Weintek official website and search for [cMT+CODESYS Package] to download and install the package.

<https://www.weintek.com/globalw/Download/Download.aspx>

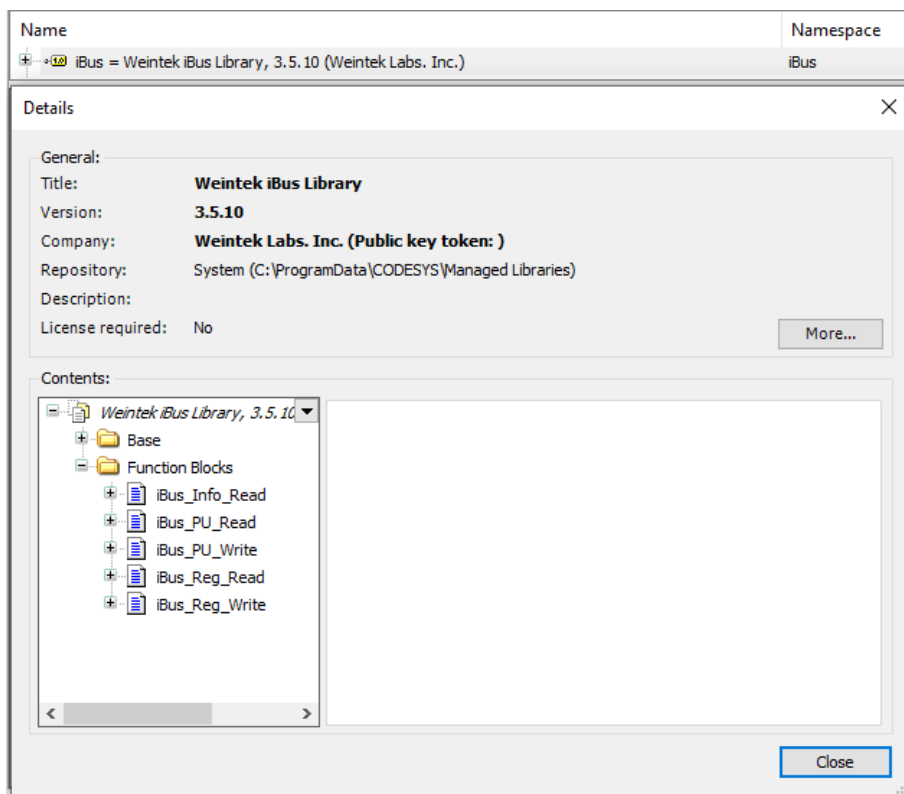
※Ver.1.0.0.280 or later and in CODESYS, Weintek_CODESYS_Library is automatically installed.



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

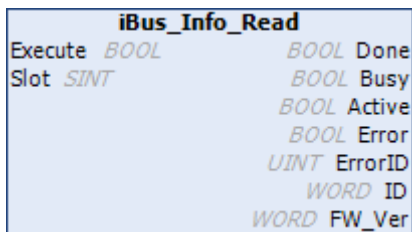


Step 3. Open Details window and the function blocks can be found.



9.2. iBus_Info_Read

Function: Specify an iR module to read its module information.



Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Read Operation	
Slot	SINT	Module Slot Number	Number: 0~15
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes reading.
Active	BOOL	Operation Status	TRUE: Function Block's command is working.
Error	BOOL	Error Status	TRUE: An error has occurred

			within the Function Block.
ErrorID	UINT	Error Code	See Appendix D in this manual.
ID	WORD	iR Module's ID Number	
FW_Ver	WORD	Firmware Version	

9.3. iBus_PU_Read

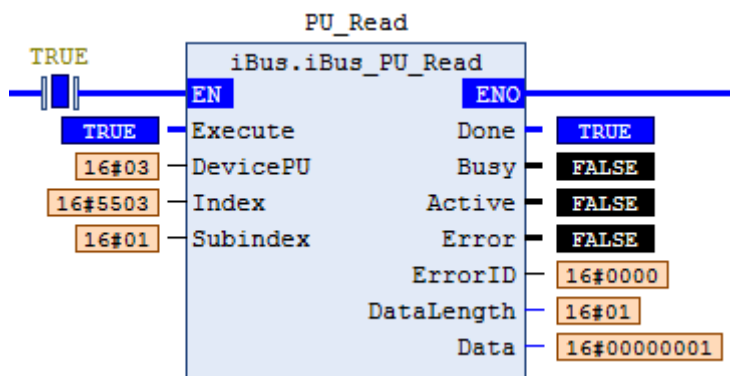
Function: Specify an iR-PU01-P module by entering its slot number, index and sub-index in order to read iR-PU parameters.

```

iBus_PU_Read
Execute BOOL                               BOOL Done
DevicePU BYTE                               BOOL Busy
Index WORD                                   BOOL Active
Subindex BYTE                               BOOL Error
                                           UINT ErrorID
                                           BYTE DataLength
                                           DWORD Data
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Read Operation	
DevicePU	BYTE	Module Slot Number	Number: 0~3
Index	WORD	Index	See Object Dictionary in iR-PU01-P user manual.
SubIndex	BYTE	Sub-index	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes reading.
Active	BOOL	Operation Status	TRUE: Function Block's command is working.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	UINT	Error Code	See Appendix D in this manual.
DataLength	BYTE	Data Length	The unit is Byte
Data	DWORD	Data	

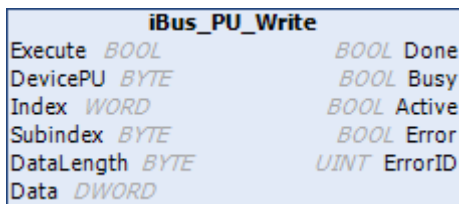
Demonstration:



Trigger “Execute” to read the data of the fourth iR-PU01-P module:
 Index = 16#5503, Sub Index = 01, Data Length = 1 (1Byte), Value = 1.

9.4. iBus_PU_Write

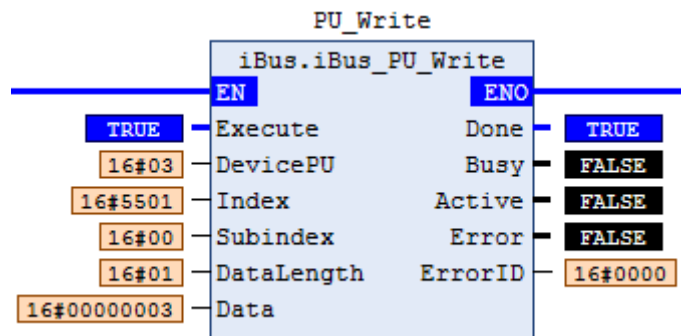
Function: Specify an iR-PU01-P module by entering its slot number, index and sub-index in order to write iR-PU parameters.



Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	
DevicePU	BYTE	Module Slot Number	Number: 0~3
Index	WORD	Index	See Object Dictionary in iR-PU01-P user manual.
SubIndex	BYTE	Sub-index	
DataLength	BYTE	Data Length	The unit is Byte
Data	DWORD	Data	
Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes reading.
Active	BOOL	Operation Status	TRUE: Function Block's command is working.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.

ErrorID	UINT	Error Code	See Appendix D in this manual.
---------	------	------------	--------------------------------

Demonstration:



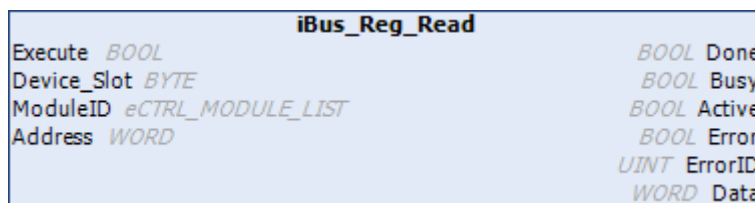
Trigger “Execute” to write the data to the fourth iR-PU01-P module:
 Index = 16#5503, Sub Index = 00, Data Length = 1 (1Byte), Value = 3.

9.5. iBus_Reg_Read

Function:

Mode 1: ModuleID=BySlot, enter the slot number and the address to read the value.

Mode 2 : ModuleID=Module ID, enter the module ID and the address to read the value.



Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Read Operation	
Device_Slot	BYTE	Module Slot Number	When Module ID=BySlot, specify the module by its slot number. When Module ID=ModuleID, specify the module by its ID number.
ModuleID	WORD	Module ID Number	
Address	BYTE	Address	See iR-AQ01-VI, iR-AM06-VI, iR-AI01-VI User Manual, and iR-AI01-TR

Output Parameters	Data Type	Definition	Description
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes reading.
Active	BOOL	Operation Status	TRUE: Function Block's command is working.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	UINT	Error Code	See Appendix D in this manual.
Data	DWORD	Data	

9.6. iBus_Reg_Write

Function:

Mode 1: ModuleID=BySlot, enter the slot number and the address to write the value.

Mode 2 : ModuleID=Module ID, enter the module ID and the address to write the value.

```

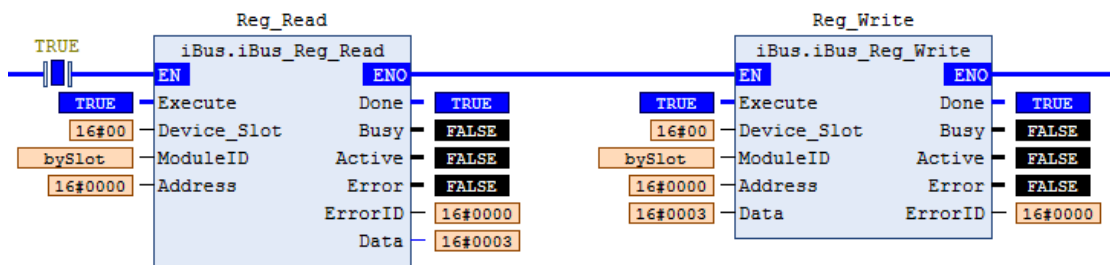
iBus_Reg_Write
Execute BOOL
Device_Slot BYTE
ModuleID eCTRL_MODULE_LIST
Address WORD
Data WORD
BOOL Done
BOOL Busy
BOOL Active
BOOL Error
UINT ErrorID
    
```

Input Parameters	Data Type	Definition	Description
Execute	BOOL	Execute Write Operation	
Device_Slot	BYTE	Module Slot Number	When Module ID=BySlot, specify the module by its slot number. When Module ID=ModuleID, specify the module by its ID number.
ModuleID	WORD	Module ID Number	
Address	BYTE	Address	See iR-AQ01-VI, iR-AM06-VI, iR-AI01-VI User Manual, and iR-AI01-TR UserManual
Data	DWORD	Data	
Output Parameters	Data Type	Definition	Description

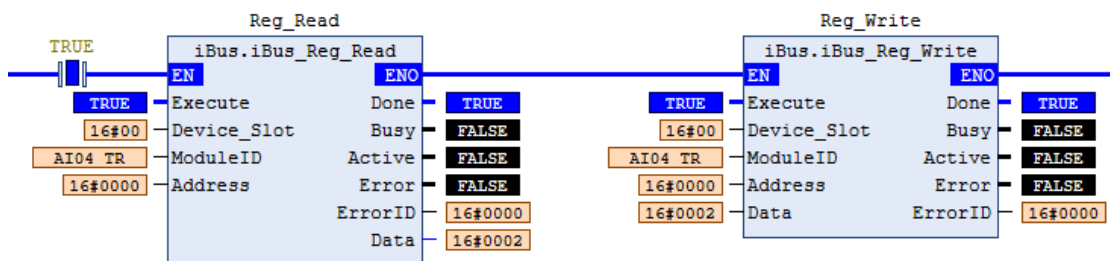
Busy	BOOL	Function Block Status	TRUE: Function Block is being executed.
Done	BOOL	Function Block Completed	TRUE: Function Block finishes reading.
Active	BOOL	Operation Status	TRUE: Function Block's command is working.
Error	BOOL	Error Status	TRUE: An error has occurred within the Function Block.
ErrorID	UINT	Error Code	See Appendix D in this manual.

Demonstration of iBus_Reg_Read & iBus_Reg_Write:

Mode 1 (ModuleID=BySlot): Trigger "Execute" to write data to the first iR module's register. Address = 0, Value = 3, and then read this value.



Mode 2 (ModuleID=ModuleID): Trigger "Execute" to write data to the first thermal module's register. Address = 0, Value = 2, and then read this value.



Appendix A. Motion Control FB Error Code

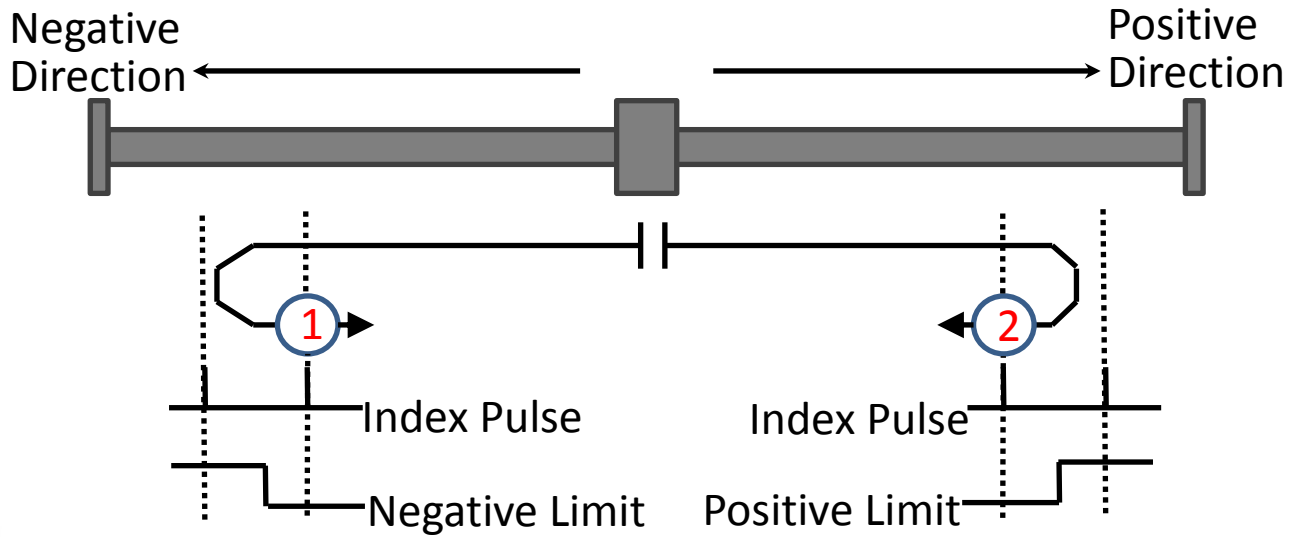
Error Code	State	Description	Error Handling
0	NO_ERROR	No Error	
1	AXIS_NOT_READY	The axis is not ready for operation.	After resolving other errors, enable MC_Power, wait until the Status turns to True, and then restart.
2	AXIS_BUFFER_FULL	Positioning Buffer is full.	Please modify the program to avoid 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.

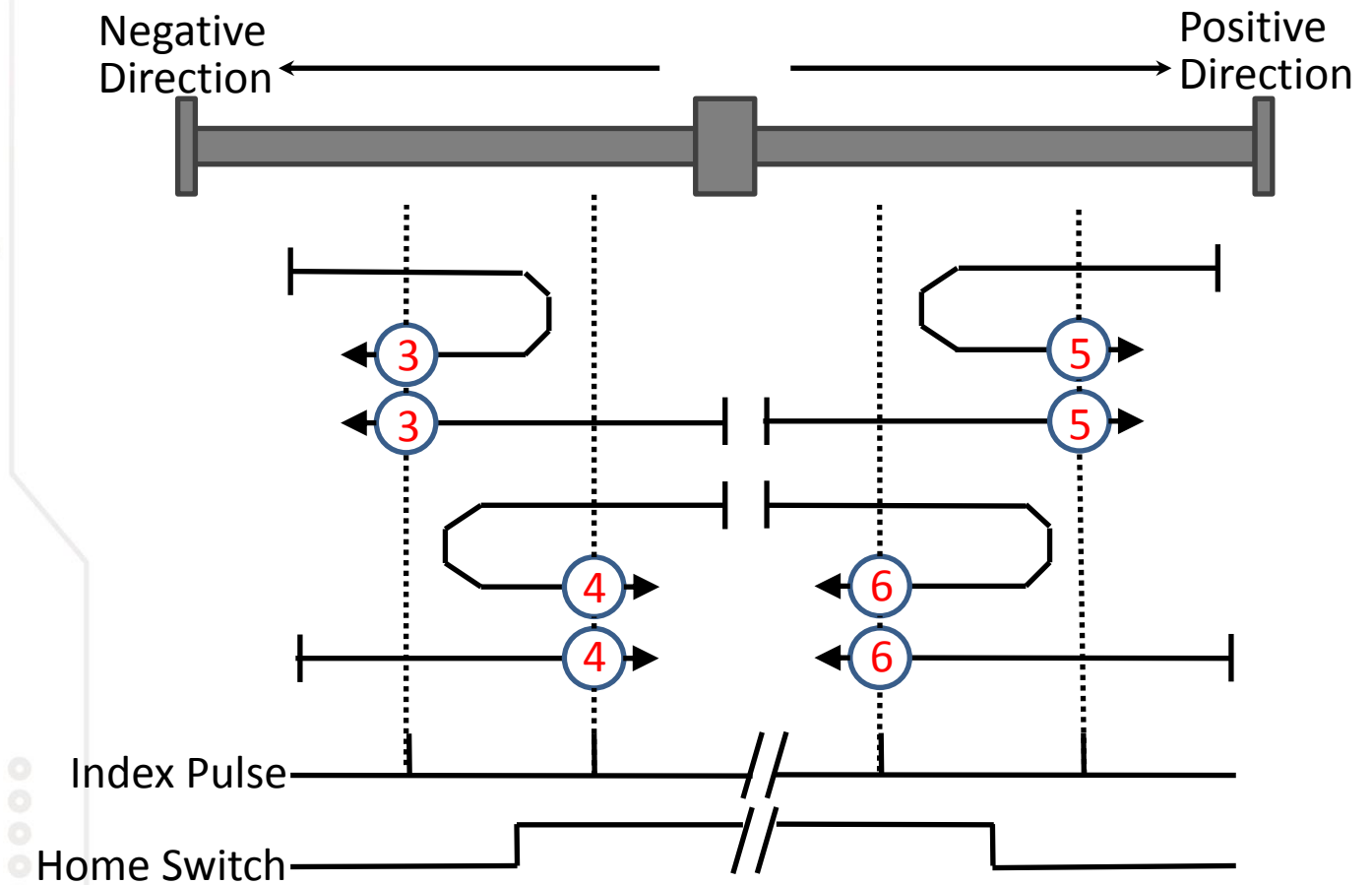
Appendix B. Homing Methods

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.

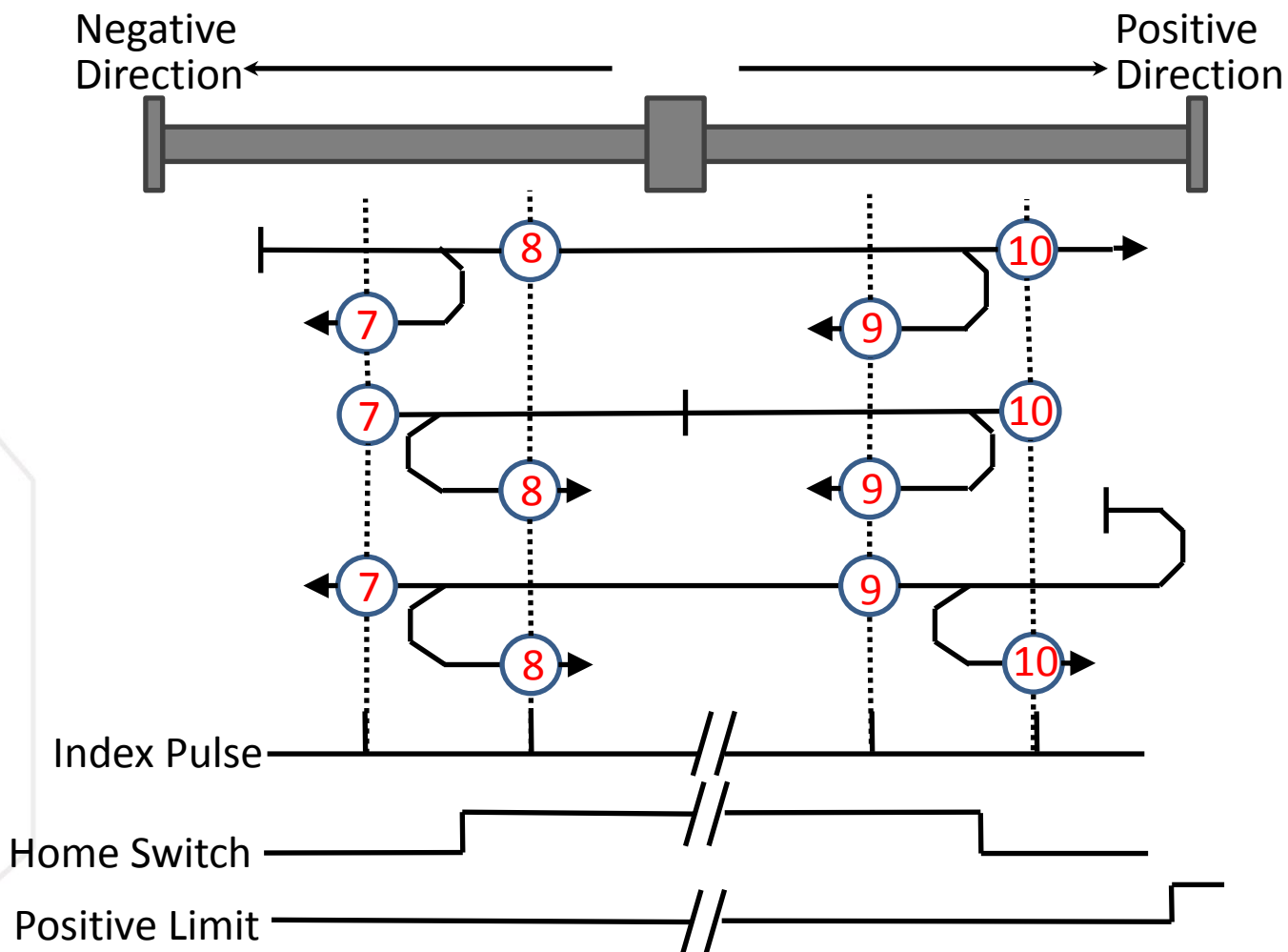


Method 3~6: Homing on positive / negative home switch and index pulse.
 The direction of movement is dependent on the state of home switch. The home position shall be at the index pulse to either to the left or right of the point where the home switch changes state.



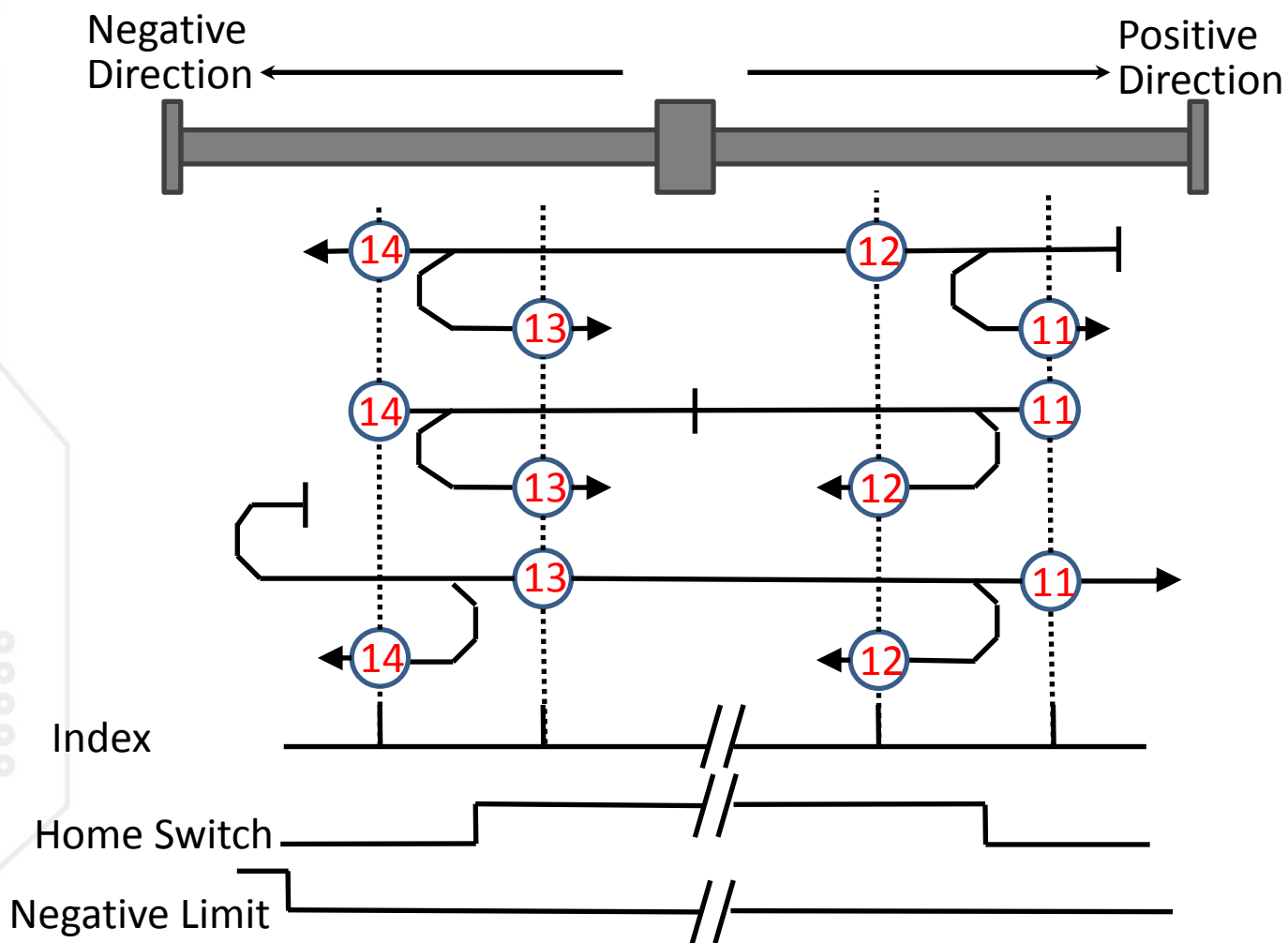
Method 7~10: Homing on home switch and index pulse – positive initial motion

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.



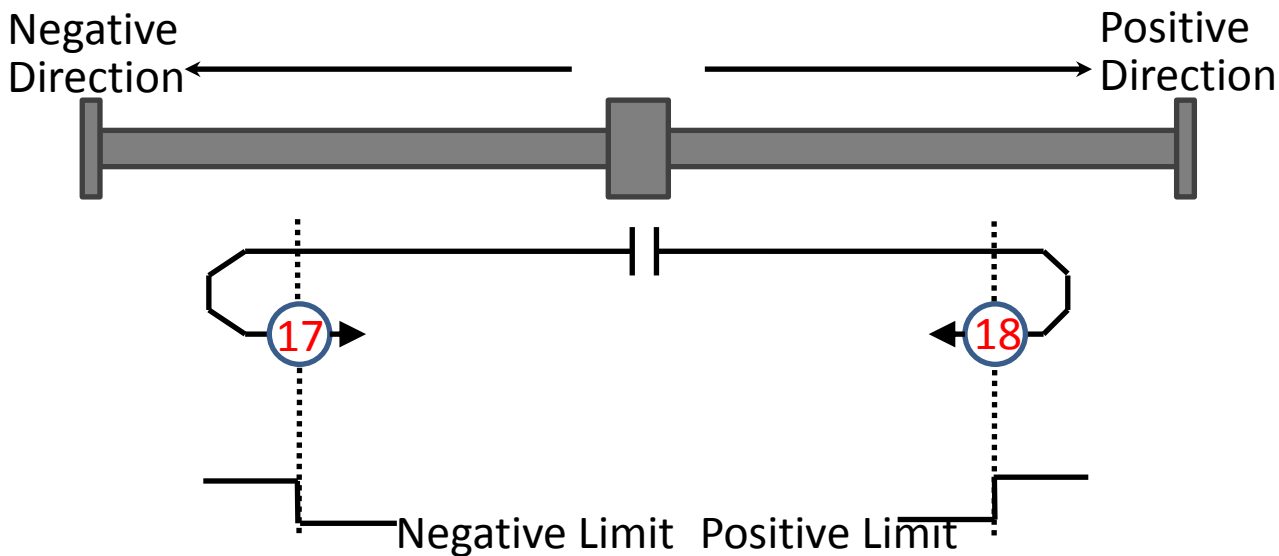
Method 11~14: Homing on home switch and index pulse – negative initial motion

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.



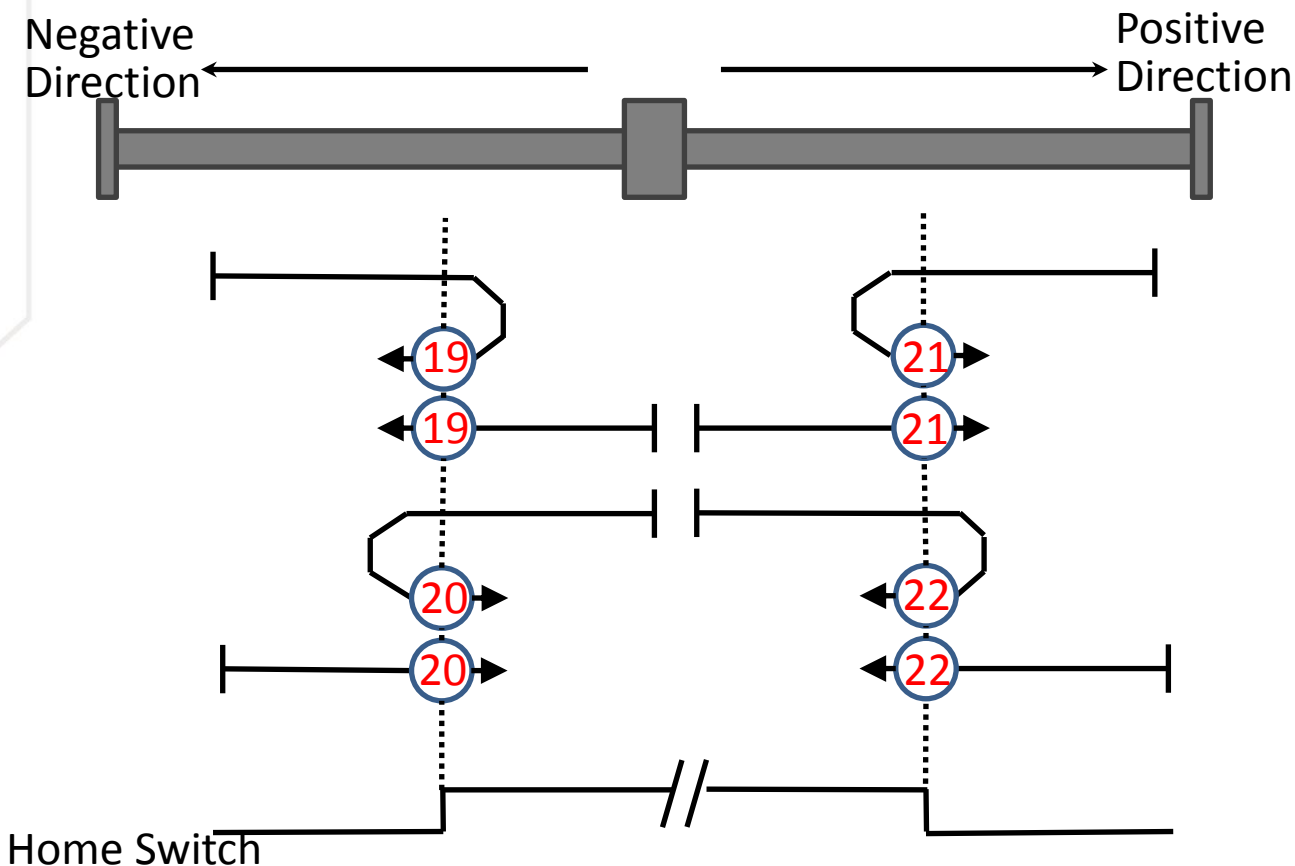
Method 17~18: Homing on negative / positive limit switch

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

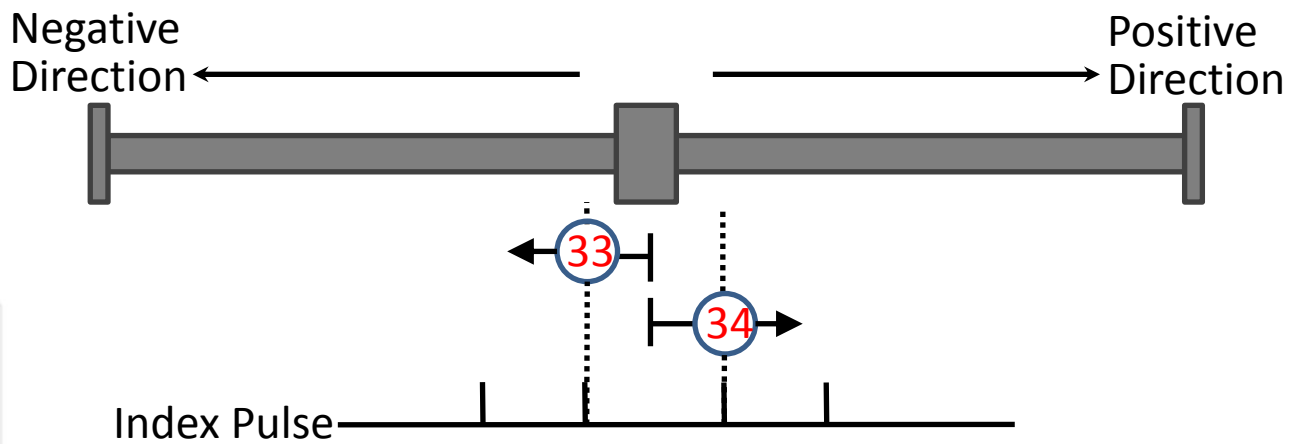
Similar to Method 7~10 except the home position is not dependent on the index pulse.

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

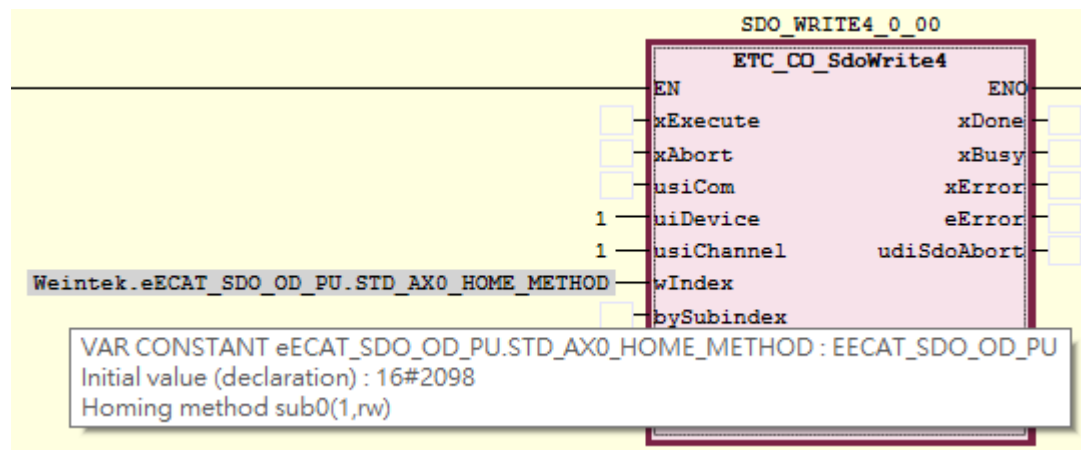
In this method, the position sensor information shall be taken to be the home position.

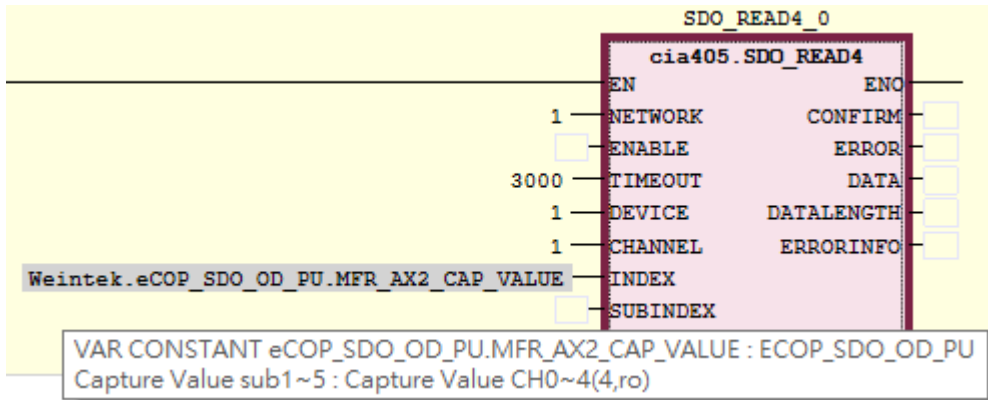
Position actual value = Home offset.

Appendix C. Enum

Name	Description
eAXIS_CiA402_Mode	Motion mode in CiA402.
eAXIS_FB_ERROR	Error code that is output when an error has occurred within the Function Block.
eAXIS_STATE	State of the axis.
eCOP_SDO_OD_PU	Specifies the Object Index of iR-PU01-P stored by SDO in CANopen.
eECAT_SDO_OD_PU	Specifies the Object Index of iR-PU01-P stored by SDO in EtherCAT.
eMC_BUFF_MODE	BufferMode of the positioning function block.
eMC_CAM_TABLEID	ID number of Cam Table for MC_CAM.
eMC_DIRECTION	Direction setting for MC_MoveAbsolute.
eMC_IO_CTRL	Specifies combination for IO control.

The read/write parameters usually used for SDO in CiA402 and iR-PU01-P user-defined object are listed in eCOP_SDO_OD_PU and eECAT_SDO_OD_PU. The notes of each parameter will show its sub-index, length, read / write property and settings information. These can be used with the SDO function block of CANopen and EtherCAT offered by CODESYS.





Appendix D.iBus FB Error Code

Error Code	Description	Error Handling
16#1	iBus command buffer is full	Use less than 256 iBus function blocks.
16#2	Maximum number of slots exceeded	Use less than 15 slots.
16#3	Timeout	Use latest version of CODESYS Runtime and iR_Slave.