User Manual

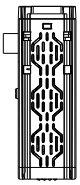
iR-PU01-P

This guide walks through important information about iR-PU01-P.

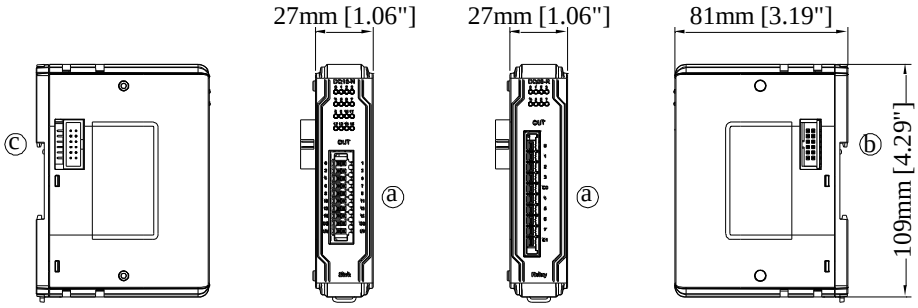
V1.00

Table of Contents

1. Product Overview



Top View



Side View

iR-DXXX
iR-PXXX

iR-DQ08-R

Side View

Front View

Bottom View

a	Terminal	*b.c*	Expansion Connector

2. Specifications

2.1 Module Specification

Module Name		iR-PU01-P
Number of Axis		1- Axis
Specification	PCB Coating	Yes
	Enclosure	Plastic
	Dimensions WxHxD	27 x 109 x 81 mm
	Weight	Approx. 0.12 kg
	Mount	35mm DIN rail mounting
Environment	Protection Structure	IP20
	Storage Temperature	-20° ~ 70°C (-4° ~ 158°F)
	Operating Temperature	0° ~ 55°C (32° ~ 131°F)
	Relative Humidity	10% ~ 90% (non-condensing)
Connection	Cross-section	AWG 28-16
Certification	EMC Immunity	Conforms to EN 55032: 2012+AC: 2013, Class A EN 61000-6-4: 2007+A1:2011 EN 55024: 2010+A1: 2015 EN 61000-6-2:2005

2.2 Digital Input Specification

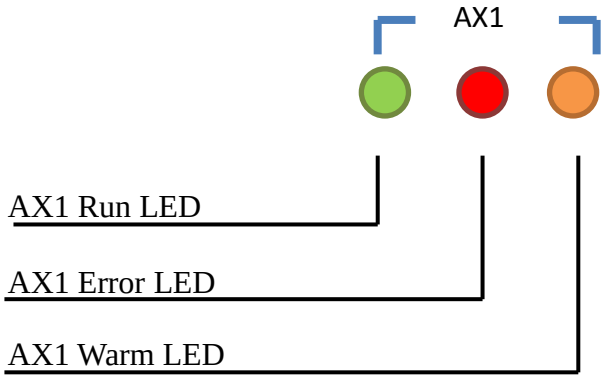
Item	Sink Input	Differential Input
Number of Inputs	4	3 (A/B/Z phase)
Input current	24 VDC, 5 mA	Meets the Requirements of ANSI Standards TIA/EIA-485-A
HIGH Level Input Voltage	15~28 VDC	-
LOW Level Input Voltage	0~5 VDC	-
Maximum input frequency	200KHz	2MHz
Input Impedance	3 KΩ	-
Indicators	Red LED Input State	

2.3 Digital Output Specification

Item	Source Output	Differential Output
Number of Outputs	4	2(A/B phase)
Output Voltage	24VDC , 50 mA	Meets the Requirements of ANSI Standards TIA/EIA-485-A
Maximum Output frequency	40KHz	2MHz
Indicators	Red LED Input State	

3. LED Indicators

3.1 AX1 LED



3.2 Run/Error/Warn LED

Run LED	Description
OFF	Axis is not ready
Blinking	Axis is ready
ON	Axis is busy
Error LED	Description
OFF	No errors
ON	Error occurred
Warn LED	Description
OFF	No warnings
ON	Warning: Unable to reach the specified velocity trajectory.

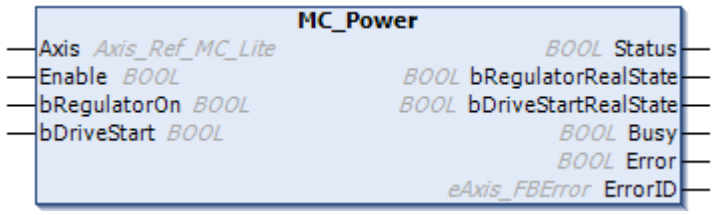
3.3 I/O LED

IN 0-3 State	Description
OFF	Digital Input OFF
ON	Digital Input ON
OUT 0-3 State	Description
OFF	Digital Output Set OFF
ON	Digital Output set ON
PA/PB State	Description
OFF	PA/PB Pulse Output OFF
ON	PA/PB Pulse Output ON
A/B/Z State	Description
OFF	A/B/Z Pulse Input OFF
ON	A/B/Z Pulse Input ON

4. Error Handling

4.1 Function Block Error

When using a function block and an error occurs, the diagnostic value is output to the pin in the function block, and ErrorID contains the error code. The following is a list showing how to handle errors.



State	Description	Error Handling
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.
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.
AXIS_MOTION_ERROR	A motion error occurs.	Please see chapter 4.3 in this manual.
AXIS_HOMING_ERROR	A homing error occurs.	Please check the homing related settings and see Chapter 4.3 in this manual.
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.

4.2 Warning

Warnings occur when:

Warn LED is on

Bit 7 in Digital Input Byte0 is 1

Digital Input Byte0		
Axis Number	Index	Sub-index
Axis 0	5500h	01h
Axis 1	5600h	01h
Axis 2	5700h	01h
Axis 3	5800h	01h

In positioning control, a warning occurs when:

- The specified acceleration/deceleration rate cannot be reached before reaching the target velocity due to jerk limitation.
- In the distance for positioning, acceleration/deceleration takes a long time so that target velocity cannot be reached.

When warning, PU will specify a lower target velocity, and remove jerk limitation and finish positioning. To keep the jerk limitation, user may adjust target velocity, acceleration/deceleration rate to avoid warning.

4.3 Error

Errors are indicated in the following ways:

Error LED is on

ErrorID of Function Block is AXIS_MOTION_ERROR

The status of the axis is ErroStop

When an error occurs, find the error code and troubleshoot the error. Use MC_Reset to clear the error, and then use MC_Power to make the axis return to Standstill state.

Error Code	Description	Cause of Error
16#6180	Motion Error 0	MC_POWER is OFF during motion. (Disconnected, or PLC Stop/Reset).
16#6181	Motion Error 1	Changes to an incorrect mode (CiA402) during motion.
16#6182	Motion Error 2	iR-PU01-P calculates trajectory incorrectly. (Including errors caused by Blending.)
16#618A	Homing Error	Incorrect Homing mode or an external signal that is not configured is used.
16#6280	Software Limitation	The position is going to exceed or already exceeds the software limitation.
16#6281	Prohibited Direction	Movement in prohibited direction.
16#8612	Exceeding Position Range	Target position exceeds software limitation or axis range.
16#6320	Function Block Error	Invalid parameters used.
16#6380	Parameter Error 0	Incorrect pulse output mode used.
16#6381	Parameter Error 1	Incorrect pulse output mode used.
16#6382	Parameter Error 2	The product of the numerator and denominator of the ratio is too large. (INT_MAX)
16#6383	Parameter Error 3	The product of the numerator and denominator of the ratio of the 1st axis is too large.
16#6384	Parameter Error 4	The product of the numerator and denominator of the ratio of the 2nd axis is too large.
16#9080	External Signal Error 0	Positive limit signal is triggered.
16#9081	External Signal Error 1	Negative limit signal is triggered.
16#9082	External Signal Error 2	Immediate stop signal is triggered.
16#7500	Communication Error	Disconnection or Heartbeat Timeout has occurred.

5. Wiring

5.1 Notes on Wiring

- Wiring for Differential Communication
  a. Wire length should be minimized (Max: 500m shielded, 300m unshielded).
  b. Please use twisted pair cables conform to the impedance matching.
  c. If wiring is to be exposed to lightning or surges, use appropriate surge suppression devices.
  d. Keep AC wiring separated from signal wires.
  e. Keep high energy and rapidly switching DC power wiring separated from signal wires.

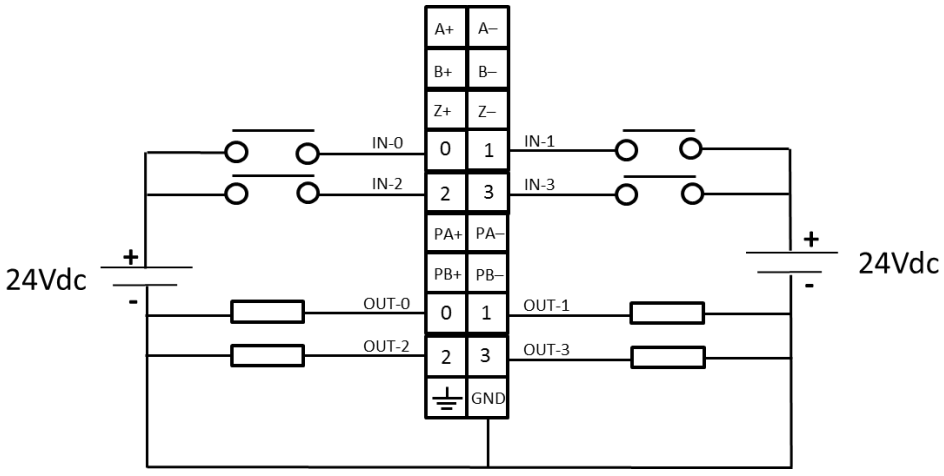
- Wiring for Digital Output

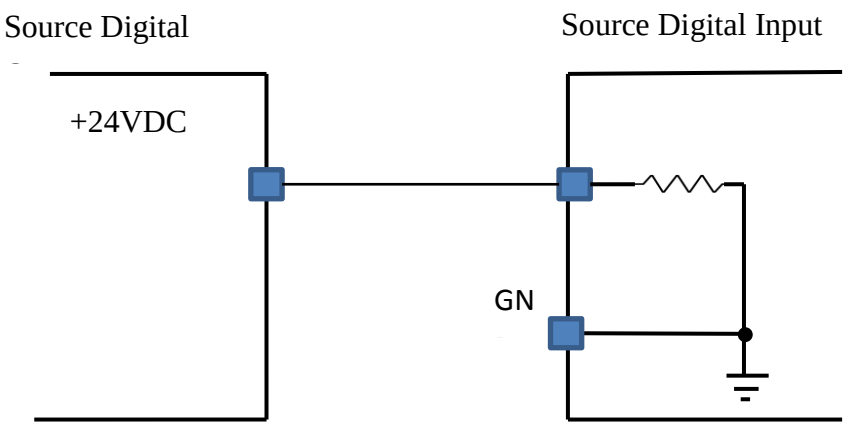
Digital output voltage range: 24VDC (-15%/+20%)
The maximum output voltage per point is 50mA, please take this into consideration when wiring.

- Wiring for Digital Input

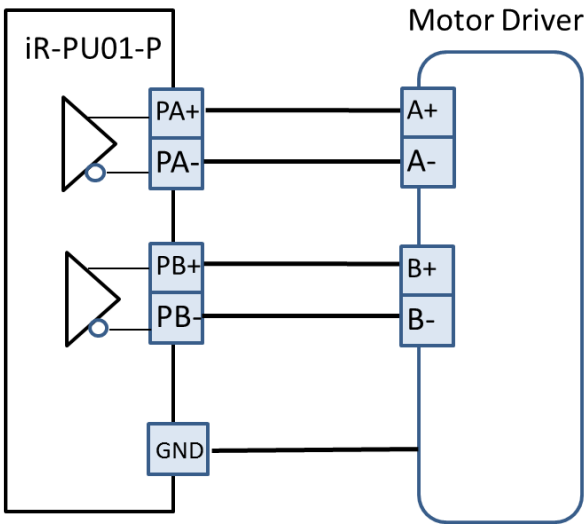
Digital input voltage range: 15~28VDC (ON), 5V (OFF)
Input impedance: 3 KΩ
Please take this into consideration when wiring.

5.2 Digital Input / Output Wiring

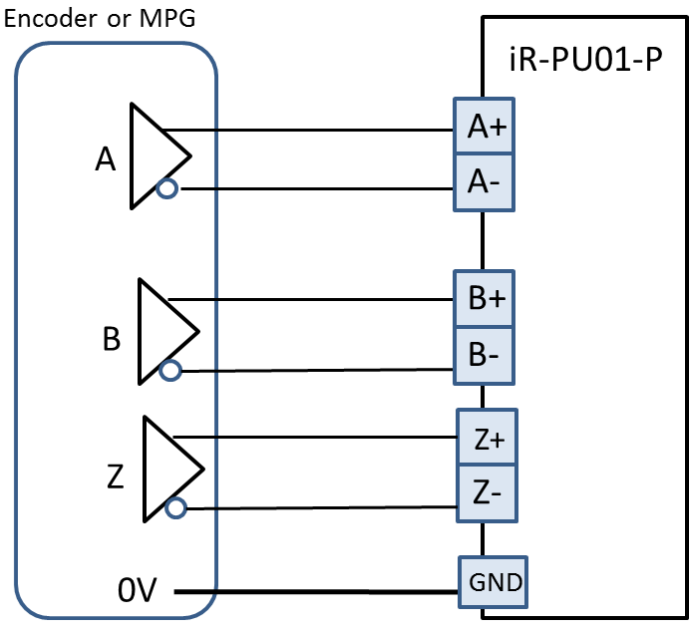




5.3 Differential Output Wiring



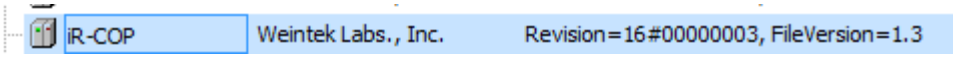
5.4 Differential Input Wiring



6. Connecting a Coupler

6.1 iR-COP

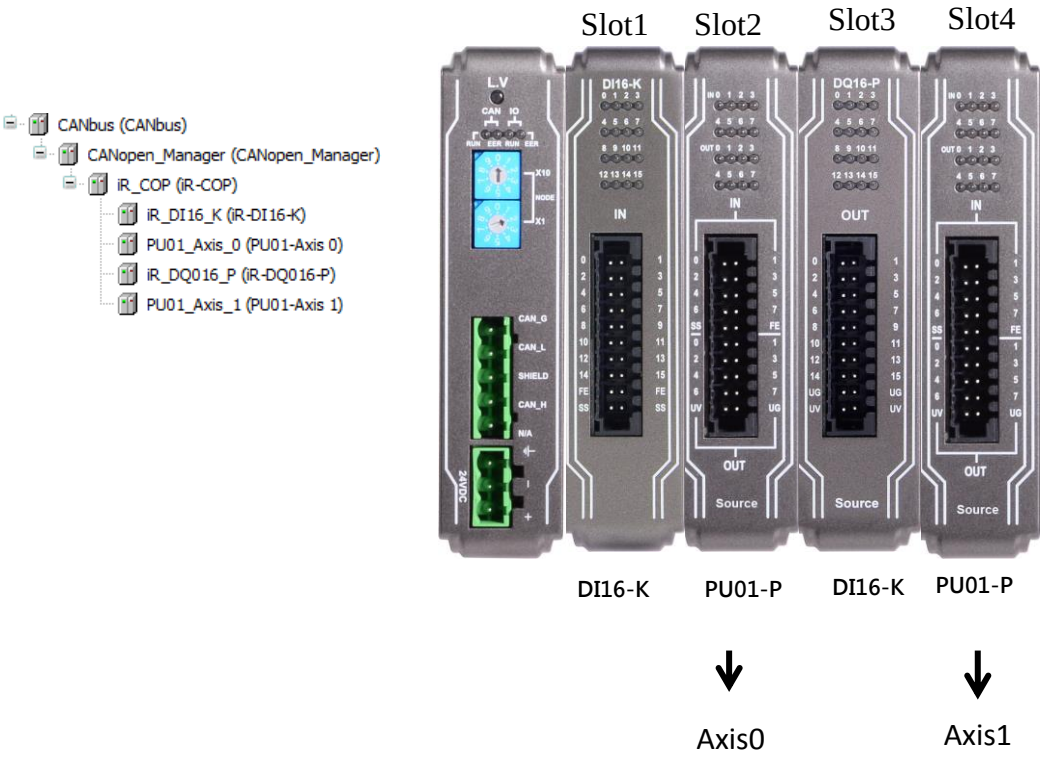
- An iR-COP coupler supports up to 4 iR-PU01-P modules at a time.
- iR-COP software version should be 1.00.3 or later.
- EDS file version should be Revision 16#00000003



6.2 Slot and Axis

An iR-COP coupler supports up to 4 iR-PU01-P modules at a time. The 4 iR-PU01-P modules use iR-COP's axes respectively, which are Axis 0~3. The iR-PU01-P module nearest to iR-COP uses Axis 0, and the second uses Axes 1, and so on.

As shown in the following figure, two iR-PU01-P modules are installed respectively on Slot 2 and Slot 4. The iR-PU01-P module installed on Slot 2 uses the first axis (Axis 0), while the iR-PU01-P installed on Slow 4 uses the second axis (Axis 1).



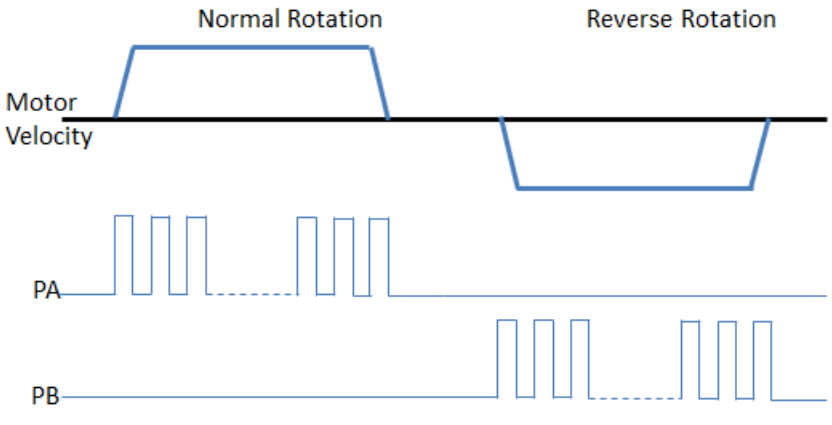
7. Features

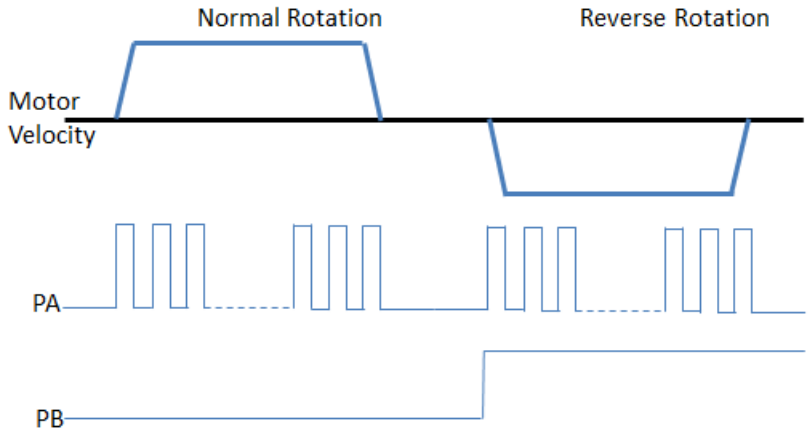
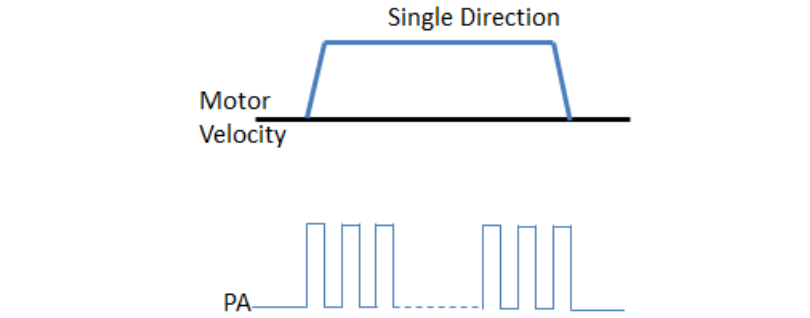
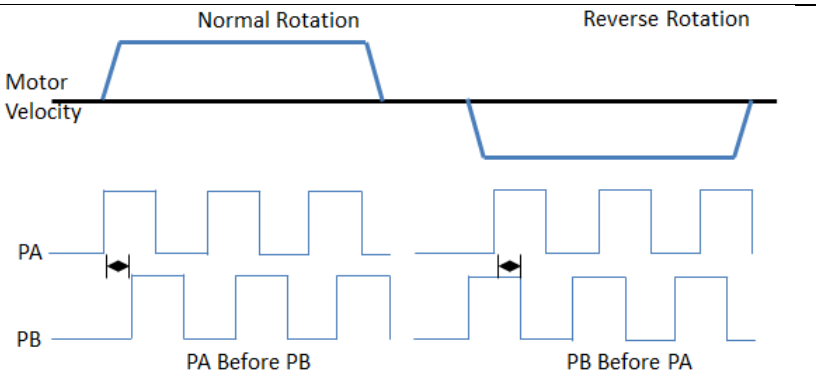
7.1 Feature List

No.	Feature
1	High Speed Pulse Output
2	High Speed Pulse Input (encoder)
3	Positioning Control (Buffer Mode supported).
4	Velocity Control allows speed specification.
5	Homing, supports over 30 modes.
6	Synchronized Motion (Gear/MPG)
7	Digital Cam Switch
8	Capture
9	Configurable I/O

7.2 High Speed Pulse Output

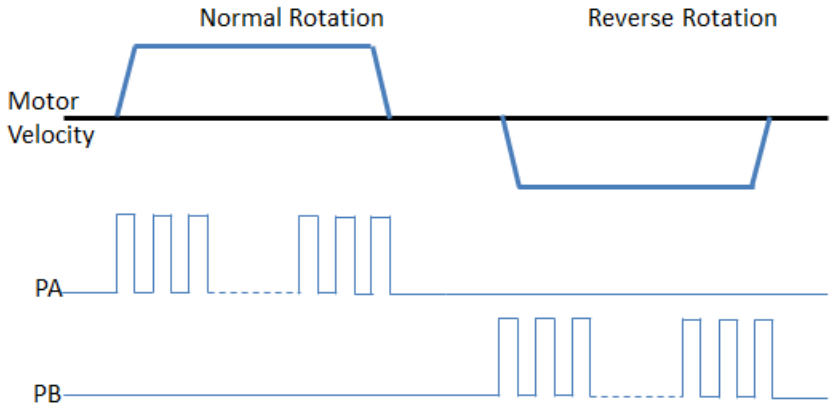
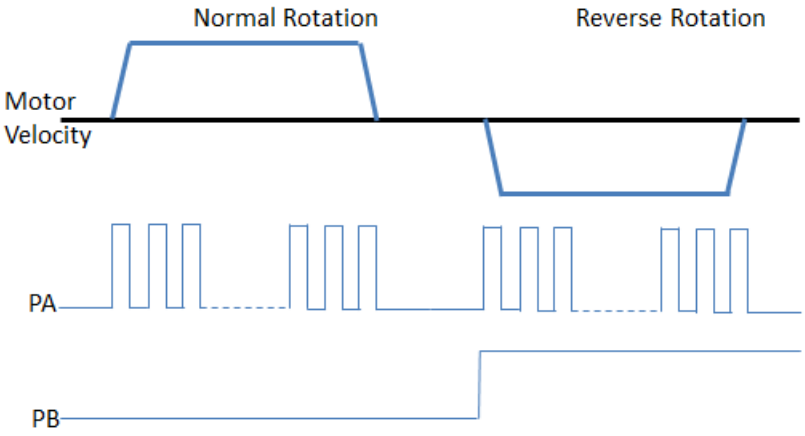
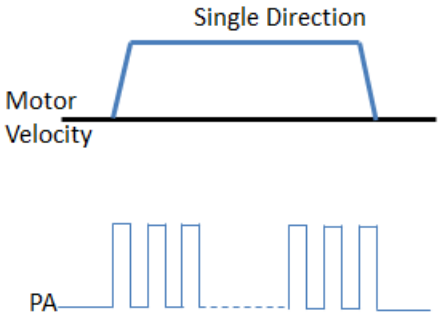
iR-PU01-P can output 2MHz pulses to control the connected servo/step motor (velocity and positioning control). Output modes include CW/CCW, Pulse/Direction, Pulse Only, A/B phase * 1、A/B phase * 2、A/B phase * 4. The output mode is configured using Object Dictionary-Index 0x5511 (Axis 0).

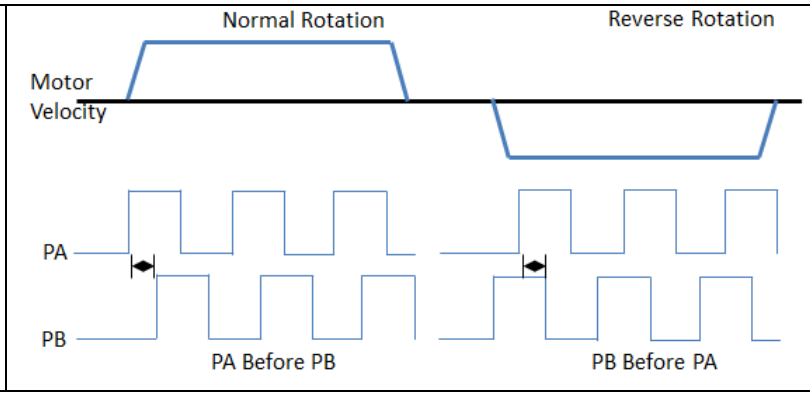
Output Modes	Schematic
CW/CCW	

Pulse/Direction	
Pulse only	
A/B Phase	

7.3 High Speed Pulse Input (Encoder)

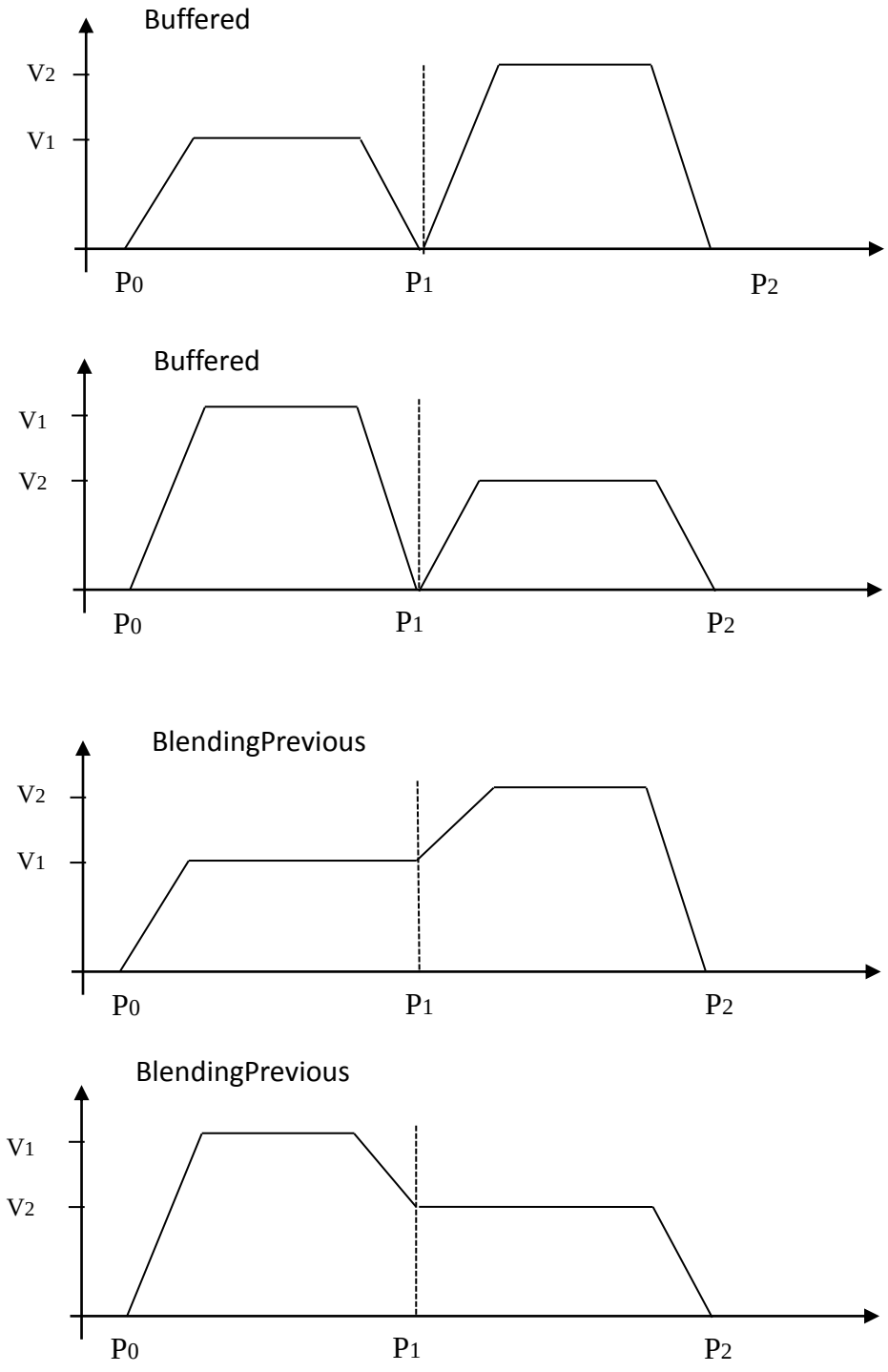
iR-PU01-P can receive up to 2MHz pulse input from the signal output by an encoder or Manual Pulse Generator (MPG). Input modes include CW/CCW, Pulse/Direction, Pulse Only, A/B phase * 1、A/B phase * 2、A/B phase * 4. The input mode is configured using Object Dictionary-Index 0x5501 (Axis 0).

Input Modes	Schematic
CW/CCW	
Pulse/Direction	
Pulse only	

A/B Phase	

7.4 Positioning Control (Buffer Mode Supported)

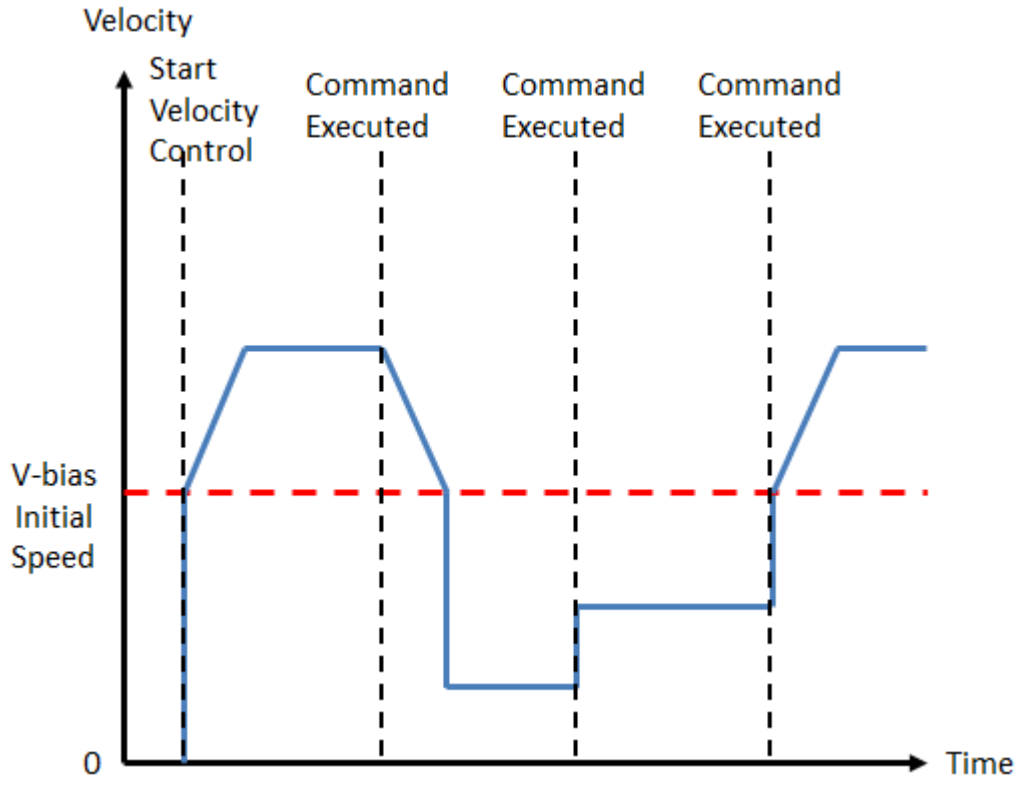
Weintek provides a library of motion control function blocks, and the function blocks relating to positioning control include: MC_MoveAbsolute and MC_MoveRelative, which can perform absolute/relative positioning for the specified target position or for the specified travel distance from current position. Buffer Mode can be used when executing more than one motion instruction.



7.5 Velocity Control

Weintek provides MC_MoveVelocity function block for controlling the speed of the motor. The module allows V-bias initial speed specification, which makes the motor rotate at specified velocity immediately regardless of acceleration/deceleration rate

when receiving a velocity command for a velocity slower than the initial speed. This can reduce resonance at low speeds. The following diagram shows how velocity changes when V-bias is used.



7.6 Homing

The module provides 37 homing modes designed according to CiA402. The homing modes include: proximity sensors, index, positive and negative limits.

7.7 Synchronized Motion (Gear / MPG)

Synchronized motion control is achieved by specifying the ratio between the master axis (Pulse Input) and the slave axis (Pulse Output). Manual Pulse Generator (MPG) is one of the applications.

7.8 Digital Cam Switch

This feature simulates mechanical cam switch using digital method, which allows settings that can by realized easier in this manner. Each track corresponds to one iR-PU01-P output point. Users can add multiple switches (16 in maximum) to a track, with each switch specifies different position and direction, in order to plan the distance and time output by a point. The position source can be a commanded position (1st axis) or an actual position (2nd axis), and ratio can be configured respectively for each axis.

7.9 Capture

Each iR-PU01-P has five Capture channels for capturing the current axis position value or the timer value of iR-PU01-P, on the rising or falling edge of the input signal. The change of position or time interval can be observed by comparing two captured values (two values captured by one channel, or compared to the latest value from other channels).

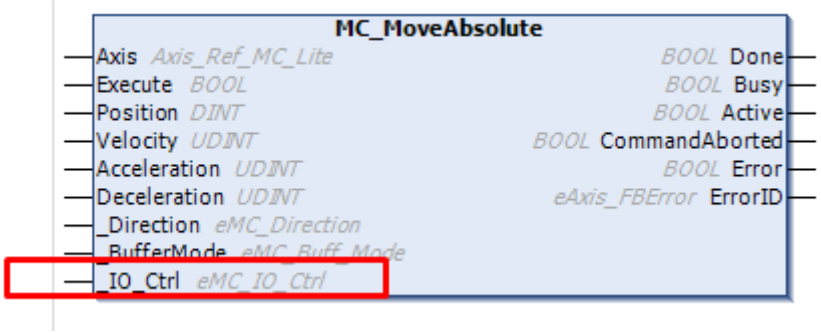
Continuous capture is also possible, to do so, the interval of the external signals should be greater than 1ms. Please take the execution cycle and communication cycle into consideration in order to read the capture value before the next capture takes place.

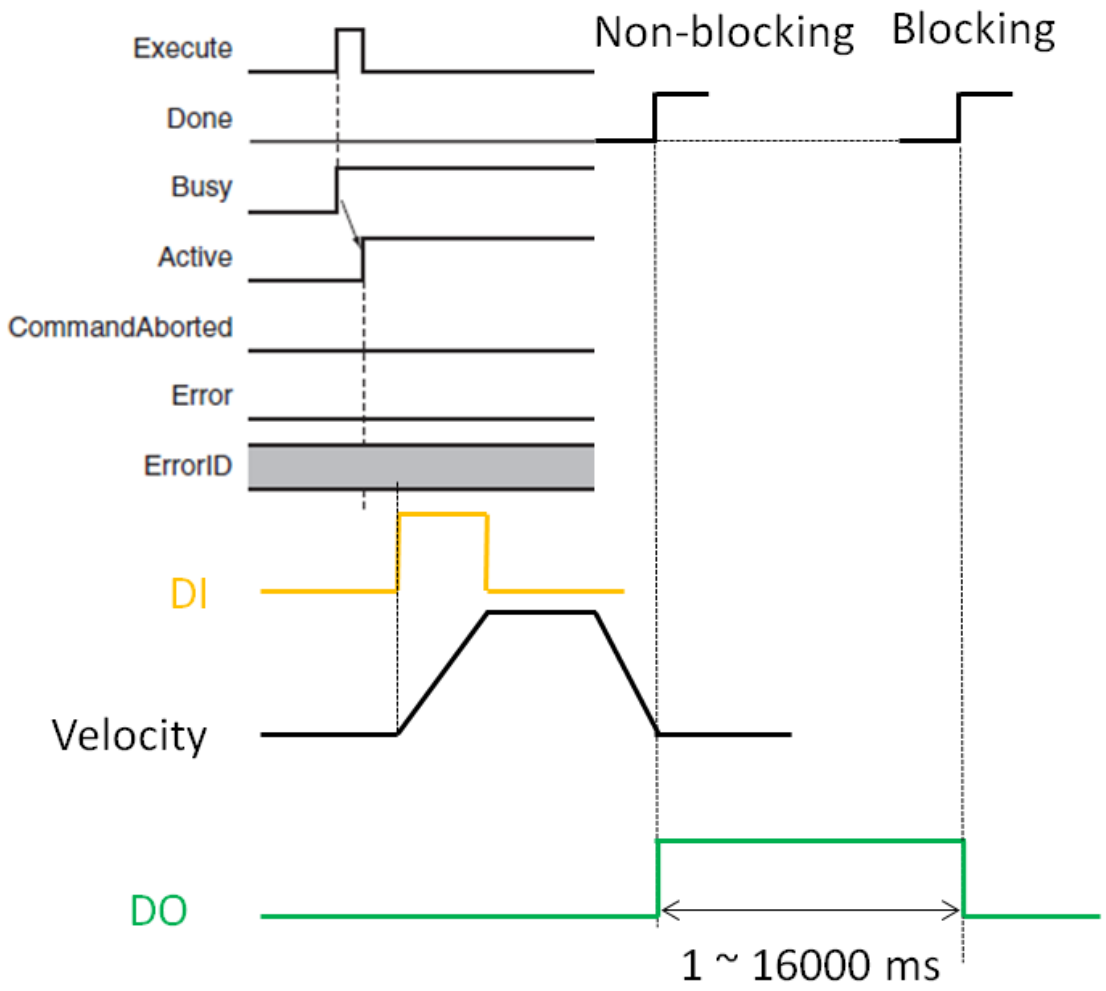
7.10 Configurable I/O

Each I/O of the module, including pulse input/output, has its unique function designed for motion control, but using them as general digital I/O points is possible.

7.11 Motion Control working with I/O Control

Apart from homing, motion control can be triggered by using an external input signal. After positioning is completed, IO_Ctrl Function Block and Object Dictionary can be used to assign multiple output points to output values at a time.





8. Object Dictionary

Data Type	Lower Limit	Upper Limit	Memory
SINT	-128	127	8bit
USINT	0	255	8bit
INT	-32768	32767	16bit
UINT	0	65535	16bit
DINT	-2147483648	2147483647	32bit
UDINT	0	4294967295	32bit

8.1 Manufacturer Specific Profile Area (5500h - 58FFh)

Item	Index Range
Axis 0(1st PU)	5500-55FF
Axis 1(2nd PU)	5600-56FF
Axis 2(3rd PU)	5700-57FF
Axis 3(4th PU)	5800-58FF

In the following list, n=0~3 which represents Axis 0~3.

Index	Sub Index	Description	Type	ro/rw	Default
5500h+n*100h		Digital Input			
	01h	DI byte 0	USINT	ro	----
5501h+ n*100h	00h	Pulse Input Method	USINT	rw	00h
5502h+ n*100h	00h	Input Polarity	UDINT	rw	00h
5503h+ n*100h		Digital Input Function			
	01h	DI 0 Function	USINT	rw	1h
	02h	DI 1 Function	USINT	rw	1h
	03h	DI 2 Function	USINT	rw	1h
	04h	DI 3 Function	USINT	rw	1h
	05h	DI A Function	USINT	rw	0h
	06h	DI B Function	USINT	rw	0h
	07h	DI Z Function	USINT	rw	1h
5504h+ n*100h		Digital Input Filter			
	01h	DI 0 Filter	USINT	rw	03h
	02h	DI 1 Filter	USINT	rw	03h
	03h	DI 2 Filter	USINT	rw	03h
	04h	DI 3 Filter	USINT	rw	03h
	05h	DI A Filter	USINT	rw	02h
	06h	DI B Filter	USINT	rw	02h
	07h	DI Z Filter	USINT	rw	02h
5510h+n*100h		Digital Output		rw	
	01h	DO byte 0	USINT	rw	0h
	02h	DO status byte 0	USINT	rw	0h
5511h+n*100h	00h	Pulse Output Method	USINT	rw	
5512h+n*100h	00h	Output Polarity	UDINT	rw	00h
5513h+n*100h		Digital Output Function			
	01h	DO 0 Function	USINT	rw	0h
	02h	DO 1 Function	USINT	rw	0h
	03h	DO 2 Function	USINT	rw	0h
	04h	DO 3 Function	USINT	rw	0h
	05h	DO A Function	USINT	rw	0h
	06h	DO B Function	USINT	rw	0h
5514h+n*100h		Digital Output Abort Connection Option			
	01h	DO 0 Option	USINT	rw	0h

	02h	DO 1 Option	USINT	rw	0h
	03h	DO 2 Option	USINT	rw	0h
	04h	DO 3 Option	USINT	rw	0h
	05h	DO A Option	USINT	rw	0h
	06h	DO B Option	USINT	rw	0h
551Ah+n*100h		PWM output setting			
	01h	PWM Output D0 setting	UDINT	rw	0h
	02h	PWM Output D1/PB setting	UDINT	rw	0h
5520h+n*100h		Axis Settings0			
	01h	Motion Cycle Time	UDINT	rw	0h
	02h	Bias Velocity	UDINT	rw	0h
5521h+n*100h		Axis Settings1			
	01h	Backlash compensation(pulse)	UINT	rw	0h
5528h+n*100h		Additional position modulo range			
	01h	1st additional position modulo range	DINT	rw	0h
	02h	2nd additional position modulo range	DINT	rw	0h
5529h+n*100h		Additional home offset			
	01h	1st additional home offset	DINT	rw	0h
	02h	2nd additional home offset	DINT	rw	0h
5530h+n*100h		Gear Motion Settings			
	01h	Master Axis Direction Limit	USINT	rw	0h
	02h	Slave Axis(PU) Direction Limit	USINT	rw	0h
	03h	Simple Moving Average Size	USINT	rw	0h
	04h	Following error window	UDINT	rw	FFFFh
	05h	Following error time out	UINT	rw	3000
5580h+n*100h		DigitalCamSwitch			
	01h	DigitalCamSwitch Enable	USINT	rw	0h
	02h	EnableMask Track 0-5	USINT	rw	0h
	03h	Valid Track 0-5	USINT	ro	0h
5581h+n*100h		DigitalCamSwitch Track Reference Source			
	01h	Track D0 Source	USINT	rw	0h
	02h	Track D1 Source	USINT	rw	0h
	03h	Track D2 Source	USINT	rw	0h
	04h	Track D3 Source	USINT	rw	0h
	05h	Track PA Source	USINT	rw	0h

	06h	Track PB Source	USINT	rw	0h
5583h+n*100h	(m=0~15)	DigitalCamSwitch MC_CAMSWITCH_REF			
	6*m+1	Switch m TrackNumber	USINT	rw	FFh
	6*m+2	Switch m FirstOnPosition	DINT	rw	0h
	6*m+3	Switch m LastOnPosition	DINT	rw	0h
	6*m+4	Switch m AxisDirection	USINT	rw	0h
	6*m+5	Switch m CamSwitchMode	USINT	rw	0h
	6*m+6	Switch m Duration(ms)	UINT	rw	0h
558Fh+n*100h		Motion Output Settings			
	01h	Motion Output Setting 0	UDINT	rw	0h
	02h	Motion Output Setting 1	UDINT	rw	0h
	02h	Motion Output Setting 2	UDINT	rw	0h
5590h+n*100h		Capture Enable			
	01h	Capture Enable Byte 0	USINT	rw	0h
5591h+n*100h		Capture Status			
	01h	Capture Status Byte 0	USINT	ro	0h
5592h+n*100h		Capture Settings			
	01h	Capture Setting Channel 0	UDINT	rw	0h
	02h	Capture Setting Channel 1	UDINT	rw	0h
	03h	Capture Setting Channel 2	UDINT	rw	0h
	04h	Capture Setting Channel 3	UDINT	rw	0h
	05h	Capture Setting Channel 4	UDINT	rw	0h
5598h+n*100h		Capture Value			
	01h	Capture Value 0	DINT	ro	0h
	02h	Capture Value 1	DINT	ro	0h
	03h	Capture Value 2	DINT	ro	0h
	04h	Capture Value 3	DINT	ro	0h
	05h	Capture Value 4	DINT	ro	0h
559Fh+n*100h		Motion Trigger Settings			
	01h	Motion Trigger Setting 0	UINT	rw	0h
	02h	Motion Trigger Setting 1	UINT	rw	0h
	03h	Motion Trigger Setting 2	UINT	rw	0h

8.1.1 Digital Input：5500h

Sub Index 01h: Input State

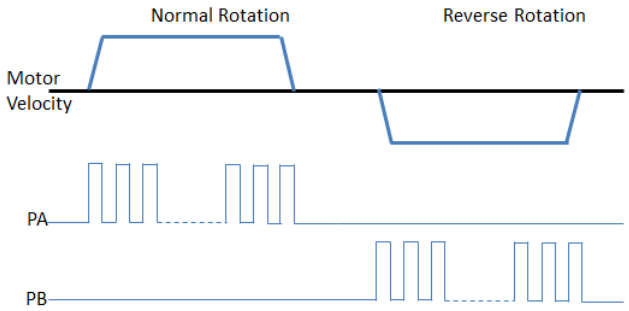
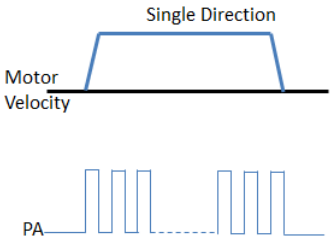
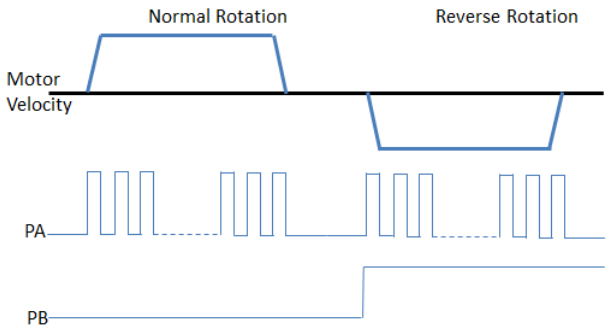
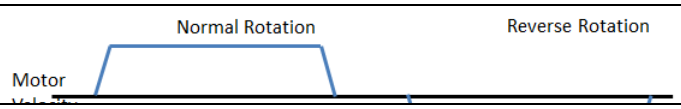
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Z	B	A	DI-3	DI-2	DI-1	DI -0

Value 0: Input state is OFF

Value 1: Input state is ON

8.1.2 Pulse Input Method：5501h

Sub Index 00h: Pulse Input Method

<table>
<tr><td colspan="5">Bit7- Bit 5 ：Reserved</td></tr>
<tr><td>Bit 4</td><td colspan="4">0: Axis Encoder
1: External Encoder(MPG..)</td></tr>
<tr><td rowspan="6">Bit3-
Bit0</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>
<tr><td>3</td><td>Pulse</td><td>Direction</td><td> 
</td></tr>
<tr><td>4</td><td>A</td><td>B</td><td rowspan="2"> 
</td></tr>
<tr><td>5</td><td>A(*2)</td><td>B(*2)</td></tr>
</table>

	6	A(*4)	B(*4)	

8.1.3 Input Polarity：5502h

Sub Index 00h: Input Polarity

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Z	B	A	DI-3	DI-2	DI-1	DI -0

Value 0: non-reverse

Value 1: reverse

8.1.4 Digital Input Function：5503h

Digital Input Functions

Sub Index	Input Point	Description
01h	DI 0	0:Normal DI 1:Home P Limit 9:In Position Signal
02h	DI 1	0:Normal DI 1:Home N Limit 9:In Position Signal
03h	DI 2	0:Normal DI 1: Force Stop 9:In Position Signal
04h	DI 3	0:Normal DI 1:Home Switch 9:In Position Signal
05h	DI A	0:Normal DI
06h	DI B	0:Normal DI
07h	DI Z	0:Normal DI 1:Index 9:In Position Signal

8.1.5 Digital Input Filter：5504h

Digital Input Filter

Sub Index	Input Point	Description
01h	DI 0	Bit7~4: Clock Divider (m), value range: 0~6 Bit3~0: Sample Clock Cycles (n), value range: 0~3, 0=bypassed Maximum pulse duration threshold: 0x63 Minimum pulse duration threshold 0x00 Pulse duration threshold: (n>0) = $\frac{2^m}{72} \times (n+1)$ Unit: us
02h	DI 1	
03h	DI 2	
04h	DI 3	
05h	DI A	
06h	DI B	
07h	DI Z	

		Rejected when input pulse duration is less than or equals to pulse duration threshold.

8.1.6 Digital Output：5510h

Sub Index 01h: Input Settings

Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Z	PB	PA	DO-3	DO-2	DO-1	DO -0

Sub Index 02h: Output State

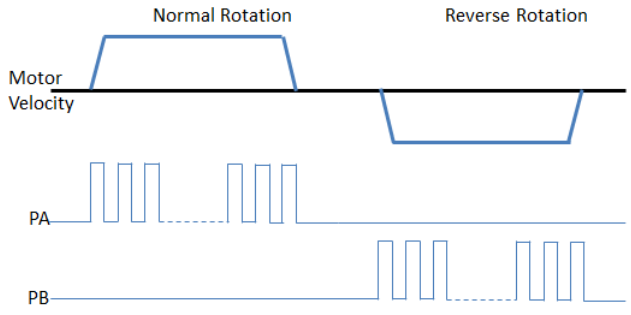
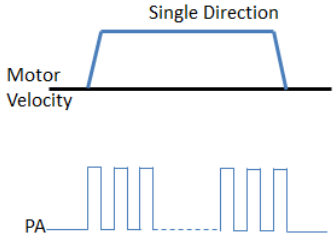
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
保留	Z	PB	PA	DO-3	DO-2	DO-1	DO -0

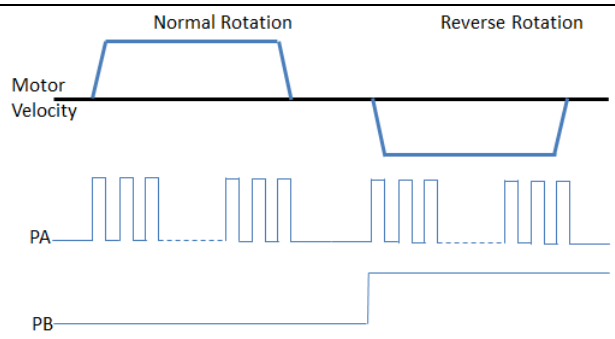
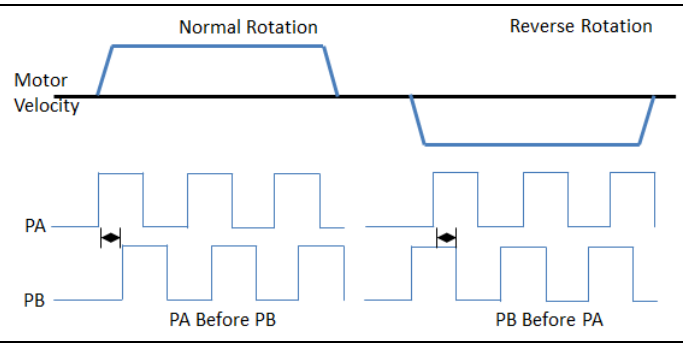
Value 0: Output state is OFF

Value 1: Output state is ON

8.1.7 Pulse Output Method：5511h

Sub Index 00h: Pulse Output Method

Bit7-Bit 4	Reserved			
Bit3-Bit 0	Value	PA	PB	
	0	Disable	Disable	
	1	CW	CCW	
	2	Pulse	NC	

	3	Pulse	Direction	
	4	A	B	
	5	A(*2)	B(*2)	
	6	A(*4)	B(*4)	

8.1.8 Output Polarity：5512h

Sub Index 00h: Output Polarity (not effective to pulse output)

Bit7-Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	B	A	DI-3	DI-2	DI-1	DI -0

Value 0: non-reverse

Value 1: reverse

8.1.9 Digital Output Function：5513h

Digital Output Functions

Sub Index	Output Point	Description
01h	DO 0	0:Normal DO 2: PWM0
02h	DO 1	0:Normal DO 2: PWM1
03h	DO 2	0:Normal DO
04h	DO 3	0:Normal DO
05h	PA	0:Normal DO
06h	PB	0:Normal DO 2:PWM1

8.1.10 Digital Output Abort Connection Option：5514h

Sub Index	Output Point	Description
01h	DO 0	0:Off
02h	DO 1	1:On

03h	DO 2	2:Keep last value Effective when output function is set to Normal, PWM will be Off , and Axis Pulse will Quick Stop
04h	DO 3	
05h	PA	
06h	PB	

8.1.11 PWM Output Setting：551Ah

Sub Index	Name	Description
01h	Output D0 setting	D0 D1 period less than 10(100k) is counted as 10. PB less than 2 is counted as 2(500k). duty cycle0~100% is adjustable, please see the spec to decide whether D0 D1 can be used. Width less than or equals to 2 will be counted as 0.
02h	Output D1/PB setting	

8.1.12 Axis Setting0：5520h

Sub Index	Name
01h	Motion Cycle Time
02h	Bias Velocity

8.1.13 Axis Setting1：5521h

Sub Index	Name	Description
01h	Backlash compensation (pulse)	Range: 0~65535

8.1.14 Additional position modulo range：5528h

Sub Index	Name	Description
01h	1st additional position modulo range	Value 0: Linear(Finite) Axis Value 1~ 2147483647:Modulo Axis
02h	2nd additional position modulo range	

8.1.15 Additional home offset：5529h

Sub Index	Name	Description
01h	1st additional home offset	Use Axis0's MC_Homing to set Offset
02h	2nd additional home offset	

8.1.16 Gear Motion Setting：5530h

Sub Index	Name	Description
01h	Master Direction Limit	bit 0: Master Direction Limit On/Off bit 1: Slave Direction Limit On/Off
02h	Slave(PU) Direction Limit	
03h	Moving Average Size	0~250
04h	Following error window	0~65535
05h	Following error time out	0~65535(ms)

8.1.17 DigitalCamSwitch Enable：5580h

Bit7-Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	B	A	DI-3	DI-2	DI-1	DI -0

Sub Index	Name	Description
01h	DigitalCamSwitch Enable	Bit5：Input B Bit4：Input A Bit3：Input DI-3 Bit2：Input DI-2 Bit1：Input DI-1 Bit0：Input DI-0 0:Disable 1:Enable
02h	EnableMask Track 0-5	Bit5：Input B

		Bit4：Input A Bit3：Input DI-3 Bit2：Input DI-2 Bit1：Input DI-1 Bit0：Input DI-0 0:Track Disable 1:Track Enable

8.1.18 DigitalCamSwitch Track Position ValueSource：5581h

Sub Index	Name	Description
01h	Track D0 ValueSource	0:Cmd Position(1st additional) 1:Act Position(2nd)
02h	Track D1 ValueSource	
03h	Track D2 ValueSource	
04h	Track D3 ValueSource	
05h	Track PA ValueSource	
06h	Track PB ValueSource	

8.1.19 DigitalCamSwitch MC_CAMSWITCH_REF：5583h

Sub Index	Name	Description
6n+01h	Switch n TrackNumber	0~5 : Track D0 ~Track PB
6n+02h	Switch n FirstOnPosition	Lower boundary where the switch is ON
6n+03h	Switch n LastOnPosition	Upper boundary where the switch is ON
6n+04h	Switch n AxisDirection	Both (=0; Default); Positive (1); Negative (2)
6n+05h	Switch n CamSwitchMode	Position based (=0; Default); Time based (=1)
6n+06h	Switch n Duration(ms)	Coupled to time based CamSwitchMode: 1~16000 ms

n=0~15

8.1.20 Motion Output Setting：558Fh

Bit31-Bit16	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reversed	B	A	DI-3	DI-2	DI-1	DI -0

Sub Index	Name	Description	
01h	Motion Output Setting0	bit31-bit16	Output Duration 1~16000 ms
		bit 15:	0=blocking (wait for Output off)
			1=non-blocking

		b14-bit6	Reserved	
		bit5	PB	0:Disable 1:Enable
		bit4	PA	
		bit3	DO 3	
		bit2	DO 2	
		bit1	DO 1	
		bit0	DO 0	

8.1.21 Capture Enable：5590h

Sub Index	Name	Description		
01h	Capture Enable Byte 0	bit31-bit6	Reserved	
		bit5	Channel 5	0:Channel Disable 1:Channel Enable
		bit4	Channel 4	
		bit3	Channel 3	
		bit2	Channel 2	
		bit1	Channel 1	
		bit0	Channel 0	

8.1.22 Capture Status：5591h

Sub Index	Name	Description		
01h	Capture Status Byte 0	bit31-bit6	Reserved	
		bit5	Channel 5	0:no value 1:got value
		bit4	Channel 4	
		bit3	Channel 3	
		bit2	Channel 2	
		bit1	Channel 1	
		bit0	Channel 0	

8.1.23 Capture Setting：5592h

The capture interval must be longer than 1ms, 16#5501 Pulse Input Method: CW_CCWcapture

target 2~4 cannot be used

Sub Index	Name	Description
01h	Capture Setting Channel 0	Please see Capture Setting list below.
02h	Capture Setting Channel 1	
03h	Capture Setting Channel 2	
04h	Capture Setting Channel 3	

05h	Capture Setting Channel 4	

Capture Setting		
bit	Name	Value
bit 31-20	Reserved	
bit 16~19	Interval	0~4 Interval between channel0~4
bit 15	Interval Mode	0 : OFF 1: On
bit 14	Reserved	
bit 13	Continuous Mode	0 : OFF 1: On
bit 12	Falling Edge Trigger	0:Falling Edge Trigger 1:Rising edge trigger
bit4~7	Signal	0：DI-0 1：DI-1 2：DI-2 3：DI-3 4：A 5：B 6：Z
bit0~3	capture target	0:Cmd pos 1: 1st addl pos 2: act positiob 3: 2nd addl pos 4: timer(unit:250ns)

8.1.24 Capture Status：5598h

Sub Index	Name	Description
01h	Capture Value 0	Capture Value
02h	Capture Value 1	
03h	Capture Value 2	
04h	Capture Value 3	
05h	Capture Value 4	

8.1.25 Motion Trigger Setting：559Fh

Bit7-Bit6	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved			Trigger	MODE			

MODE: 0~6 = DI0~Z

Trigger: 1:Rising edge trigger 0:Falling Edge Trigger

Sub Index	Name	Description
01h	Motion Trigger Setting0	Trigger ：1:Rising edge trigger 0:Falling Edge Trigger MODE ：0~6 = DI0~Z
02h	Motion Trigger Setting1	
03h	Motion Trigger Setting2	

8.2 Standardized device profile Area (6000h - 7FFFh)

Item	Index Range
Axis 0(1st PU)	6000-67FF*
Axis 1(2nd PU)	6800-6FFF
Axis 2(3rd PU)	7000-77FF
Axis 3(4th PU)	7800-7FFF

Axis 0's Object will be in the same index range 6000-67FF as AIO and DIO. Please see Cia402's document for more information on Object Dictionary.

In the following list, n=0~3 which represents Axis 0~3.

Index	Sub Index	Description	Type	ro/rw	Default
6007h+n*800h	00h	Abort connection option code	INT	rw	1h
603Fh+n*800h	00h	Error code	UINT	ro	----
6040h+n*800h	00h	Control word	UINT	rw	0h
6041h+n*800h	00h	Status word	UINT	ro	----
605Eh+n*800h	00h	Fault reaction option code	INT	rw	0h
6060h+n*800h	00h	Modes of operation	SINT	rw	0h
6061h+n*800h	00h	Modes of operation display	SINT	ro	0h
6062h+n*800h	00h	Position demand value	DINT	ro	0h
6063h+n*800h	00h	Position actual internal value	DINT	ro	0h
6064h+n*800h	00h	Position actual value	DINT	ro	0h
606Bh+n*800h	00h	Velocity demand value	DINT	ro	0h
606Ch+n*800h	00h	Velocity actual value	DINT	ro	0h
607Ah+n*800h	00h	Target Position	DINT	rw	0h
607Bh+n*800h		Position range limit			
	01h	Min position range limit	DINT	ro	0h
	02h	Max position range limit	DINT	rw	0h
607Ch+n*800h	00h	Home offset	DINT	rw	0h
607Dh+n*800h		Software position limit			
	01h	Min position limit	DINT	rw	0h
	02h	Max position limit	DINT	rw	0h
607Fh+n*800h	00h	Max profile velocity	UDINT	rw	2000000
6080h+n*800h	00h	Max motor speed	UDINT	rw	2000000
6081h+n*800h	00h	Profile velocity	UDINT	rw	0h
6083h+n*800h	00h	Profile acceleration	UDINT	rw	0h
6084h+n*800h	00h	Profile deceleration	UDINT	rw	0h
6085h+n*800h	00h	Quick stop deceleration	UDINT	rw	10000000
608Fh+n*800h		Position encoder resolution			
	01h	Encoder increments	UDINT	rw	1h
	02h	Motor revolutions	UDINT	rw	1h

6091h+n*800h		Gear ratio			
	01h	Motor shaft revolutions	UDINT	rw	1h
	02h	Driving shaft revolutions	UDINT	rw	1h
6092h+n*800h		Feed constant			
	01h	Feed	UDINT	rw	1h
	02h	Shaft revolutions	UDINT	rw	1h
6098h+n*800h	00h	Homing method	SINT	rw	37
6099h+n*800h		Homing speeds			
	01h	Speed during search for switch	UDINT	rw	1000
	02h	Speed during search for zero	UDINT	rw	500
609Ah+n*800h	00h	Homing acceleration	UDINT	rw	1000
60A4h+n*800h		Profile jerk			
	01h	Profile jerk 1	UDINT	rw	50000000
60C5h+n*800h	00h	Max acceleration		rw	10000000
60C6h+n*800h	00h	Max deceleration		rw	10000000
60E4h+n*800h		Additional position actual value			
	01h	1st additional position actual value	DINT	ro	0
	02h	2nd additional position actual value	DINT	ro	0
60E6h+n*800h		Additional position encoder resolution - encoder increments			
	01h	1st additional position encoder resolution -encoder increments	UDINT	rw	0
	02h	2nd additional position encoder resolution - encoder increments	UDINT	rw	0
60E8h+n*800h		Additional gear ratio -motor shaft revolutions			
	01h	1st additional gear ratio -motor shaft revolutions	UDINT	rw	1
	02h	2nd additional gear ratio -motor shaft revolutions	UDINT	rw	1
60E9h+n*800h		Additional feed constant -feed			
	01h	1st additional feed constant -feed	UDINT	rw	1
	02h	2nd additional feed constant -feed	UDINT	rw	1
60EBh+n*800h		Additional position encoder resolution -motor revolutions			
	01h	1st additional position encoder resolution -motor revolutions	UDINT	rw	1
	02h	2nd additional position encoder resolution -motor revolutions	UDINT	rw	1
60EDh+n*800h		Additional gear ratio -driving shaft revolutions			
	01h	1st additional gear ratio -driving shaft revolutions	UDINT	rw	1
	02h	2nd additional gear ratio -driving shaft revolutions	UDINT	rw	1

60EEh+n*800h		Additional feed constant -driving shaft revolutions			
	01h	1st additional feed constant -driving shaft revolutions	UDINT	rw	1
	02h	2nd additional feed constant -driving shaft revolutions	UDINT	rw	1
60FCh+n*800h	00h	Position demand internal value	DINT	ro	0h
60FDh+n*800h	00h	Digital inputs	UDINT	ro	0h
60FFh+n*800h	00h	Target velocity	DINT	rw	0
6502h+n*800h	00h	Supported drive modes	UDINT	ro	25h
67FFh+n*800h	00h	Device type	UDINT	ro	FFFF0192h

9. Motion Control Function Blocks

Weintek Motion Control Function Blocks designed according to PLCopen Motion Control makes it easy to give motion control instructions to iR-PU01-P.

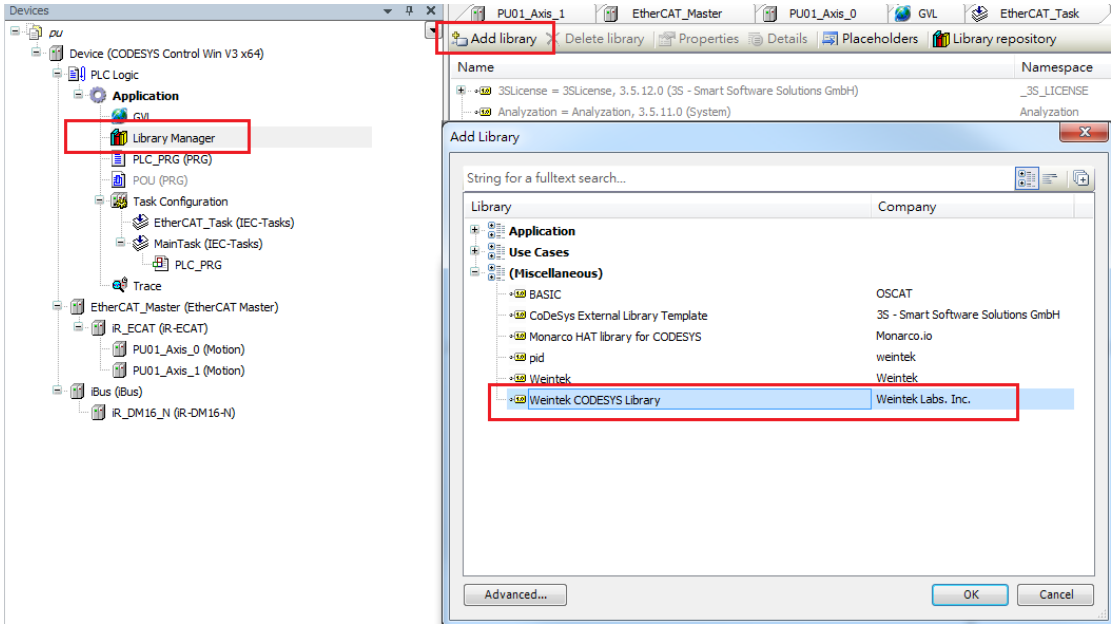
9.1 Motion Control Function Block List

Item	Name	Description
1	AXIS_REF_LITE	Information on the corresponding axis.
2	MC_Power	Starts or stops the system.
3	MC_Home	Performs homing.
4	MC_MoveVelocity	Performs velocity control.
5	MC_MoveAbsolute	Performs positioning for the specified absolute target position.
6	MC_MoveRelative	Performs positioning for a relative position.
7	MC_Gear	Specifies the gear ratio between the master axis and the slave axis and starts gear operation.
8	MC_Stop	Decelerates an axis to a stop. Motion instructions can only be given after the axis stops.
9	MC_Halt	Stops axis operation and ends all motion control function blocks. The speed returns to 0. Motion instructions can be given to interrupt MC_Halt.
10	MC_Reset	Clears axis error and make the axis return to Standstill state.

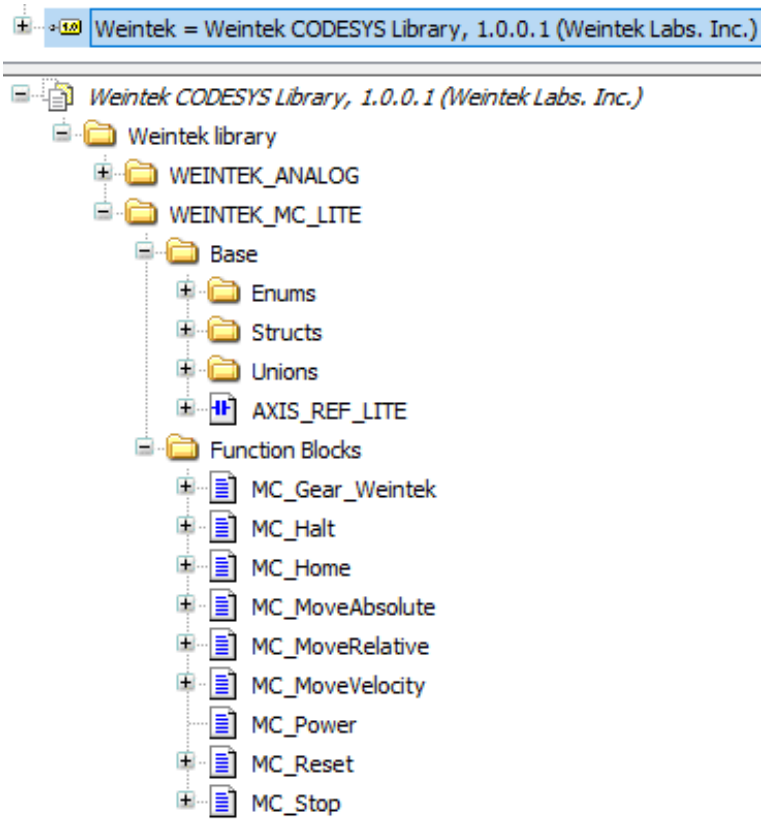
9.2 Download and Install

Step 1. Open the Download page in Weintek official website, search for cMT+CODESYS Package, download and install the package.
https://www.weintek.com/globalw/Download/Download.aspx

Step 2. In CODESYS software interface, add Weintek CODESYS Library.



Step 3. Motion Function Blocks are ready for use now.



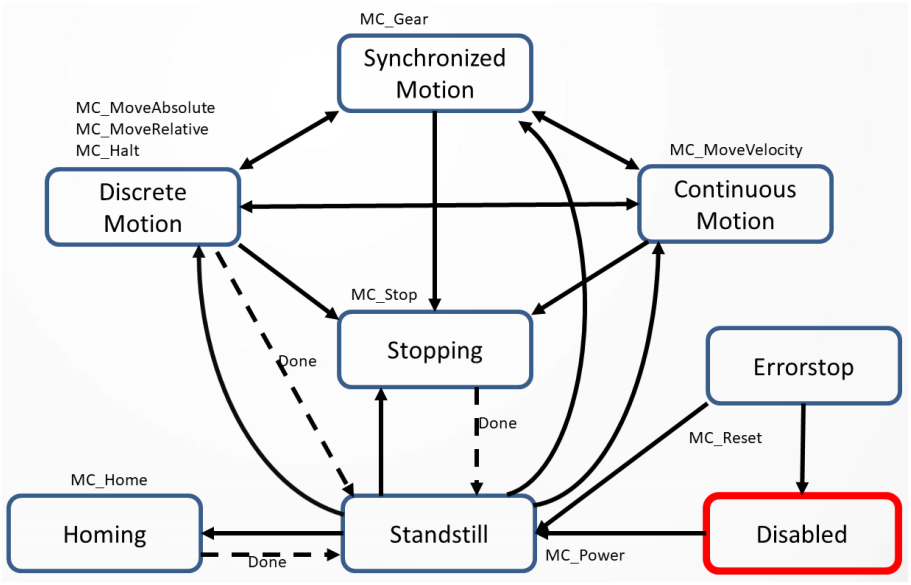
9.3 MC_Status

The PLCopen motion standard provides a way to have standard application libraries that are reusable for multiple hardware platforms, which reduces costs during development, maintenance and training. The states of axes and state transitions caused by the execution of instructions are based on the PLCopen specifications for

motion control.

The operation of an axis when motion control instructions are executed for it is shown in the following figure, and the arrows show state transitions. When any error occurs, the state changes to Errorstop.

※Status in the red frame in the following figure is the initial state.



Declare AXIS_REF_LITE.

```
PROGRAM PLC_PRG
VAR
    Axis000 : Weintek.Axis_REF_Lite ;
    MC_Power_0: weintek.MC_Power ;
    MC_MoveVelocity_0: weintek.MC_MoveVelocity;
    MC_Stop_0: weintek.MC_Stop;
    MC_Reset_0: weintek.MC_Reset;
```

MC_Status can be found under AXIS_REF_LITE after login.

Expression	Type	Value
Axis000	Weintek.Axis_REF_Lite	
_Delay_Cycles	BYTE	0
_CMPT_PV	BOOL	FALSE
Mapping_Q	unAXIS_VAR_OUT	
Mapping_I	unAXIS_VAR_IN	
_MC_Status	EAXIS_STATE	Standstill

9.4 Execution of Function Blocks

- Execute and Enable are two input variables that can start function block execution.

  Execute: Starts execution of current function block used at the moment when Execute changes from OFF to ON. Execution of the function block will continue

until another instruction is executed and interrupts operation.

Enable: When Enable variable is ON, execution of function block continues, and the execution stops when Enable if OFF. Modifications to the parameters during execution are usually effective.

- Busy, Active, Done, In***, CommandAbort, and Error are output variables that indicate the execution status of function blocks.

  Busy: Function block is executing.

  Active: Function block obtains permission to control the applicable axis.

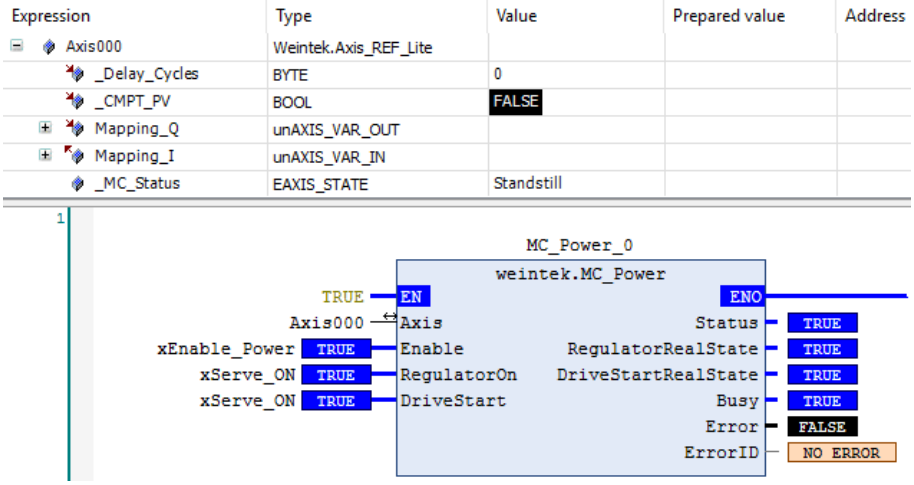
  Done and In*** (asterisk stands for any string of characters): Function block operation ends or when the commanded condition is reached.

  CommandAbort: Another operation instruction interrupts the commended condition.

  Error: An error occurred during the execution of function block.
- Triggering Execute variable during the execution is ineffective.
- Only function blocks that support ContinuousUpdate can be updated during execution (Execute variable is in ON state); other function blocks (ContinuousUpdate included) are triggered at the moment when Execution variable turns ON.

9.5 MC_Power

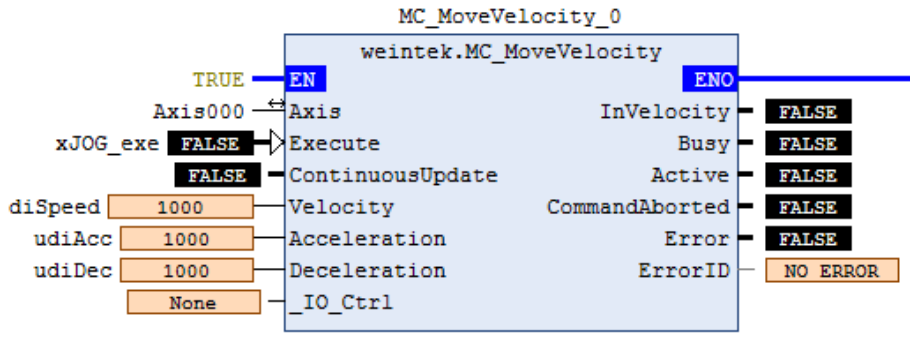
Executing the MC_Power function block makes the Servo ready to operate. The Power function block should be executed before using any Motion function blocks. After executing the Power function block, the axis enters Standstill state.



As shown in the figure above, after triggering MC_Power.Enable, MC_Status enters Standstill state, which means the axis is ready for motion instructions.

9.6 MC_MoveVelocity

MC_MoveVolocity function block performs velocity control for the specified axis. The following parameters are used when executing MC_MoveVelocity.



Velocity: Specify the target velocity and the rotation direction. Positive velocity = positive direction, negative velocity = negative direction.

Acceleration: Specify the acceleration rate, the value cannot be 0.

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

ContinuousUpdate: Continuously updates the velocity. TRUE= the target velocity, acceleration rate and deceleration rate can be changed when the axis is operating. An axis that is operating and is in Continuous Motion state can only be stopped using MC_Stop or MC_Halt.

IO_Ctrl: Trigger execution using digital input.

9.7 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.

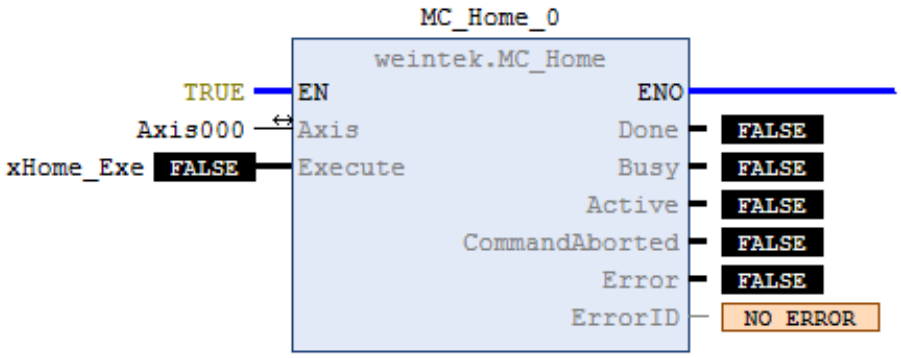
607C: Home offset.

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

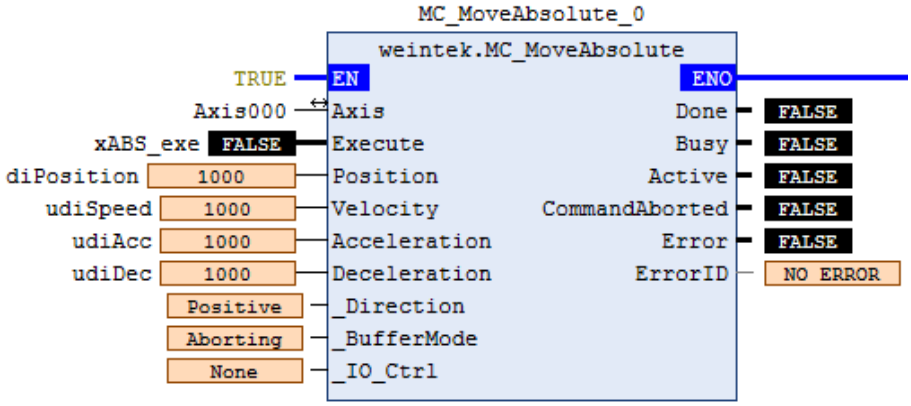
Executing MC_Home when the axis is in Stanstill state performs homing using the parameters shown above. The axis changes to Homing state when this function block

is executed, and returns to Standstill state after homing is completed.



9.8 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.

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

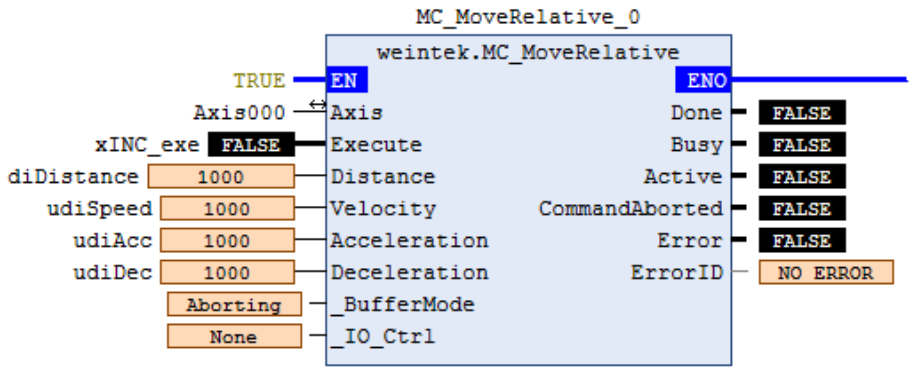
Direction: Specify the direction and the shortest path.

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.

9.9 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.

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

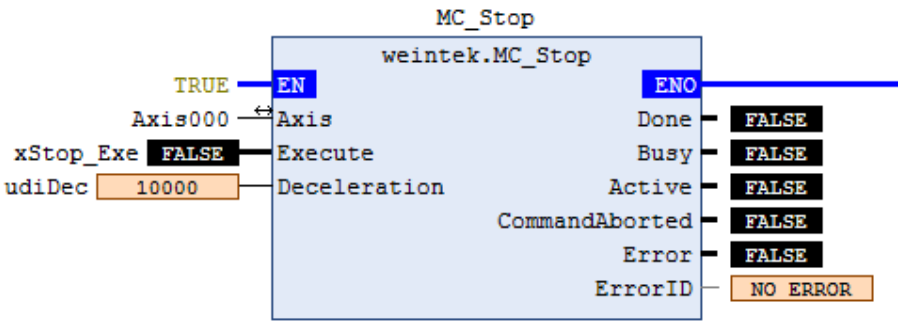
BufferMode: Continuously executes the next instruction after the ongoing motion is completed.

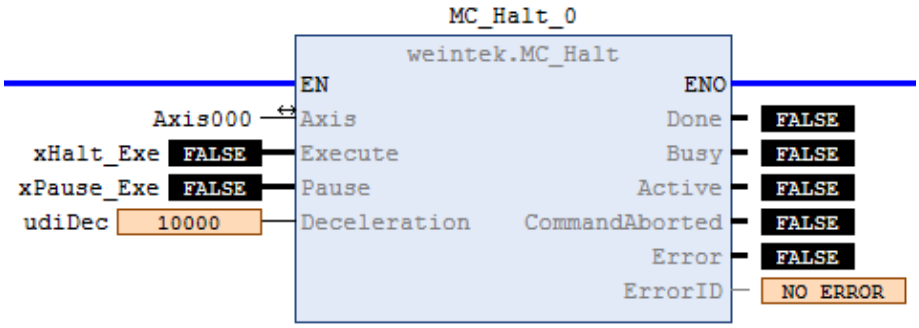
IO_Ctrl: Trigger execution using digital input, and then output digital signal after the motion is completed.

9.10 MC_STOP and MC_Halt

MC_STOP and MC_Halt function blocks can stop axis operation. When using MC_Halt, instructions can still be given to the axis before it stops. 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 or MC_Halt.

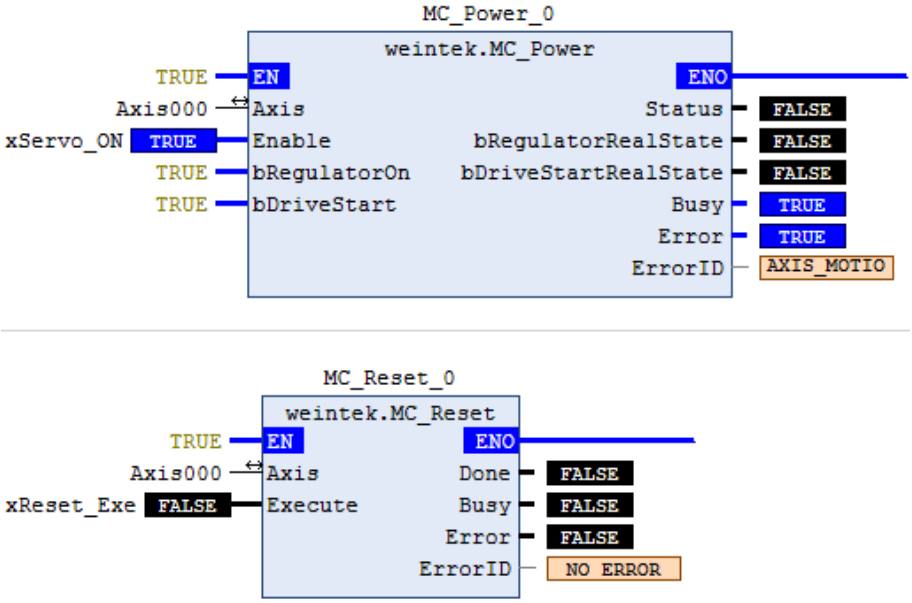




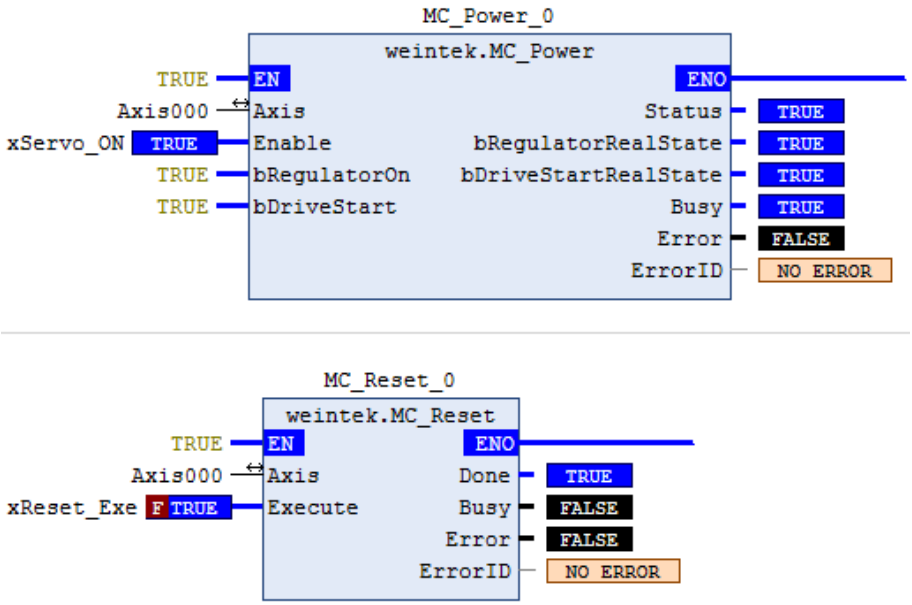
Deceleration: Specify the deceleration rate, the value cannot be 0.
The axis enters Standstill state after it stops.

9.11 MC_Reset

Triggering MC_Reset function block can reset the errors when the axis turns into Errorstop state due to error. The axis enters Disabled state when MC_Power is FALSE, and the axis enters Standstill state when MC_Power is TRUE.
If the axis stays in Errorstop state after triggering MC_Reset, please check the cause of error again, in order to clear the error.



When an error occurs, MC_Power.Error=TRUE, please execute MC_Reset to change the axis state from Errorstop to Standstill, to continue operation.



9.12 MC_Gear

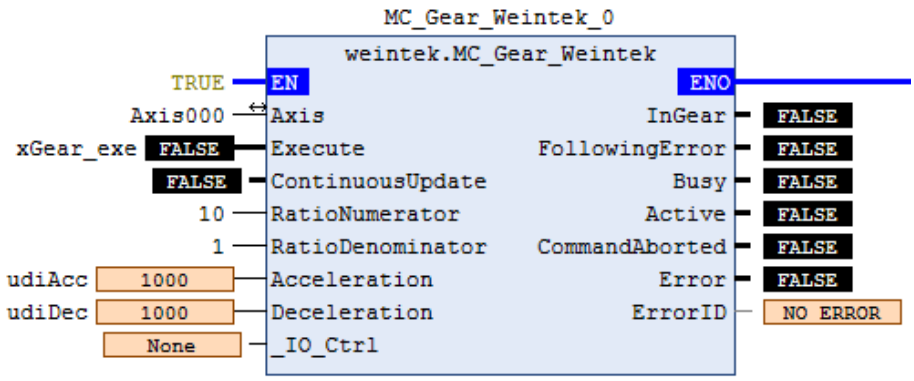
MC_Gear function block converts the input pulse from MPG (Manual Pulse Generator) into output pulse.

Since pulse input and output use different axes (master axis and slave axis), Pulse Input Method 5501h must be configured:

Bit 4: 0 (axis encoder), 1 (external encoder), please set Bit 4 to 1 (external encoder) to use MPG.

Bit 0~3: Please see chapter 8.1.2 in this manual.

The following parameters are used when executing MC_Gear.



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

Acceleration/Deceleration: Specify the acceleration / deceleration rate, the value cannot be 0.

RatioNumerator: Specify the numerator of the gear ratio between the master and slave axes.

RatioDenominator: Specify the denominator of the gear ratio between the master

and slave axes.

Output Pulse = Input Pulse $*\frac{RatioNumerator}{RatioDenominator}$

After executing this function block the axis enters Synchronized Motion state and can only be stopped using MC_Stop or MC_Halt.
IO_Ctrl: Trigger execution using digital input.

10.Quick Start of iR-PU01-P in CODESYS CANopen

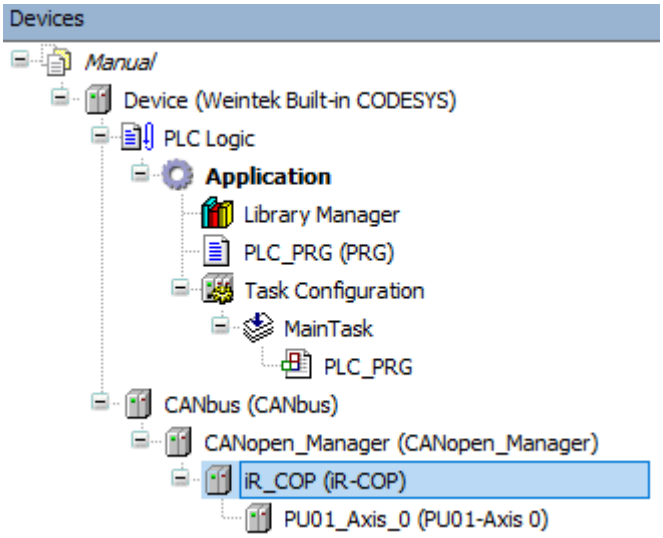
iR-PU01-P supports high speed pulse output (PA, PB). Pulse output modes include: A/B phase (*1/*2/*4), CW/CCW, Pulse/Direction, Pulse Only. Please check the input method used by the motor, and configure iR-PU01-P in accordance. Please also take wiring into consideration.
The following steps explain how to start iR-PU01-P module.

10.1 Install and Add Weintek Library

Please see Chapter 9.1 in this manual to download and install Weintek Library.
Open [Library Manager] -> [Add Library] to add Weintek Library.

10.2 Launch New Project and Add iR-PU01-P

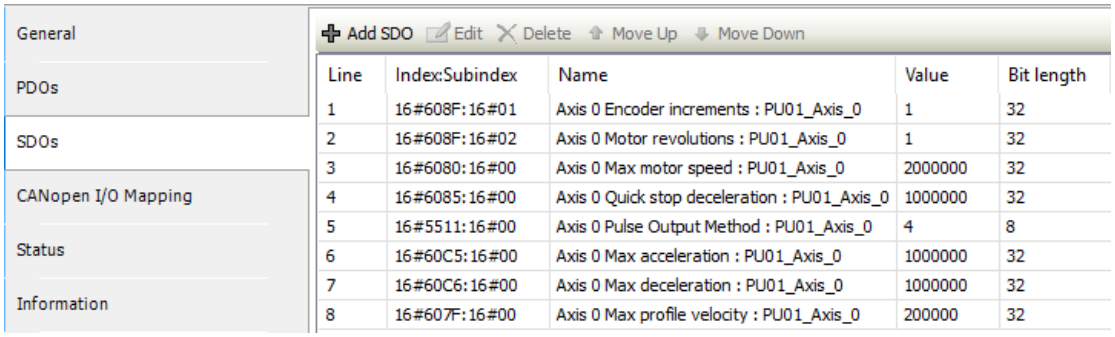
Add CANbus device:
[Device]->[Add Device]->[Fieldbesses]->[CANbus]
Add CANopen_Manager device:
[CANbus]->[Add Device]->[CANopen_Manager]
Add iR-COP coupler:
[CANopen_Manager] ->[Add Device]->[iR-COP] (V1.3)
Add iR-PU01-P module:
[iR-COP]->[Add Device]->[PU01-Axis 0]



10.3 Configuring Motion Control Parameters

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

Parameters	Name	Index	Sub Index	Value	Bit Length
Motor	Encoder Increments	16#608F	16#01	1	32
	Motor Revolutions	16#608F	16#02	1	32
Pulse	Pulse Output Method	16#5511	16#00	4(=AB phase)	8
Velocity	Max. Motor Speed	16#6080	16#00	2000000	32
	Max. Acceleration	16#60C5	16#00	1000000	32
	Max. Deceleration	16#60C6	16#00	1000000	32
	Max. Profile Velocity	16#607F	16#00	200000	32
Quick Stop	Quick Stop Deceleration	16#6085	16#00	1000000	32



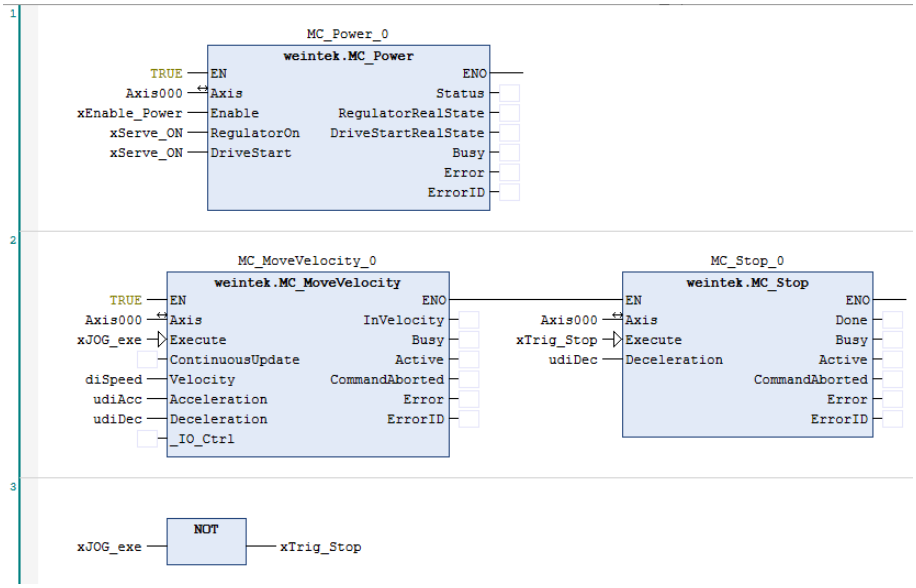
The above parameters should be configured in advance for iR-PU01-P to perform pulse control.

10.4 Declaration and Programming

Declare Axis000 and run trial operation of Function Blocks.

```
VAR
    // Axis reference
    Axis000 : Weintek.Axis_REF_Lite ;
    // Motion Control Function Block
    MC_Power_0: weintek.MC_Power ;
    MC_MoveVelocity_0: weintek.MC_MoveVelocity;
    MC_Stop_0: weintek.MC_Stop;
    MC_Reset_0: weintek.MC_Reset;
    // JOG Button
    xEnable_Power, xServe_ON, xJOG_exe, xTrig_Stop, xTrig_Reset : BOOL ;
    // JOG parameter
    diSpeed : DINT := 1000 ;
    udiAcc : UDINT := 1000 ;
    udiDec : UDINT := 1000 ;
```

Use FBD (Function Block Diagram) programming.



The following three function blocks are used for JOG operation:

MC_Power: Starts motion control system

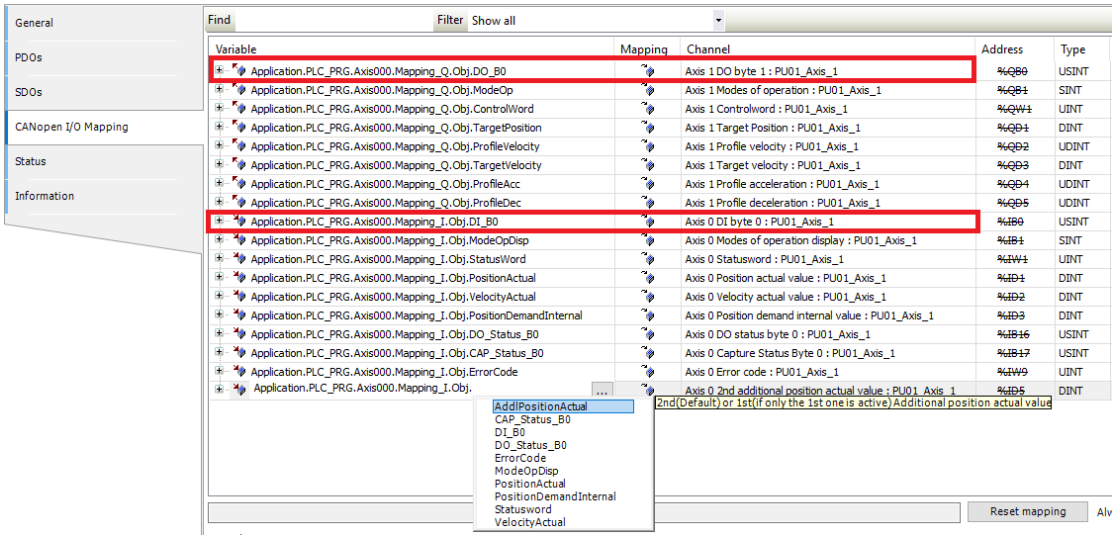
MC_MoveVelocity: Performs velocity control.

MC_Stop: Decelerates until stop.

10.5 Axis I/O Mapping

Mapping of input variables can be found in Axis000.Mapping_I, and mapping of output variables can be found in Axis000.Mapping_Q, the user only need to fill in the string of characters used in Channel.

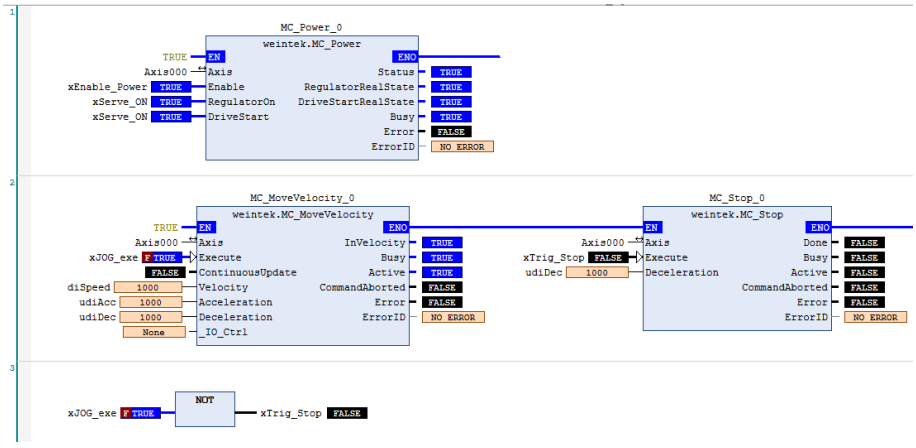
I/O mapping variables are shown below:



Please make sure that the Variable is identical to the Channel, and Mapping_Q should be completely mapping to Mapping_I.

10.6 Login and Run Trial Operation

After the settings explained in the preceding steps are completed, you can now log in and run trial operation of function blocks.



Press xPEnable_Power & xServe_ON button to execute MC_Power function block to start iR-PU01-P.

Press xJOG_exe button to execute MoveVelocity function block to make iR-PU01-P output pulse for velocity control.

Release xJOG_exe to execute MC_Stop function block to decelerate pulse output to a stop.