User Manual

iR-PU01-P

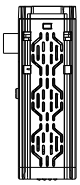
This guide walks through important information about iR-PU01-P.

UM019004E_20231116

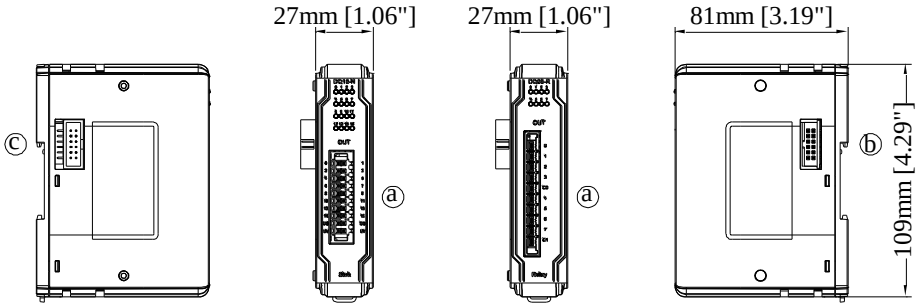
Table of Contents

目錄

1. Product Overview



Top View



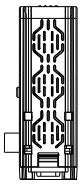
Side View

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

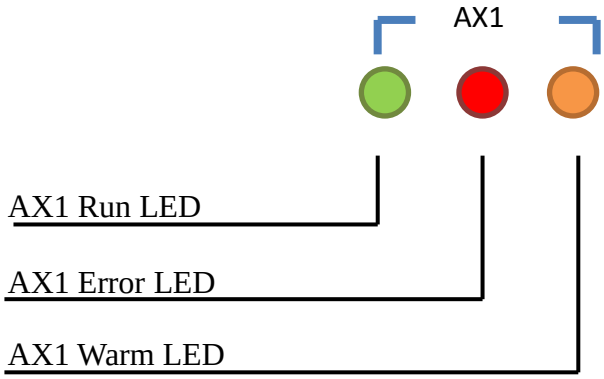
Item	Sink Input	Differential Input
Number of Inputs	4	3 (A/B/Z phase)
Input current	24 VDC, 5 mA	Meets the Requirements of ANSI Standards TIA/EIA-485-A
HIGH Level Input Voltage	15~28 VDC	-
LOW Level Input Voltage	0~5 VDC	-
Maximum input frequency	200KHz	2MHz
Input Impedance	3 KΩ	-
Indicators	Red LED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

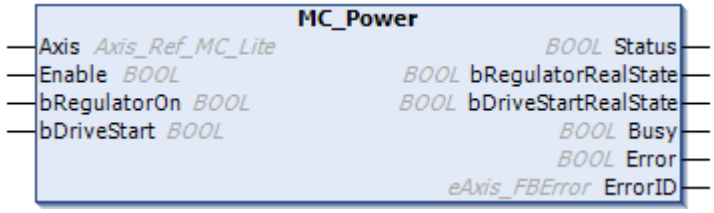
3.3 I/O LED

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 prevent MC_HOME from switching to other motions, and avoid associating the positioning in buffered mode 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

The status of the axis is ErroStop

The cause of Error is indicated in the Axis Error Code. To see more details on the cause of error, read Sub Error Code 553Fh of the axis using function block (e.g. SDO_READ4).

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.

4.3.1 Error Codes:

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#6385	Parameter Error 5	Incorrect setting of Gear related parameters.

16#6386	Parameter Error 6	Incorrect setting of CAM related parameters.
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.

4.3.2 Sub Error Code

When an error occurs, read object dictionary address 5X3F* to find the Sub Error Code for troubleshooting.

*X represents the sequence number of the axis where X = 5 indicates the 0th axis, X = 6 indicates the 1st axis, X = 7 indicates the 2nd axis, X = 8 indicates the 3rd axis.

Error Code	Sub Error Code	Cause of Error
16#618A	1	Wrong value in Homing method.
	2	Positive Limit is not configured.
	3	Negative Limit is not configured.
	4	Index is not configured.
	5	Home Switch is not configured.
	6	Unused Limit Switch in Homing procedure is contacted.
	7	Limit Switch is contacted while Homing is done.
Error Code	Sub Error Code	Cause of Error
16#6320	1	V bias is over maximum velocity.
	2	Velocity settings error.
	3	Acceleration settings error.
	4	Deceleration settings error.
Error Code	Sub Error Code	Cause of Error
16#6382	1	Product of Sub Index 1 of 608F, 6091 and 6092 is over 2147483647.
	2	Product of Sub Index 2 of 608F, 6091 and 6092 is over 2147483647.
Error Code	Sub Error Code	Cause of Error
16#6383	1	Product of 60E6, 60E8 and 60EE is over 2147483647.
	2	Product of 60EB, 60ED and 60E9 is over 2147483647.
Error Code	Sub Error Code	Cause of Error
16#6384	1	Product of 60E6, 60E8 and 60EE is over 2147483647.
	2	Product of 60EB, 60ED and 60E9 is over 2147483647.
Error Code	Sub Error Code	Cause of Error
16#6385	1	Master Encoder is not configured.
	2	Bidirection of Master is prohibited.

	3	Bidirection of Slave is prohibited
	4	Ratio Denominator is 0 in function block.
	5	Acceleration exceeds the maximum acceleration on Function Block.
	6	Deceleration exceeds the maximum Deceleration on Function Block.
Error Code	Sub Error Code	Cause of Error
16#6386	1	Master Encoder is not configured.
	2	MasterScaling setting error.
	3	SlaveScaling setting error.
	4	StartMode setting error.
	5	CAMTableID setting error.
	6	CAM table is not correctly specified. (Master isn't strictly increasing)
	7	EngageMode setting error
	8	EngageDirection setting error
Error Code	Sub Error Code	Cause of Error
16#7500	1	Communication error occurs during motion.
	2	Communication error occurs in Standstill state or error state.

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.

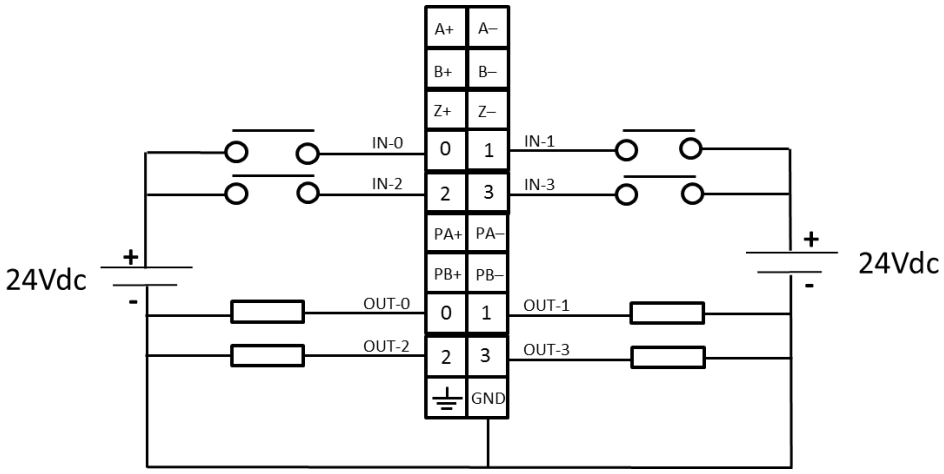
- Wiring for Digital Output

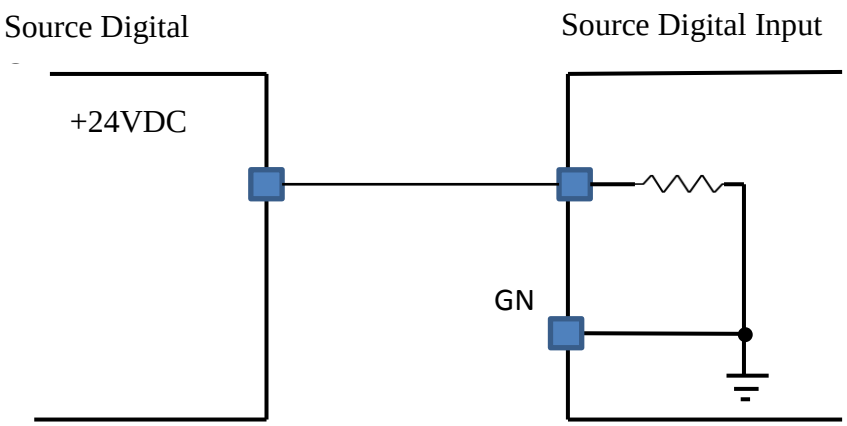
Digital output voltage range: 24VDC (-15%/+20%)
The maximum output voltage per point is 50mA, please take this into consideration when wiring.

- Wiring for Digital Input

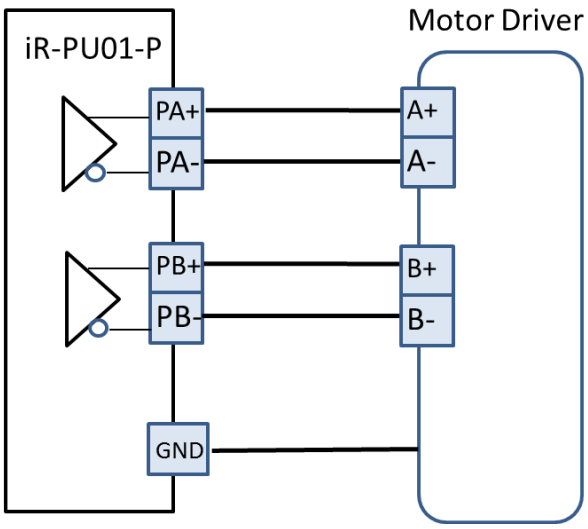
Digital input voltage range: 15~28VDC (ON), 5V (OFF)
Input impedance: 3 KΩ
Please take this into consideration when wiring.

5.2 Digital Input / Output Wiring

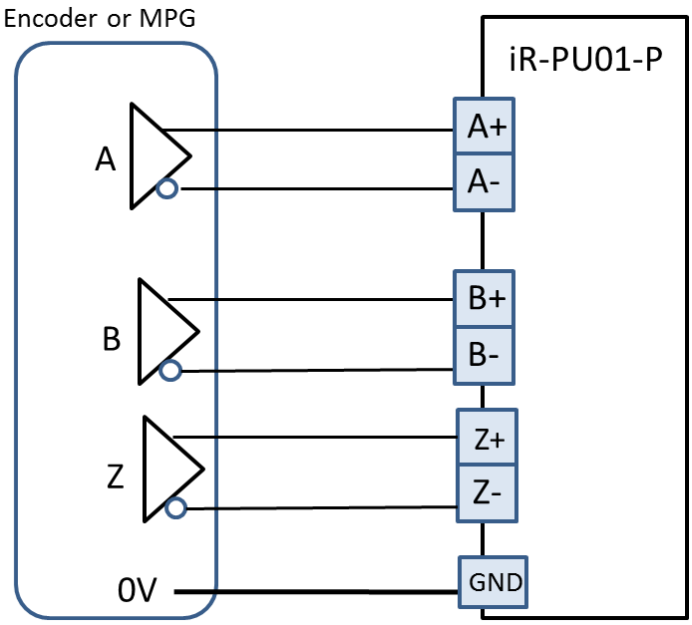




5.3 Differential Output Wiring



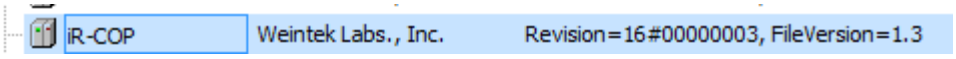
5.4 Differential Input Wiring



6. Connecting a Coupler

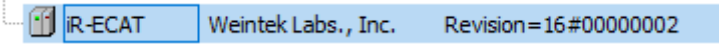
6.1 iR-COP

- 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 iR-ECAT

- An iR-ECAT coupler supports up to 4 iR-PU01-P modules at a time.
- iR-ECAT software version should be 1.00.2 or later.
- ESI file version should be Revision 16#00000002



6.3 iR-ETN

- An iR-ETN coupler supports up to 4 iR-PU01-P modules at a time.
- iR-ETN software version should be 1.0.2.0 or later.
- iR-ETN can be connected to iR-PU01-P modules for motion control. Modbus addresses specific to motion control are explained in chapter 6.3.1~6.3.3 in this user manual.

6.3.1 iR-ETN Axis Variable Instance Mapping Area

Name	Modbus Address	Size
Axis 0 Input	40000~40015	32bytes
Axis 1 Input	40016~40031	32bytes
Axis 2 Input	40032~40047	32bytes
Axis 3 Input	40048~40063	32bytes
Axis 0 Output	40500~40515	32bytes
Axis 1 Output	40516~40531	32bytes
Axis 2 Output	40532~40547	32bytes
Axis 3 Output	40548~40563	32bytes

In the following tables, axis 0 is used.

Axis 0 Input:

Item	Address (Dec)	Description				
1	40000	High Byte	Axis 0 Mode of Operation Display	USINT	Unsigned 8	Dec
		Low Byte	Axis 0 Digital Input	BYTE	Unsigned 8	Hex
2	40001	Axis 0 Status word		UINT	Unsigned 16	Hex

3	40002	Axis 0 Position actual value (Lo word)	DINT	Signed 32	Dec
4	40003	Axis 0 Position actual value (Hi word)			
5	40004	Axis 0 Velocity actual value(Lo word)	DINT	Signed 32	Dec
6	40005	Axis 0 Velocity actual value(Hi word)			
7	40006	Axis 0 Position demand internal value(Lo word)	DINT	Signed 32	Dec
8	40007	Axis 0 Position demand internal value(Hi word)			
9	40008	High Byte / Axis 0 Digital Output Status	BYTE	Unsigned 8	Hex
		Low byte / Axis 0 Capture Channel Status	BYTE	Unsigned 8	Hex
10	40009	Axis 0 Error code	UINT	Unsigned 16	Hex
11	40010	Axis 0 2nd additional position actual value (Lo word)	DINT	Signed 32	Dec
12	40011	Axis 0 2nd additional position actual value(Hi word)			
	40012 ~40015	Reserved			

Axis 0 Output:

Item	Address (Dec)	Description			
1	40500	High Byte / Axis 0 Mode of Operation	USINT	Unsigned 8	Dec
		Low byte / Axis 0 Digital Output	BYTE	Unsigned 8	Hex
2	40501	Axis 0 Control word	UINT	Unsigned 16	Dec
3	40502	Axis 0 Target Position (Lo word)	DINT	Signed 32	Dec
4	40503	Axis 0 Target Position (Hi word)			
5	40504	Axis 0 Profile velocity (Lo word)	DINT	Signed 32	Dec
6	40505	Axis 0 Profile velocity (Hi word)			
7	40506	Axis 0 Target velocity (Lo word)	DINT	Signed 32	Dec
8	40507	Axis 0 Target velocity (Hi word)			
9	40508	Axis 0 Profile acceleration (Lo word)	DINT	Signed 32	Dec
10	40509	Axis 0 Profile acceleration (Hi word)			
11	40510	Axis 0 Profile deceleration(Lo word)	DINT	Signed 32	Dec
12	40511	Axis 0 Profile deceleration (Hi word)			
	40512 ~40515	Reserved			

6.3.2 iR-ETN Access Method

Please see the following table for more information on how iR-ETN reads or writes iR-PU01-P's parameters.

R/W	Address (Hex)	Description
Write Object	0xFFF0	Index
	0xFFF1	Sub-index (High Byte)

		Length (Low Byte)				
	0xFFF2	Hi Byte	0x56		WORD	DWORD
		Lo Byte	0x78	BYTE		
	0xFFF3	Hi Byte	0x12			
		Lo Byte	0x34			
	iR-ETN sequentially writes data into 0xFFF0~0xFFF3. Data will be written to iR-PU01-P when iR-ETN writes data into 0xFFF3.					
Read Object	0xFFF4	Index				
	0xFFF5	Sub-index (High Byte) Length (Low Byte)				
	0xFFF6	Hi Byte	0x56		WORD	DWORD
		Lo Byte	0x78	BYTE		
	0xFFF7	Hi Byte	0x12			
		Lo Byte	0x34			
	Step1: iR-ETN sequentially writes data into 0xFFF4~0xFFF5. iR-ETN will start reading iR-PU01-P object when writing data into 0xFFF5, and the data will be placed in 0xFFF6~0xFFF7. Step 2: Read data of 0xFFF6~0xFFF7 Object.					

6.3.3 NMT Control Address 0xFFF8(65528)

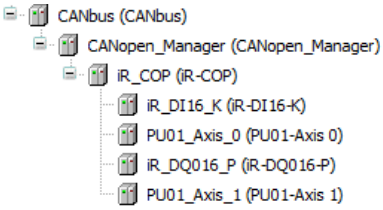
To activate data exchange between Axis Instance Variable (virtual axis) and iR-PU01-P (real axis), please set NMT control address to 2 (NMT Operation).

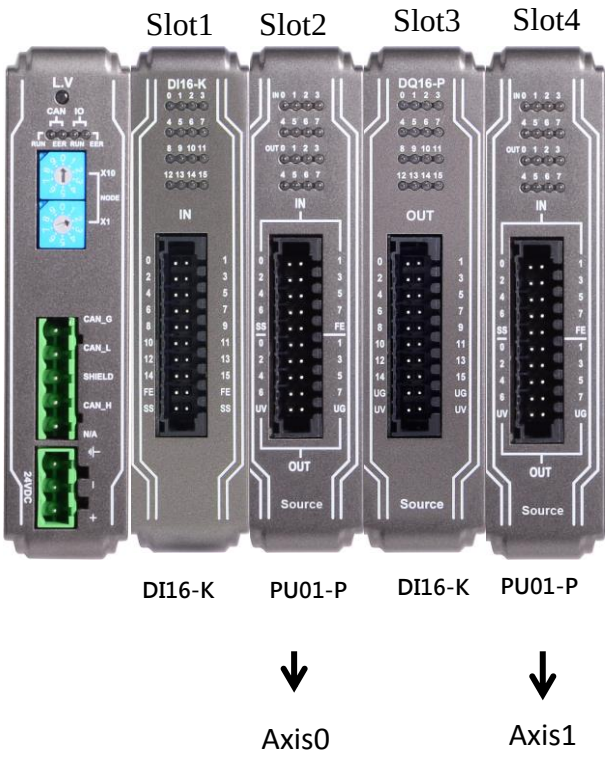
NMT	NMT Stop	0x0001
	NMT Operation	0x0002
	NMT Pre-operational	0x0080
	NMT Reset application	0x0081
	NMT Reset communication	0x0082

6.4 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).


CANbus (CANbus)
CANopen_Manager (CANopen_Manager)
iR_COP (iR-COP)
iR_DI16_K (iR-DI16-K)
PU01_Axis_0 (PU01-Axis 0)
iR_DQ016_P (iR-DQ016-P)
PU01_Axis_1 (PU01-Axis 1)


Slot1
Slot2
Slot3
Slot4
DI16-K
PU01-P
DI16-K
PU01-P
Axis0
Axis1

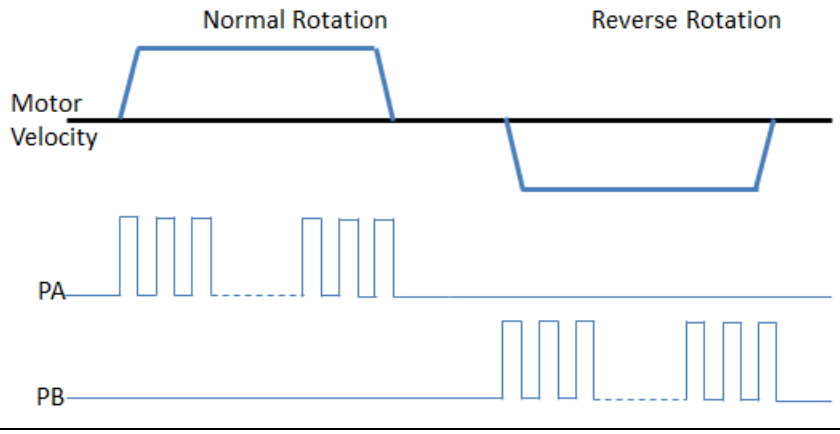
7. Features

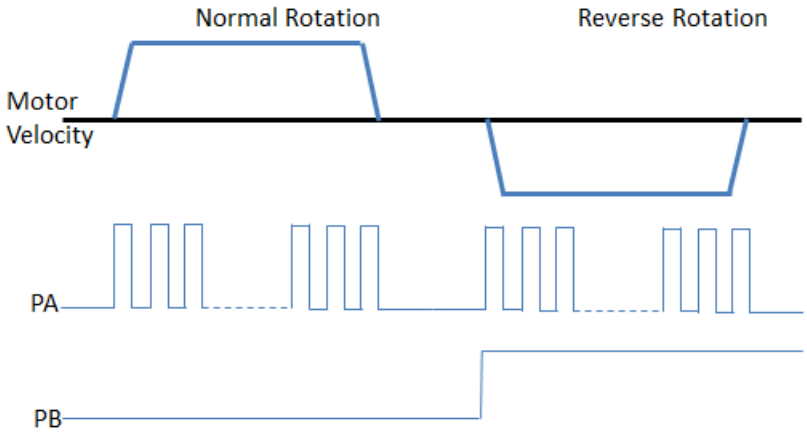
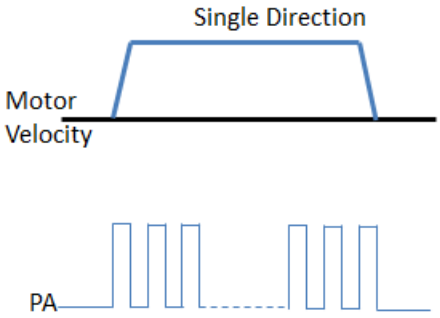
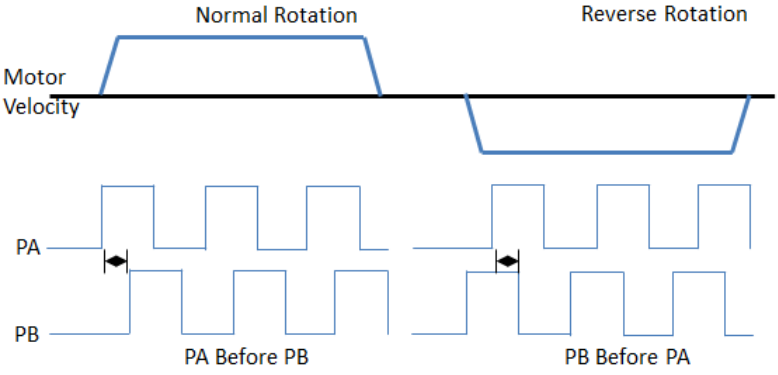
7.1 Feature List

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	Synchronized Motion (CAM)
8	Digital Cam Switch
9	Capture
10	Configurable I/O
11	Motion Control working with I/O Control
12	24V PWM

7.2 High Speed Pulse Output

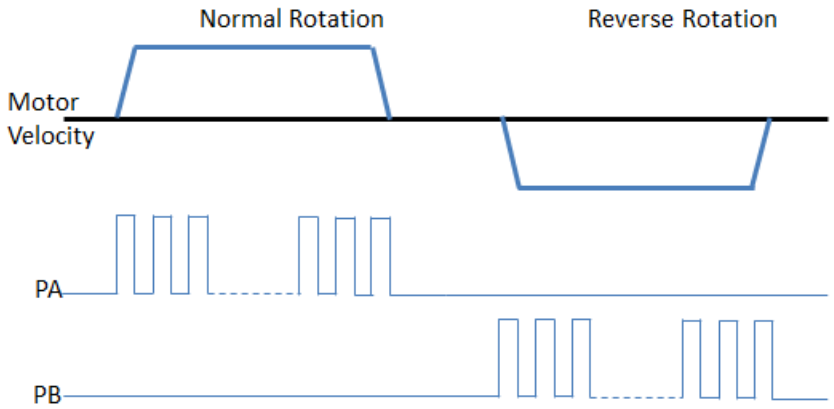
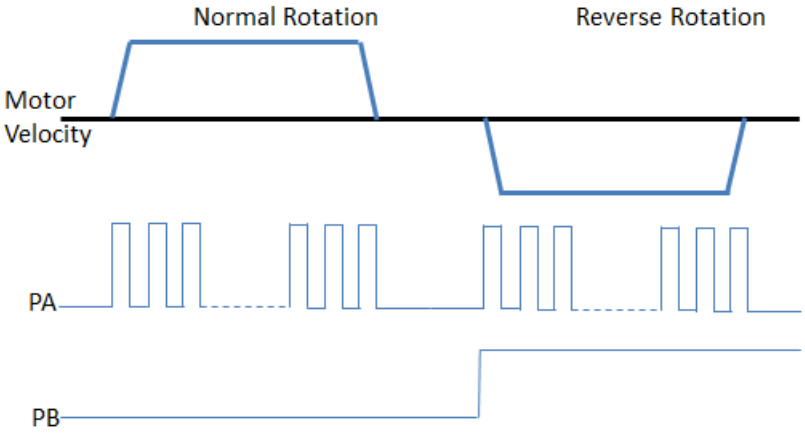
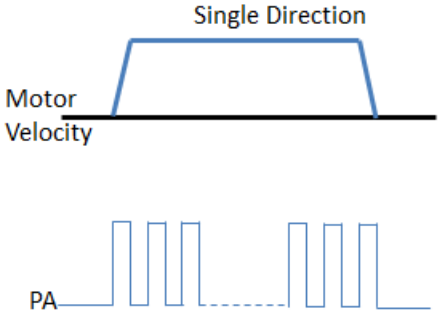
iR-PU01-P can output 2MHz pulses to control the connected servo/step motor (velocity and positioning control). Output modes include CW/CCW, Pulse/Direction, Pulse Only, A/B phase * 1、A/B phase * 2 (4MHz equivalent)、A/B phase * 4 (8MHz equivalent). The output mode is configured using Object Dictionary-Index 0x5511 (Axis 0).

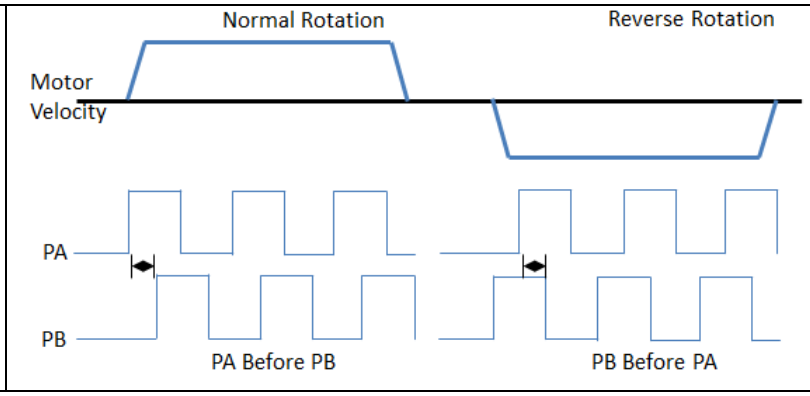
Output Modes	Schematic
CW/CCW	

Pulse/Direction	
Pulse only	
A/B Phase	

7.3 High Speed Pulse Input (Encoder)

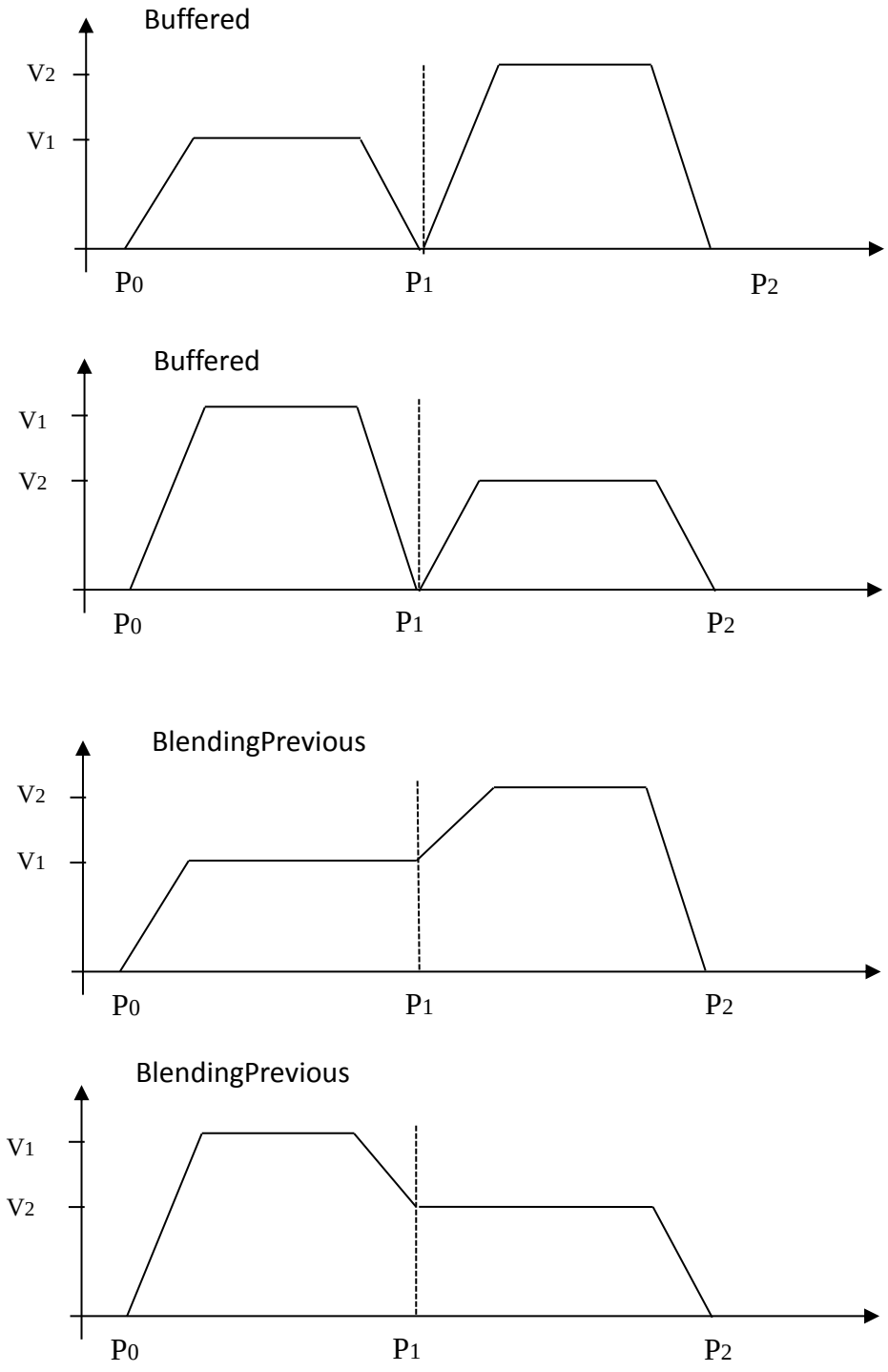
iR-PU01-P can receive up to 2MHz pulse input from the signal output by an encoder or Manual Pulse Generator (MPG). Input modes include CW/CCW, Pulse/Direction, Pulse Only, A/B phase * 1、A/B phase * 2 (4MHz equivalent)、A/B phase * 4 (8MHz equivalent). The input mode is configured using Object Dictionary-Index 0x5501 (Axis 0).

Input Modes	Schematic
CW/CCW	
Pulse/Direction	
Pulse only	

A/B Phase	

7.4 Positioning Control (Buffer Mode Supported)

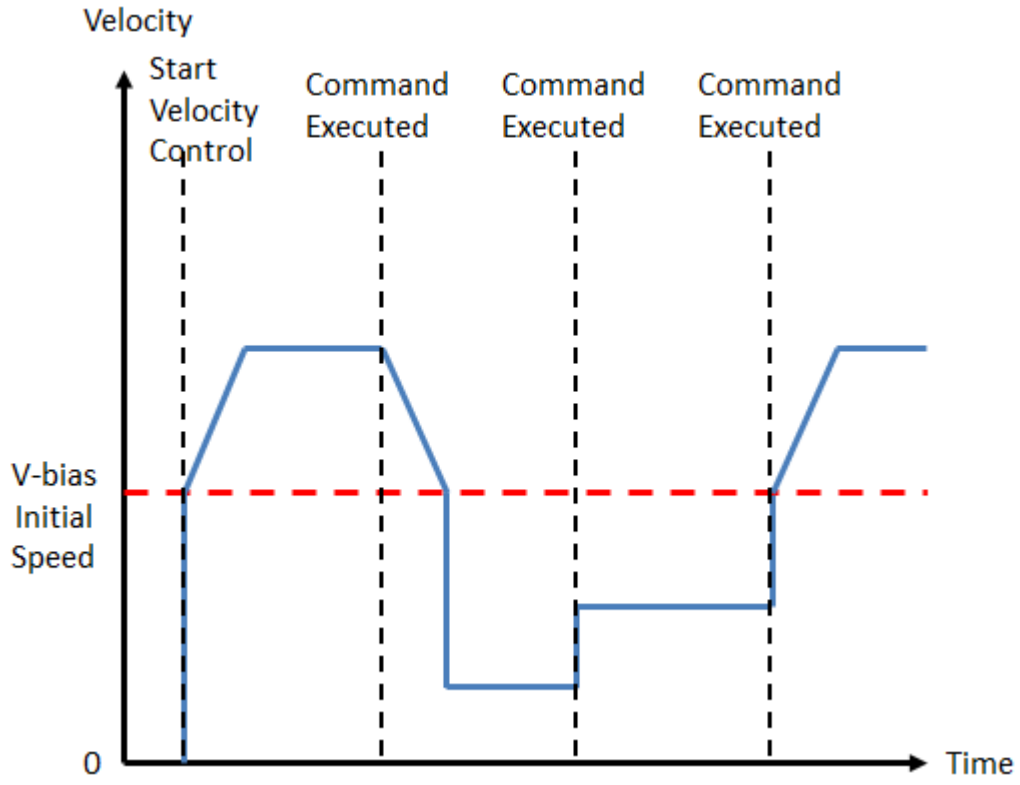
Weintek provides a library of motion control function blocks, and the function blocks relating to positioning control include: MC_MoveAbsolute and MC_MoveRelative, which can perform absolute/relative positioning for the specified target position or for the specified travel distance from current position.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



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 more than 30 homing modes designed according to CiA402.

The following are the references for finding Home.

P = Positive Limit

N = Negative Limit

I = Index

H = Home Switch

The following table shows the references used by each homing method. For more information, see Appendix B in Weintek Library user manual.

After homing process is completed, the Position Actual Value = Home Offset.

No.	P	N	I	H	References for Home
1		*	*		References to Negative Limit and Index.
2	*		*		References to Positive limit and Index.
3			*	*	References to the left switching edge of the Home Switch and Index.
4			*	*	References to the left switching edge of the Home Switch and Index.
5			*	*	References to the right switching edge of the Home Switch and Index.
6			*	*	References to the right switching edge of the Home Switch and Index.

7	*		*	*	Same as No. 3
8	*		*	*	Same as No. 4
9	*		*	*	Same as No. 5
10	*		*	*	Same as No. 6
11		*	*	*	Same as No. 3
12		*	*	*	Same as No. 4
13		*	*	*	Same as No. 5
14		*	*	*	Same as No. 6

The difference between No. 17~30 and No. 1~14 is that for No. 17~30, the home position is not dependent on the Index, and the rest remain the same as No. 1~14.

No.	P	N	I	H	References for Home
17		*			References to the Negative Limit.
18	*				References to the Positive Limit.
19				*	References to the left switching edge of the Home Switch.
20				*	References to the left switching edge of the Home Switch.
21				*	References to the right switching edge of the Home Switch.
22				*	References to the right switching edge of the Home Switch.
23	*			*	Same as No. 19
24	*			*	Same as No. 20
25	*			*	Same as No. 21
26	*			*	Same as No. 22
27		*		*	Same as No. 19
28		*		*	Same as No. 20
29		*		*	Same as No. 21
30		*		*	Same as No. 22
33			*		References to the next Index in the negative direction.
34			*		References to the next Index in the positive direction.
35					References to the current position (without moving).
37					References to the current position (without moving).

No. -35 and -37 are similar to No. 35 and 37. For No. -35, after homing is completed, the position demand value remain the same, and the position actual value is made equal to the position demand value, where in No.-37, the position demand value is made equal to the position actual value.

No.	P	N	I	H	References for Home
-35					References to the current position (without moving).
-37					References to the current position (without moving).

7.7 Synchronized Motion (Gear / MPG)

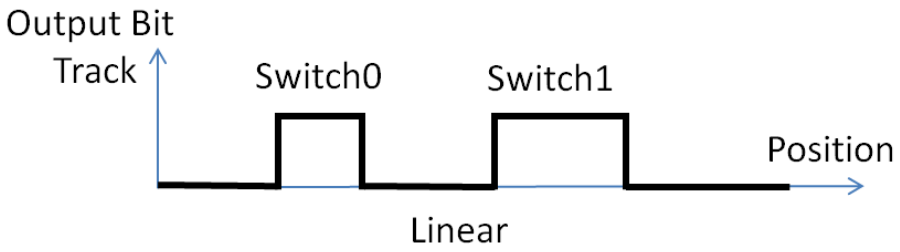
Synchronized motion control is achieved by specifying the ratio between the master axis (Pulse Input) and the slave axis (Pulse Output). Manual Pulse Generator (MPG) is one of the applications.

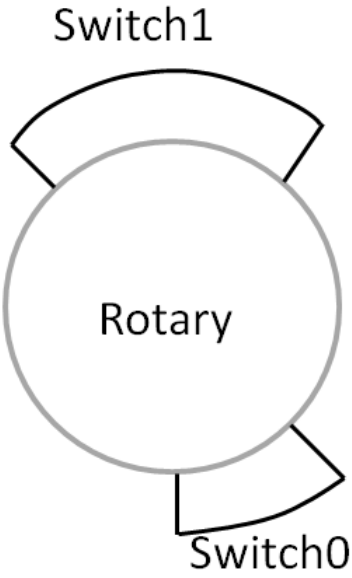
7.8 Synchronized Motion (CAM)

Traditional mechanical cam is used especially in transforming rotary motion into linear motion, and is also capable of non-linear motion and discontinuous motion. Electronic cam can achieve the same, but it has the ability to change motion profiles simply by creating and maintaining a Cam Table, without having to make any mechanical adjustments. With electronic cam, computing complex motion profiles such as interpolated motion is also possible.

7.9 Digital Cam Switch

This feature simulates mechanical cam switch using digital method, which allows settings that can by realized easier in this manner, such as direction and time settings. 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 additional axis) or an actual position (2nd additional axis), and ratio can be configured respectively for each axis. Shown below are a linear axis and a rotary axis that can be defined using Digital Cam Switch.
See settings of 5583h in chapter 8. Users can decide which switches are added to a track, the first and last ON position of the switch, the duration, or whether the switch follows axis direction.





7.10 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.

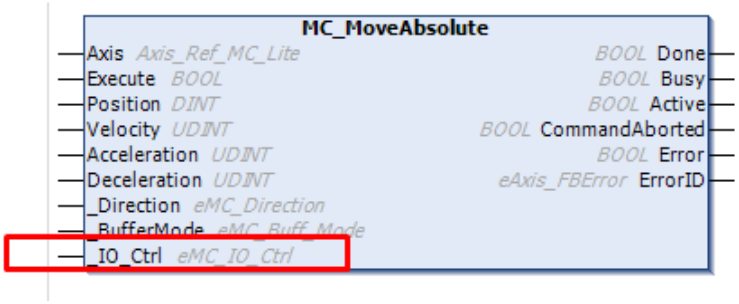
See settings of 5590h ~ 5598h in chapter 8. After completing capture settings (5592h Capture Setting), enable capture for the corresponding channels (5590h Capture Enable), see capture status (5591h Capture Status) to find out if any value is captured by a channel, and then read the captured value (5598h Capture Value) of the channel.

7.11 Configurable I/O

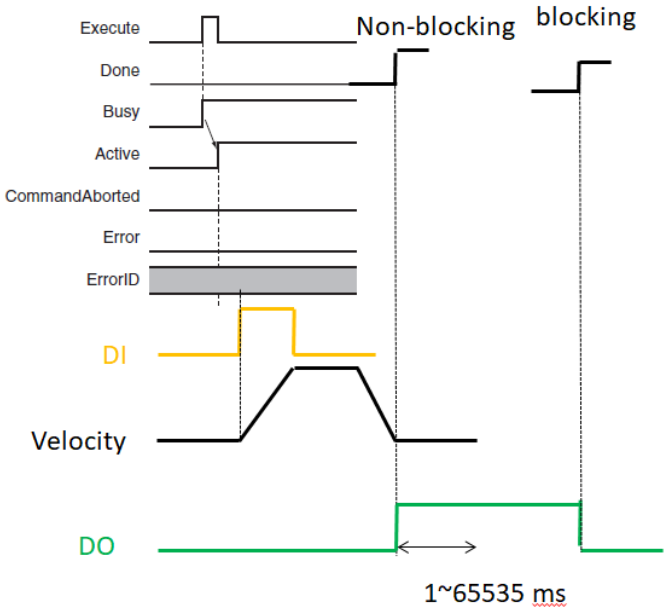
Each I/O of the module, including pulse input/output, has its unique function designed for motion control, but using them as general digital I/O points is possible. See settings of 5503h and 5513h in chapter 8. Although by default Digital Input is used for homing, it can be used as a general digital input point, or for judging whether positioning is completed. Digital Output can be used for PWM output. Digital Input DI-2 can be configured for 24V simple counter.

7.12 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 one or multiple output points to output values at a time.

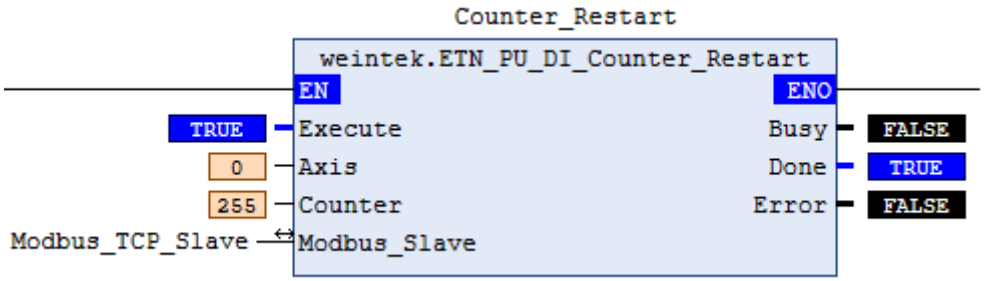


See settings of 558Fh and 559Fh in chapter 8. In 559Fh Motion Trigger Settings, three trigger signals can be pre-defined, and one of them can be used to trigger motion control when function block starts executing. In 558Fh Motion Output Settings, three output modes can be pre-defined, and one of them can be used after the movement is completed. "Done" can be configured either Blocking (digital output completed) or Non-Blocking(movement completed).



7.13 4-Channel 24V High Speed Counter

Starting with firmware V1.03.0, iR-PU01-P can be configured for a 4-channel 24V high speed counter. After selecting counter mode in address 55F0 Module Mode, the DI 0~3 outputs can be used for high speed counter. The related function blocks in Weintek Library can be used in CODESYS to configure the counters and read counter values.



Device.Application.PLC_PRG

Expression	Type	Value
Axis0	weintek.AXIS_REF_...	
_Delay_Cycles	BYTE	0
_CMPT_PV	BOOL	FALSE
_CMPT_PT	BOOL	FALSE
_CMPT_Home	BOOL	FALSE
_Mode_Simple	BOOL	FALSE
Mapping_Q	unAXIS_VAR_OUT	
Torque_Q	stAxis_Torque_Out	
Mapping_I	unAXIS_VAR_IN	
Obj	stAxis_Mapping_In	
Reg	ARRAY [1..12] OF ...	
Counter_Mode	stCounter_Mapping_In	
DI_B0	USINT	4
ModeOpDisp	SINT	1
Statusword	UINT	567
PositionActual	DINT	0
CounterValue_0	UDINT	13421
CounterValue_1	UDINT	13421
CounterValue_2	UDINT	13377
CounterValue_3	UDINT	13376

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
553Fh+n*100h	00h	Sub Error code	USINT	ro	
5540h+n*100h		CAM Motion Settings			
	03h	Moving Average Size	USINT	rw	0h
	04h	MasterOffset	DINT	rw	0h
	05h	SlaveOffset	DINT	rw	0h
	06h	StartMode(Slave Start Direction)	USINT	rw	0h
	07h	EngageMode(Master)	USINT	rw	0h
	08h	EngagePosition(Master)	DINT	rw	0h
		EngageDirection(Master)	USINT	rw	0h
5541h+n*100h	(m=0~50)	CAM Table 0 Settings (i=0~2)			
	1h	Mode	USINT	rw	0h
	2h	Periodic	USINT	rw	0h
	3h	MasterAbsolute	USINT	rw	0h
	4h	SlaveAbsolute	USINT	rw	0h
	5h	Transition Direction(Slave)	USINT	rw	0h
	m+10	Reg m	DINT	rw	0h
5542h+n*100h	(m=0~50)	CAM Table 0 X(Master)			
	m+1	X point NO.m	UDINT	rw	0h
5543h+n*100h	(m=0~50)	CAM Table 0 Y(Slave)			
	m+1	Y point NO m	DINT	rw	0h
5544h+n*100h	(m=0~50)	CAM Table 0 V			
	m+1	V point NO m	REAL	rw	0h
5545h+n*100h	(m=0~50)	CAM Table 0 A			
	m+1	A point NO m	REAL	rw	0h
5546h+n*100h		CAM Table 1 Settings			
5547h+n*100h		CAM Table 1 X(Master)			
5548h+n*100h		CAM Table 1 Y(Slave)			
5549h+n*100h		CAM Table 1 V			
554Ah+n*100h		CAM Table 1 A			
554Bh+n*100h		CAM Table 2 Settings (i=0~2)			
554Ch+n*100h		CAM Table 2 X(Master)			

554Dh+n*100h		CAM Table 2 Y(Slave)			
554Eh+n*100h		CAM Table 2 V			
554Fh+n*100h		CAM Table 2 A			
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
55C0h+n*100h		HW_Counter0			
	01h	Counter Value	UDINT	ro	0h
	02h	Computed Value	UDINT	ro	0h
	03h	Control Bit	USINT	rw	0h
	04h	Initial Value	UDINT	rw	0h
	06h	Computed Mode	USINT	rw	0h
	07h	Sampling Time	UINT	rw	1000
55F0h+n*100h	00h	Module Mode	USINT	rw	0h
55D0h+n*100h~ 55D3h+n*100h		Digital Input High Speed Counter Function			
	01h	Counter Value	UDINT	ro	0h
	02h	Computed Value	UDINT	ro	0h
	03h	Control Bit	USINT	rw	0h
	04h	Initial Value	UDINT	rw	0h
	06h	Computed Mode	USINT	rw	0h
	07h	Sampling Time	UINT	rw	1000

8.1.1 Digital Input：5500h

Sub Index 01h: Input State

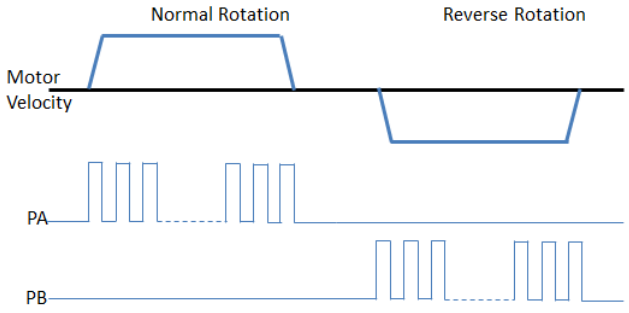
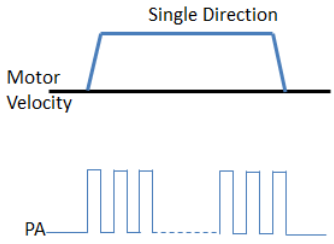
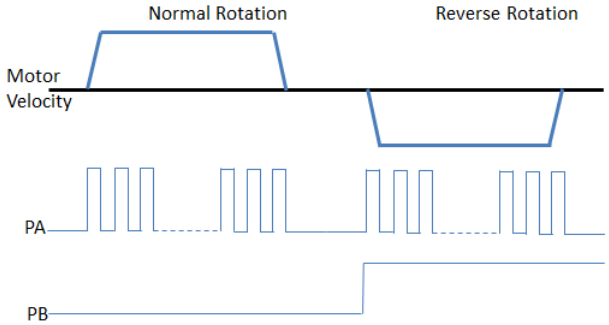
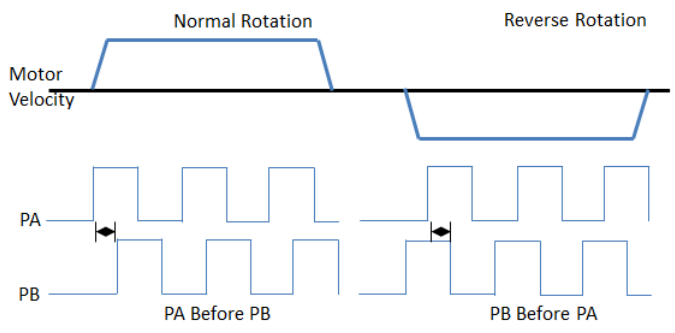
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Z	B	A	DI-3	DI-2	DI-1	DI -0

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="8">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="3"> 
</td></tr>
<tr><td>5</td><td>A(*2)</td><td>B(*2)</td></tr>
<tr><td>6</td><td>A(*4)</td><td>B(*4)</td></tr>
</table>

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 10: Simple Counter
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 Rejected when input pulse duration is less than or equals to pulse duration threshold.
02h	DI 1	
03h	DI 2	
04h	DI 3	
05h	DI A	
06h	DI B	
07h	DI Z	

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

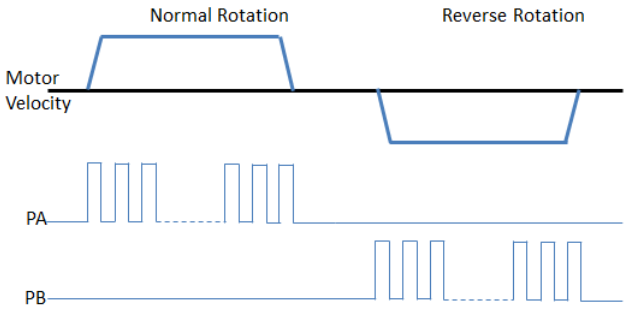
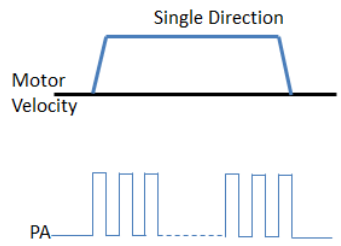
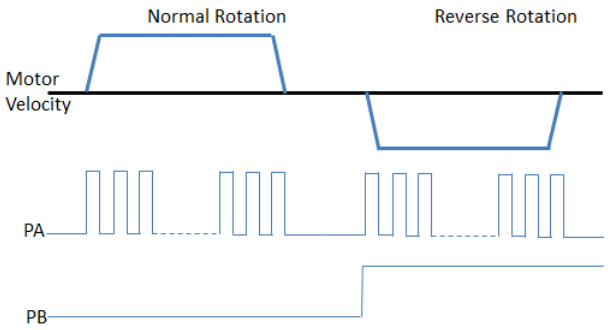
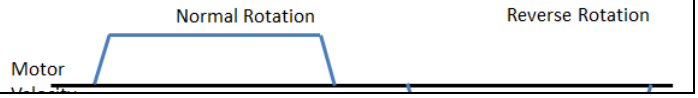
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Reserved	Z	PB	PA	DO-3	DO-2	DO-1	DO -0

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	DO-3	DO-2	DO-1	DO-0

Value 0: non-reverse

Value 1: reverse

8.1.9 Digital Output Function：5513h

Digital Output Functions

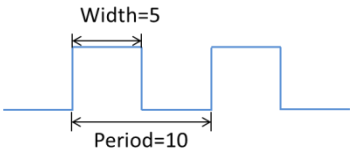
Sub Index	Output Point	Description
01h	DO 0	0:Normal DO 2: PWM0
02h	DO 1	0:Normal DO 2: PWM1
03h	DO 2	0:Normal DO
04h	DO 3	0:Normal DO
05h	PA	0:Normal DO
06h	PB	0:Normal DO 2:PWM1

8.1.10 Digital Output Abort Connection Option：5514h

Sub Index	Output Point	Description
01h	DO 0	0:Off 1:On 2:Keep last value Effective when output function is set to Normal, PWM will be Off , and Axis Pulse will Quick Stop
02h	DO 1	
03h	DO 2	
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 cycle 0~100% 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	

		$$\text{PWM duty cycle} = \frac{Width(us)[High\ word]}{Period(us)[Low\ word]}$$ E.g. To get a result where PWM frequency = 100k and duty cycle = 50%, the setting value is 16#0005000A  To stop PWM output, the duty cycle = 0% and the setting value is 16#0000000A.

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 Sub Error Code：553Fh

Please see Chapter 4.3 in this user manual.

8.1.18 CAM Motion Settings：5540h

Sub Index	Name	Description
03h	Moving Average Size	0~250
04h	MasterOffset	Add offset to the master axis (X axis) when using CAM table.
05h	SlaveOffset	Add offset to the slave axis (Y axis) when using CAM table.
06h	StartMode(Slave Start Direction)	0:Positive 1:ShortestWay 2:Negative 3:Current
07h	EngageMode	0:Instantaneous 1:Master_Distance 2:Master_Position
08h	EngagePosition	Engage position of master axis.
09h	EngageDirection	0:Both 1:Positive 2:Negative

8.1.19 CAM Table 0 Settings：5541h

Sub Index	Name	Description
01h	Mode	0:Line 1:Poly5 2:Mixed
02h	Periodic	0:False 1:True
03h	MasterAbsolute	0:False 1:True
04h	SlaveAbsolute	0:False 1:True
05h	Transition Direction(Slave)	0:Positive 1:Negative
10-60	Reg	When the Mode is Mixed: 0:Line 1:Poly5

8.1.20 CAM Table 0 X (Master)：5542h

Sub Index	Name	Description
1-51	X point 0 ~ 50	Value X of points 0~50 in CAM Table.

8.1.21 CAM Table 0 Y (Slave)：5543h

Sub Index	Name	Description
1-51	Y point 0 ~ 50	Value Y of points 0~50 in CAM Table.

8.1.22 CAM Table 0 V ：5544h

Sub Index	Name	Description
1-51	V point 0 ~ 50	Value V of points 0~50 in CAM Table. (Float)

8.1.23 CAM Table 0 A ：5545h

Sub Index	Name	Description
1-51	A point 0 ~ 50	Value A of points 0~50 in CAM Table. (Float)

8.1.24 CAM Table 1 Settings : 5546h

The number of points in CAM table is 20.

Sub Index	Name	Description
01h	Mode	0:Line 1:Poly5 2:Mixed
02h	Periodic	0:False 1:True
03h	MasterAbsolute	0:False 1:True
04h	SlaveAbsolute	0:False 1:True
05h	Transition Direction(Slave)	0:Positive 1:Negative
10-30	Reg	When the Mode is Mixed: 0:Line 1:Poly5

8.1.25 CAM Table 1 X(Master) : 5547h

The number of points in CAM table is 20.

Sub Index	Name	Description
1-21	X point 0 ~ 20	Value X of points 0~20 in CAM Table.

8.1.26 CAM Table 1 Y(Slave) : 5548h

The number of points in CAM table is 20.

Sub Index	Name	Description
1-21	Y point 0 ~ 20	Value Y of points 0~20 in CAM Table.

8.1.27 CAM Table 1 V : 5549h

The number of points in CAM table is 20.

Sub Index	Name	Description
1-21	V point 0 ~ 20	Value V of points 0~20 in CAM Table. (Float)

8.1.28 CAM Table 1 A : 554Ah

The number of points in CAM table is 20.

Sub Index	Name	Description
1-21	A point 0 ~ 20	Value A of points 0~20 in CAM Table. (Float)

8.1.29 CAM Table 2 Settings : 554Bh

Same as 5546h

8.1.30 CAM Table 2 X(Master) : 554Ch

Same as 5547h

8.1.31 CAM Table 2 Y(Slave) : 554Dh

Same as 5548h

8.1.32 CAM Table 2 V : 554Eh

Same as 5549h

8.1.33 CAM Table 2 A : 554Fh

Same as 5545A

8.1.34 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

8.1.35 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	

8.1.36 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

8.1.37 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-03h	Motion Output Setting0-2	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	

8.1.38 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	

8.1.39 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	

8.1.40 Capture Settings：5592h

The capture interval must be longer than 1ms. When 16#5501 Pulse Input Method is CW_CCW, capture target cannot be set to 2~4.

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)

8.1.41 Capture Value：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	

8.1.42 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	

8.1.43 HW Counter Function: 55C0h

iR-PU01-P's Digital Input DI-2 can be configured for 24V high speed counter (DI-2 Function = 10). When DI-2 is used as high-speed counter, Pulse Input Method(5501h) cannot be set to 1.

Sub Index	Name	Description
01h	Counter Value	Counter value
02h	Computed Value	Computed value
03h	Control Bit	Encoder control bit: Bit-0: Enable Bit-7: Restart (Auto clear)
04h	Initial Value	Initial value
05h	Mode	Reserved
06h	Computed Mode	Pulse computed mode: 0: Speed (Frequency) 1: Difference
07h	Sampling Time	Sampling time, unit: ms (Default: 1000)

8.1.44 Module Mode: 55F0h

iR-PU01-P's module mode must be selected before MC_Power.Status=TRUE.

Sub Index	Name	Description
00h	Module Mode	Select a module mode: 0: Motion (Default) 1: Counter

8.1.45 Digital Input High Speed Counter Function: 55D0h~55D3h

iR-PU01-P's Digital Input DI-0~DI-3 can be configured for 24V high speed counter. Pulse Input / Output may not be used when DI-0~DI-3 are used for high-speed counter.

Sub Index	Name	Description
01h	Counter Value	Counter value
02h	Computed Value	Computed value
03h	Control Bit	Encoder control bit: Bit-0: Enable Bit-7: Restart (Auto clear)
04h	Initial Value	Initial value
05h	Mode	Reserved
06h	Computed Mode	Pulse computed mode: 0: Speed (Frequency) 1: Difference
07h	Sampling Time	Sampling time, unit: ms (Default: 1000)

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 6000h-67FFh as AIO and DIO. Please see Cia402's document for more information on Object Dictionary.

On iR-ECAT, the index range is 2000h-3FFFh (offset 4000h from Cia402's)

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	UDINT	rw	1

		-driving shaft revolutions			
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.

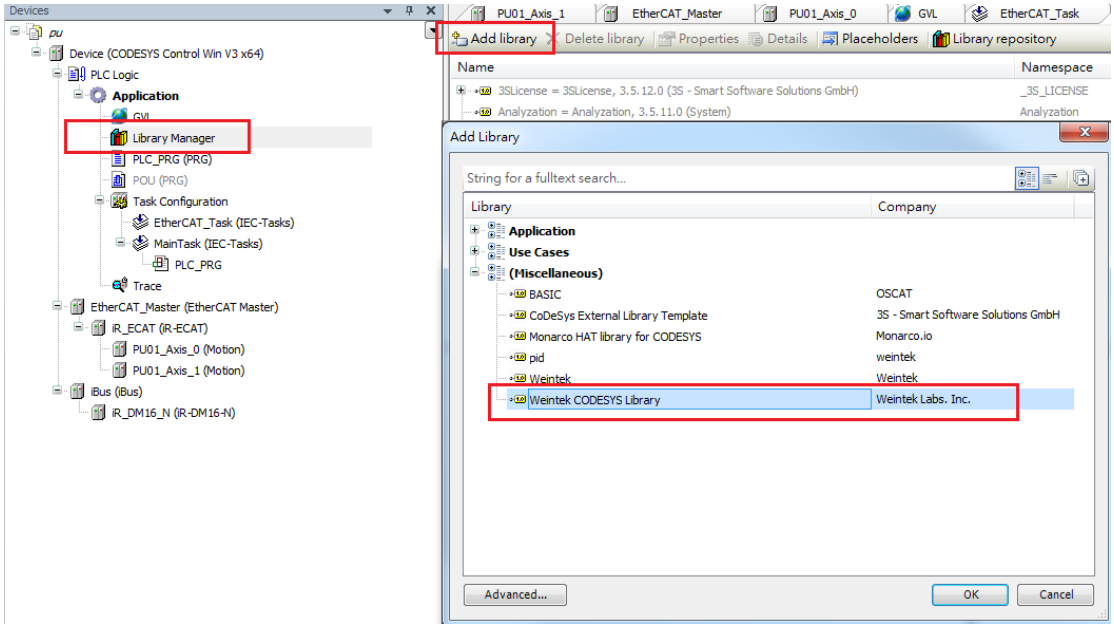
9.1 Motion Control Function Block List

Item	Name	Description
1	AXIS_REF_LITE	Object data type of the 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_Weintek	Specifies the gear ratio between the master axis and the slave axis and starts gear operation.
8	MC_CAM_Weintek	Synchronizes the position of the slave axis with the master axis according to the CAM table.
9	MC_Stop	Forces an axis to decelerate to a stop. Motion instructions can only be given after the axis stops.
10	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.
11	MC_Reset	Clears axis error and make the axis return to Standstill state.

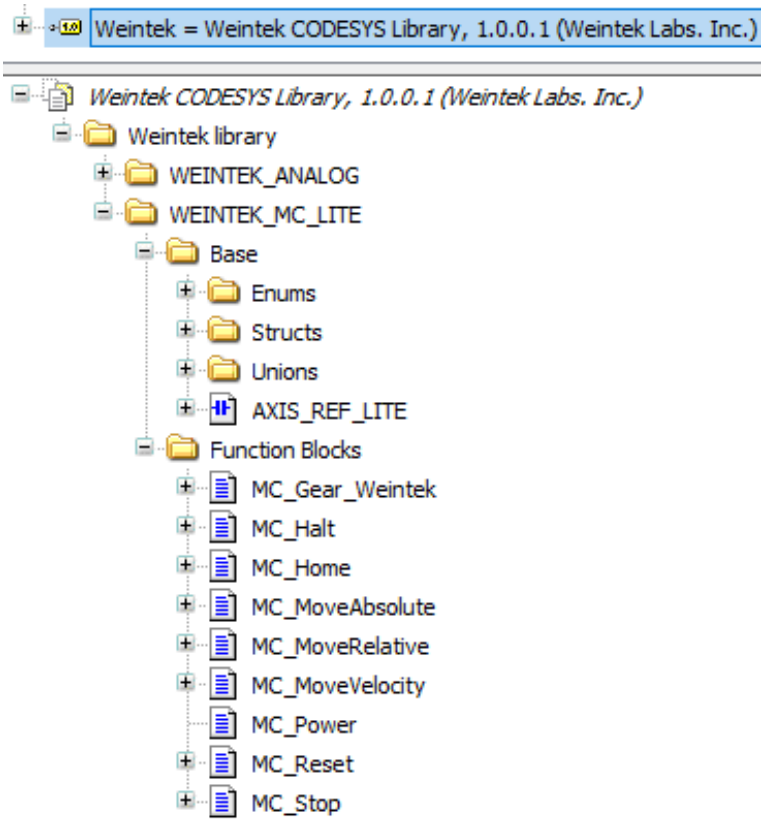
9.2 Download and Install

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.



Step 4. Explanations on the Library can be found in CODESYS software. For more information please see the user manual.

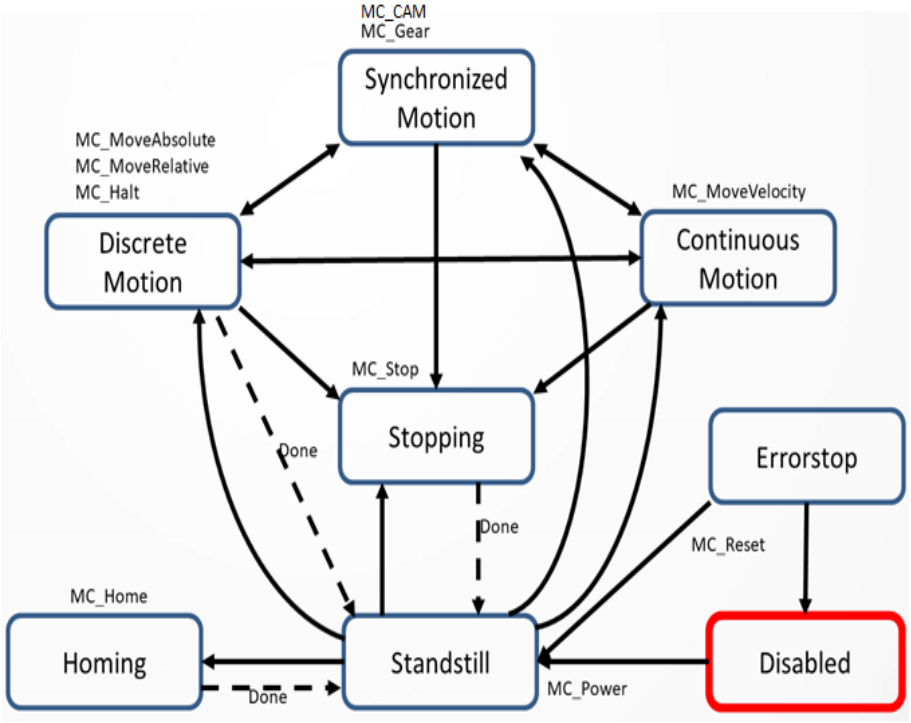
9.3 MC_Status

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 (Axis Variable Instance)

```
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 Creating and Setting an Axis

Creating an axis:

The type of an axis object is AXIS_REF_LITE, its Mapping_Q and Mapping_I will be mapped to iR-PU01-P's I/O as shown below.

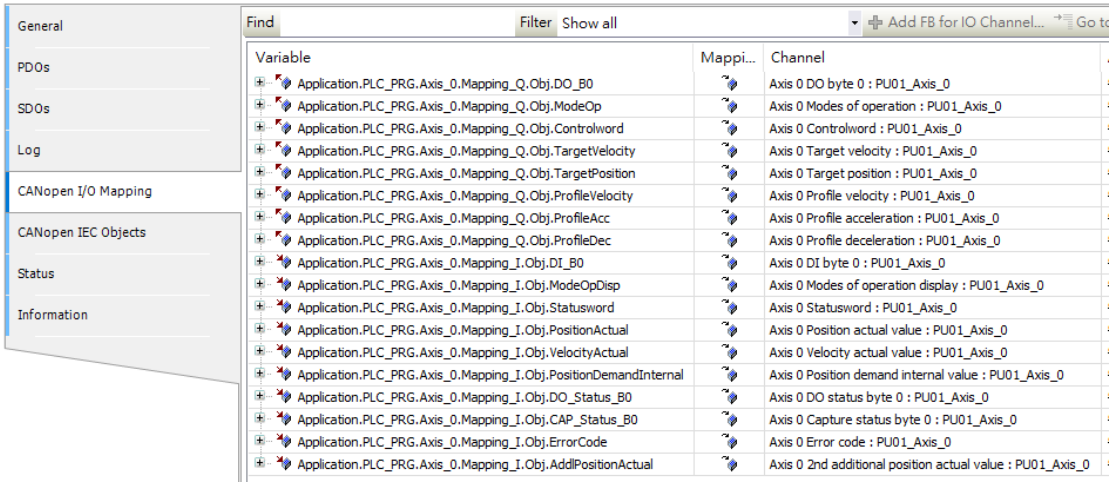
FUNCTION_BLOCK AXIS_REF_LITE

Name	Type	Inherited from	Address	Initial	Comment
_Delay_Cycles	BYTE				
_CMPT_PV	BOOL				
Mapping_Q	unAXIS_VAR_OUT				Axis Output Mappings
Mapping_I	unAXIS_VAR_IN				Axis Input Mappings

In the program create an axis object. Declare variable Axis_0, the type is AXIS_REF_Lite. Variable Axis_0 stands for an axis object and can be used in the program and function block.

AXIS_0: AXIS REF_LITE ;

In the program, map the virtual axis (Variable) to the real axis (Channel). The real axis can be an iR-PU01-P or a servo motor. By mapping their I/O and exchanging data, Axis_0 can be connected to a device in the network and control the device. As shown in the following figure, Axis_0 is connected to the first iR_PU01_P that is connected to an iR-COP in CANopen network.

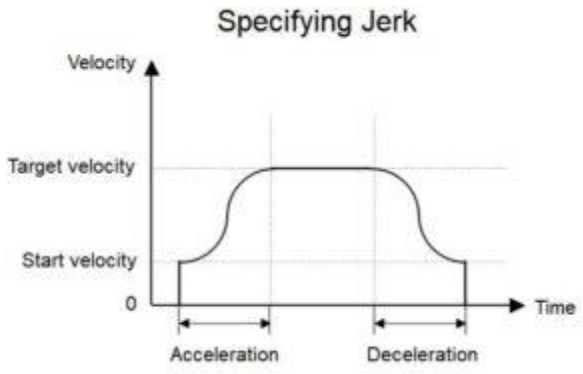


Fundamental Axis settings:

- Pulse Input Method and Pulse Output Method: 5501h and 5511h
- Unit conversion: the user can define the unit of the axis length (e.g. mm, cm…), and the conversion ratio of pulse unit, and then set parameters according to the units.
  - Drive axis: 608Fh, 6091h, 6092h

- Additional encoder axis: 60E6h, 60E8h, 60E9h, 60EBh, 60EDh, 60EEh
  Master axis unit conversion in motion synchronization will use the 2nd additional encoder where Capture and Digital Cam Switch can use both additional encoders.

- Limitations:
  - Hardware Limitation: Use 5503h to set limits or force stop.
  - Software Limitation: 607Dh
  - Quick stop deceleration: 6085h: Decelerate the device to stop when error occurs due to preset limits, forced stop, or exceeded setting limitations.
  - Max. motor speed: 6080h. For an iR-PU01-P, the maximum motor speed means the maximum pulse output frequency. iR-PU01-P can output 2MHz pulses, if the receiver can accept 100Khz in maximum, then 6080h should be set to 100000, so that when exceeding this limit, iR-PU01-P will report error.
  - Max. profile velocity: 607Fh is not the pulse speed. It is the converted user unit of speed.
  - Max. Acceleration: 60C5h is in user unit.
  - Max. Deceleration: 60C6h is in user unit.
  - Jerk and Bias Velocity: 5520h and 5521h. Please see Chapter 7.5 for more information on Bias Velocity, it can reduce resonance when motor is at low speeds. Profile jerk specifies the rate of change in acceleration / deceleration rate. By specifying jerk, the waveform during acceleration / deceleration will be an S-curve, which will reduce the vibration on the machine.



9.5 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

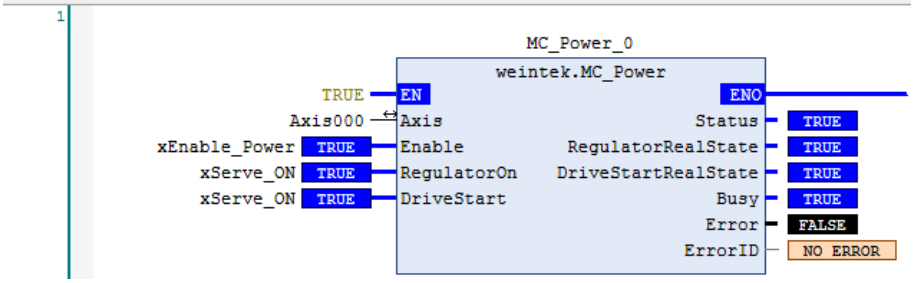
Execute changes from OFF to ON. Execution of the function block will continue until another instruction is executed and interrupts operation.
Enable: When Enable variable is ON, execution of function block continues, and the execution stops when Enable if OFF. Modifications to the parameters during execution are usually effective.

- Motion function blocks other than positioning function block do not have BufferMode specification available, but the behavior is similar to BufferMode's Aborting, which stops executing current instruction and executes a new one.
  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 or an unexpected event 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. Please see Weintek CODESYS Library user manual for more information on when is the time to update parameters.

9.6 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.

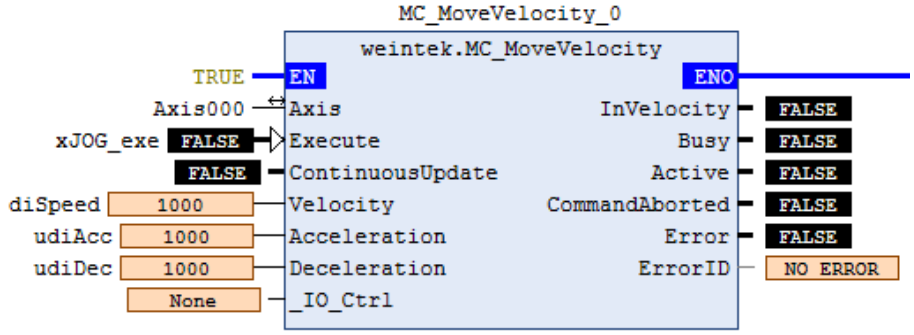
Expression	Type	Value	Prepared value	Address
Axis000	Weintek.Axis_REF_Lite			
_Delay_Cycles	BYTE	0		
_CMPT_PV	BOOL	FALSE		
Mapping_Q	unAXIS_VAR_OUT			
Mapping_I	unAXIS_VAR_IN			
_MC_Status	EAXIS_STATE	Standstill		



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.

9.7 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.

9.8 MC_Home

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

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

6099h#1: Homing at low speed.

6099h#2: Homing at high speed.

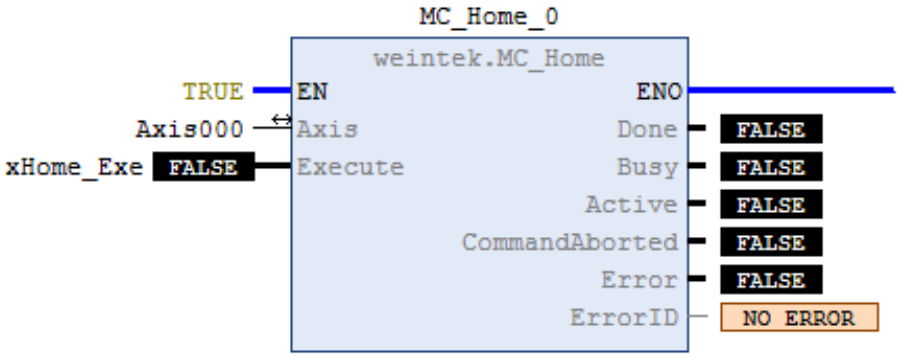
609Ah: Homing acceleration.

607Ch: Home offset.

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

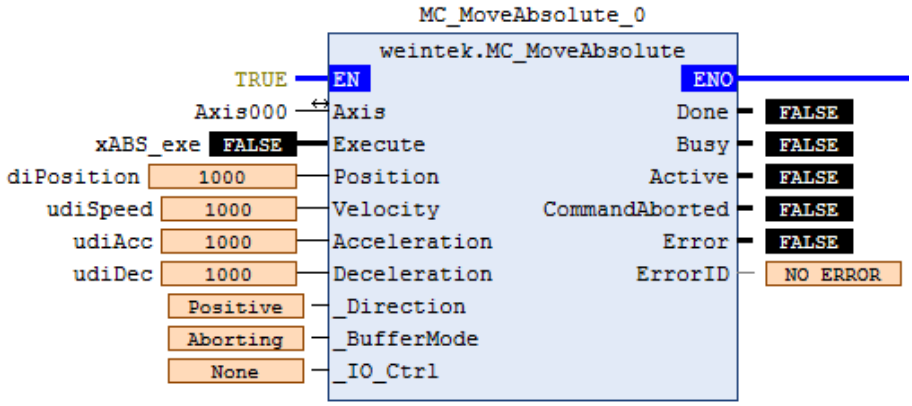
Line	Index:Subindex	Name	Value	Bit length
1	16#6098:16#00	Axis 1 Homing method : PU01_Axis_1	27	8
2	16#6099:16#01	Axis 1 Speed during search for switch : PU01_Axis_1	2000	32
3	16#6099:16#02	Axis 1 Speed during search for zero : PU01_Axis_1	10000	32
4	16#609A:16#00	Axis 1 Homing acceleration : PU01_Axis_1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



9.9 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.

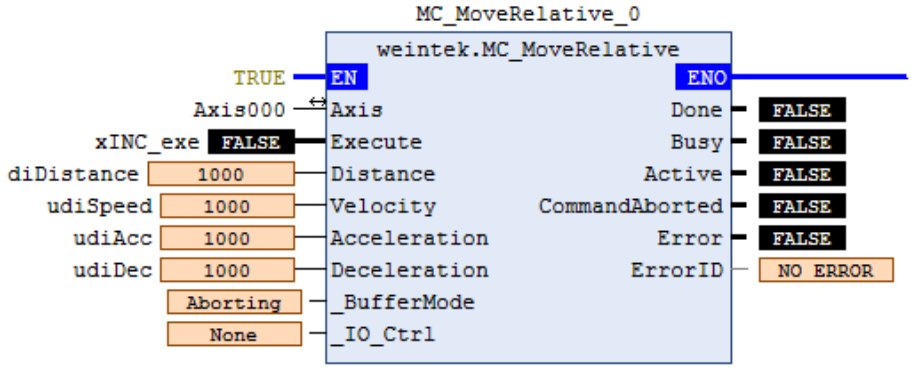
Direction: Specify the direction and the shortest path.

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.

9.10 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.



Distance: 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.

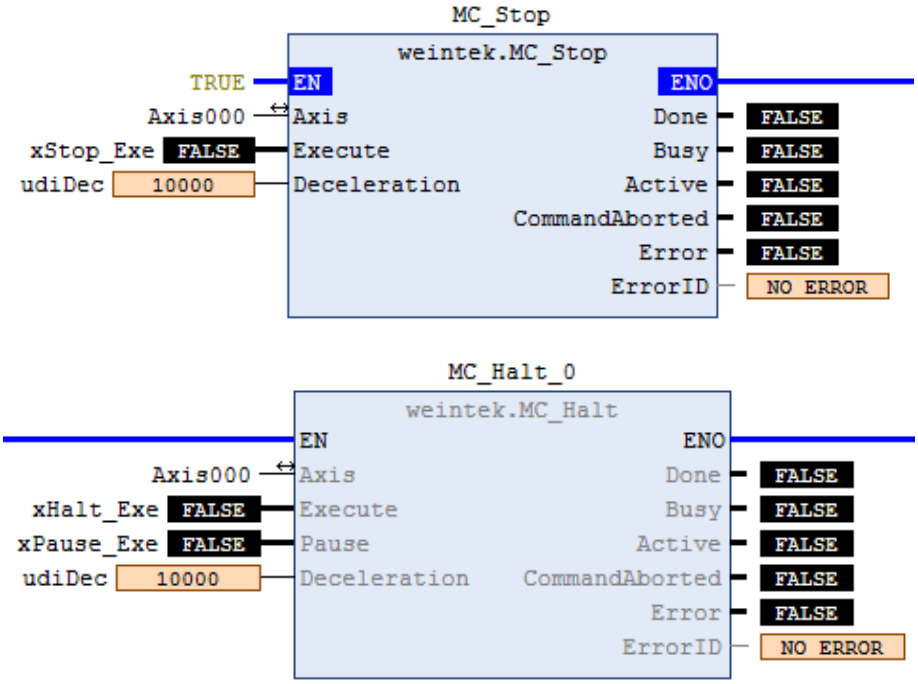
BufferMode: Continuously executes the next instruction after the ongoing motion is completed.

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

9.11 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.



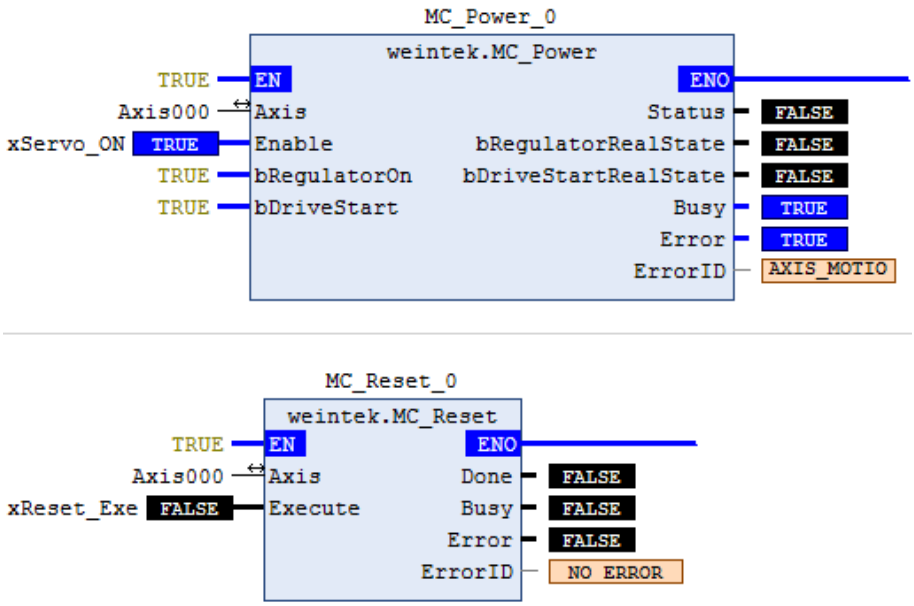
Deceleration: Specify the deceleration rate, the value cannot be 0.

The axis enters Standstill state after it stops.

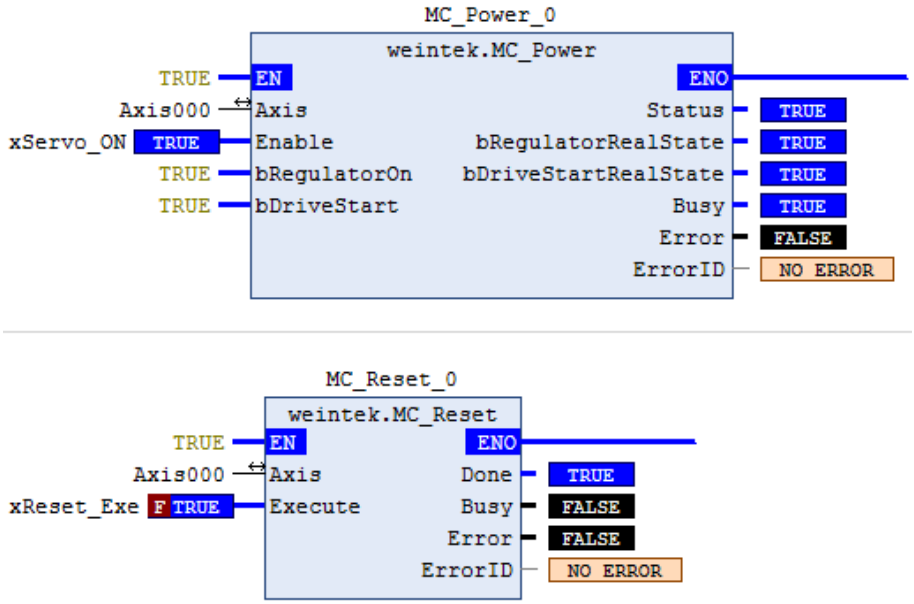
9.12 MC_Reset

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.



9.13 MC_Gear_Weintek(MPG)

MC_Gear_Weintek(MPG) function block obtains output pulse (slave axis) by multiplying the input pulse (master axis) by electronic gear ratio.

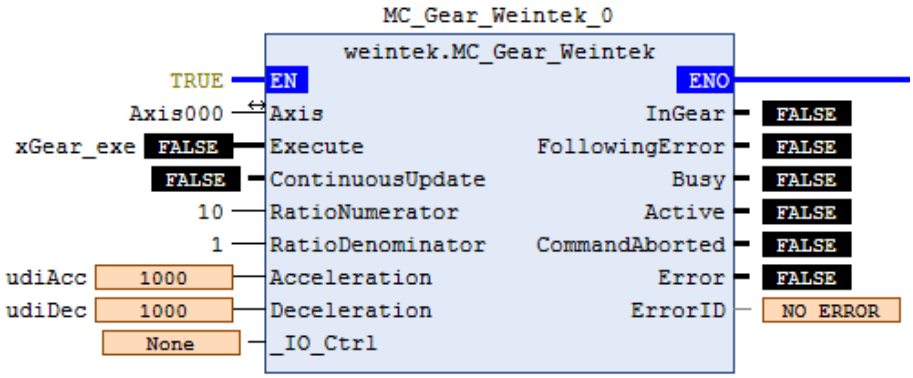
The master axis encoder must be configured using 5501h, and MPG can also be a source of input pulses.

Pulse Input Method 5501h:

Bit 4: 0 (drive axis encoder), 1 (master axis encoder), please set Bit 4 to 1 (master axis encoder) when using MPG.

Bit 0~3: Please see chapter 8.1.2 in this manual.
Convert the input pulse of the 2nd additional encoder in 60E6h, 60E8h, 60E9h, 60EBh, 60EDh, and 60EHh to master user unit.

The following parameters are used when executing MC_Gear_Weintek.



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.

Slave user unit = Master user unit $* \frac{RatioNumerator}{RatioDenominator}$

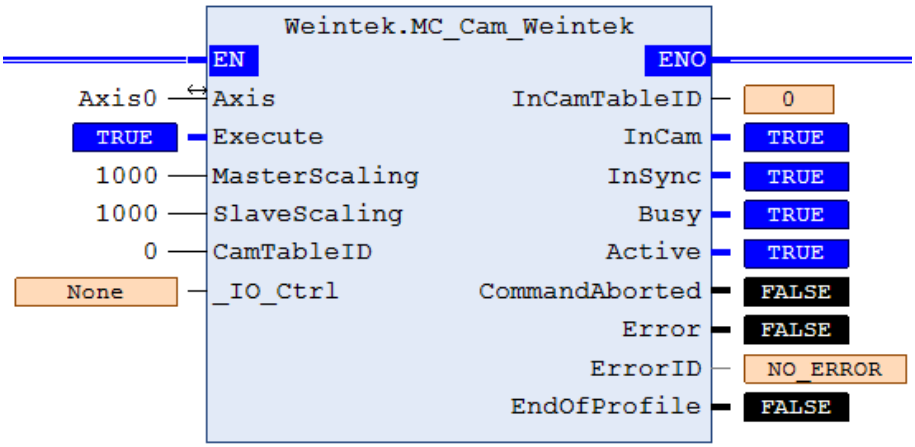
IO_Ctrl: Trigger execution using digital input.
The rest of the settings of MC_Gear_Weintek can be found in: Gear Motion Setting：5530h.
Master Direction Limit: Specify the effective direction of master axis.
Slave(PU) Direction Limit: Specify the effective direction of slave axis.
Moving Average: Filter unstable input pulses.
Following Error: It is TRUE when the window size is exceeded for a period of time that reaches time out. By default it is disabled (where both window size and time out are 0).

9.14 MC_CAM_Weintek

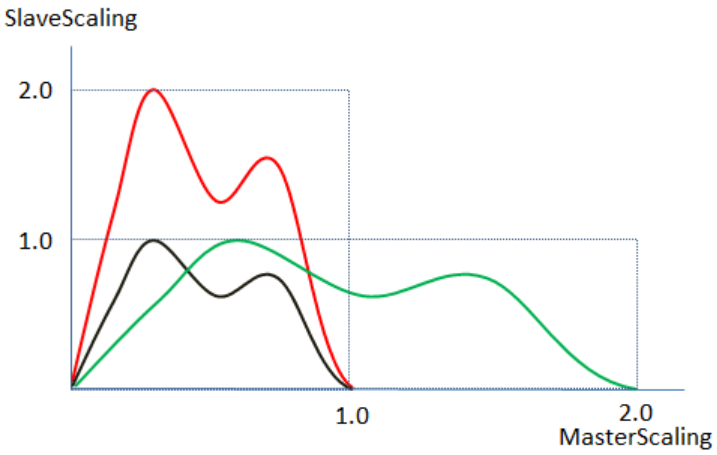
MC_CAM_Weintek function block obtains output pulse (slave axis) by converting input pulse (master axis) according to the CAM Table created by the user.
The master axis encoder must be configured using 5501h

Pulse Input Method 5501h:
Bit 4: 0 (drive axis encoder), 1 (master axis encoder), please set Bit 4 to 1 (master axis encoder) when using Electronic CAM.
Bit 0~3: Please see chapter 8.1.2 in this manual.

Convert the input pulse of the 2nd additional encoder in 60E6h, 60E8h, 60E9h, 60EBh, 60EDh, and 60EHh to master user unit.



MasterScaling: The phase of the master axis is extended or contracted by using the specified scale. The unit is 1/1000.
SlaveSclaing: The displacement of the slave axis is extended or contracted by using the specified scale. The unit is 1/1000.
CamtableID: Specify the CAM Table to be used by its number (0~2)
IO_Ctrl: Trigger execution using digital input.
In MC_CAM_Weintek function block, MasterScaling adjusts the scale of the horizontal axis while SlaveScalign adjusts the scale of the vertical axis.



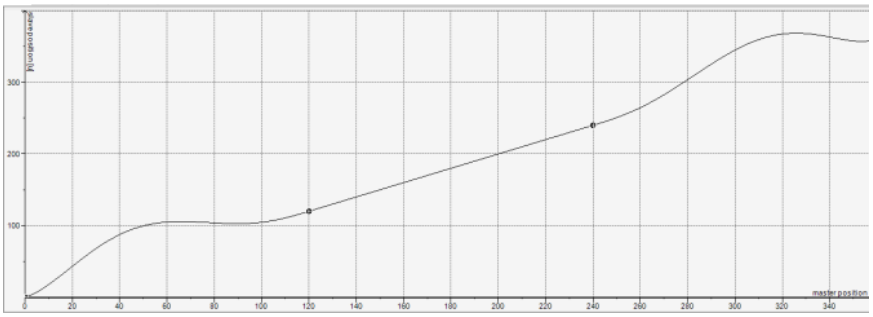
For iR-PU01-P, three CAM Tables can be used: 5541h-5545h, 5546h-554Ah, and 554Bh-554Fh.

When MC_CAM_Weintek function block is executed, the user can decide which CAM table is to be used, or change to another CAM Table at the beginning of the next cam cycle. Similar to CODESYS, the table is created by defining the X, Y coordinates of the points (5542h, 5543h) and then select the way to link the points from Line or 5th Degree Polynomial (5541h). If the later one is selected, the velocity (5544h) and the acceleration rate (5545h) of the point can be defined.

Please do the following things first before using MC_CAM_Weintek.

(Axis 0's object address used in the following example).

A. Create CAM Table



Motion calculations:

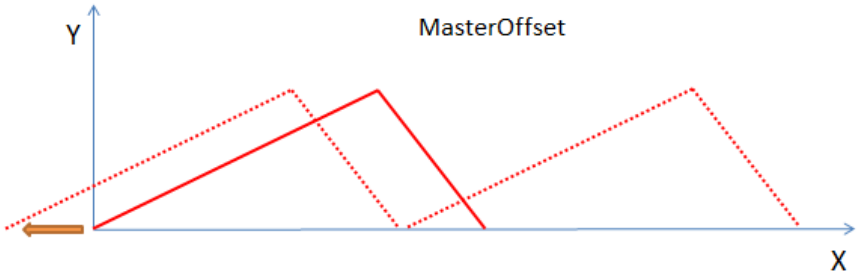
- Position of Master axis X (5542h, 5547h, 554Ch):
  X = ((MasterPosition / MasterScaling) + MasterOffset) % (CAM profile length)
  If the point is out of the range of CAM, then take the remainder of the Master max position.

- Position of Salve axis Y (5543h, 5548h, 554Dh):
  Y = CAM(X)
  SlavePosition = (Y x SlaveScale) + SlaveOffset

B. Set CAM motion parameters (5540h)

- Engage master and slave axes to CAM motion:
  Set EngageMode, EngagePosition, and EngageDiretion to engage the master axis so that it moves in the specified direction to an absolute or relative position.
- Offset:
  MasterOffset / SlaveOffset: Offset the CAM table on the master (X) / slave (Y) side.
  A positive master offset shifts the entire camming profile in the negative direction, and a negative master offset shifts the camming profile in the positive direction.
  As shown in the following figure, when MasterOffset is positive, it shifts the

entire camming profile in the negative direction (red solid line moves to red dotted line).



SlaveOffset shifts the camming profile upwards or downwards.

As shown in the following figure, when SlaveOffset is positive, it shifts the camming profile upward.

C. Set CAM Table (5541h, 5546h, 554Bh)

- Mode: The way to link the points.

  0: Line (default). Link the points using straight lines to form the camming profile.

  1: 5th Degree Polynomial. Form a curve line as camming profile by adjusting velocity and acceleration rate.

  2: Mixed. Straight lines and curves exist in camming profile.

- StartMode: Similar to MC_MoveAbsolute's Direction, this determines the direction in which the axis engages to CAM.

- MasterAbsolute and SlaveAbsolute:

  If MasterAbsolute is TRUE, then the CAM is started at the current master position.

  If MasterAbsolute is FALSE, then the CAM is relocated to the current position. The zero point of the master is also shifted to the current master position.

  If SlaveAbsolute is True, when starting a new cycle, the cam is evaluated independent of the current position of the slave. This can lead to jumps if the slave position to the master start position deviates from that of the master end position.

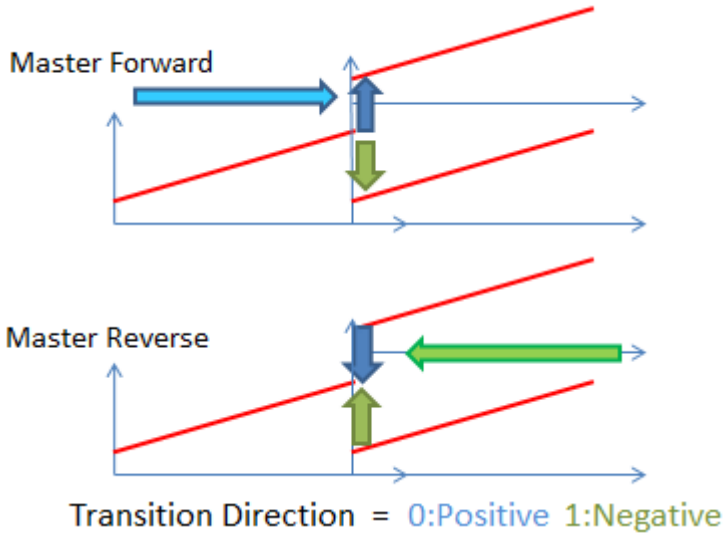
  If SlaveAbsolute is False, then the new CAM is started allowing for the current slave position. The position that the slave has after the end of the previous cycle is added as a slave offset to the new evaluations of the CAM. Jumps can also

occur if the slave position at the master start position is not 0.

The following figures show how the camming profile is shifted:

- Transition Direction:

  Under SlaveAbsolute mode, the value at the end point of the first CAM may not agree with the value at the start point of the second CAM, and the transition direction changes accordingly.



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 9.2 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]

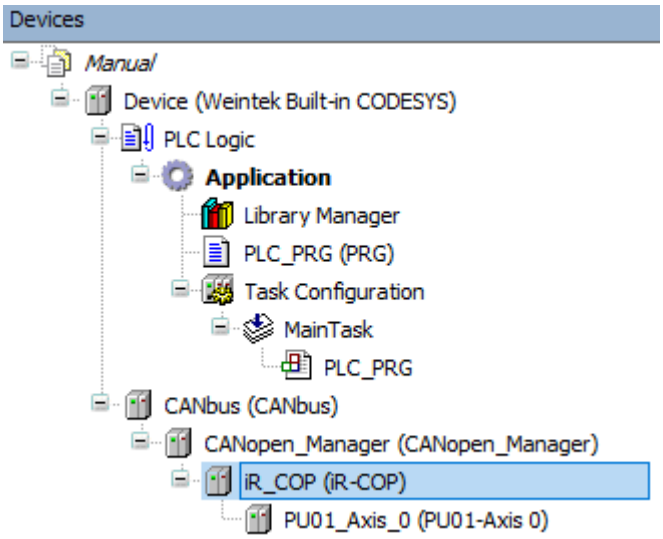
Add iR-COP coupler:

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

Add iR-PU01-P module:

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

Instead of doing the steps above, PLCopen_XML which is created in advance can be imported, and settings can be copied from other project files.

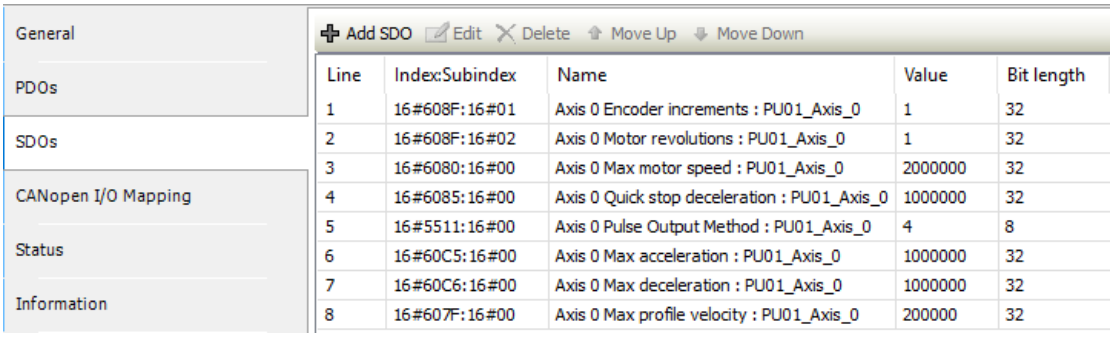


10.3 Configuring Motion Control Parameters

[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

Basic and necessary parameters will be marked with a * sign for users to find and configure them first.



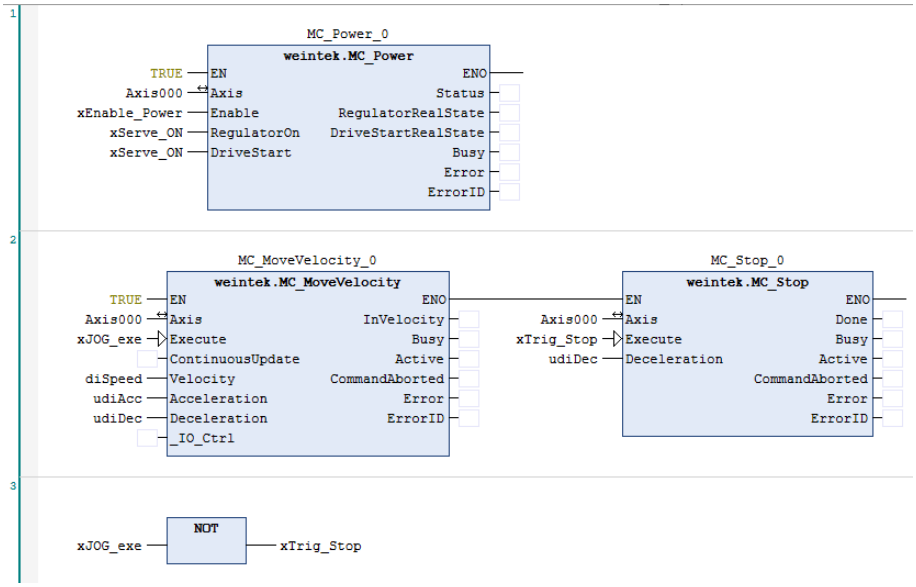
The above parameters should be configured in advance for the modules to perform motion control properly.

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

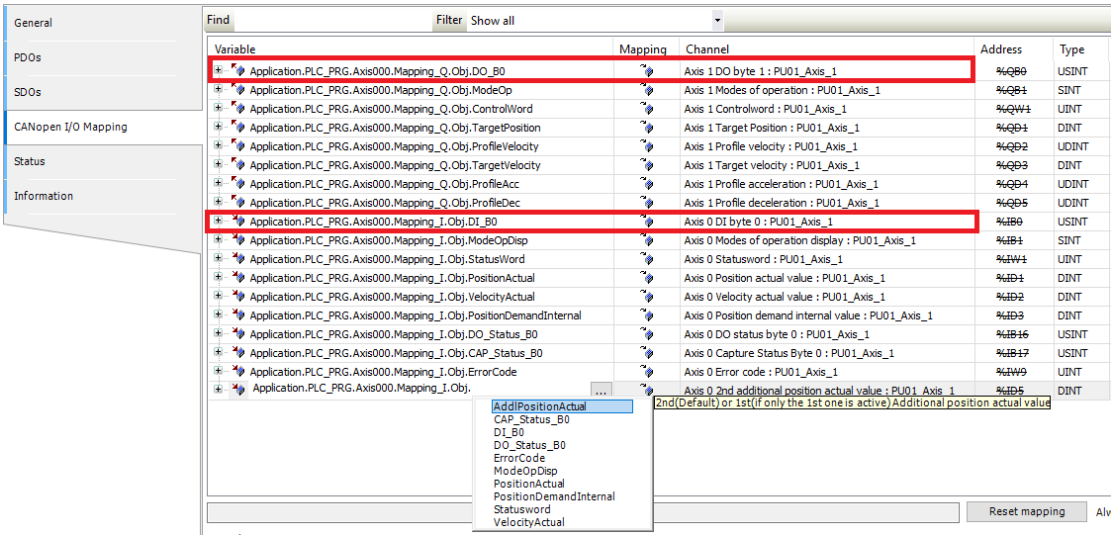
MC_MoveVelocity: Performs velocity control.

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:



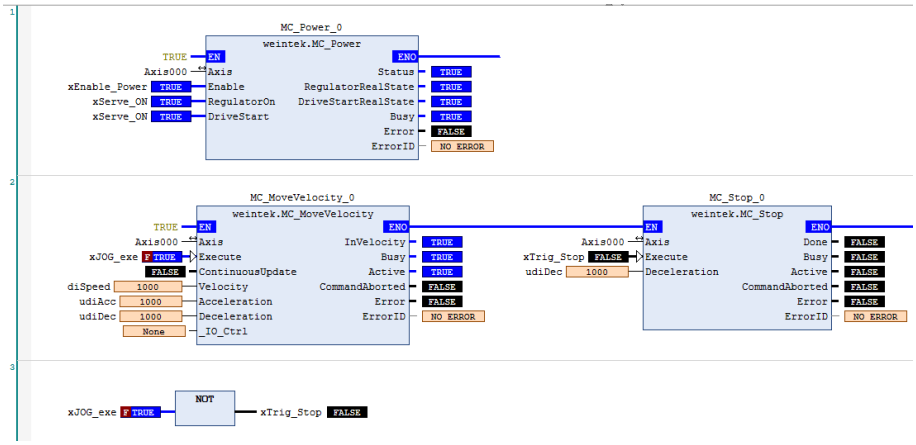
Please make sure that the Variable is identical to the Channel, and Mapping_Q should be completely mapping to Mapping_I.

If the settings were imported or pasted before, then use Replace Active Editor to

replace the names.

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.

11. Setting iR-PU01-P in CODESYS PLCopenXML

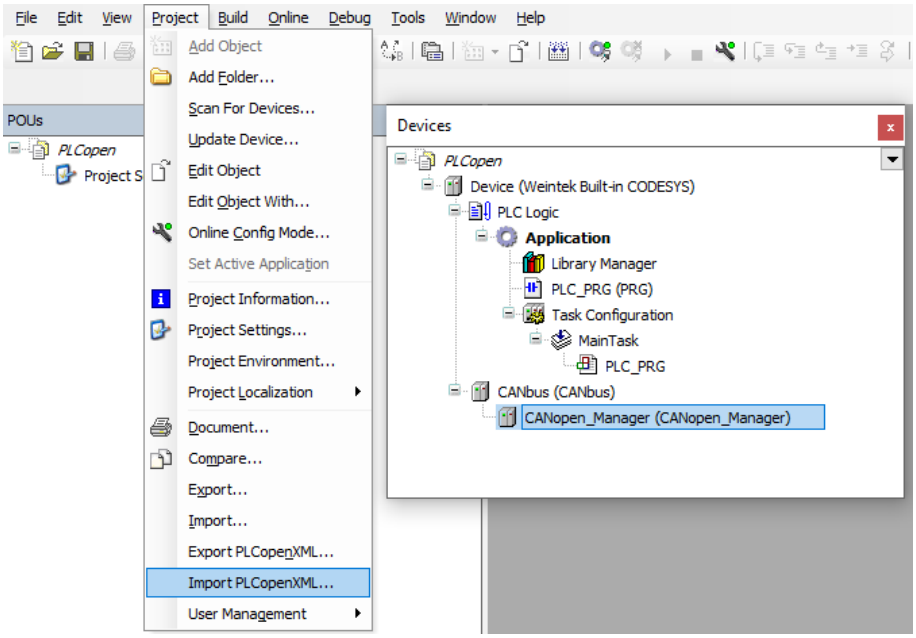
11.1 Install and Add Weintek Library

Please see Chapter 9.2 in this manual to download and install Weintek Library.

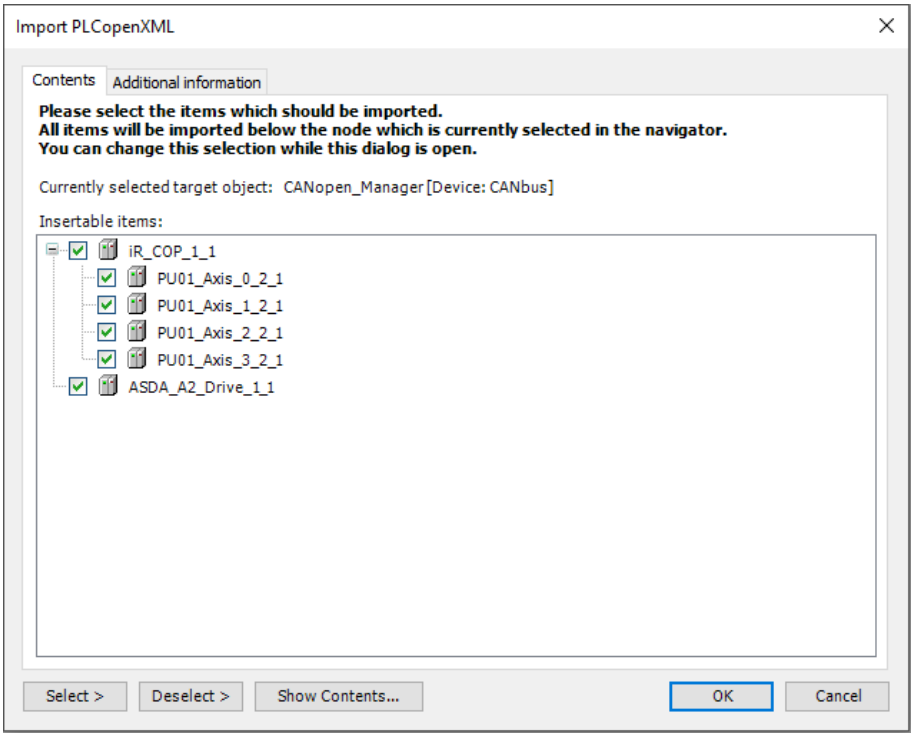
Path: C:\Users\PC\Weintek CODESYS and Remote IO\PLCopen Template

11.2 Import PLCopenXML

Click [CANopen_Manager] and then select [Project] » [Import PLCopenXML].



Import [Weintek_Axis_Template] file.



※EtherCAT_Master can import PLCopenXML in the same way.

12. Configuration Steps of iR-PU01-P's PWM

PWM output can be done using iR-PU01-P's DO.0 & DO.1, or PB.
The maximum frequency for DO.0 and DO.1 is 100kHz.
The maximum frequency for PB is 500kHz.
Users can dynamically adjust PWM parameters by using Weintek_CODESYS_Library and Weintek_iBus_Library function blocks.

12.1 Setting Digital Output Function

Set Digital Output Function (5514h) as shown below.
Set the highlighted functions to 2 for PWM output.

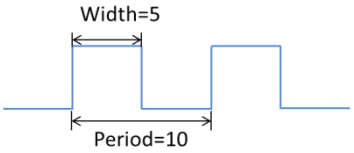
O-Pulse Output Method*	USINT	0
O-Digital Output Polarity	UDINT	0
O-Digital Output 0 Function	USINT	2
O-Digital Output 1 Function	USINT	2
O-Digital Output 2 Function	USINT	0
O-Digital Output 3 Function	USINT	0
O-Digital Output PA Function	USINT	0
O-Digital Output PB Function	USINT	0
O-DO 0 abort connection option	USINT	0
O-DO 1 abort connection option	USINT	0

12.2 Setting PWM Parameters

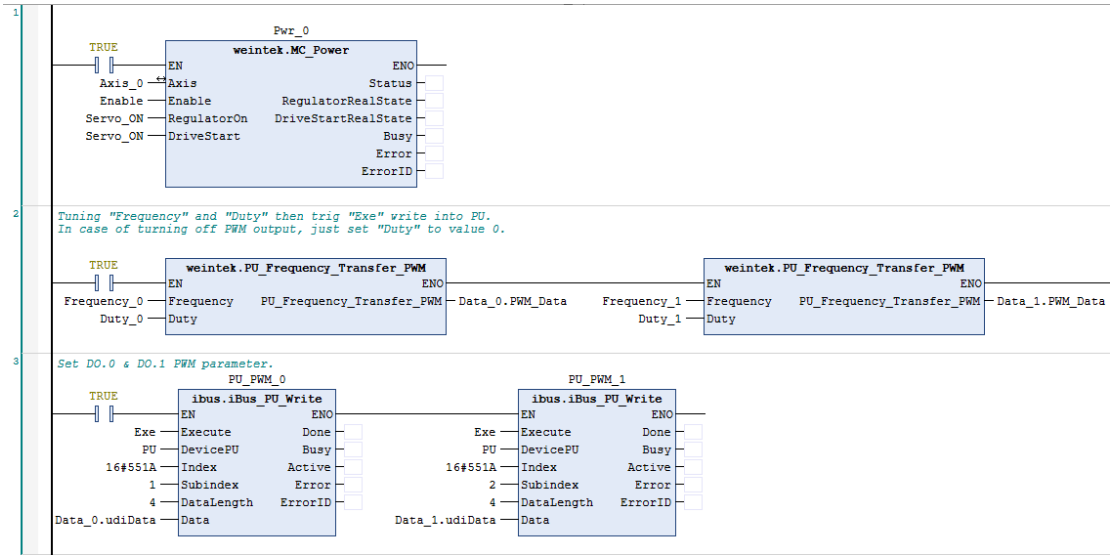
Please see Object Dictionary to find explanation for settings below.

$$\text{PWM duty cycle} = \frac{Width(us)[High\ word]}{Period(us)[Low\ word]}$$

E.g. To get a result where PWM frequency = 100k and duty cycle = 50%, the setting value is 16#0005000A



Frequency_Transfer_PWM function block can be used to directly convert input value to PWM parameter as shown below.



The result from Frequency_Transfer_PWM is written to the object – PWM Output Setting.

Parameters can be written to iR-PU01-P using the following function blocks:

1. cMT-CTRL01 - iBus_PU_Write function block in Weintek_iBus_Library
2. iR-COP – SDO_WRITE_DATA function block.
3. iR-ECAT – ETC_CO_SdoWrite function block.

12.3 Starting PWM Output

After writing PWM parameters and execute MC_Power, DO.0 and DO.1 will immediately output PWM signal.

※Please note that PWM output will be stopped when duty cycle = 0.

13. 4-Channel 24V High Speed Counter

Starting with firmware V1.03.0, iR-PU01-P can be configured for a 4-channel 24V high speed counter. When doing so, A/B/Z pulse input and PA/PB pulse output will turn into general input / output; therefore, they may not be used for pulse input / output. Below is an example showing how to configure iR-ETN+iR-PU01-P for a 4-channel 24V high speed counter.

13.1 Setting Module Mode

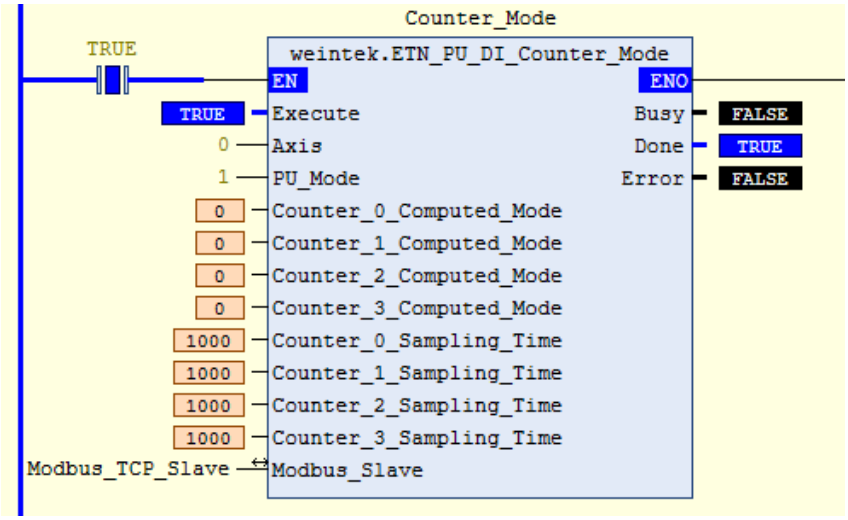
iR-PU01-P's module mode can be set in object dictionary: Index=55F0h, SubIndex=00h. The 24V counter may be used when module mode is set to 1.

Settings:

iR-ETN:

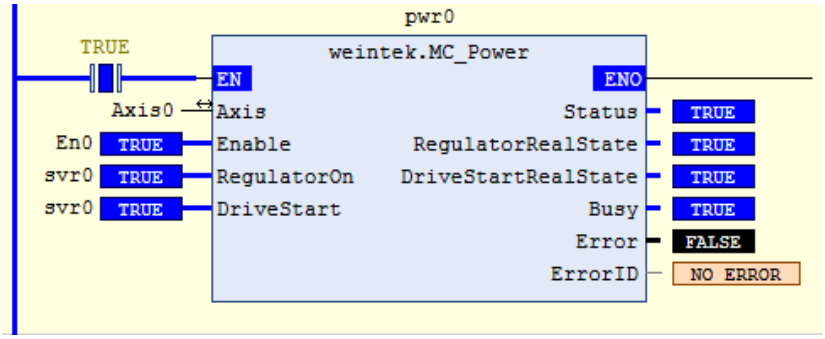
ModbusTCPSlave I/O Mapping: Select Mapping_I.Reg and Mapping_Q.Reg.

In the program, execute ETN_PU_Counter_Mode function block to set iR-PU01-P's counter mode.



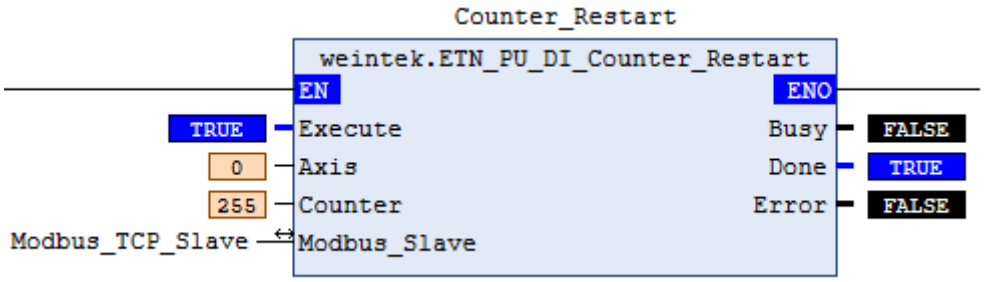
13.2 Power ON

Execute MC_Power function block, when Power.Status=TRUE, iR-PU01-P will be in counter mode.



13.3 Start Counter

Execute Counter_Restart function block to start DI-0~3 high speed counter.



When a pulse is sent, the high speed counter value will be recorded in: AXIS_REF_LITE.Mapping_I.Counter_Mode.CounterValue_0~3.

Device.Application.PLC_PRG

Expression	Type	Value
Axis0	weintek.AXIS_REF_...	
_Delay_Cycles	BYTE	0
_CMPT_PV	BOOL	FALSE
_CMPT_PT	BOOL	FALSE
_CMPT_Home	BOOL	FALSE
_Mode_Simple	BOOL	FALSE
Mapping_Q	unAXIS_VAR_OUT	
Torque_Q	stAxis_Torque_Out	
Mapping_I	unAXIS_VAR_IN	
Obj	stAxis_Mapping_In	
Reg	ARRAY [1..12] OF ...	
Counter_Mode	stCounter_Mapping_In	
DI_B0	USINT	4
ModeOpDisp	SINT	1
Statusword	UINT	567
PositionActual	DINT	0
CounterValue_0	UDINT	13421
CounterValue_1	UDINT	13421
CounterValue_2	UDINT	13377
CounterValue_3	UDINT	13376

Execute Counter_Restart function block again to reset to initial value.