WEINTEK LABS., INC.

Positioning Control

Absolute/Relative

Function Block

MC_MoveAbsolute/MC_MoveRelative

Demo Project

Contents

1. Overview and Operation

Overview

This demo project introduces how to use Weintek Library Function Block and iR-PU01-P to perform positioning control by outputting pulse signals to servo/stepper motors.

CODESYS can be used to control iR-PU01-P to output pulse signals to a servo/stepper motor, which determines the distance and speed in which the motor rotates.

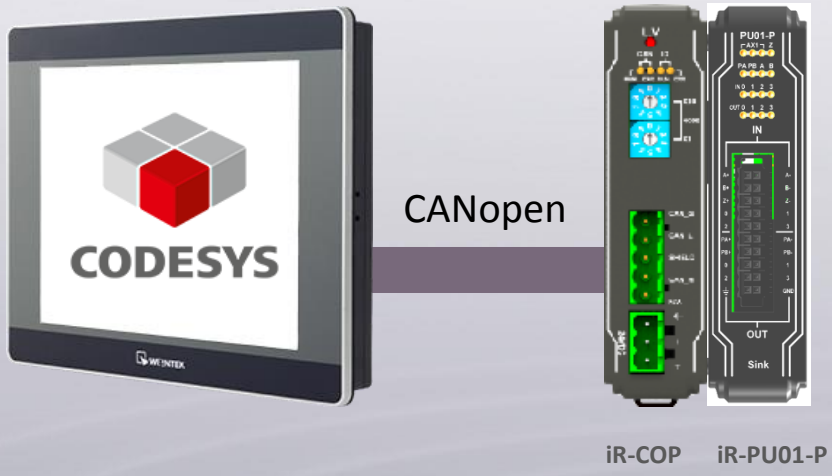
Use iR-COP V1.03 and please see the related demo projects according to the device used:

Using iR-COP: See iR_Application_Positioning_Demo_20190906

Using cMT-CTRL01: See iR_Application_ Positioning _Demo_CTRL_20200708

Using cMT Series HMI: See iR_Application_ Positioning _Demo_HMI_20200708

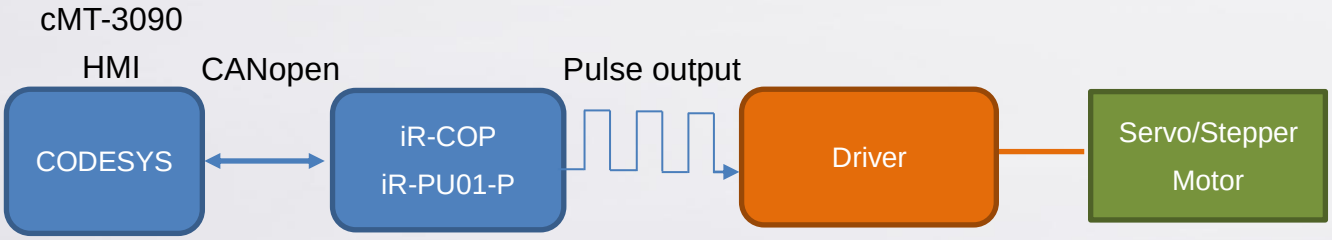
System

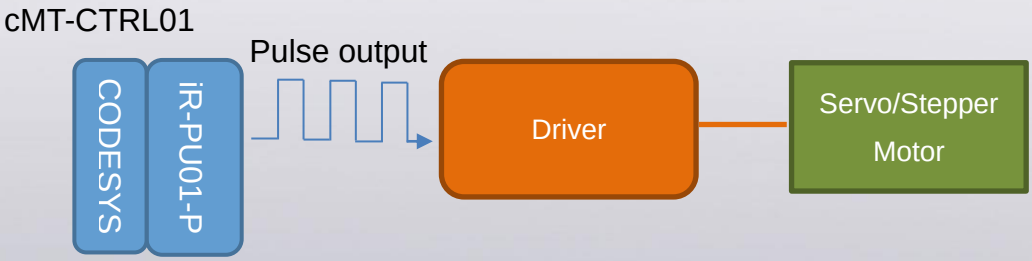


iR-COP iR-PU01-P

cMT-CTRL01 iR-PU01-P

Operation





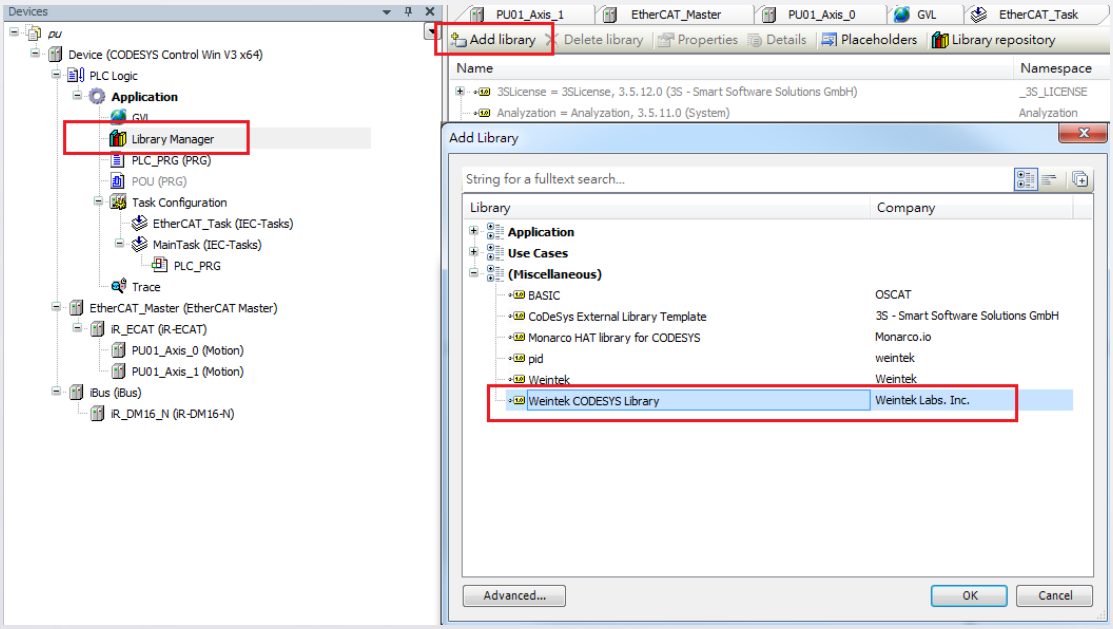
2. Installing Weintek Library

Step 1. In the download page in Weintek official website, search for [cMT+CODESYS Package], and then download and install the package.

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

(The description file of iR-PU01-P is included in the package)

Step 2. In CODESYS interface add Weintek CODESYS Library.



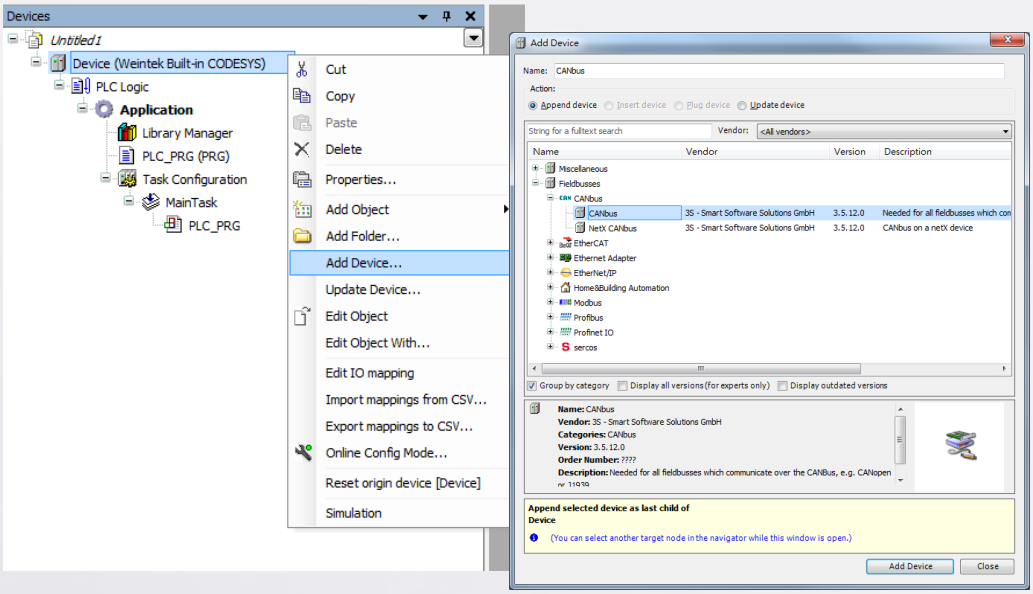
Step 3. Motion Function Block is ready for use after installation.

3. Adding iR-PU01-P to CODESYS Project

Adding iR-PU01-P by using Weintek Built-in CODESYS:

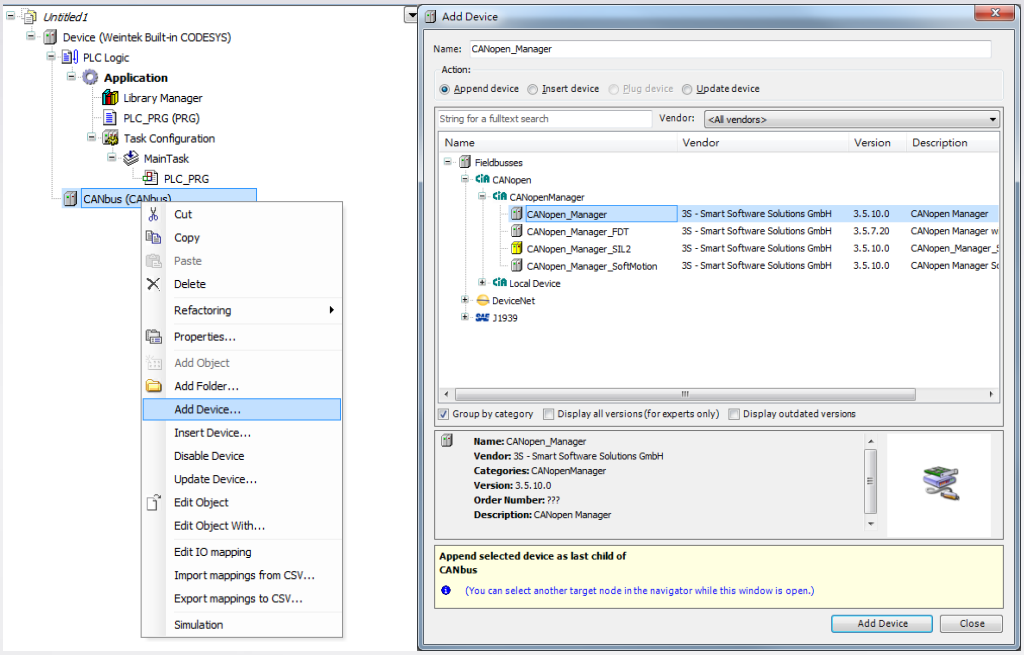
- Add CANbus device:

  [Device]->[Add Device]->[Fieldbusses]->[CANbus]



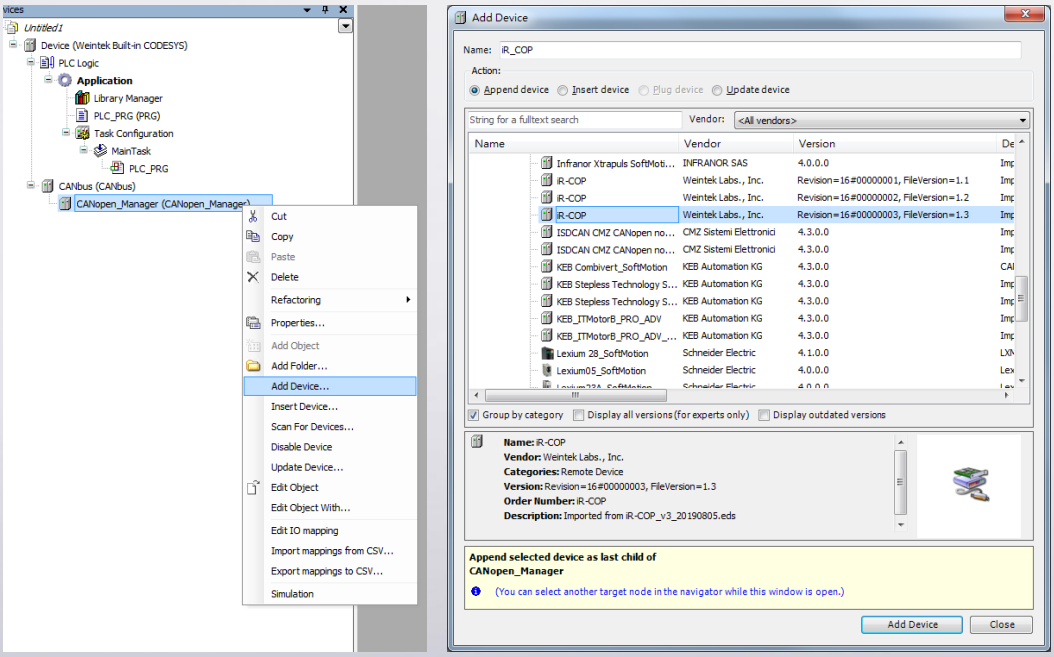
- Add CANopen_Manager device:

  [CANbus]->[Add Device]->[CANopen_Manager]



- Add iR-COP module:

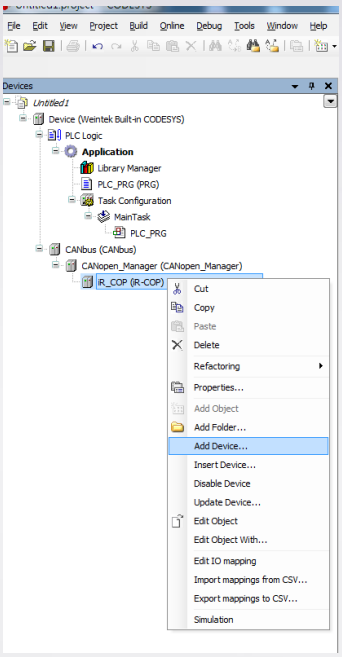
  [CANopen_Manager] ->[Add Device]->[iR-COP] (V1.3)

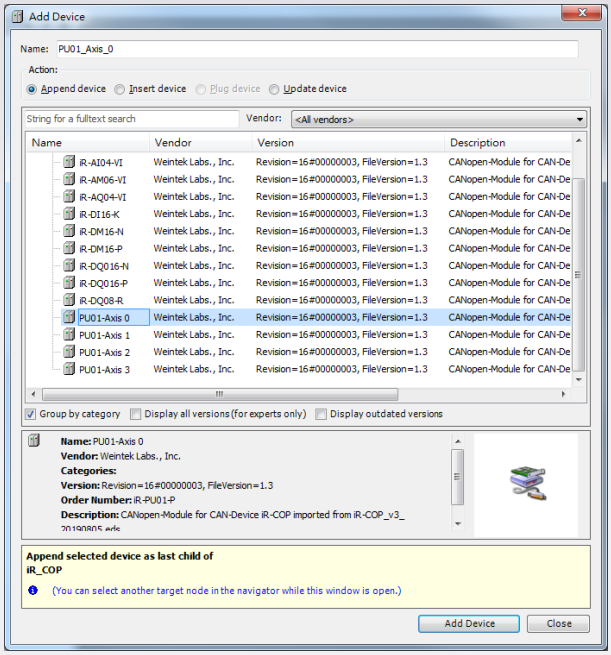


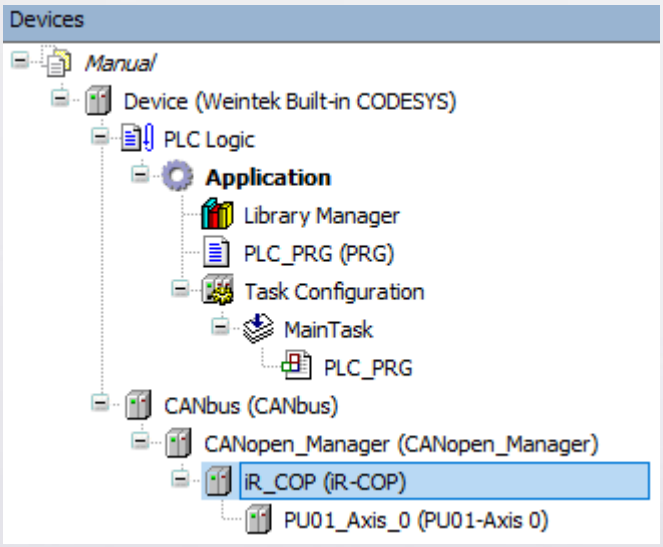
- Add iR-PU01-P module:

  [iR-COP]->[Add Device]->[PU01-Axis 0]

Positioning Control Absolute/Relative



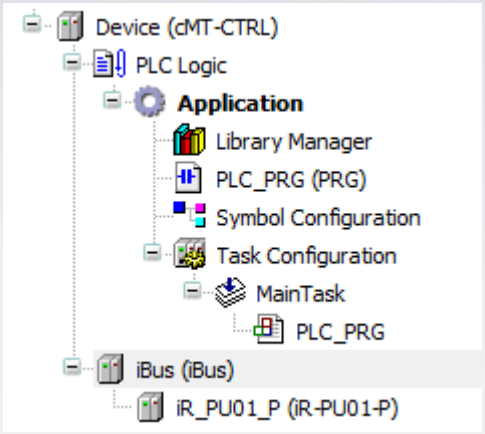




Adding iR-PU01-P by using cMT-CTRL:

- Add iR-PU01-P device:

  [iBus]->[Add Device]->[Miscellaneous]->[iR-PU01-P]


Device (cMT-CTRL)
PLC Logic
Application
Library Manager
PLC_PRG (PRG)
Symbol Configuration
Task Configuration
MainTask
PLC_PRG
iBus (iBus)
iR_PU01_P (iR-PU01-P)

4. iR-PU01-P Parameter Settings

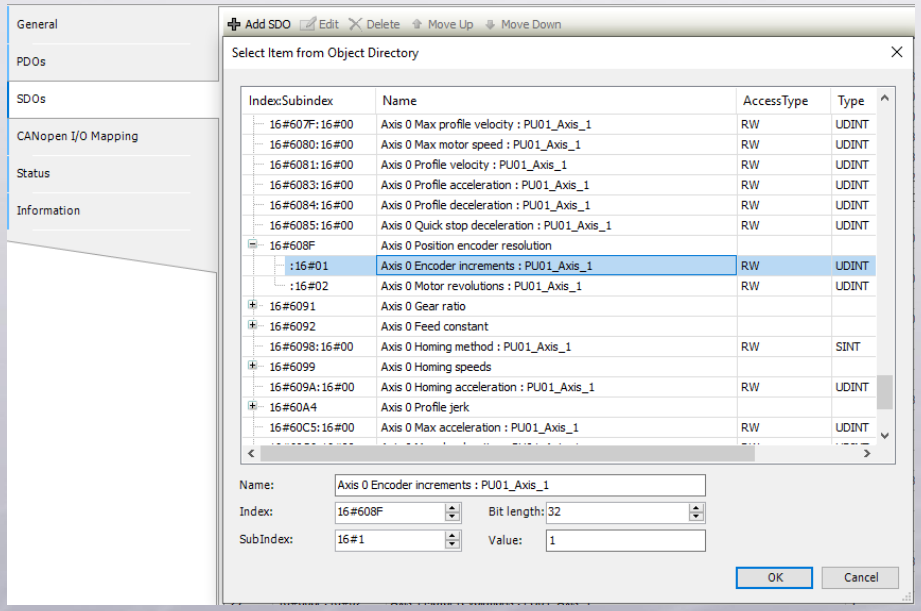
Settings relating to positioning control:

Line	Index:Subindex	Name	Value	Bit length
1	16#608F:16#01	Encoder increments : AX1_PU01	16#1	32
2	16#608F:16#02	Axis 1 Motor revolutions : PU01_Axis_1	1	32
3	16#5511:16#00	Axis 1 Pulse Output Method : PU01_Axis_1	4	8
4	16#6080:16#00	Axis 1 Max motor speed : PU01_Axis_1	200000	32
5	16#607F:16#00	Axis 1 Max profile velocity : PU01_Axis_1	200000	32
6	16#60C5:16#00	Axis 1 Max acceleration : PU01_Axis_1	100000	32
7	16#60C6:16#00	Axis 1 Max deceleration : PU01_Axis_1	100000	32
8	16#6085:16#00	Axis 1 Quick stop deceleration : PU01_Axis_1	100000	32
9	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
10	16#607C:16#00	AX1 Home offset	0	32
11	16#6099:16#01	Speed during search for switch : AX1_PU01	10000	32
12	16#6099:16#02	Speed during search for zero : AX1_PU01	2000	32
13	16#609A:16#00	AX1 Homing aceleration : AX1_PU01	10000	32

Before controlling a motor using a motion control module, please configure the settings relating to protection and unit carefully. iR-PU01-P's LED may show error state when skipping these settings and directly using function blocks.

Parameter settings:

[iR-COP]->[SDOs]->[Add SDO]



The SDO settings will be written to iR-PU01-P after login.

- Motor Resolution Setting: 608Fh

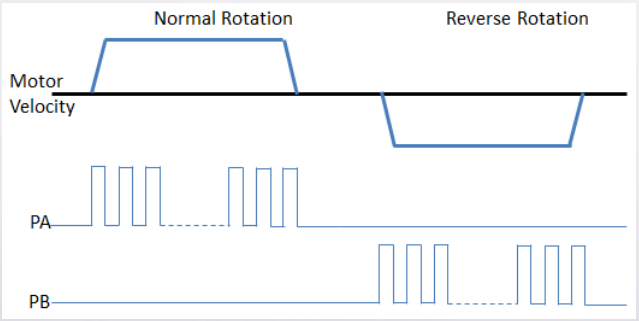
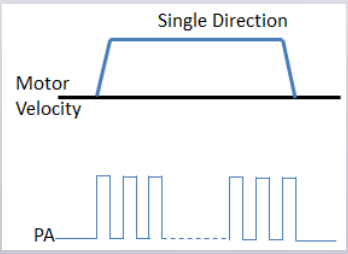
  Motor resolution: number of pulses per revolution. In the demonstration the values are set to 1.

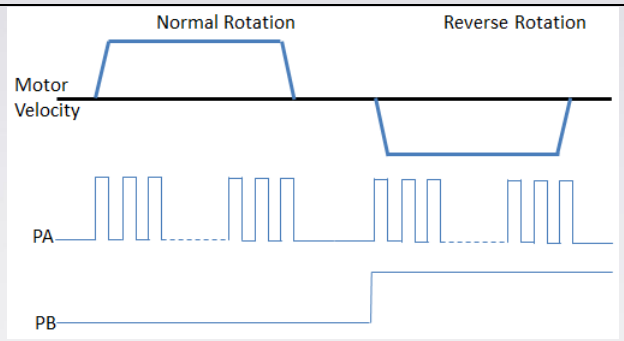
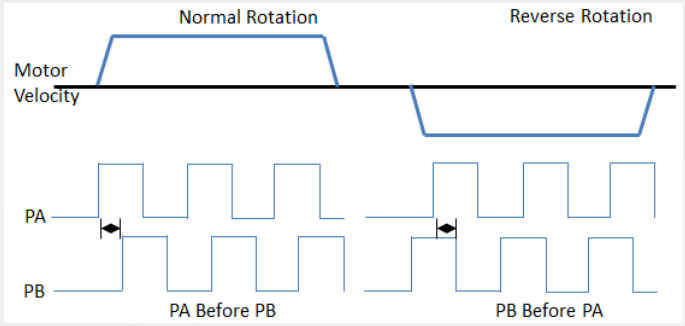
$$Position\ encoder\ resolution = \frac{encoder\ increments}{motor\ revolution}$$

- Pulse Output Method: 5511h

  Pulse output method is determined by the pulse type supported by the driver. The pulse type of both the driver and iR-PU01-P should be identical for the motor to rotate in the desired direction and distance.

  Sub Index 00h: Pulse Output Method

<table>
<tr><td>Bit7-
Bit 4</td><td colspan="4">Reserved</td></tr>
<tr><td rowspan="4">Bit3-
Bit 0</td><td>Value</td><td>PA</td><td>PB</td><td></td></tr>
<tr><td>0</td><td>Disable</td><td>Disable</td><td></td></tr>
<tr><td>1</td><td>CW</td><td>CCW</td><td> 
</td></tr>
<tr><td>2</td><td>Pulse</td><td>NC</td><td> 
</td></tr>
</table>

	3	Pulse	Direction	
	4	A	B	

- Max. Velocity: 6080h, 607Fh, 60C5h, 60C6h

  Max. Motor Speed: 6080h

  Enter the value according to the motor specification. Generally, the unit is RPM (Round Per Minute), but for this parameter, the unit is PPS (Pulse Per Second). Please convert the unit before entering the value.

  Max. Profile Velocity: 607Fh

  This is the maximum allowable velocity for the velocity profile. If 607Fh conflicts with 6080h, the lower value will be the maximum velocity.

  Max. Acceleration/Deceleration: 60C5h/60C6h

  When specifying a value greater than the value of 60C5h/60C6h, the value of 60C5h/60C6h will be the maximum acceleration/deceleration rate.

- Quick Stop Deceleration: 6085h

  When an error occurs during the runtime of iR-PU01-P, or when the limit sensor is encountered, this setting can decelerate the motor to stop at the specified deceleration rate.

- Homing Method: 6098h

  iR-PU01-P provides 37 homing methods. In this demonstration, homing method number 27 is used: Rapidly moves toward the negative limit until encountering Home sensor, and then slowly moves away from Home sensor in an opposite direction.

- Home Offset: 607Ch

  Use home offset to specify a start position after homing.

- Homing Speed: 6099h

  Use [Speed during search for switch] when start homing.

  Use [Speed during search for zero] after reaching the first reference point.

- Homing Acceleration: 609Ah

  Specify the acceleration rate for homing.

5. Function Blocks

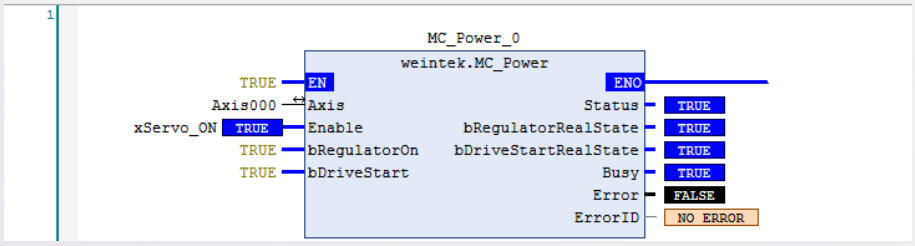
For more information on Weintek Library, please see this manual:

UM018017E_CODESYS_Weintek_Library_UserManual_20190305_eng

MC_Power

MC_Power must be executed before giving any motion instruction. When it is successfully executed and no error occurs, the axis enters Standstill state.

Expression	Type	Value	Prepared value	Address	Comment
Axis000	weintek.Axis_REF				
Mapping_Q	unAxis_Data_Out				
delay_cycles	BYTE	0			
cpty_v	BOOL	FALSE			
Mapping_I	unAxis_Data_In				
_MC_Status	EAXIS_STATE	Standstill			PLCopen State



As shown above, MC_Status is in Standstill state, which means the axis is ready for any motion instruction given to it.

MC_Home

Motion Function Block provides 37 homing methods which can be selected using MC_Home function block. Please see the following parameters:

6098: Homing method. (Use one of the 37 homing methods designed according to CiA402)

6099#1: Homing at low speed.

6099#2: Homing at high speed.

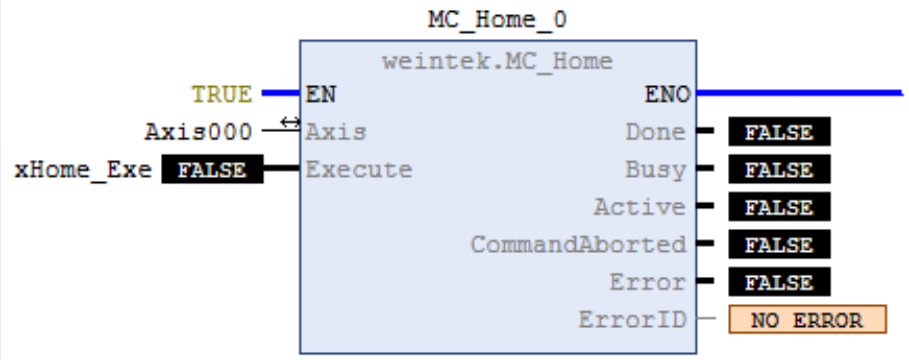
609A: Homing acceleration.

607C: Home offset.

Add SDO | Edit | Delete | Move Up | Move Down

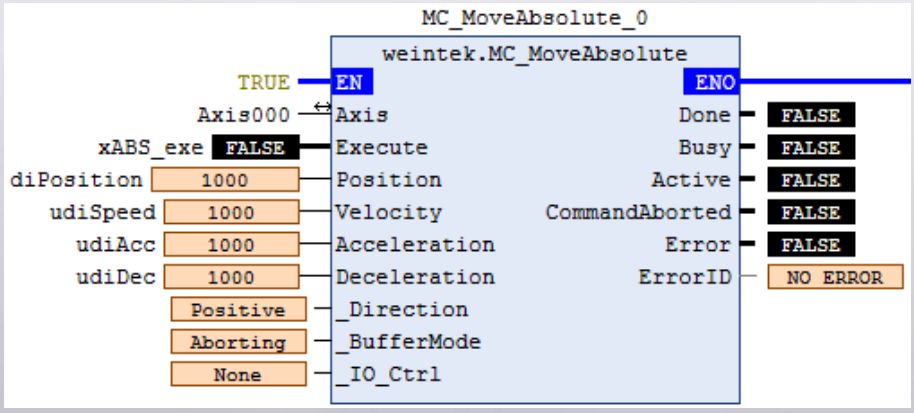
Line	Index:Subindex	Name	Value	Bit length
1	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
2	16#6099:16#01	Axis 1 Speed during search for switch : PU01_Axis_1	2000	32
3	16#6099:16#02	Axis 1 Speed during search for zero : PU01_Axis_1	10000	32
4	16#609A:16#00	Axis 1 Homing acceleration : PU01_Axis_1	10000	32
5	16#607C:16#00	Axis 1 Home offset : PU01_Axis_1	1000	32

Executing MC_Home when the axis is in Stanstill state performs homing using the parameters shown above. The axis changes to Homing state when this function block is executed, and returns to Standstill state after homing is completed.



MC_MoveAbsolute

The MC_MoveAbsolute function block moves the axis to a specified absolute target position. The following parameters are used when executing MC_MoveAbsolute.



- Position: Specify the absolute target position.
- Velocity: Specify the target velocity, the value cannot be 0.
- Acceleration/Deceleration: Specify the acceleration / deceleration rate,

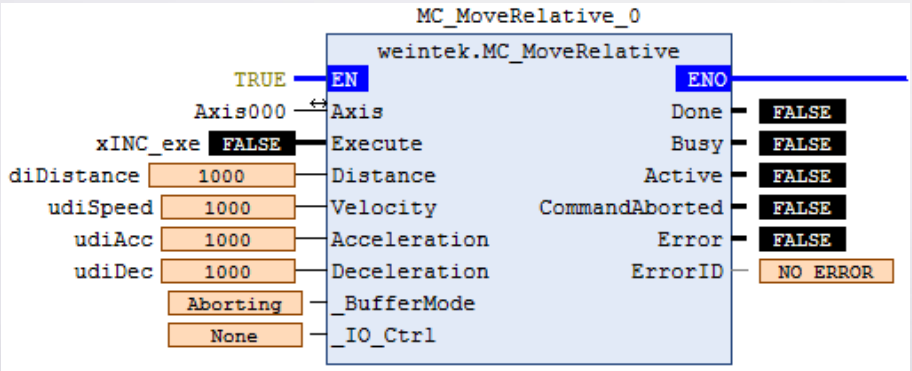
the value cannot be 0.

- Direction: Specify the direction and the shortest path.
- BufferMode: Continuously executes the next instruction after the ongoing motion is completed.
- IO_Ctrl: Trigger execution using digital input, and then output digital signal after the motion is completed.

After executing this function block, the axis enters Discrete Motion state, and returns to Standstill state after positioning is completed.

MC_MoveRelative

The MC_MoveRelative function block performs positioning for a specified travel distance from the current position. The following parameters are used when executing MC_MoveRelative.



- Position: Specify the target position, which equals to current position + specified distance.
- Velocity: Specify the target velocity, the value cannot be 0.
- Acceleration/Deceleration: Specify the acceleration / deceleration rate, the value cannot be 0.
- BufferMode: Continuously executes the next instruction after the ongoing motion is completed.

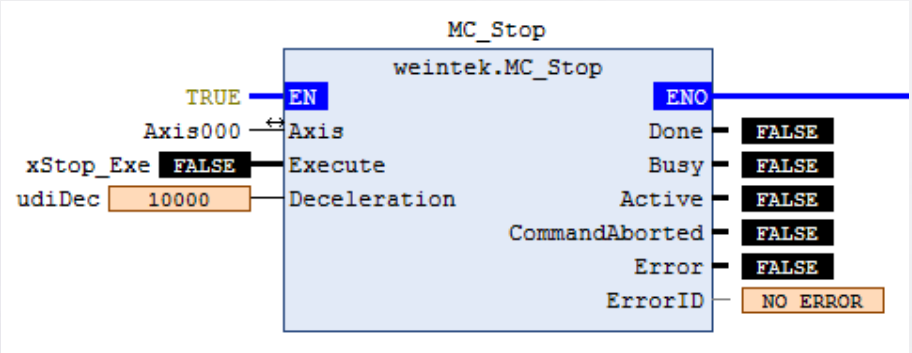
- IO_Ctrl: Trigger execution using digital input, and then output digital signal after the motion is completed.

After executing this function block, the axis enters Discrete Motion state, and returns to Standstill state after positioning is completed.

MC_STOP

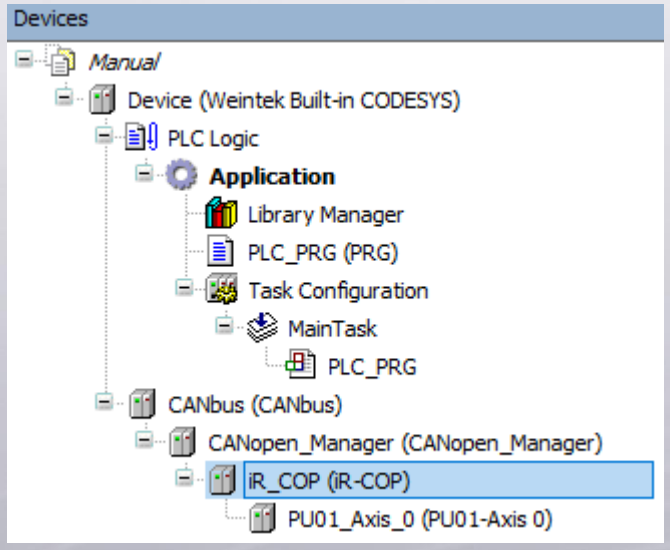
MC_STOP can stop axis operation. When using MC_STOP, it decelerates the axis to stop, and instructions can only be given after the axis stops.

The following parameters are used when executing MC_STOP.



- Deceleration: Specify the deceleration rate, the value cannot be 0.

The axis enters Standstill state after it stops.



6. Demo Project Settings

Absolute/Relative Positioning

xEnable_Power: Start the system.

xServe_ON: Turn on server.

xHome_exe: TRUE = Executing Homing.

xABS_exe: TRUE = Execute absolute positioning.

xINC_exe: TRUE = Execute relative positioning.

xTrig_Stop: TRUE= Stop during motion.

xTrig_Reset: TRUE = Reset iR-PU01-P.

diPosition: Specify an absolute target position.

diDistance: Specify distance.

udiSpeed: Specify speed.

udiAcc: Specify acceleration rate.

udiDec: Specify deceleration rate.

eBuffer_Mode： Buffer the next motion instruction and continuously executes the buffered instruction after the ongoing motion is completed.

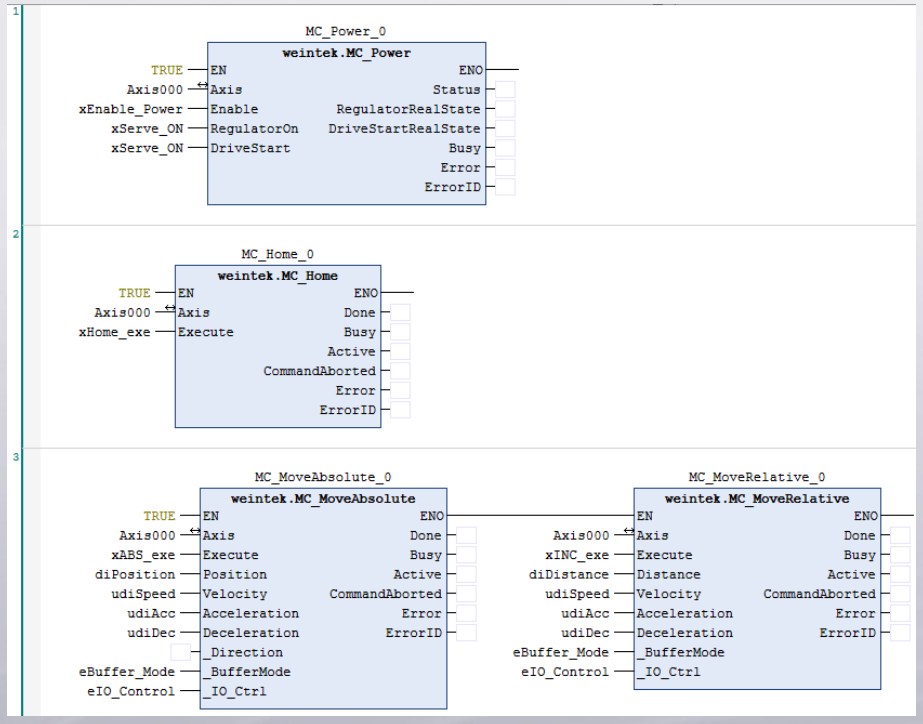
eIO_Control: Trigger motion control using iR-PU01-P's built-in digital input.

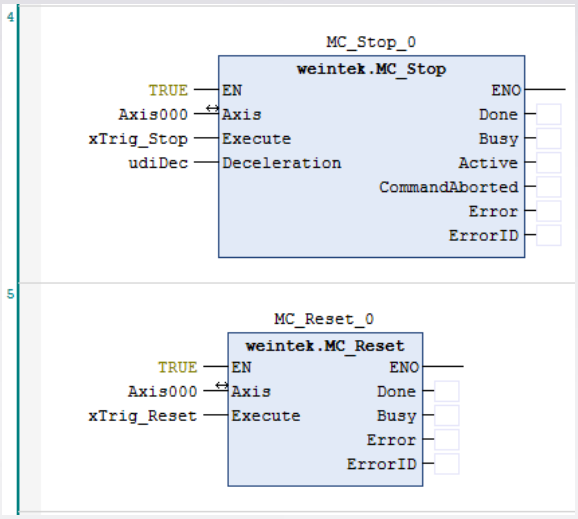
Declaration

```
VAR
    // Axis reference
    Axis000 : Weintek.Axis_REF_Lite ;
    // Motion Control Function Block
    MC_Power_0: weintek.MC_Power ;
    MC_MoveAbsolute_0 : weintek.MC_MoveAbsolute ;
    MC_MoveRelative_0 : weintek.MC_MoveRelative ;
    MC_Home_0 : weintek.MC_Home ;
    MC_Stop_0: weintek.MC_Stop;
    MC_Reset_0: weintek.MC_Reset;
    // Positioning control Button
    xEnable_Power, xServe_ON, xABS_exe, xINC_exe, xHome_exe : BOOL ;
    xTrig_Stop, xTrig_Reset : BOOL;
    // Positioning parameter
    diPosition : DINT := 10000 ;
    diDistance : DINT := 10000 ;
    udiSpeed : UDINT := 1000 ;
    udiAcc : UDINT := 1000 ;
    udiDec : UDINT := 1000 ;
    eIO_Control : weintek.eMC_IO_CTRL ;
    eBuffer_Mode : weintek.eMC_BUFF_MODE ;
    // Variable
END_VAR
```

Declare necessary variables and give initial value.

FBD





1: Starting motion control system: xEnable_Power & xServe_ON must be TRUE and no error occurs.

2: Homing function block.

3: Absolut/Relative positioning function block.

4: Stop function block, the current motion will be stopped when triggering xTrig_Stop during motion.

5: When an error occurs during motion, triggering xTrig_Reset can reset iR-PU01-P.

7. Login and Operate

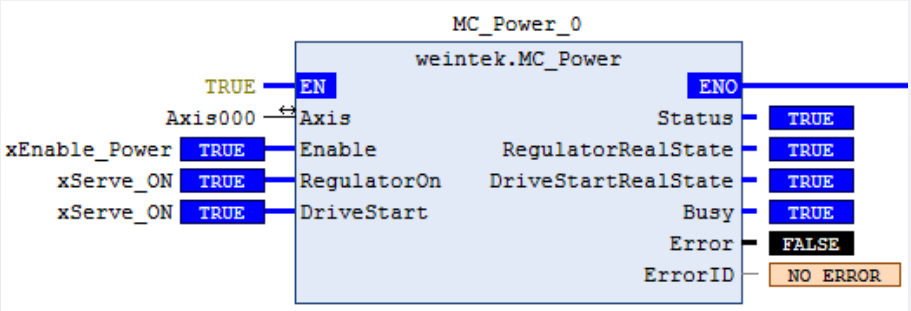
The following demonstrates how the project works.

Start Motion Control System

xEnable_Power & xServe_ON = TRUE, start controlling the axis.

Status, RegulatorRealState, DriveStartRealState, Busy = TRUE: No error.

Error = TRUE: An error occurs.



Homing

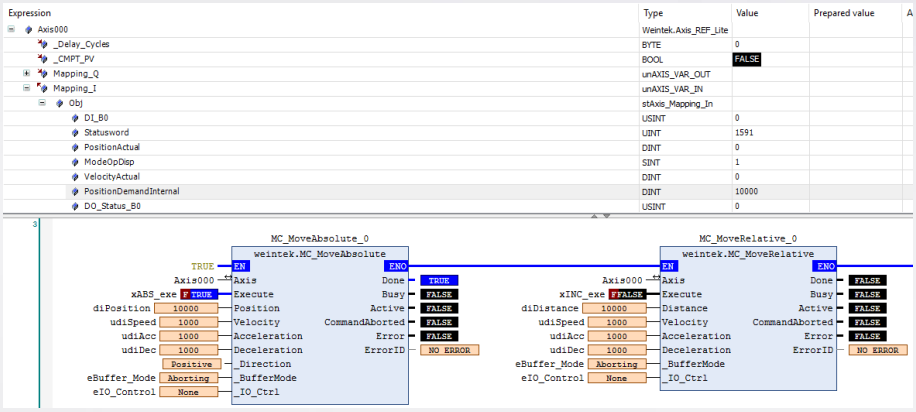
Triggering xHome_exe can start homing using the following settings.

9	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
10	16#607C:16#00	AX1 Home offset	0	32
11	16#6099:16#01	Speed during search for switch : AX1_PU01	10000	32
12	16#6099:16#02	Speed during search for zero : AX1_PU01	2000	32
13	16#609A:16#00	AX1 Homing aceleration : AX1_PU01	10000	32

The axis moves to the negative limit at the speed specified in [Speed during search for switch]. After it encounters the homing sensor, it moves at the speed specified in [Speed during search for zero] to the positive limit and away from the homing sensor. After it leaves the homing sensor, homing is completed, and Home Offset specifies a start position.

Absolute Positioning

Triggering xABS_exe can start absolute positioning. In this demo project, the absolute position is specified as 10000.

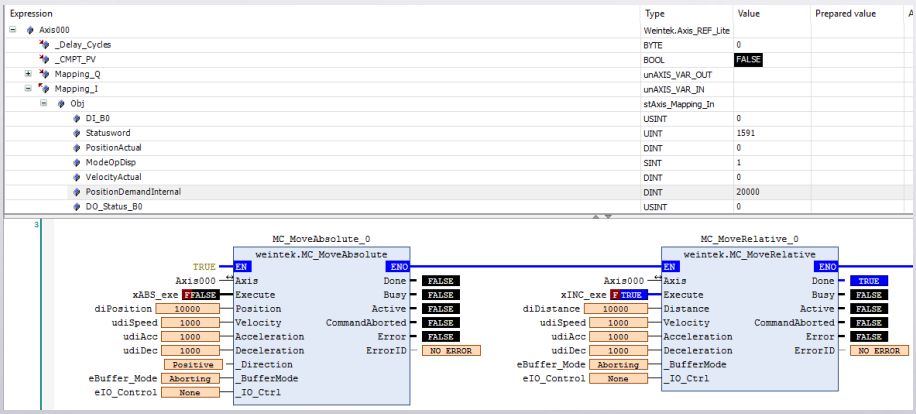


After absolute positioning is completed (MC_Move_Absolut_0.Done=TRUE), the motor is positioned as specified. The current position can be found in Axis000.Mapping_I » Obj » PositionDemandInternal.

Relative Positioning

Triggering xINC_exe can start relative positioning.

The new position is current position (10000) + diDistance (10000) = 20000



After relative positioning is completed (MC_MoveRelative_0.Done=TRUE), the motor is positioned as specified. The current position can be found in Axis000.Mapping_I » Obj » PositionDemandInternal.