User Manual

QEC-RXXMV3S

EtherCAT Pulse Output module.

Up to 3-axis Pulse Output Control.

Support CiA 402 and G-code Profiles.

(Revision 2.0)

REVISION

Date	Version	Description
2024/8/29	Version 1.0	New Release.
2026/3/5	Version 2.0	• Change the term “Slave” to “SubDevice”. • Update firmware version to 1.06. • Change CiA 402 X/Y/Z axis to M1/M2/M3 axis. • Add index objects 0x50C1~0x50C6 to specify the workpiece coordinates' offset position in G-code. • Supported homing methods (0x60E3 / 0x68E3 / 0x70E3), add Homing Method 37. • Change minimum cycle time from 125 to 250 μs. • Update Synchronization Modes diagram • Added object with supported homing methods (0x60E3/0x68E3/0x70E3). • Modify Storable Objects and update according to previous versions. Add 0x5011 and 0x5014 to the Common field, and remove 0x608C, 0x688C, and 0x708C from the CiA-402 field.
2026/7/1	Version 2.1	• Add index object 0x5015 (PulseFrequency) to set the maximum output pulse frequency of each axis (M1/M2/M3) in CiA-402 mode. • Update firmware version to 1.08.

COPYRIGHT

The information in this manual is subject to change without notice for continuous improvement in the product. All rights are reserved. The manufacturer assumes no responsibility for any inaccuracies that may be contained in this document and makes no commitment to update or to keep current the information contained in this manual.

No part of this manual may be reproduced, copied, translated or transmitted, in whole or in part, in any form or by any means without the prior written permission of ICOP Technology Inc.

©Copyright 2026 ICOP Technology Inc.
Ver.2.1 July, 2026

TRADEMARKS ACKNOWLEDGMENT

ICOP® is the registered trademark of ICOP Corporation. Other brand names or product names appearing in this document are the properties and registered trademarks of their respective owners. All names mentioned herewith are served for identification purpose only.

For more detailed information or if you are interested in other ICOP products, please visit our official websites at:

- Global: www.icop.com.tw
- USA: www.icoptech.com
- Japan: www.icop.co.jp
- Europe: www.icoptech.eu
- China: www.icop.com.cn

For technical support or drivers download, please visit our websites at:

- https://www.icop.com.tw/resource_entrance

For EtherCAT solution service, support or tutorials, 86Duino Coding IDE 500+ introduction, functions, languages, libraries, etc. Please visit the QEC website:

- QEC: https://www.qec.tw/

This Manual is for the QEC series.

SAFETY INFORMATION

- Read these safety instructions carefully.
- Please carry the unit with both hands and handle it with caution.
- Power Input voltage +19 to +50 VDC Power Input (Typ. +24 VDC)
- Make sure the voltage of the power source is appropriate before connecting the equipment to the power outlet.
- To prevent the QEC device from shock or fire hazards, please keep it dry and away from water and humidity.
- Operating temperature between -20 to +70°C.
- When using external storage as the main operating system storage, ensure the device's power is off before connecting and removing it.
- Never touch un-insulated terminals or wire unless your power adaptor is disconnected.
- Locate your QEC device as close as possible to the socket outline for easy access and avoid force caused by the entangling of your arms with surrounding cables from the QEC device.
- If your QEC device will not be used for a period of time, make sure it is disconnected from the power source to avoid transient overvoltage damage.

WARNING!

DO NOT ATTEMPT TO OPEN OR TO DISASSEMBLE THE CHASSIS (ENCASING) OF THIS PRODUCT. PLEASE CONTACT YOUR DEALER FOR SERVICING FROM QUALIFIED TECHNICIAN.

CONTENT

Ch. 1 General Information 1
1.1 Introduction 2
1.2 Specifications 3
1.2.1 General Information 3
1.2.2 Driver Information 4
1.3 Dimension 5
1.4 Mounting Instruction 6
1.5 Ordering Information 7
1.5.1 Ordering Part Number 7
Ch. 2 Hardware System 8
2.1 General Technical Data 9
2.2 Connector Summary 10
2.2.1 EtherCAT Interface 11
2.2.1.1 EC IN 11
2.2.1.2 EC OUT 11
2.2.2 Power Connector 13
2.2.3 Power and Connection Status LEDs 14
2.2.3.1 Power Status LED 14
2.2.3.2 Connection Status LEDs 16
2.2.4 Drive Status LEDs 17
2.2.5 M1-axis Connector 18
2.2.6 M2-axis Connector 20
2.2.7 M3-axis Connector 22
2.2.8 DIN-Rail installation 24
2.3 Wiring to the Connector 25
2.3.1 Connecting the wire to the connector 25
2.3.2 Removing the wire from the connector 25
2.3.3 Pulse Signal and Sing Signal Wiring 26
2.3.4 Home Signal Wiring 27
2.3.5 SVON Signal Wiring 28
2.3.6 Alarm Signal Wiring 29
2.3.7 Differential Encoder (A/B/Z) Wiring 30
Ch. 3 Hardware Installation 31
3.1 DIN-Rail installation 32
3.2 Removing QEC-RXXMV Unit 33
Ch. 4 EtherCAT Communication 34
4.1 EtherCAT Basics 35
4.2 EtherCAT Cabling 35
4.3 EtherCAT State Machine 36
4.4 Process Data Object 38

4.4.1 PDO Mapping 39
4.4.2 PDO Assign 41
4.4.3 PDO Operation Process 42
4.4.3.1 For G-code Mode Operation Process: 42
4.5 CAN application protocol over EtherCAT 44
4.6 Synchronization Modes 45
4.6.1 Free Run 45
4.6.2 SM2 Event 45
4.6.3 Distributed Clock 46
Ch. 5 Getting Started 47
5.1 Introduction 49
5.2 TwinCAT (PP Mode) 50
5.2.1 Install the ESI File 50
5.2.2 Add the QEC-R11MP3S to the Project 51
5.2.3 Start to Configure the QEC-R11MP3S 53
5.2.4 Control the QEC-R11MP3S 56
5.3 86Duino Coding IDE (PP Mode) 61
5.3.1 Connection and wiring hardware 62
5.3.2 Download Software 63
5.3.3 Connect to your PC and set up the environment 64
5.3.4 Write code 65
5.3.4.1 Single Axis 66
5.3.4.2 Three Axes 70
Ch. 6 SubDevice Information 75
6.1 ESI (EtherCAT SubDevice Information) file 76
6.2 Object Dictionary 76
6.2.1 Standard Objects (0x1000-0x1FFF) 77
6.2.1.1 RxPDO Mapping Objects 79
6.2.1.2 TxPDO Mapping Objects 82
6.2.1.3 Sync Manager Objects 87
6.2.2 Manufacturer Objects (0x5000-0x5FFF) 90
6.2.2.1 Motor Objects 91
6.2.2.2 G code Objects 102
6.2.3 Especial Objects (0x6000-0xFFFF) 119
6.2.3.1 CiA-402 Objects 119
6.2.3.2 Index 0x6nnx Input Data of the Module (0x6000 - 0x6FFF) 121
6.2.3.3 Index 0x7nnx Output Data of the Module (0x7000 - 0x7FFF) 136
6.2.3.4 Index 0xAnnx Device Object (0xA000 – 0xAFFF) 141
6.2.3.5 Index 0xFnnx Device Object (0xF000 – 0xFFFF) 142
Warranty 143

Ch. 1

General Information

1.1 Introduction

The QEC-RXXMV3S series is a high-performance EtherCAT pulse-output motion module designed for traditional automation systems that require precise, synchronized multi-axis control. It supports 3-axis independent control with configurable pulse output up to 4 MHz per channel (100 kHz / 500 kHz / 4 MHz, selectable via object 0x5015), and achieves high-speed synchronization with a minimum communication cycle of 250 µs. Built for seamless integration, it is compatible with CiA 402 and G-code drive profiles, and supports multiple synchronization modes, including DC, SM2, and FreeRun.

To cover a wide range of motion tasks, the QEC-RXXMV3S supports Profile Position (PP), Cyclic Synchronous Position (CSP), and Cyclic Synchronous Velocity (CSV), along with Homing (Methods 19, 20, 21, 22, and 37). Each axis provides complete committed I/O signals—including Pulse±, Sing±, Home±, SVON±, Alarm±, and A/B/Z encoder interfaces—making it ideal for precise motion control, machine management, and retrofit automation projects. The series has been verified by conformance testing tools, ensuring reliable EtherCAT communication and robust operation in industrial environments.

Additionally, the encoder interfaces (A, B, Z) with differential signaling deliver accurate position feedback, superior noise immunity, and enhanced motion control, making the QEC-RXXMV3S a reliable and precise solution for high-performance industrial applications.

The QEC-RXXMV3S series is compact at 107.45 x 66 x 30 mm (Without DIN-Rail), making it very convenient for system installation via DIN-Rail mounting. It operates within a temperature range of -20°C to +70°C and is equipped with two network ports for EtherCAT network redundancy, enhancing system reliability and stability.

1.2 Specifications

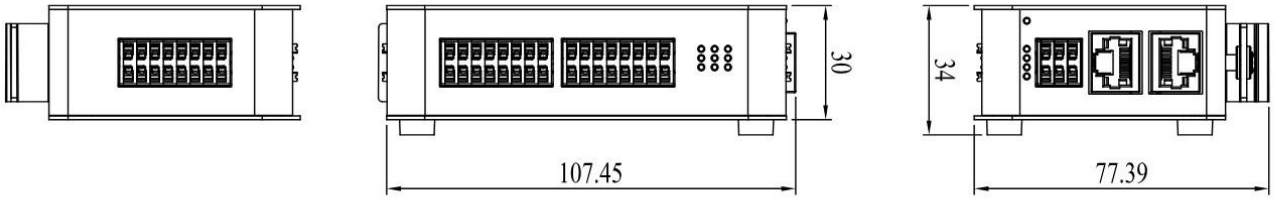
1.2.1 General Information

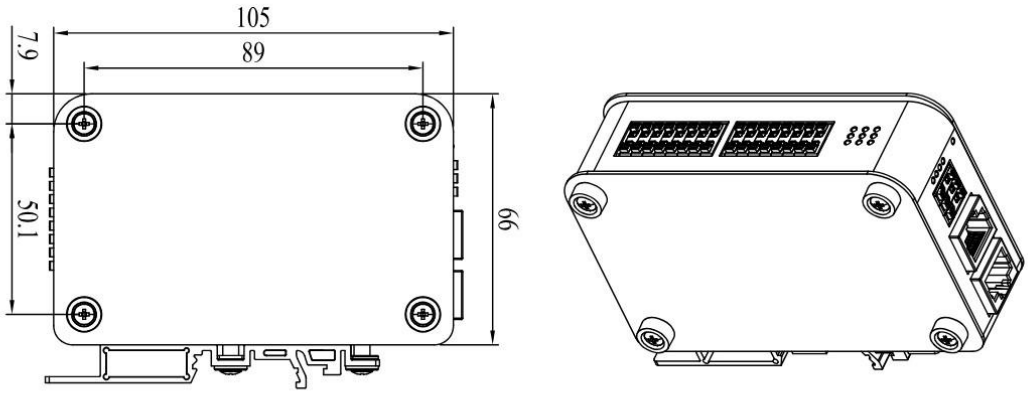
General	
Connector	Push-in Terminal (Euroblock)
Connector Color	Pulse-: White, Pulse+: Red Sing-: White, Sing+: Black Home-: White, Home+: Blue SVON-: White, SVON+: Orange Alarm-: White, Alarm+: Dark Red Encoder A-: White, A+: Blue Encoder B-: White, B+: Green Encoder Z-: White, Z+: Dark Brown
Protocol	EtherCAT (RJ-45 x2)
Ethernet Standard	IEEE 802.3
Transmission Rate	100 Mbps
Power Connector	4-pin Power Input/Output & 2-pin FGND
Power Requirement	+19 to +50 VDC Power Input (Typ. +24 VDC @ 300mA)
Power Consumption	Min. 7.2 W
LED Indicator	PWR, RUN, LINK, ERROR, Alarm, Home, Motor
Certifications	CE, FCC, VCCI
Environment	
Operating Temperature	-20 to +70 °C
Drive Protection	Thermal shutdown (TSD) circuit Under voltage lock out (UVLO) circuit Over-current detection (ISD) circuit
Hardware	
Dimension	107.45 x 66 x 30 mm (Without DIN-Rail)
Weight	370 g
Installation	DIN-Rail
Internal Monitoring	Temperature, Voltage, Current, Startup time

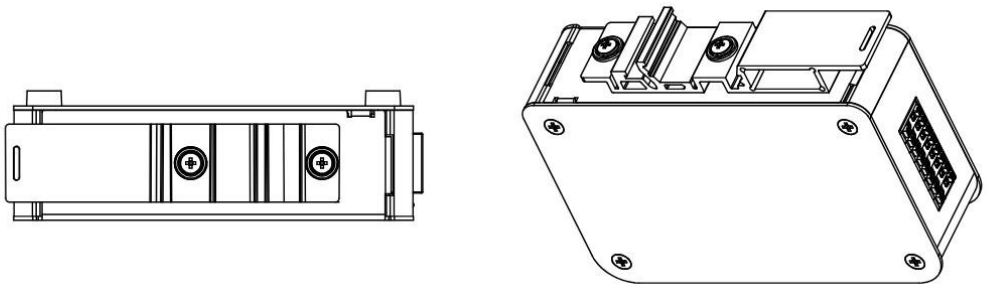
1.2.2 Driver Information

Driver	
Interface	EtherCAT
Drive Profile	CiA402, G-code
Minimum Communication Cycle	250 μs
Synchronization Mode	DC, SM2, FreeRun
Compatible Operation Mode	Profile Position (PP) Cyclic Synchronous Position (CSP) Cyclic Synchronous Velocity (CSV) Homing (HM: Support Method 19, 20, 21, 22, 37)
Motor driver control interface	3-channel Pulse-Dir output(±5V differential output)
Voltage Requirement	+8 to +50 VDC
Pulse command Output	100 kHz / 500 kHz / 4 MHz (configurable via object 0x5015, default 100 kHz)
Committed I/O Signal	Pulse± / Sing± / Home± / SVON± / Alarm± / A± / B± / Z± for each channel
Encoder	
Encoder Inputs	3 x Encoder counter (A, B, Z), differential
Maximum Encoder pulse frequency	14 MHz
Positioning Range	-2,147,483,648 through 2,147,483,647 pulse (32-bit)
Differential Voltage of Encoder Input	±5V
Encoder Type	AB Phase x4 AB Phase x2 CW/CCW x2 CW/CCW STEP/DIR x2 STEP/DIR

1.3 Dimension





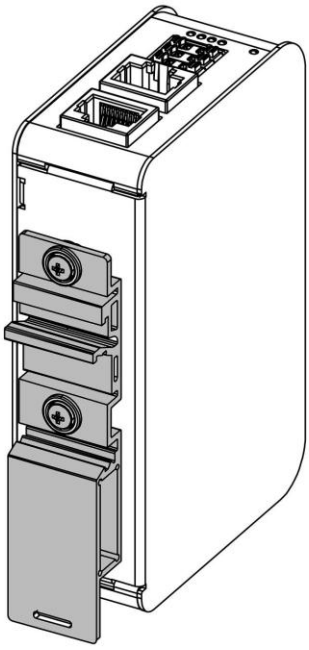


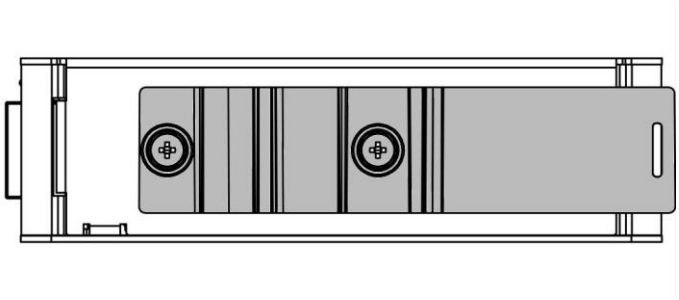
(Unit: mm)

1.4 Mounting Instruction

QEC-RXXMV series is an easy-install design to help you set-up your modules easily. Please refer to Ch.3.1 DIN-Rail installation.

- **DIN-Rail**





1.5 Ordering Information

Type	RJ45 power source		Functions		Feature	-	Coating
	EtherCAT Input	EtherCAT Output	Pulse Output	Functions	Standard		
QEC-R	X	X	MV	X	S		X

1. Type: Code 1~4

R: EtherCAT SubDevice

2. RJ45 Power source: Code 5~6

0: RJ45 In/Out w/o power

1: RJ45 In/Out – Power Device

3. Functions: Code 7~9

MV: Pulse Output

X: 1 or 3, different stepper motor axis

4. Feature: Code 10

S: Standard

5. Coating: Code 11

C: Yes / N: Normal

Q E C - R X X M V X S - X

1.5.1 Ordering Part Number

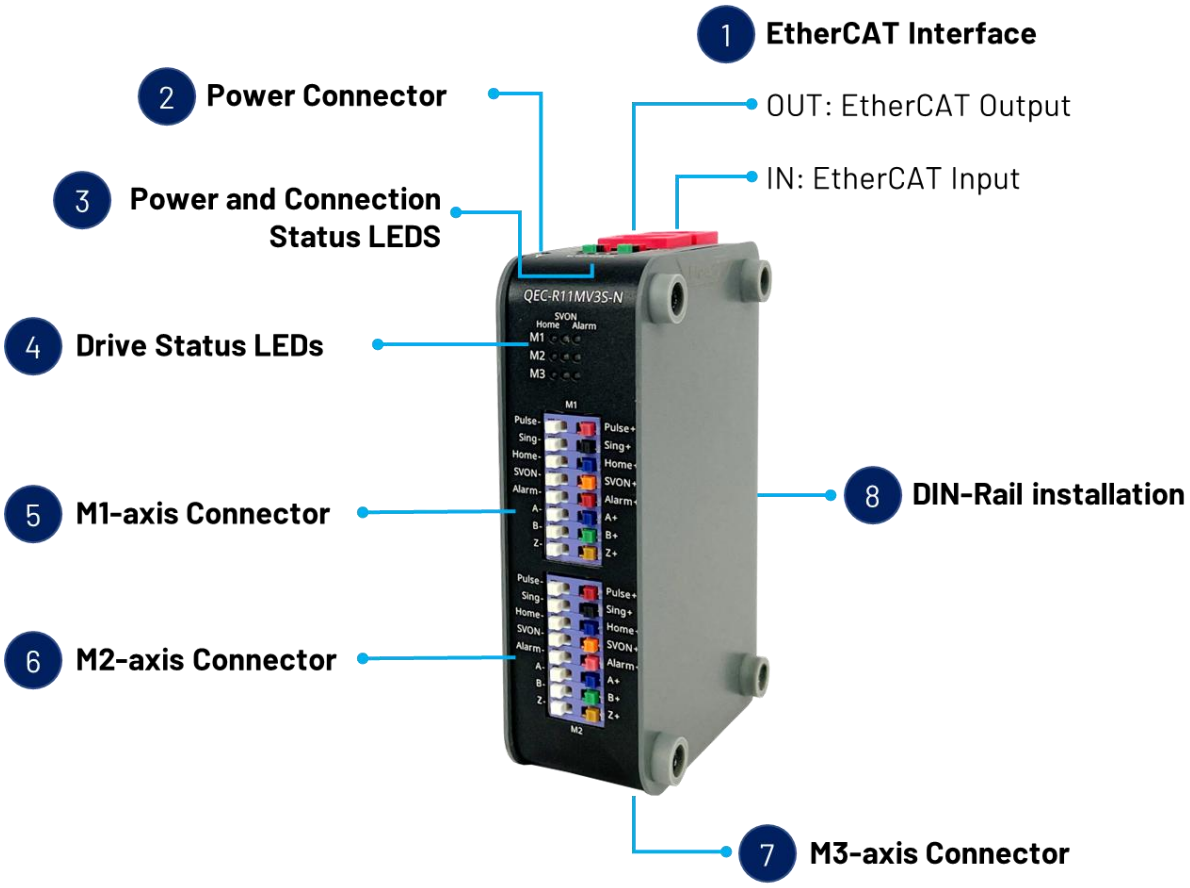
Above is the standard Part Number, please contact our sales if you need to order other part number.

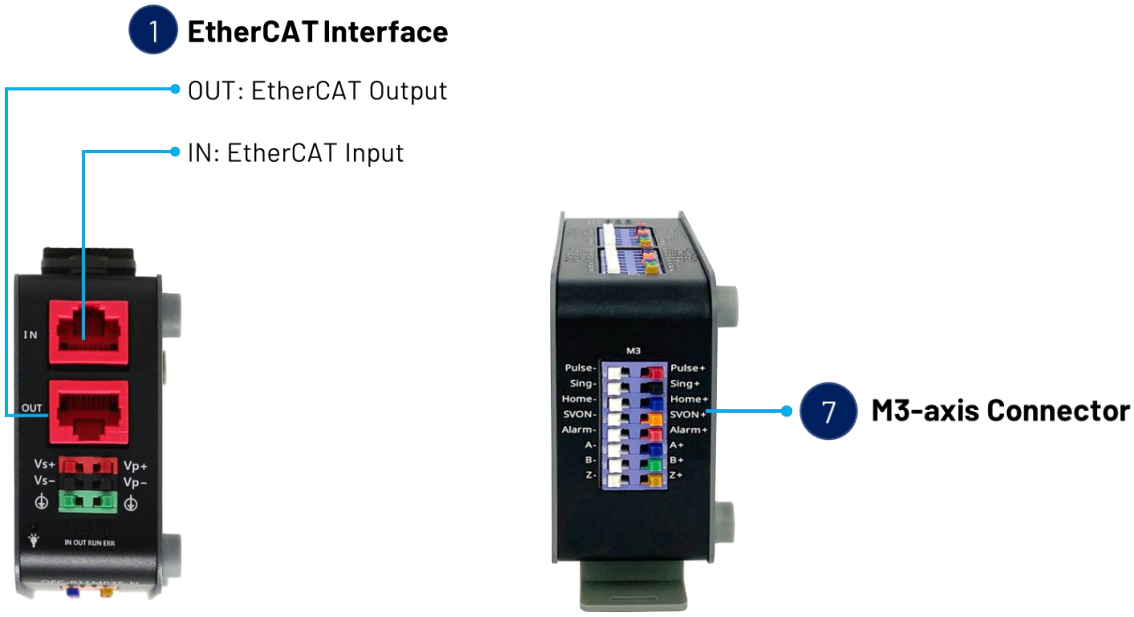
- **QEC-R00MV3S-N**: EtherCAT SubDevice 3 axis Pulse Output module.
- **QEC-R00MV3S-C**: EtherCAT SubDevice 3 axis Pulse Output module (board with coating).
- **QEC-R11MV3S-N**: EtherCAT SubDevice 3 axis Pulse Output module/PoE.

Ch. 2

Hardware System

2.1 General Technical Data



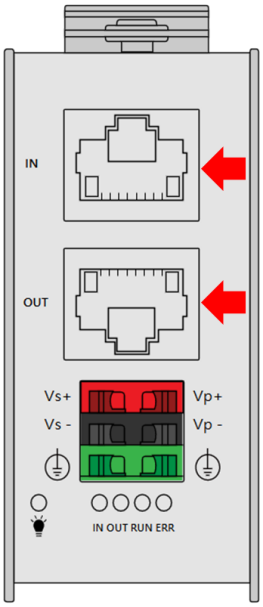


2.2 Connector Summary

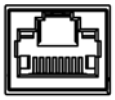
No.	Description		Type Narrative	Num #
1	EtherCAT Interface	OUT	RJ45 Connector (Gold finger)	8-pin
		IN		8-pin
2	Power Connector		Power Socket	6-pin
3	Power and Connection Status LEDs		Status LEDs	-
4	Drive Status LEDs		Status LEDs	-
5	M1-axis Connector		Drive M1 axis.	16-pin
6	M2-axis Connector		Drive M2 axis.	16-pin
7	M3-axis Connector		Drive M3 axis.	16-pin
8	DIN-Rail		-	-

2.2.1 EtherCAT Interface

RJ45 Connectors.



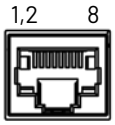
2.2.1.1 EC IN



Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS- (GND)	8	VP- (GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.
* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

2.2.1.2 EC OUT

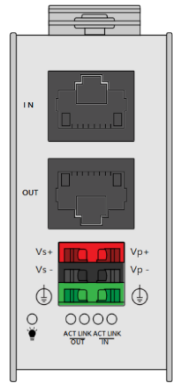
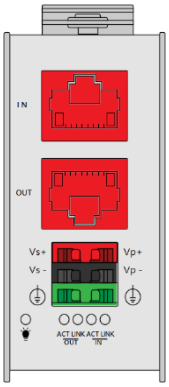


Pin #	Signal Name	Pin #	Signal Name
1	LAN2_TX+	2	LAN2_TX-
3	LAN2_RX+	4	VS+
5	VP+	6	LAN2_RX-
7	VS- (GND)	8	VP- (GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.
* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

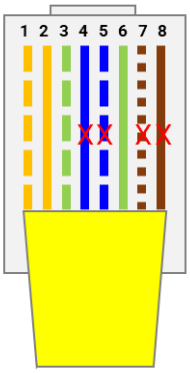
Notes QEC's PoE (Power over Ethernet)

In QEC product installations, users can easily distinguish between PoE and non-PoE: if the RJ45 house is red, it is PoE type, and if the RJ45 house is black, it is non-PoE type.

	
Non-PoE type	PoE type

PoE (Power over Ethernet) is a function that delivers power over the network.
QEC can be equipped with an optional PoE function to reduce cabling. In practice, PoE is selected based on system equipment, so please pay attention to the following points while evaluating and testing:

1. The PoE function of QEC is different and incompatible with EtherCAT P, and the PoE function of QEC is based on PoE Type B, and the pin functions are as follows:



T568B Color

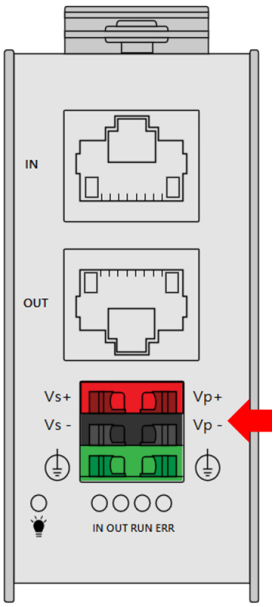
PINS	10/100 DC on Spares (Mode B)
1	Rx+
2	Rx-
3	Tx+
4	DC+
5	DC+
6	Tx-
7	DC-
8	DC-

2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

3. QEC's PoE power supply is up to 24V/3A.

2.2.2 Power Connector

Euroblock Connectors.
4-pins Power Input/Output & 2-pins FGND.



Vs for system power; Vp for peripheral power, motor power, and backup power.

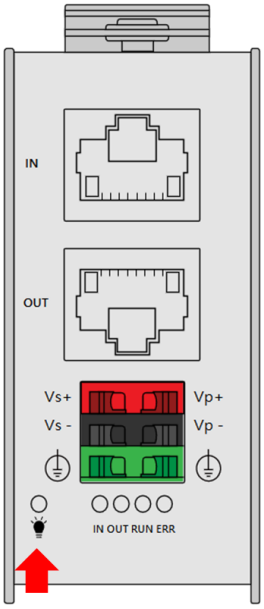
	Pin #	Signal Name	Pin #	Signal Name
	1	Vs+	2	Vp+
	3	Vs- (GND)	4	Vp- (GND)
	5	F.G	6	F.G

* **Note:** Power Input voltage +19 to +50VDC Power Input (Typ. +24VDC)

2.2.3 Power and Connection Status LEDs

QEC-RXXMV3S's power and connection status LEDs.

2.2.3.1 Power Status LED



Power input is 24V (typical).
The LED status provide high/low voltage warning.

Notation	States	Condition	Description
PWR	Green LED On	Voltage <= 50V and >= 45V Voltage <= 26V and >= 19V	When Vs and Vp voltages are confirmed to be normal, the Green LED will remain steady on.
	Green LED On Red LED On	Voltage < 45V and > 26V Voltage < 19V and > 12V	LEDs will alternately flash (at 0.3-second intervals) until the Vs and Vp voltages are correct.
	Orange LED On	Voltage > 50V or < 12V	Orange LED (Green + Red) will continuously flash (at 0.3-second intervals) until the Vs and Vp voltages are correct.

* **Note:** Vs power status will be displayed first.

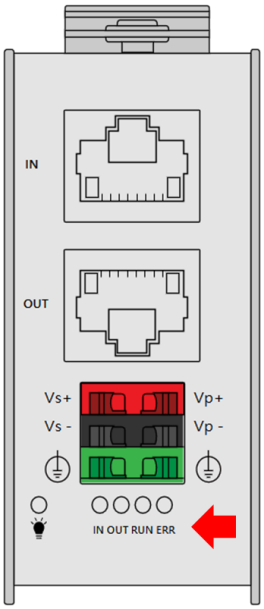
Power ERROR Code table (Red LED Flashing Display (2 seconds/cycle)):

Long Light	Short Flash	Description
0 Long Light	After microchip completes the BootLoad test, it proceeds to the APP program stage.	
	1 short flash	microchip communication with the EtherCAT chip failed.
	2 short flashes	EtherCAT chip internal RAM test failed.
	5 short flashes	Quartz oscillator on the board abnormality.
	6 short flashes	Quartz oscillator on the board abnormality.
1 Long Light	Indicates the microchip BootLoad stage during startup, APP program not yet executed.	
	1 short flash	microchip internal SRAM failed.
	2 short flashes	APP software CHECKSUM failed.
2 Long Lights	Not yet defined.	

* **Note**: If you encounter any of the above abnormal states, please write to info@icop.com.tw.

2.2.3.2 Connection Status LEDs

There are EtherCAT In, Out, Run, and Error Status LEDs.

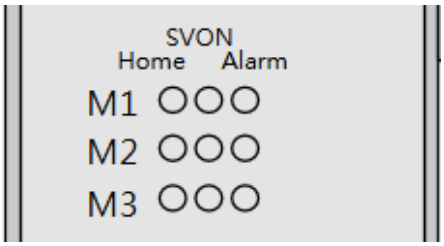


Status LEDs connection table:

Notation	Color	States	Description
In	Green	Off	No link
		Blinking	Link and activity
		On	Link without activity
Out	Green	Off	No link
		Blinking	Link and activity
		On	Link without activity
Run	Green	Off	The device is in state INIT
		Blinking	The device is in state Pre-Operation
		Single Flash	The device is in state Safe-Operation
		On	The device is in state Operation
Err	Red	Off	No error
		Blinking	Invalid Configuration
		Single Flash	Local Error
		Double Flash	Process Data Watchdog Timeout EtherCAT Watchdog Timeout
		On	The device is in state Error

2.2.4 Drive Status LEDs

There is Home, SVON, and Alarm Status LEDs.

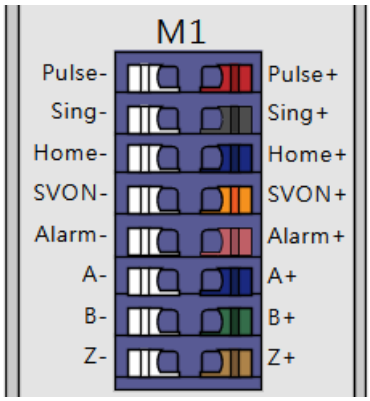


Drive Status LEDs table:

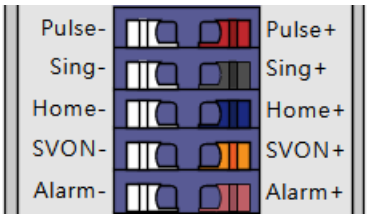
Notation	Color	Description
M1 Home	Orange	M1 axis Machine Home signal.
M1 SVON	Orange	M1 axis Machine Servo-ON (SVON) signal.
M1 Alarm	Orange	M1 axis Machine Alarm signal.
M2 Home	Orange	M2 axis Machine Home signal.
M2 SVON	Orange	M2 axis Machine Servo-ON (SVON) signal.
M2 Alarm	Orange	M2 axis Machine Alarm signal.
M3 Home	Orange	M3 axis Machine Home signal.
M3 SVON	Orange	M3 axis Machine Servo-ON (SVON) signal.
M3 Alarm	Orange	M3 axis Machine Alarm signal.

2.2.5 M1-axis Connector

M1 axis pulse output connectors (Euroblock).
Signals: Pulse± / Sing± / Home± / SVON± / Alarm± / A± / B± / Z±.

Signal Name		Signal Name
Pulse-		Pulse+
Sing-		Sing+
Home-		Home+
SVON-		SVON+
Alarm-		Alarm+
A-		A+
B-		B+
Z-		Z+

Pulse Output signal Description:



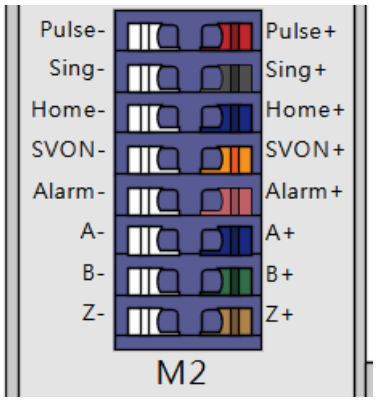
Name	Connector Color	Signal	Signal Description
Pulse+	Red	Output	Pulse differential output signal (positive, 5V)
Pulse-	White	Output	Pulse differential output signal (negative, 5V)
Sing+	Black	Output	Direction differential signal (positive, 5V)
Sing-	White	Output	Direction differential signal (negative, 5V)
Home+	Blue	Input	Homing signal (positive)
Home-	White	Input	Homing signal (negative)
SVON+	Orange	Output	Servo-on signal (positive)
SVON-	White	Output	Servo-on signal (negative)
Alarm+	Red	Input	Alarm signal (positive)
Alarm-	White	Input	Alarm signal (negative)

Encoder connectors Description:

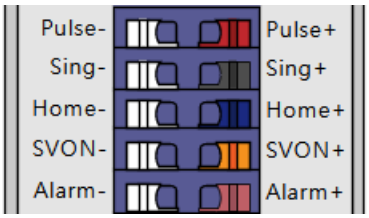
Name	Connector Color	Signal	Signal Description
A+	Blue	Input	Differential Encoder A phase (positive, 5V)
A-	White	Input	Differential Encoder A phase (negative, 5V)
B+	Green	Input	Differential Encoder B phase (positive, 5V)
B-	White	Input	Differential Encoder B phase (negative, 5V)
Z+	Brown	Input	Differential Encoder Z phase (positive, 5V)
Z-	White	Input	Differential Encoder Z phase (negative, 5V)

2.2.6 M2-axis Connector

M2 axis pulse output connectors (Euroblock).
Signals: Pulse± / Sing± / Home± / SVON± / Alarm± / A± / B± / Z±.

Signal Name		Signal Name
Pulse-		Pulse+
Sing-		Sing+
Home-		Home+
SVON-		SVON+
Alarm-		Alarm+
A-		A+
B-		B+
Z-		Z+

Pulse Output signal Description:



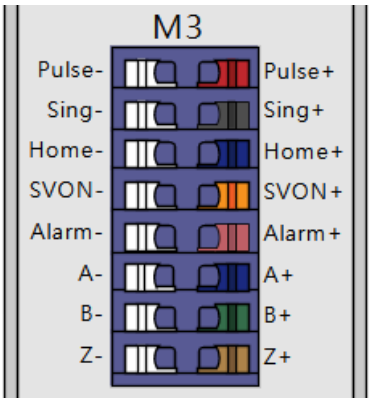
Name	Connector Color	Signal	Signal Description
Pulse+	Red	Output	Pulse differential output signal (positive, 5V)
Pulse-	White	Output	Pulse differential output signal (negative, 5V)
Sing+	Black	Output	Direction differential signal (positive, 5V)
Sing-	White	Output	Direction differential signal (negative, 5V)
Home+	Blue	Input	Homing signal (positive)
Home-	White	Input	Homing signal (negative)
SVON+	Orange	Output	Servo-on signal (positive)
SVON-	White	Output	Servo-on signal (negative)
Alarm+	Red	Input	Alarm signal (positive)
Alarm-	White	Input	Alarm signal (negative)

Encoder connectors Description:

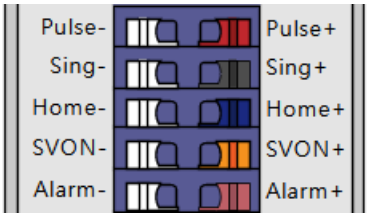
Name	Connector Color	Signal	Signal Description
A+	Blue	Input	Differential Encoder A phase (positive, 5V)
A-	White	Input	Differential Encoder A phase (negative, 5V)
B+	Green	Input	Differential Encoder B phase (positive, 5V)
B-	White	Input	Differential Encoder B phase (negative, 5V)
Z+	Brown	Input	Differential Encoder Z phase (positive, 5V)
Z-	White	Input	Differential Encoder Z phase (negative, 5V)

2.2.7 M3-axis Connector

M3 axis pulse output connectors (Euroblock).
Signals: Pulse± / Sing± / Home± / SVON± / Alarm± / A± / B± / Z±.

Signal Name		Signal Name
Pulse-		Pulse+
Sing-		Sing+
Home-		Home+
SVON-		SVON+
Alarm-		Alarm+
A-		A+
B-		B+
Z-		Z+

Pulse Output signal Description:



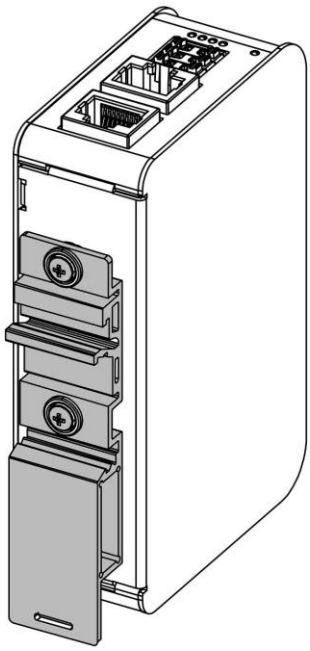
Name	Connector Color	Signal	Signal Description
Pulse+	Red	Output	Pulse differential output signal (positive, 5V)
Pulse-	White	Output	Pulse differential output signal (negative, 5V)
Sing+	Black	Output	Direction differential signal (positive, 5V)
Sing-	White	Output	Direction differential signal (negative, 5V)
Home+	Blue	Input	Homing signal (positive)
Home-	White	Input	Homing signal (negative)
SVON+	Orange	Output	Servo-on signal (positive)
SVON-	White	Output	Servo-on signal (negative)
Alarm+	Red	Input	Alarm signal (positive)
Alarm-	White	Input	Alarm signal (negative)

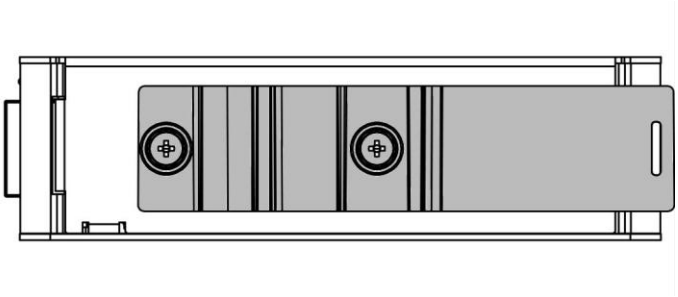
Encoder connectors Description:

Name	Connector Color	Signal	Signal Description
A+	Blue	Input	Differential Encoder A phase (positive, 5V)
A-	White	Input	Differential Encoder A phase (negative, 5V)
B+	Green	Input	Differential Encoder B phase (positive, 5V)
B-	White	Input	Differential Encoder B phase (negative, 5V)
Z+	Brown	Input	Differential Encoder Z phase (positive, 5V)
Z-	White	Input	Differential Encoder Z phase (negative, 5V)

2.2.8 DIN-Rail installation

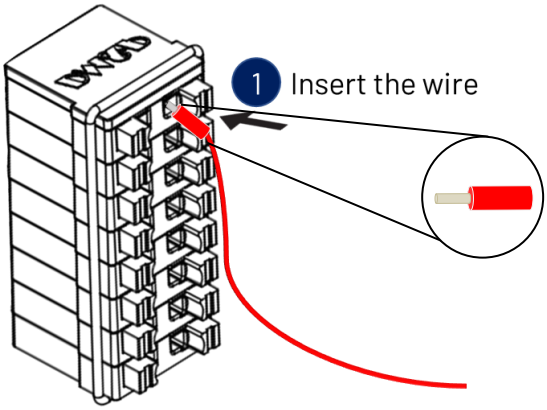
Please refer to Ch.3.1 DIN-Rail installation.

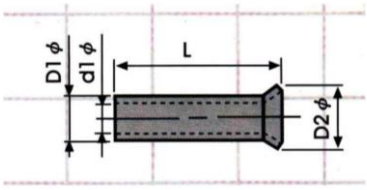




2.3 Wiring to the Connector

2.3.1 Connecting the wire to the connector

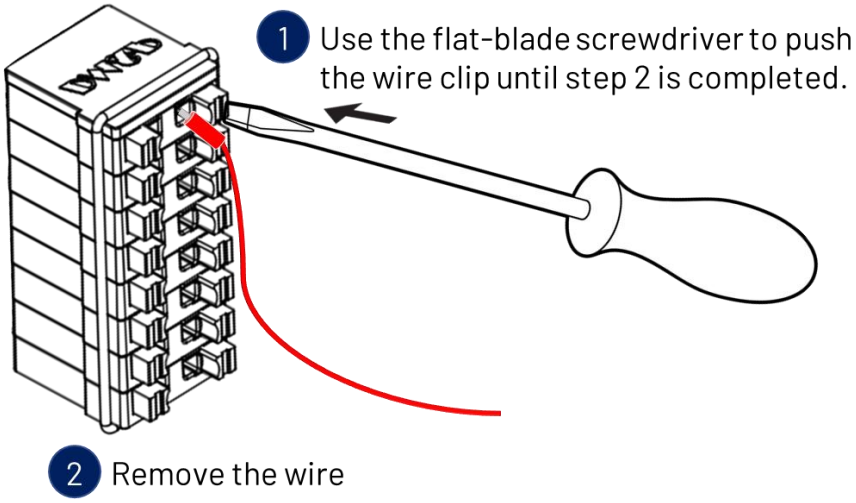




Insulated Terminals Dimensions (mm)

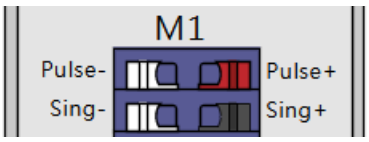
Position	L	ØD1	Ød1	ØD2
CN 0.5-6	6.0	1.3	1.0	1.9
CN 0.5-8	8.0	1.3	1.0	1.9
CN 0.5-10	10.0	1.3	1.0	1.9

2.3.2 Removing the wire from the connector



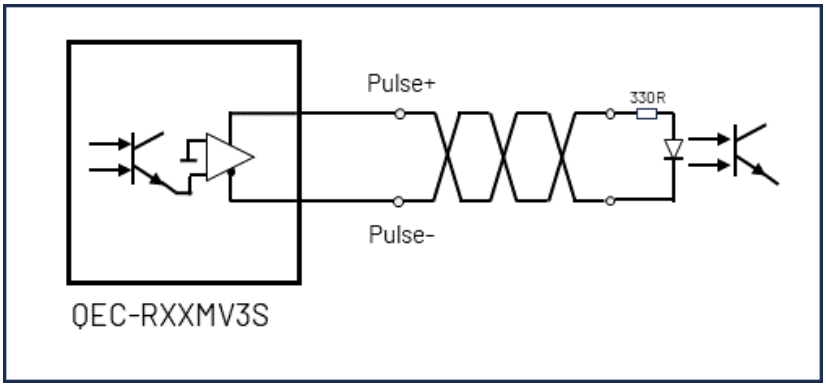
2.3.3 Pulse Signal and Sing Signal Wiring

Pulse signal and Sing signal.

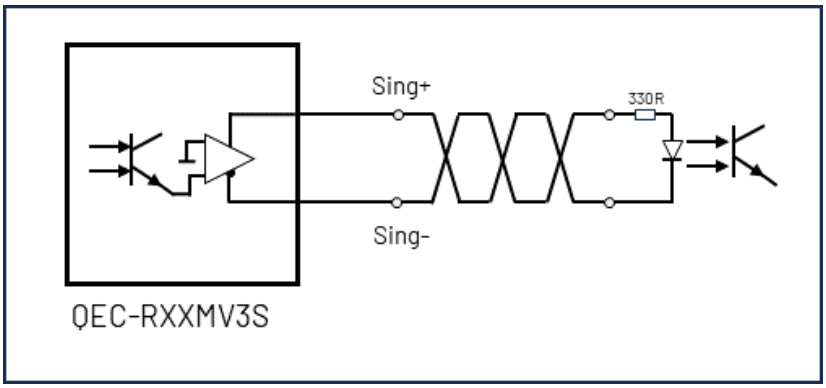


Circuit examples for Pulse +/- and Sing +/- of M1 (The same principles apply to M2 and M3).
The QEC-RXXMV3S generates the necessary differential signals for Pulse+/- and Sing+/-.

- Pulse +/- signal application wiring:



- Sing +/- signal application wiring:



The diagram shows the correct wiring for a 5V differential signal input for the QEC-RXXMV3S. The connection includes a 330Ω resistor to ensure proper signal conditioning. The differential signals are correctly routed to the input terminals of the device.

This setup is intended to receive 5V differential signals effectively, ensuring accurate signal transmission and reception.

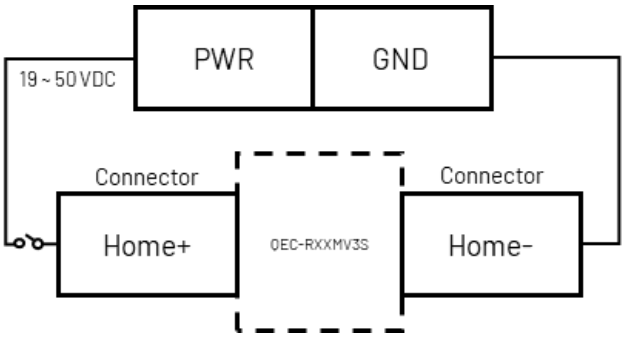
2.3.4 Home Signal Wiring

Home signal.

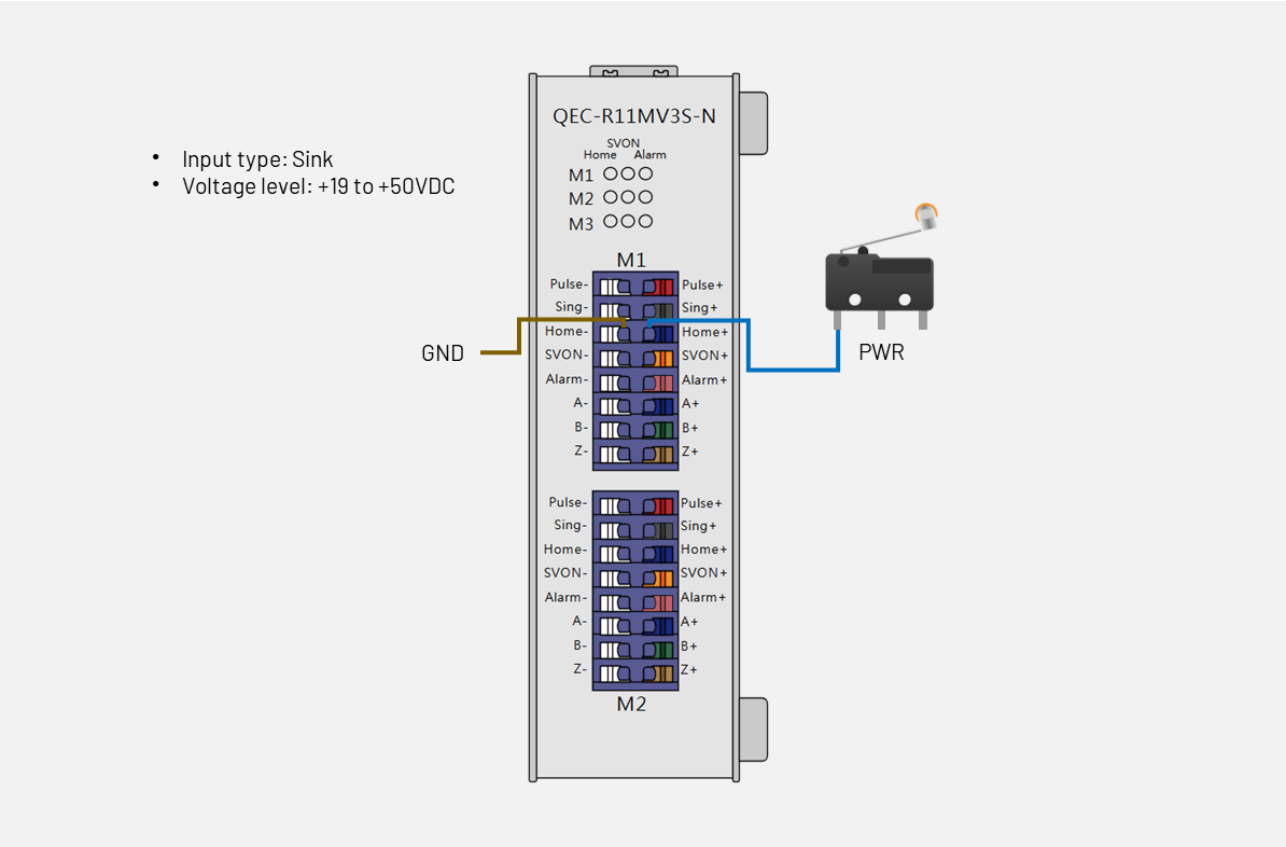
Home signal is digital input.

Digital Input	
Digital Input channels	3
Input type	Sink
Voltage level	+19 to +50VDC
Isolation Voltage Protection	2500 Vrms

Application wiring:



Circuit examples for Home of M1. The same principles apply to M2 and M3.



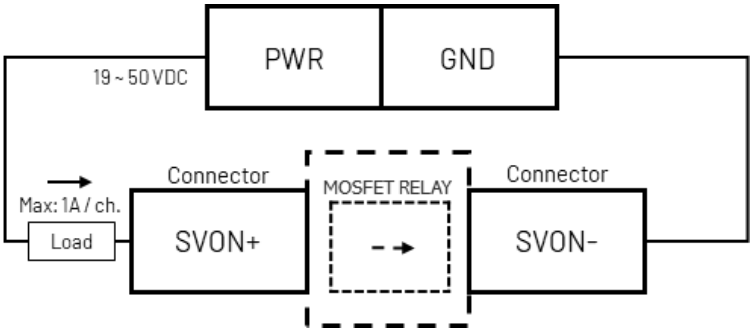
2.3.5 SVON Signal Wiring

SVON (Servo On) signal.

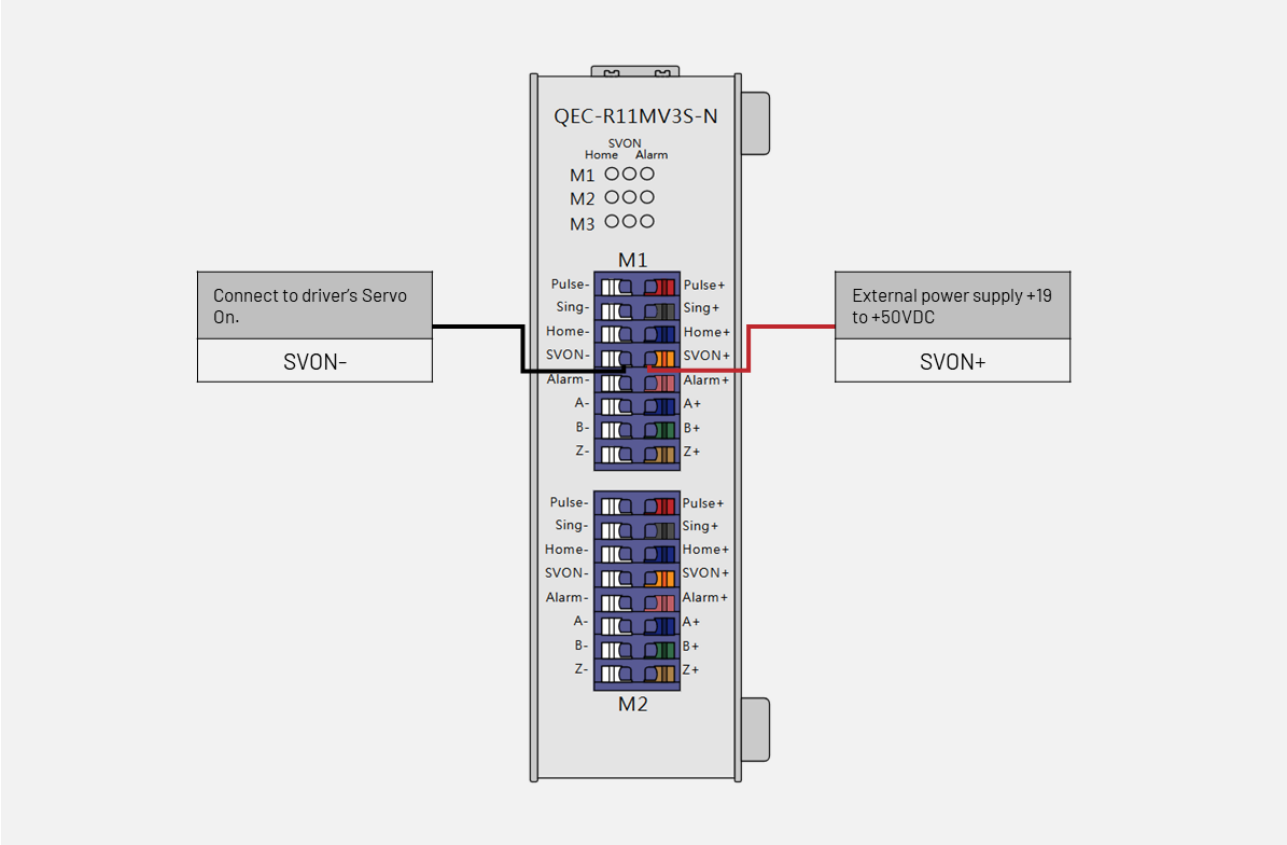
SVON signal is digital output.

Digital Output	
Digital Output channels	3
Output type	Sink
Voltage level	+19 to +50VDC
Isolation Voltage Protection	2500 Vrms

Application wiring:



Circuit examples for SVON of M1. The same principles apply to M2 and M3.



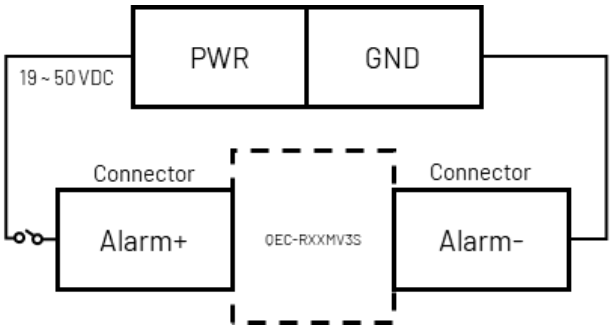
2.3.6 Alarm Signal Wiring

Alarm signal.

Alarm signal is digital input.

Digital Input	
Digital Input channels	3
Input type	Sink
Voltage level	+19 to +50VDC
Isolation Voltage Protection	2500 Vrms

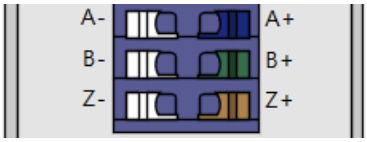
Application Wiring:



Circuit examples for Alarm of M1. The same principles apply to M2 and M3.

2.3.7 Differential Encoder (A/B/Z) Wiring

Differential encoder (A/B/Z).

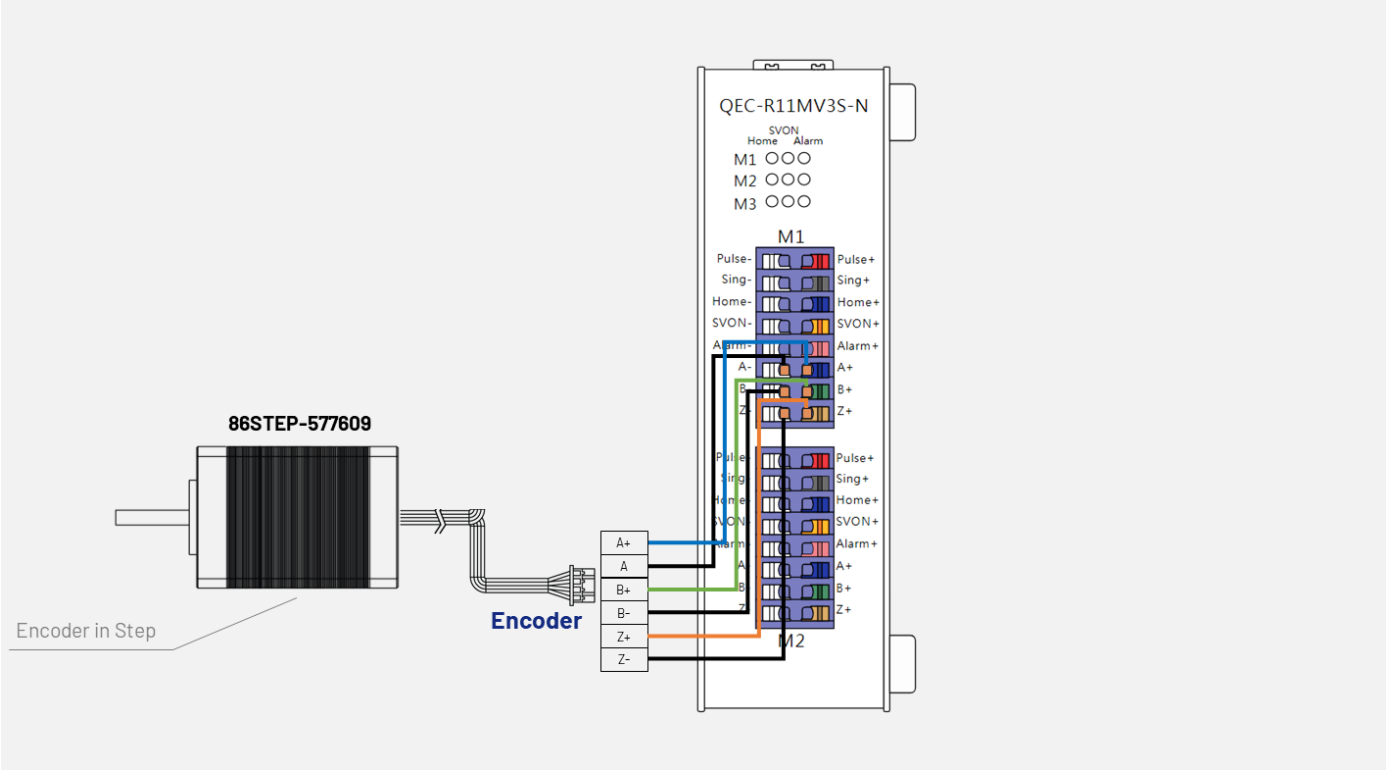


The QEC-RXXMV3S supports differential encoder by default.

Encoder connectors Description:

Name	Connector Color	Signal	Signal Description
A+	Blue	Input	Differential Encoder A phase (positive, 5V)
A-	White	Input	Differential Encoder A phase (negative, 5V)
B+	Green	Input	Differential Encoder B phase (positive, 5V)
B-	White	Input	Differential Encoder B phase (negative, 5V)
Z+	Brown	Input	Differential Encoder Z phase (positive, 5V)
Z-	White	Input	Differential Encoder Z phase (negative, 5V)

Circuit examples for Encoder of M1. The same principles apply to M2 and M3.



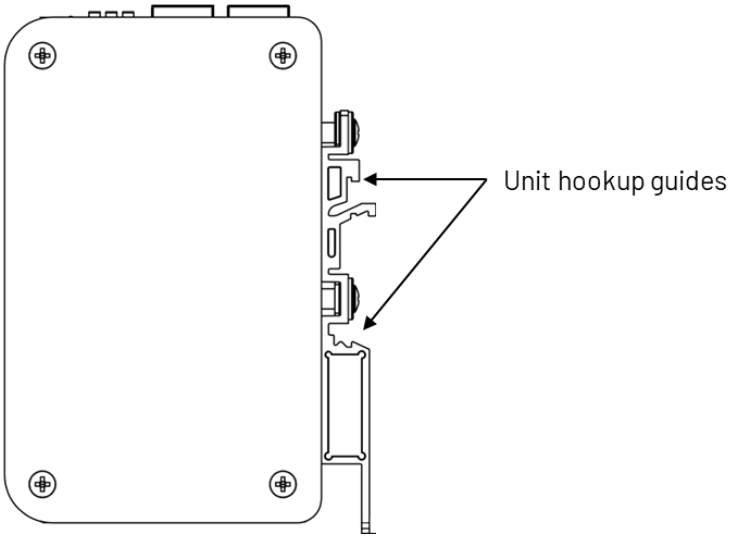
The encoder in the QEC-RXXMV3S can read various signals, including A, B, and Z. It's capable of interpreting pulse signals and determining motor rotation direction, CW (clockwise) or CCW (counter clockwise).

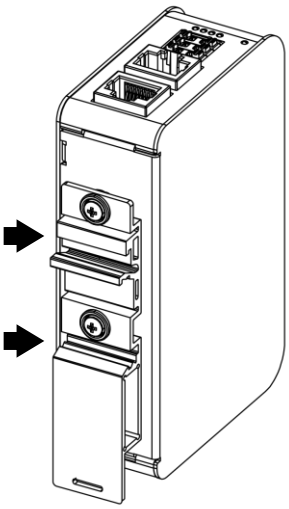
Ch. 3

Hardware Installation

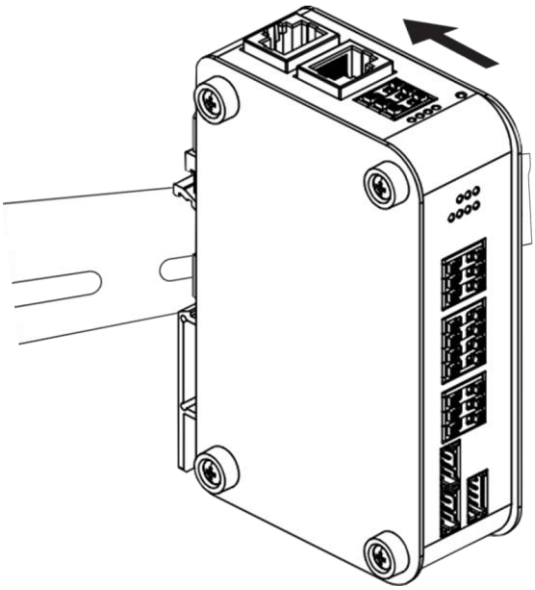
3.1 DIN-Rail installation

Slide in the QEC-RXXMV on the hookup guides and press the QEC-RXXMV with a certain amount of force against the DIN track until the DIN Track mounting hook lock into place.





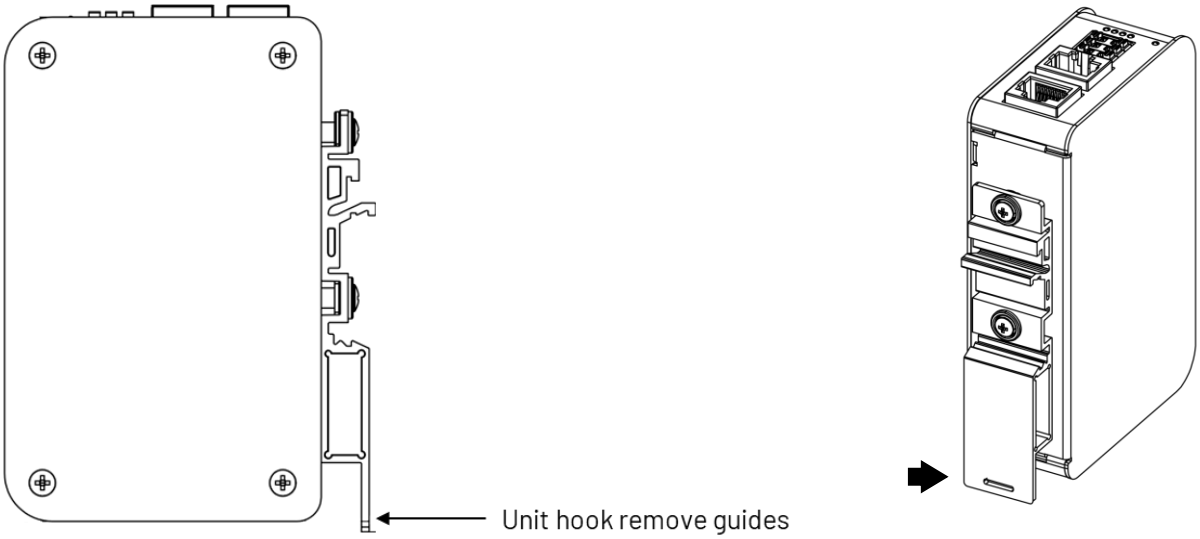
When you mount the QEC-RXXMV, releasing the DIN track mounting hook on the QEC-RXXMV is unnecessary. After you mount the QEC-RXXMV, make sure it is locked to the DIN Track.



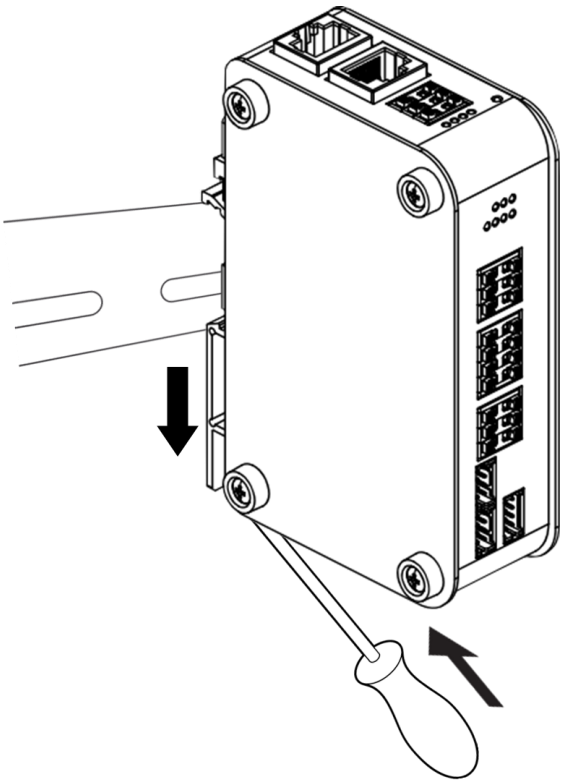
* **Note:** Always turn OFF the Unit power supply and I/O power supply before connecting and removing the QEC-RXXMV.

3.2 Removing QEC-RXXMV Unit

Use a flat-blade screwdriver to remove the DIN Track mounting hook on the unit.



Pull down and out the flat-blade screwdriver with force against the DIN track until you hear the DIN Track remove the hook.



Ch. 4

EtherCAT Communication

4.1 EtherCAT Basics

EtherCAT (Ethernet for Control Automation Technology) is an Ethernet-based fieldbus system developed by Beckhoff Automation. The protocol is standardized in IEC 61158 and is suitable for both hard and soft real-time computing requirements in automation technology.

The goal during the development of EtherCAT was to apply Ethernet for automation applications requiring short data update times (also called cycle times; ≤ 100 µs) with low communication jitter (for precise synchronization purposes; ≤ 1 µs) and reduced hardware costs. Typical application fields for EtherCAT are machine controls (e.g., semiconductor tools, metal forming, packaging, injection molding, assembly systems, printing machines, robotics). Remote-controlled hump yard facilities are used in the railroad industry.

4.2 EtherCAT Cabling

The cable length between two EtherCAT devices must not exceed 100 m.

Cables and connectors

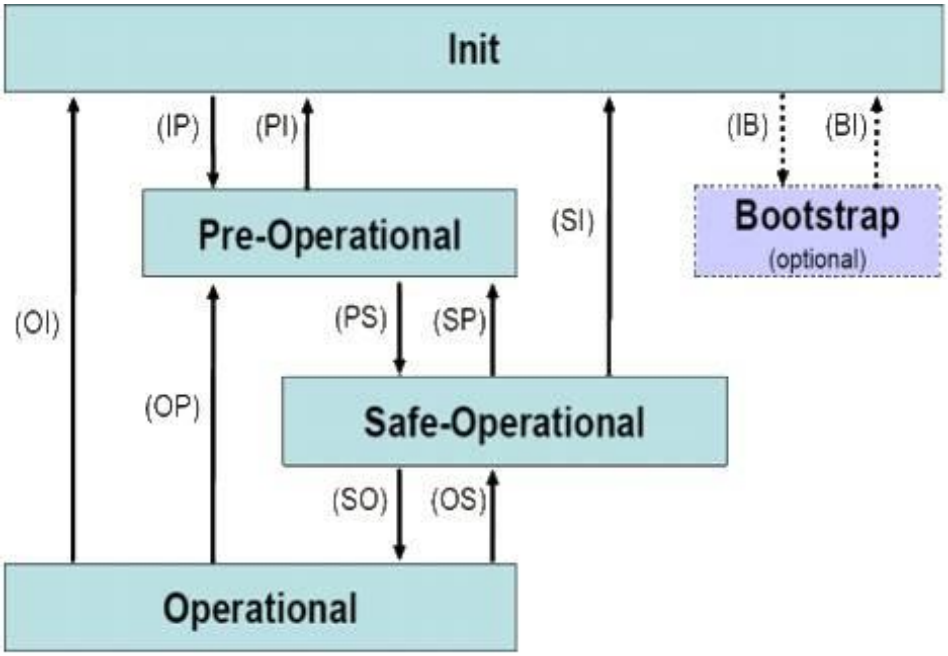
For connecting EtherCAT devices, only Ethernet connections (cables + plugs) that meet the requirements of at least category 5 (CAT5) according to EN 50173 or ISO/IEC 11801 should be used. EtherCAT uses 4 wires for signal transfer.
The pin assignment is compatible with the Ethernet standard (ISO/IEC 8802-3).

Ethernet pin signals and color description table:

Pin	Color of conductor	Signal	Description
1	Yellow	TD+	Transmission Data+
2	Orange	TD-	Transmission Data-
3	White	RD +	Receiver Data+
6	Blue	RD -	Receiver Data-

4.3 EtherCAT State Machine

The state of the EtherCAT SubDevice is controlled via the EtherCAT State Machine (ESM). Depending upon the state, different functions are accessible or executable in the EtherCAT SubDevice. Specific commands must be sent by the EtherCAT MDevice to the device in each state, particularly during the bootup of the SubDevice.



A distinction is made between the following states:

- Init
- Pre-Operational
- Safe-Operational and
- Operational
- Boot

The regular state of each EtherCAT SubDevice after bootup is the OP state.

Init

After switch-on the EtherCAT SubDevice in the Init state. No mailbox or process data communication is possible. The EtherCAT MDevice initializes sync manager channels 0 and 1 for mailbox communication.

Pre-Operational (Pre-Op)

During the transition between Init and Pre-Op the EtherCAT SubDevice checks whether the mailbox was initialized correctly. In Pre-Op state mailbox communication is possible, but not process data communication. The EtherCAT MDevice initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the SubDevice supports configurable mapping, PDO mapping or the sync manager PDO assignment. In this state the settings for the process data transfer and perhaps terminal-specific parameters that may differ from the default settings are also transferred.

Safe-Operational (Safe-Op)

During transition between Pre-Op and Safe-Op the EtherCAT SubDevice checks whether the sync manager channels for process data communication and, if required, the distributed clocks settings are correct. Before it acknowledges the change of state, the EtherCAT SubDevice copies current input data into the associated DP-RAM areas of the EtherCAT SubDevice controller (ECSC). In Safe-Op state mailbox and process data communication is possible, although the SubDevice keeps its outputs in a safe state, while the input data are updated cyclically

* **Note:** Outputs in SAFEOP state
The default set watchdog monitoring sets the outputs of the module in a safe state - depending on the settings in SAFEOP and OP - e.g. in OFF state. If this is prevented by deactivation of the watchdog monitoring in the module, the outputs can be switched or set also in the SAFEOP state.

Operational (Op)

Before the EtherCAT MDevice switches the EtherCAT SubDevice from Safe-Op to Op it must transfer valid output data. In the Op state the SubDevice copies the output data of the MDevice to its outputs. Process data and mailbox communication is possible.

Boot

In the Boot state the SubDevice firmware can be updated. The Boot state can only be reached via the Init state. In the Boot state mailbox communication via the file access over EtherCAT (FoE) protocol is possible, but no other mailbox communication and no process data communication

4.4 Process Data Object

Process Data Communication (PDO Communication) commands and receives Process Data Objects (PDO) with MDevice periodically. Data that will be delivered and received is already defined at the initial stage of communication by PDO Mapping.
PDO communication is categorized as transmission PDO (following TxPDO) delivers controller status information and Receipt PDO (following RxPDO) delivers command from MDevice.

This communication can be used under Operational status of controller and TxPDO is only available for Safe-Operational. PDO Mapping is to set Application Objects that will be delivered and received by PDO communication.

When EtherCAT SubDevice performs PDO transmission, the actions performed after receiving a network packet can be divided into three parts:

a. PDO_OutputMapping
b. ECAT_Application
c. PDO_InputMapping

Description as follows:

1. **PDO_OutputMapping:**
   Read the Output PDO content (command issued by the user) from the packet. The reading time varies depending on the device. The user can know the time required to read the Output PDO from the packet by reading object 0x1C32.6.

2. **ECAT_Application:**
   Execute the user command read from the packet. Taking MV3S (CiA-402 mode) as an example, MV3S will start to control the 3-axis motor to rotate to the position specified by the user in ECAT_Application.

3. **PDO_InputMapping:**
   Upload read-only parameters (current motor position and speed, Digital Input Level, ADC reading value) to Input PDO. The user can know the time required to upload to Input PDO by reading object 0x1C33.6.

4.4.1 PDO Mapping

TxPDO Mapping information to be delivered to the MDevice is to be set at 1A00, 1A10, 1A20, and 1A40 Objects, and RxPDO Mapping information to receive a command from the MDevice is to be set at 1600, 1610, and 1620 Objects.

RxPDO Mapping Table:

PDO Map Object		Object Contents		
Index	**Sub**	**Object**	**Sub**	**Data Type**
0x1600	M1 Axis RxPdoMapping0			
0x1600	1	0x6040	0x00	UINT16
0x1600	2	0x607A	0x00	INT32
0x1600	3	0x60FF	0x00	INT32
0x1600	4	0x6060	0x00	INT8
0x1610	M2 Axis RxPdoMapping0			
0x1610	1	0x6840	0x00	UINT16
0x1610	2	0x687A	0x00	INT32
0x1610	3	0x68FF	0x00	INT32
0x1610	4	0x6860	0x00	INT8
0x1620	M3 Axis RxPdoMapping0			
0x1620	1	0x7040	0x00	UINT16
0x1620	2	0x707A	0x00	INT32
0x1620	3	0x70FF	0x00	INT32
0x1620	4	0x7060	0x00	INT8

Application Object List:

Object	**Sub**	**Name**
0x6040	0x00	Control Word
0x607A	0x00	Target Position
0x60FF	0x00	Target velocity
0x6060	0x00	Mode of Operation

TxPDO Mapping Table:

PDO Map Object		Object Contents			
Index	**Sub**	**Object**	**Sub**	**Data Type**	**Name**
0x1A00	M1 Axis TxPdoMapping0				
0x1A00	1	0x6041	0x00	UINT16	Statusword
0x1A00	2	0x6064	0x00	INT32	Position actual value
0x1A00	3	0x606C	0x00	INT32	Velocity Actual Value
0x1A00	4	0x60E4	0x01	-	Additional position encoder value
0x1A00	5	0x60FD	0x00	UINT32	Digital inputs
0x1A00	6	0x6061	0x00	INT8	Modes of operation display
0x1A10	M2 Axis TxPdoMapping0				
0x1A10	1	0x6841	0x00	UINT16	Statusword
0x1A10	2	0x6864	0x00	INT32	Position actual value
0x1A10	3	0x686C	0x00	INT32	Velocity Actual Value
0x1A10	4	0x68E4	0x01	-	Additional position encoder value
0x1A10	5	0x68FD	0x00	UINT32	Digital inputs
0x1A10	6	0x6861	0x00	INT8	Modes of operation display
0x1A20	M3 Axis TxPdoMapping0				
0x1A20	1	0x7041	0x00	UINT16	Statusword
0x1A20	2	0x7064	0x00	INT32	Position actual value
0x1A20	3	0x706C	0x00	INT32	Velocity Actual Value
0x1A20	4	0x70E4	0x01	-	Additional position encoder value
0x1A20	5	0x70FD	0x00	UINT32	Digital inputs
0x1A20	6	0x7061	0x00	INT8	Modes of operation display

4.4.2 PDO Assign

PDO Assign is to set PDO Mapping Object will be assigned at SyncManager.

SyncManager PDO Assign Object:

Index	Sub	Object
0x1C12	SM2 assignment	
0x1C12	1	0x1600
0x1C12	2	0x1610
0x1C12	3	0x1620
0x1C13	SM3 assignment	
0x1C13	1	0x1A00
0x1C13	2	0x1A10
0x1C13	3	0x1A20

PDO Mapping Object:

Object	Name
0x1600	Rx PDO Map0
0x1610	Rx PDO Map1
0x1620	Rx PDO Map2
…	
0x1A00	Tx PDO Map0
0x1A10	Tx PDO Map1
0x1A20	Tx PDO Map2

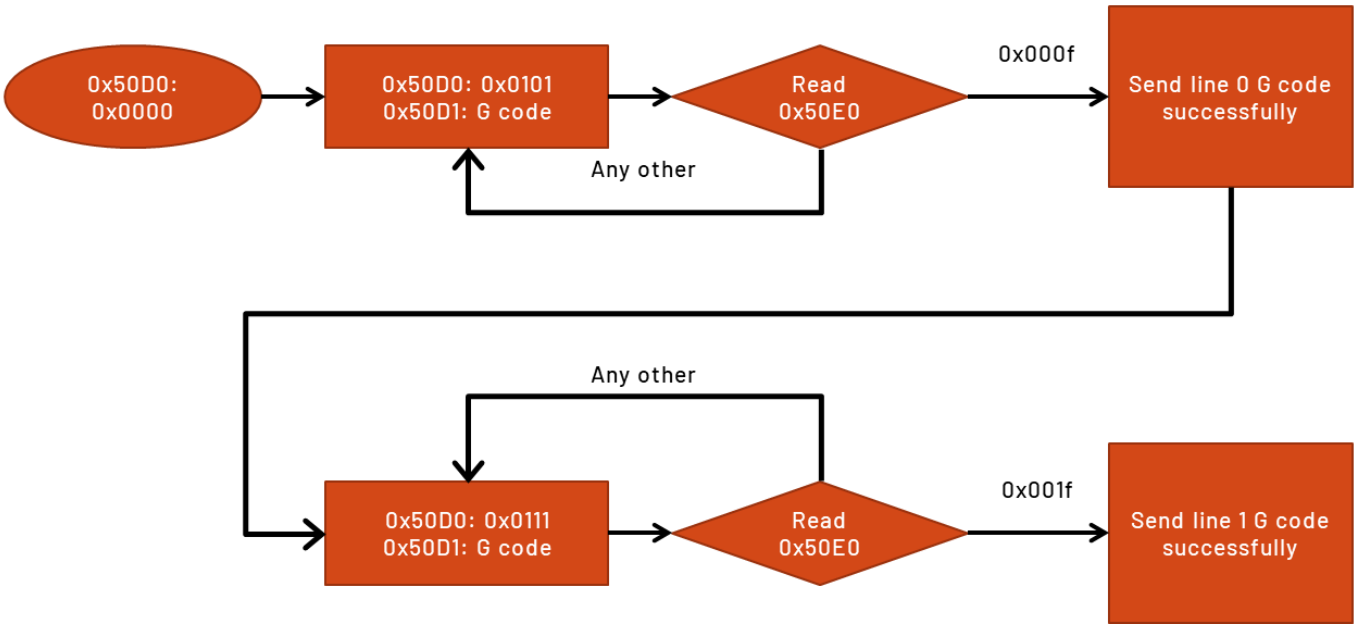
- 1C12h is object to assign RxPDO and can assign one object among RxPDO 1600, 1610, or 1620 Objects.
- 1C13h is object to assign TxPDO and can assign one object among TxPDO 1A00, 1A10, or 1A20 Objects.

4.4.3 PDO Operation Process

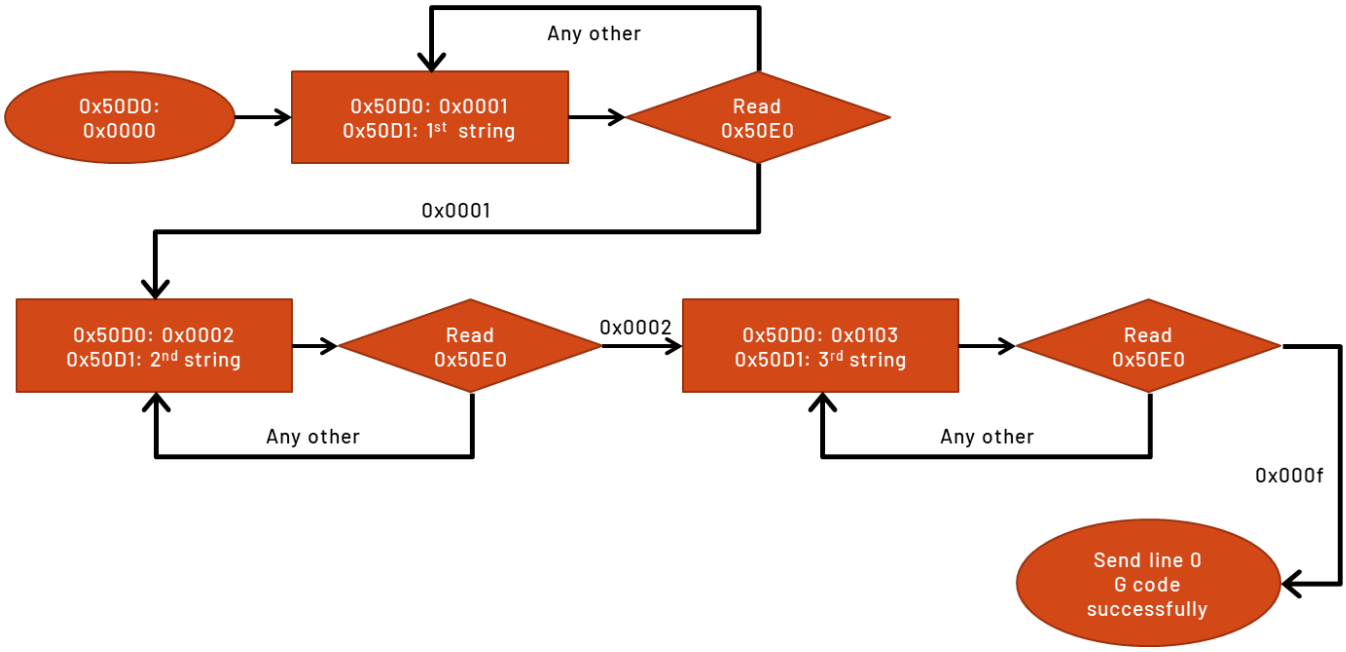
PDO Operation Process.

4.4.3.1 For G-code Mode Operation Process:

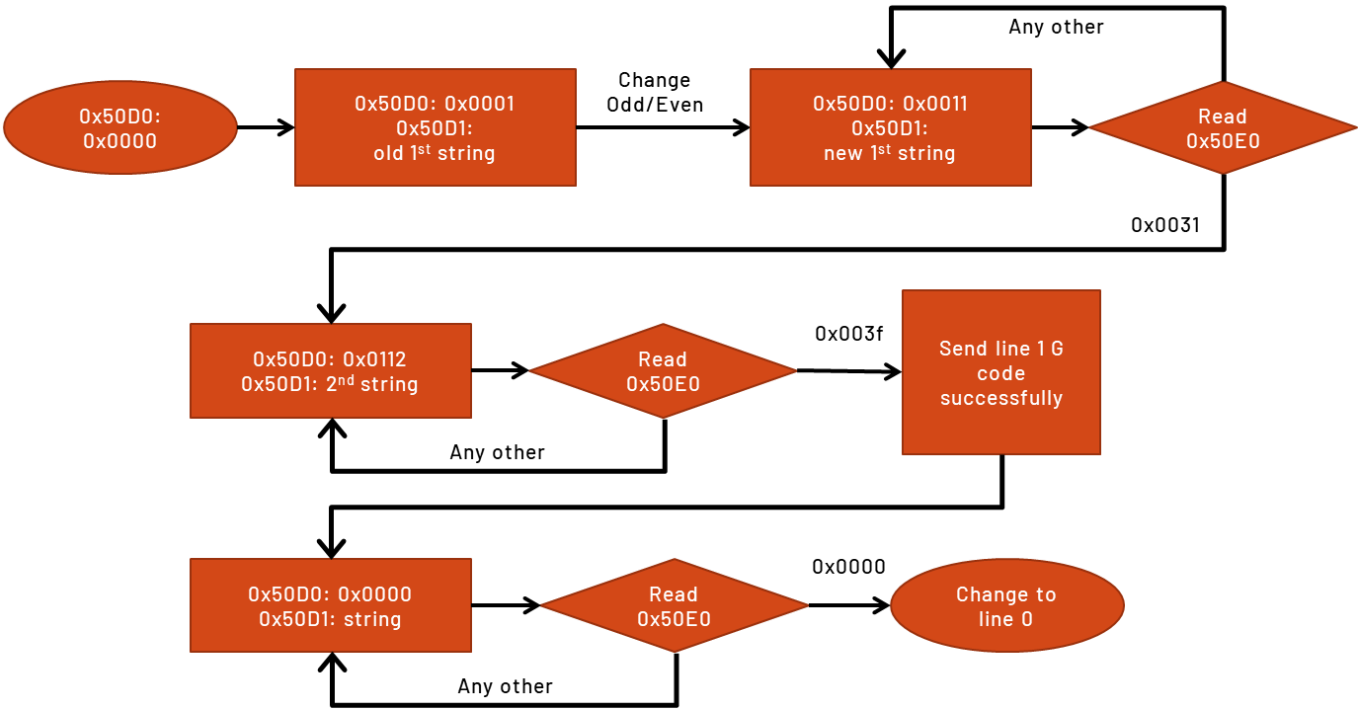
Single PDO transmission 2 lines of G code:



3 times PDO transmits 1 line of G code:



Line number reset:

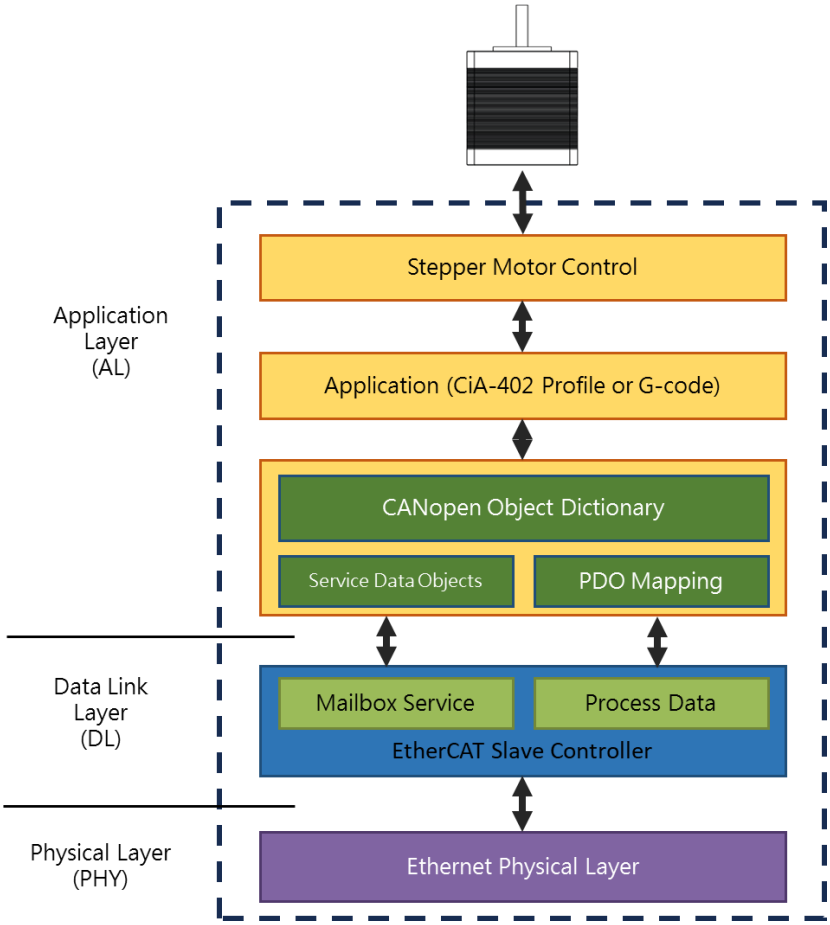


4.5 CAN application protocol over EtherCAT

The CoE interface (CAN application protocol over EtherCAT) is used for parameter management of EtherCAT devices. EtherCAT SubDevices or the EtherCAT MDevice manage fixed (read-only) or variable parameters required for operation, diagnostics, or commissioning.
CoE parameters are arranged in a table hierarchy. In principle, the user has read access via the fieldbus.

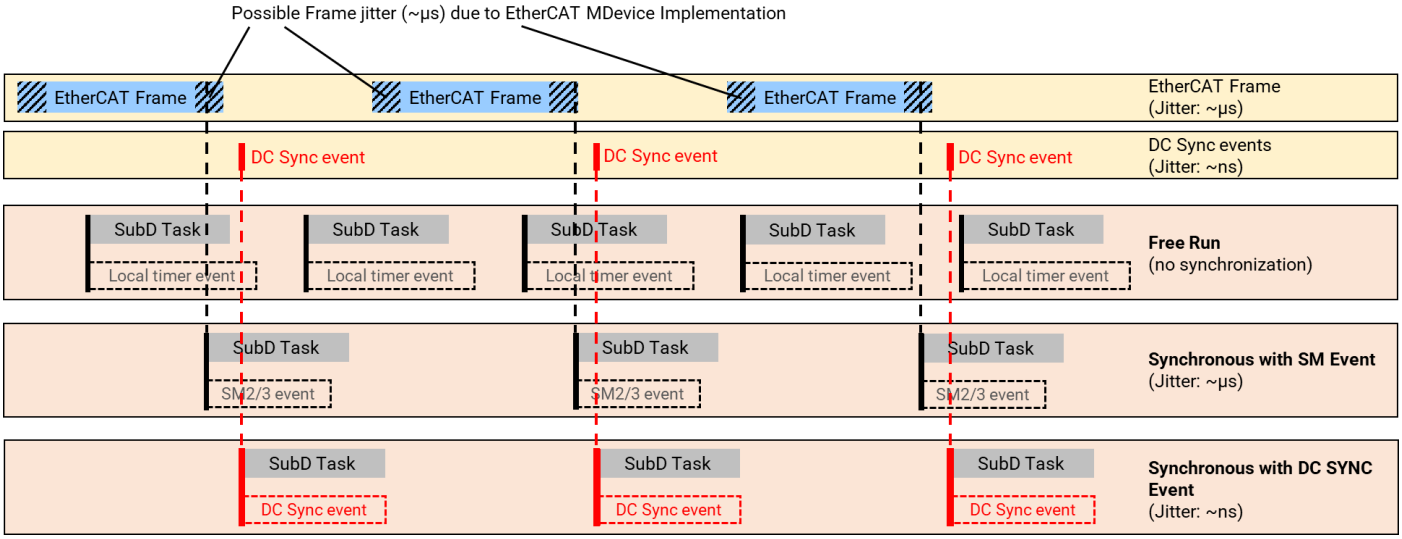
QEC-RXXMV3S supports CAN application protocol over EtherCAT (CoE).

EtherCAT SubDevice structure is as follows.



4.6 Synchronization Modes

Synchronization modes provided by QEC-RXXMV3S are as follows.



4.6.1 Free Run

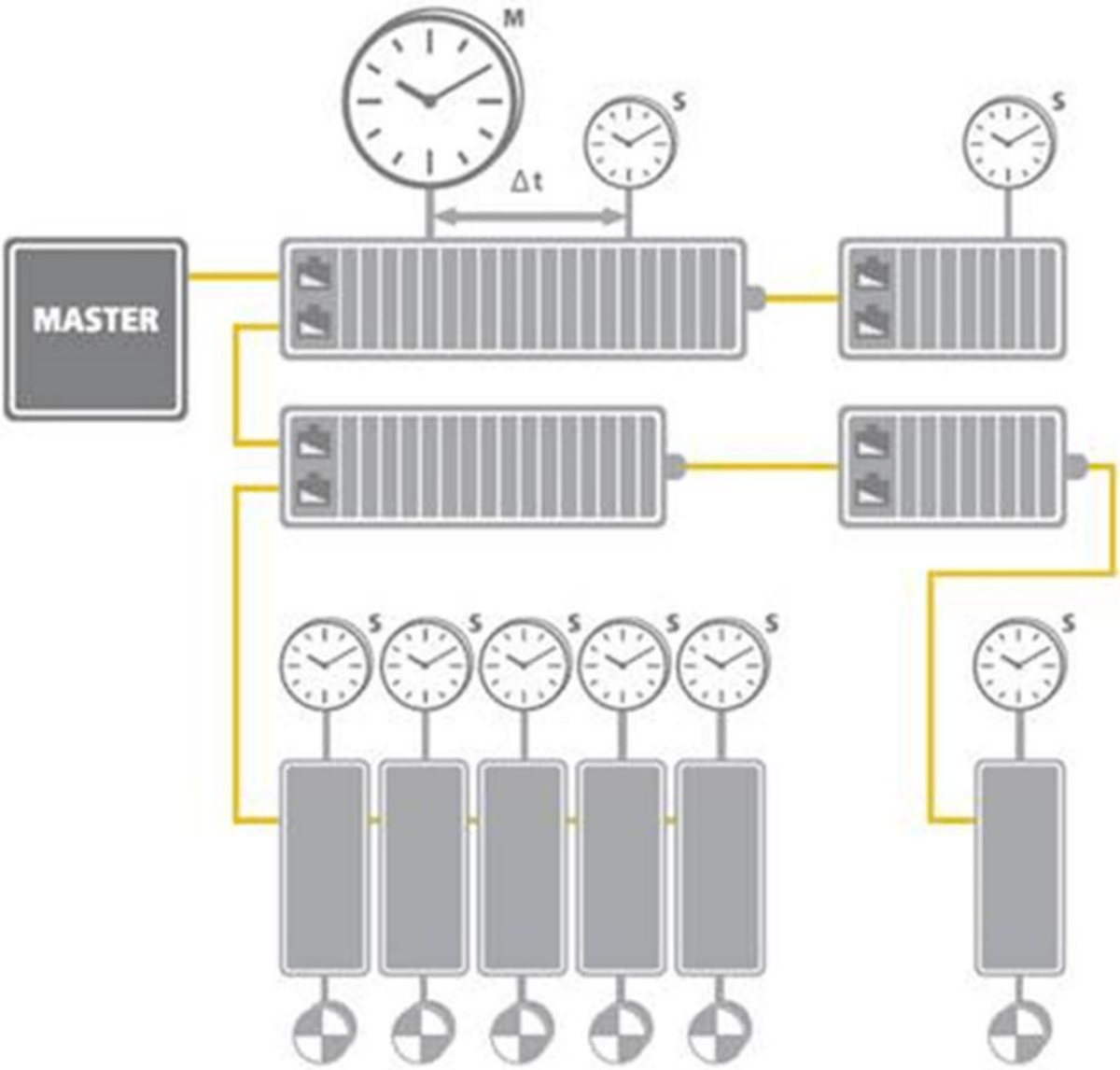
In this mode, the EtherCAT SubDevice operates independently of the EtherCAT MDevice's timing. The SubDevice does not synchronize its operations with the MDevice's clock or any other synchronization signals in the EtherCAT network. This mode is typically used in applications where precise timing alignment with other devices in the network is not critical. The device processes its tasks based on its internal clock or triggers, without waiting for external synchronization signals.

4.6.2 SM2 Event

In the SM2 synchronization mode, the EtherCAT SubDevice synchronizes its operations based on the SyncManager 2 events. These events are triggered when the SyncManager processes a passing frame, typically associated with cyclic data exchanges. This means that the SubDevice aligns its tasks, such as data acquisition or actuator control, with the specific timing of the SM2 events. This mode ensures that the SubDevice's operations are tightly coordinated with the data communication cycle, improving the consistency and predictability of the device's behavior in the network.

4.6.3 Distributed Clock

The Distributed Clock mode allows for precise synchronization of the EtherCAT SubDevice with the distributed clock system of the EtherCAT network. In this mode, the SubDevice aligns its operations with either the SYNC0 or SYNC1 events, which are part of the distributed clock system. This synchronization is crucial for applications that require highly accurate timing alignment between multiple devices. The distributed clock system ensures that all participating devices in the network are synchronized to a common time reference, minimizing the timing variations and achieving coordination with nanosecond-level precision. This mode is essential for complex motion control tasks and synchronized operations across multiple devices in an automated system.



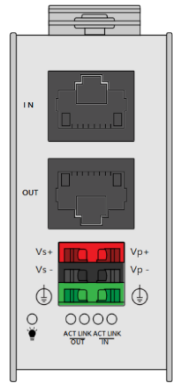
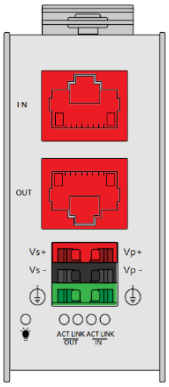
EtherCAT: Illustration of Distributed Clock (DC). (Source of information: http://www.ethercat.org/)

Ch. 5

Getting Started

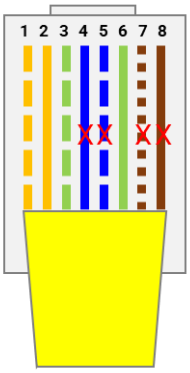
Notes QEC's PoE (Power over Ethernet)

In QEC product installations, users can easily distinguish between PoE and non-PoE: if the RJ45 house is red, it is PoE type, and if the RJ45 house is black, it is non-PoE type.

	
Non-PoE type	PoE type

PoE (Power over Ethernet) is a function that delivers power over the network.
QEC can be equipped with an optional PoE function to reduce cabling. In practice, PoE is selected based on system equipment, so please pay attention to the following points while evaluating and testing:

1. The PoE function of QEC is different and incompatible with EtherCAT P, and the PoE function of QEC is based on PoE Type B, and the pin functions are as follows:



T568B Color

PINS	10/100 DC on Spares (Mode B)
1	Rx+
2	Rx-
3	Tx+
4	DC+
5	DC+
6	Tx-
7	DC-
8	DC-

2. When connecting PoE and non-PoE devices, make sure to disconnect Ethernet cables at pins 4, 5, 7, and 8 (e.g., when a PoE-supported QEC EtherCAT MDevice connects with a third-party EtherCAT SubDevice).

3. QEC's PoE power supply is up to 24V/3A.

5.1 Introduction

Welcome to the Quick Start Guide for the QEC-RXXMV3S EtherCAT Pulse Output Module. This section helps you complete the basic setup, verify EtherCAT communication, and run a first motion test.

To streamline the process, this guide uses two software tools:

- TwinCAT (PC-based) for EtherCAT network configuration and diagnostics.
- 86Duino IDE (QEC-M Series) for building and running the sample application on the QEC-M controller.

The following pages provide step-by-step instructions for device discovery, PDO/SDO verification, and basic motion operation, so you can quickly start evaluating the module's capabilities in your system.

***Note**: This section uses QEC-RXXMP3S as an example. The setup procedure and configuration workflow are identical for QEC-RXXMV3S

5.2 TwinCAT (PP Mode)

This section describes how to commission the QEC-RXXMV3S in TwinCAT using the CiA 402 drive profile in Profile Position (PP) mode. You will complete basic EtherCAT setup, verify communication, and prepare the axis for position commands.

***Note**: For demonstration purposes, screenshots and examples in this section may reference QEC-RXXMP3S. The commissioning workflow and settings are the same for QEC-RXXMV3S. Please make sure to install the ESI file that matches your actual device model.

5.2.1 Install the ESI File

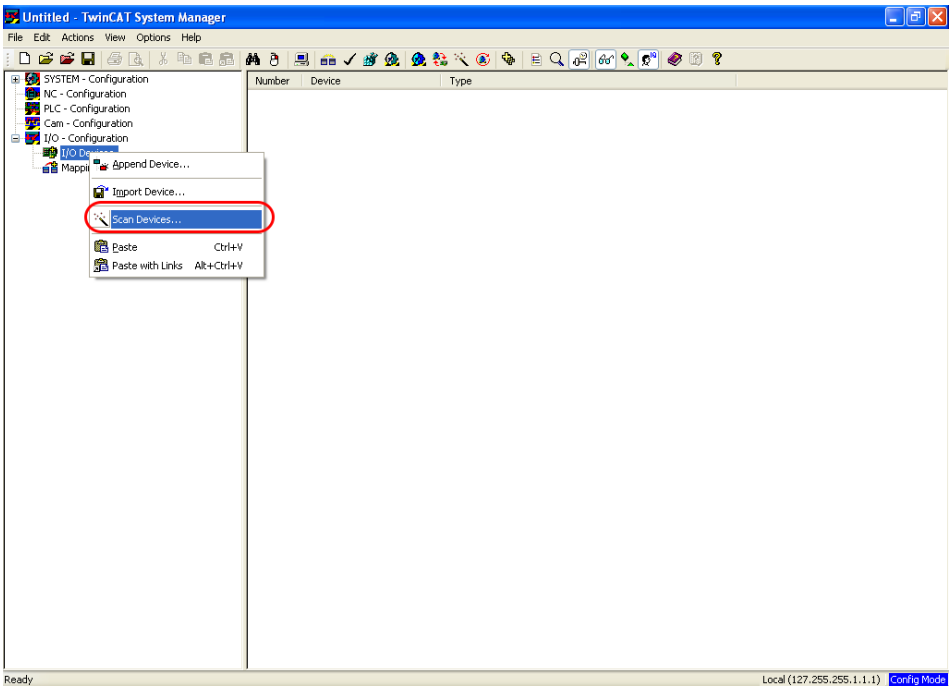
1. Download the ESI file
   Download the ESI package (ZIP) from the corresponding product page and extract it:
   - QEC-RXXMV3S (this user manual):
     https://www.qec.tw/product/qec-rxxmv3s/
   - QEC-RXXMP3S (example used in this section):
     https://www.qec.tw/product/qec-rxxmp3s/
2. Install the ESI file
   Copy the extracted ESI .xml file to the TwinCAT EtherCAT device description directory (administrator permission may be required).
   e.g. C:\TwinCAT\3.1\Config\Io\EtherCAT\
3. Restart TwinCAT (if required)
   If TwinCAT was running while the ESI file was copied, restart TwinCAT XAE (and/or restart the TwinCAT System service) so the new device description can be loaded.
   If the device still does not appear, reboot the PC.

***Note**: Ensure the ESI (.xml) version matches the module firmware/ESI version provided by ICOP.

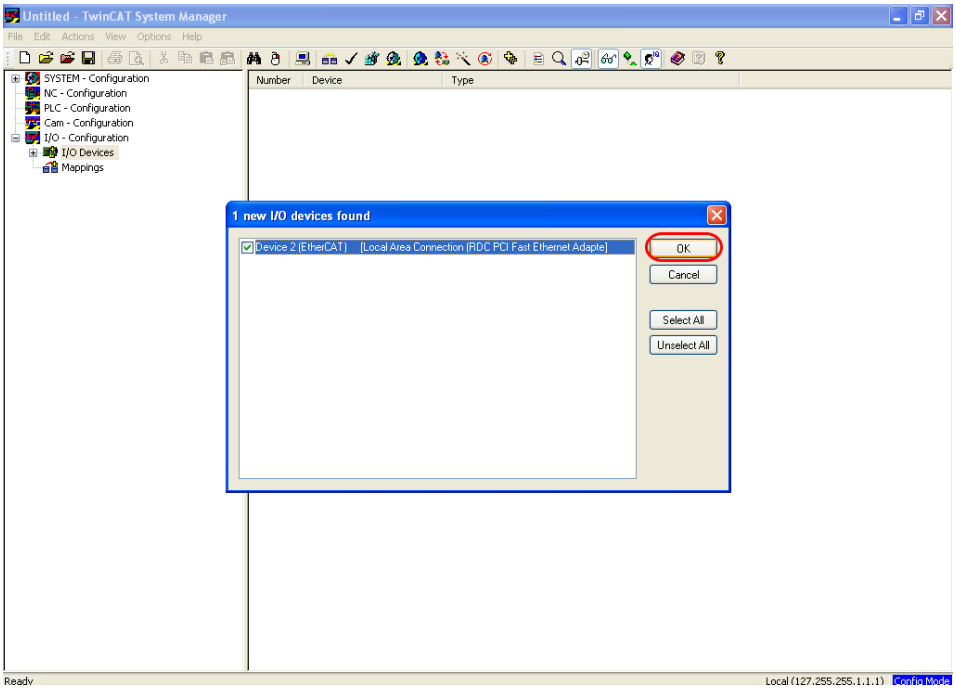
5.2.2 Add the QEC-R11MP3S to the Project

This section assumes that the TwinCAT software is in Config Mode.

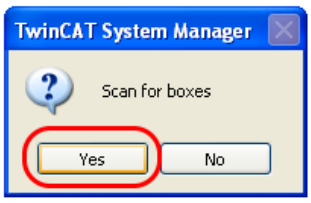
1. Scan for the QEC-RXXMP3S device.
   Right-click on the EtherCAT adapter that the QEC-RXXMP3S is attached to. In the drop-down menu that opens, select the "**Scan Device**" or "**Scan**" option.
   (If the "**Scan**" option is not available, the TwinCAT software is not in Config Mode.)



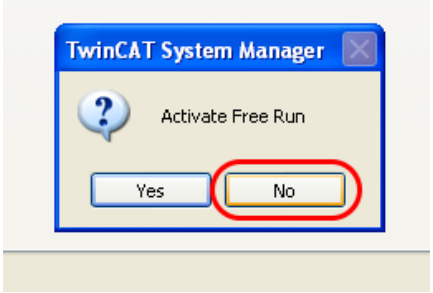
2. Choose the EtherCAT connection network.



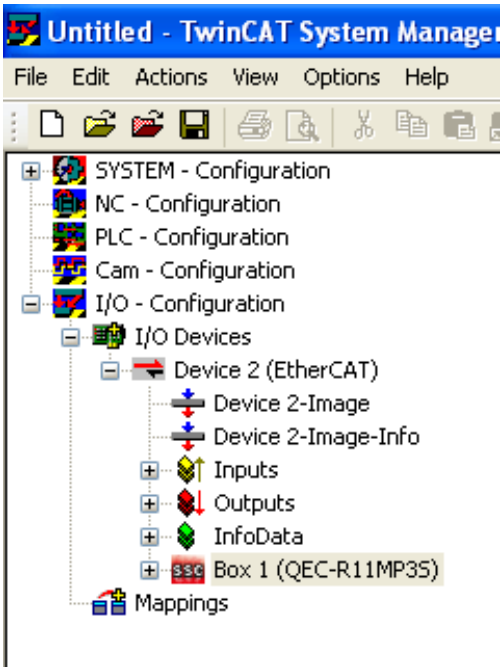
3. Confirm “**Yes**” to start the scan.



4. Choose “**No**” when TwinCAT asks you to activate Free Run.
   We need to set up the PP mapping to the PDO before operation.

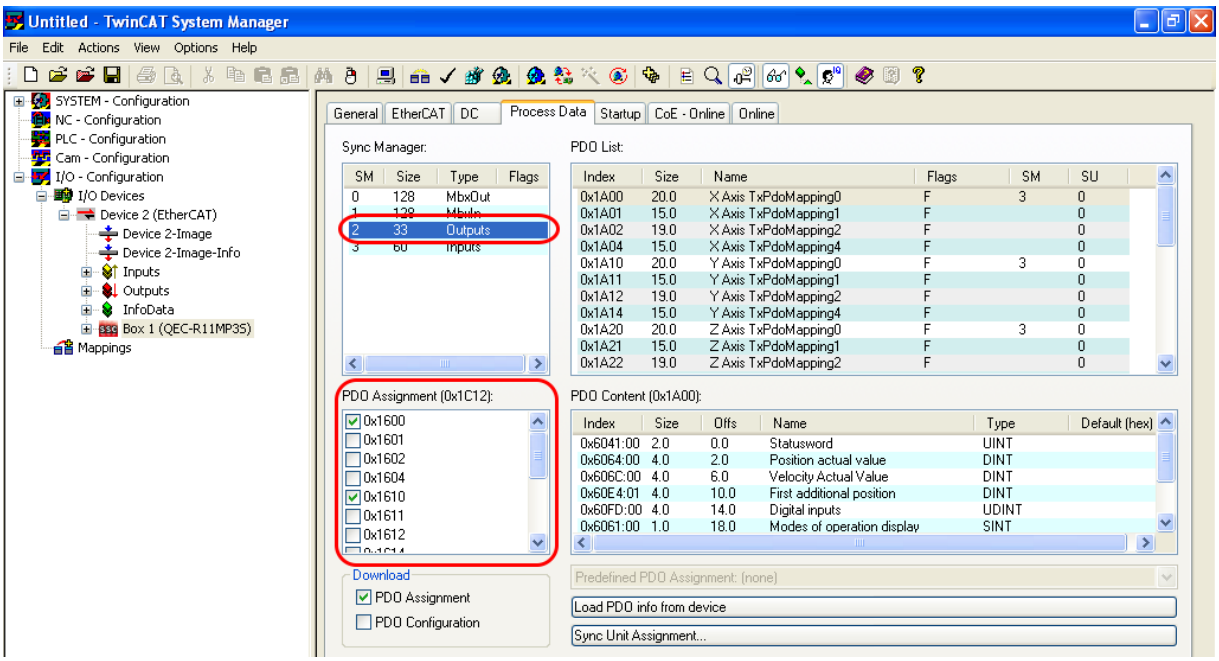


5. The QEC-R11MP3S will appear in the device tree and the name will typically begin with “**Box**”.

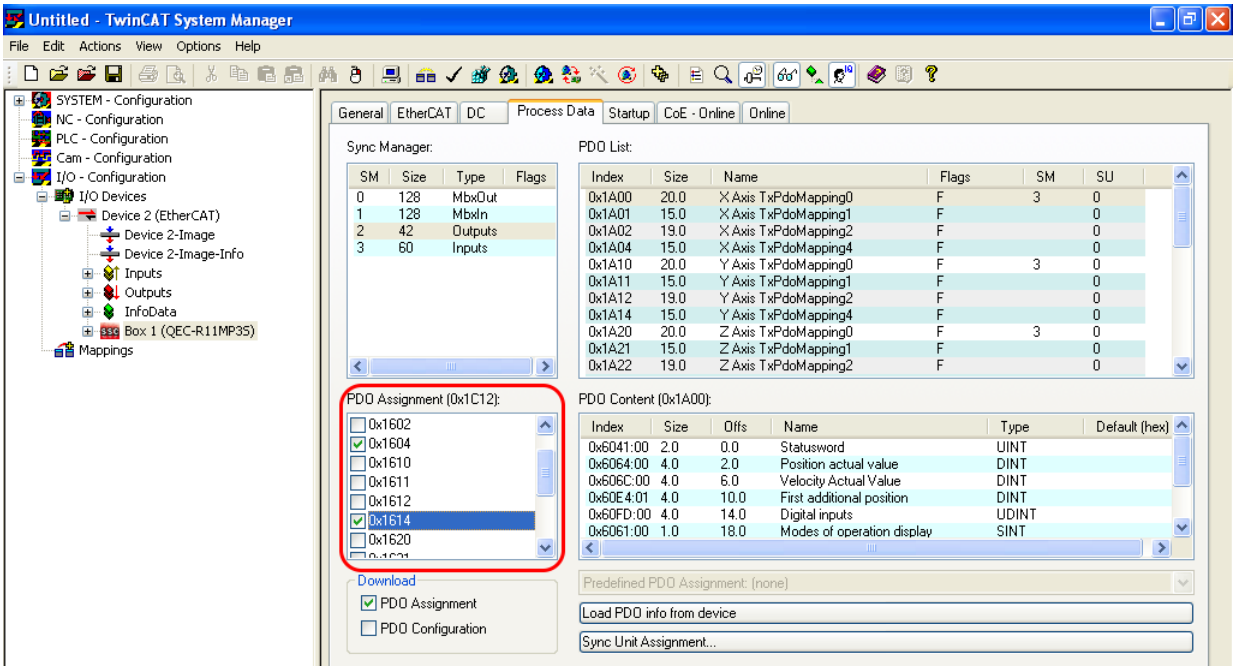


5.2.3 Start to Configure the QEC-R11MP3S

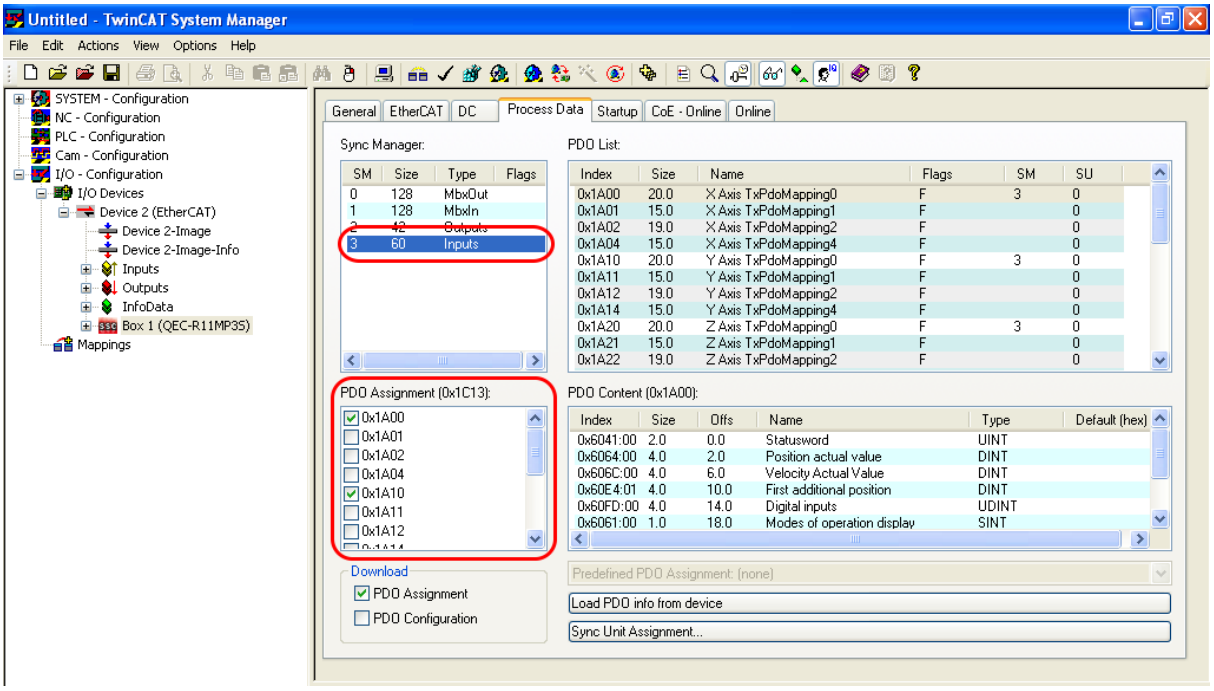
1. Click on the “**Process Data**” tab, and click "**2 33 Outputs**" in the "**Sync Manager:**" field to set Outputs PDO Mapping.
   The original check boxes in the "**PDO Assignment (0x1C12):**" field are 0x1600, 0x1610, and 0x1620. The default check settings are is used in CSP or CSV mode.



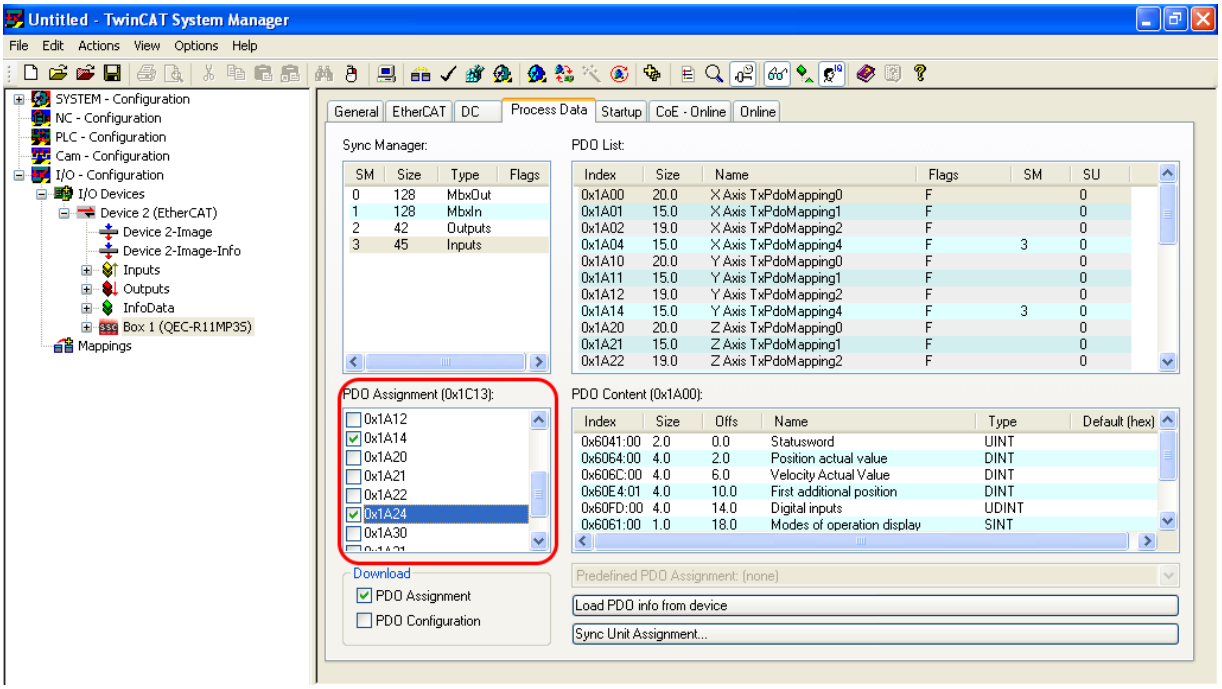
2. Change the checked items to **0x1604**, **0x1614**, and **0x1624**, which modify the Outputs PDO Mapping to suit PP mode.



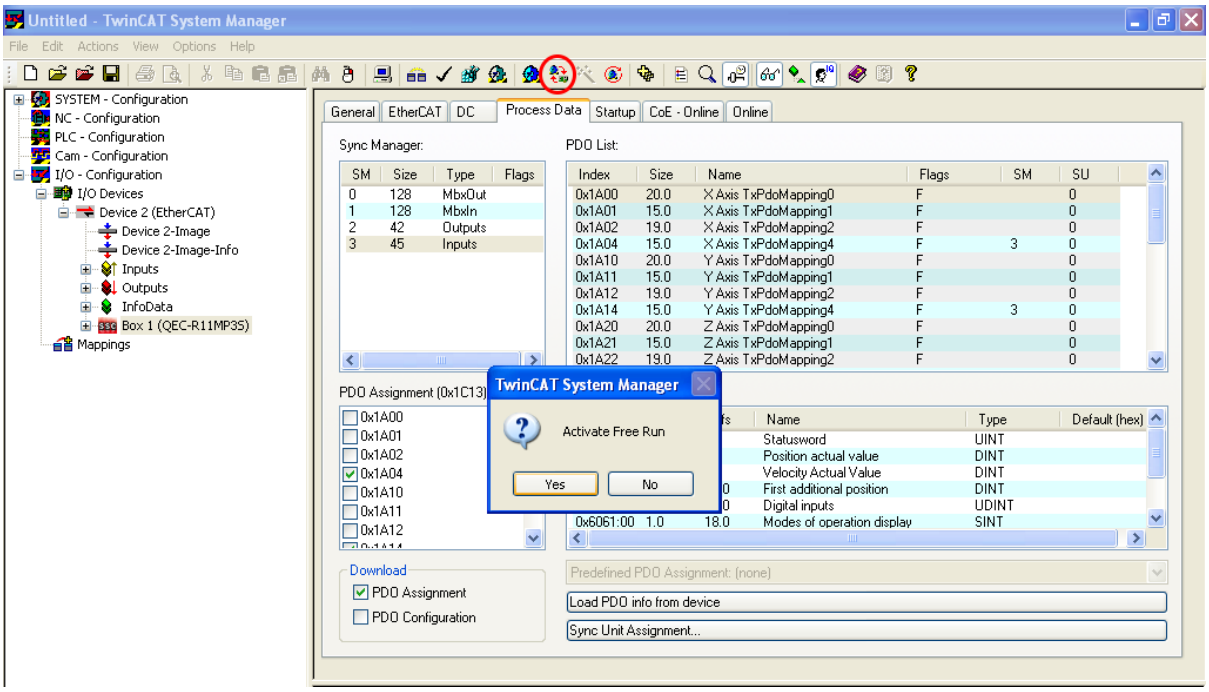
3. Click "**3 60 Inputs**" in the "**Sync Manager:**" column to set Inputs PDO Mapping. The original check in the "**PDO Assignment (0x1C13):**" column is 0x1A00, 0x1A10, 0x1A20, and the default check setting is used in **CSP** or **CSV** mode.



4. Change the checked items to **0x1A04**, **0x1A14**, and **0x1A24**, which modify the Inputs PDO Mapping to suit PP mode.



5. After setting, click "**Reload I/O Devices (F4),**" marked in red in the picture below, to update the PDO Mapping settings.

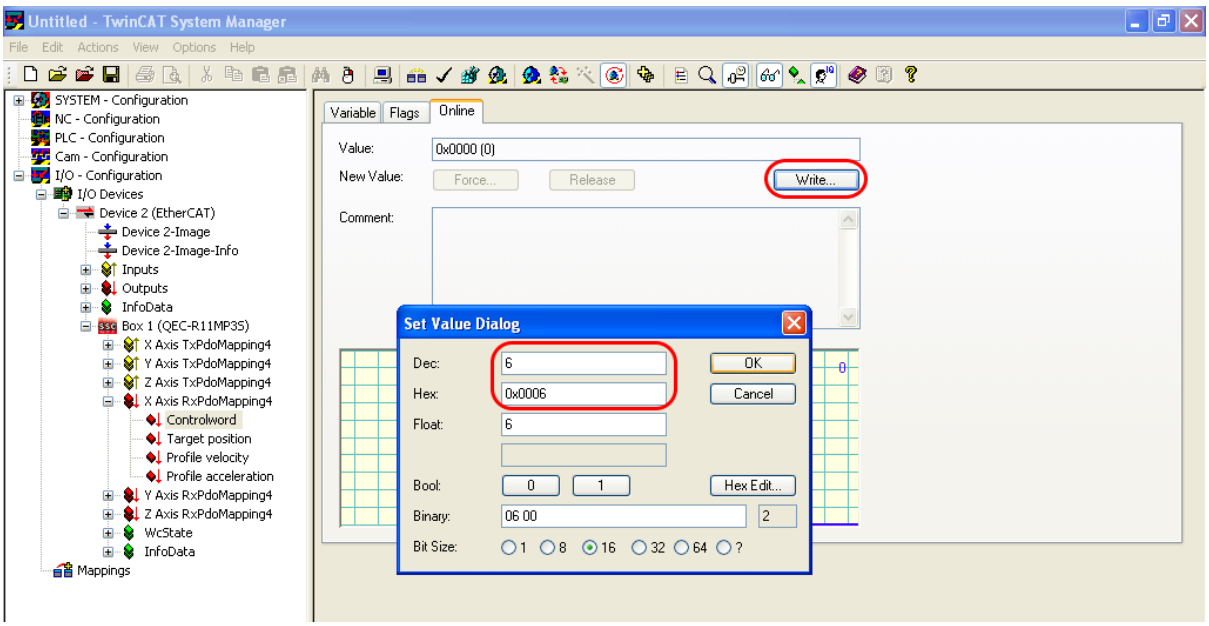


6. After clicking, a window will pop up asking whether to "**Activate Free Run.**" Please click "**Yes**".

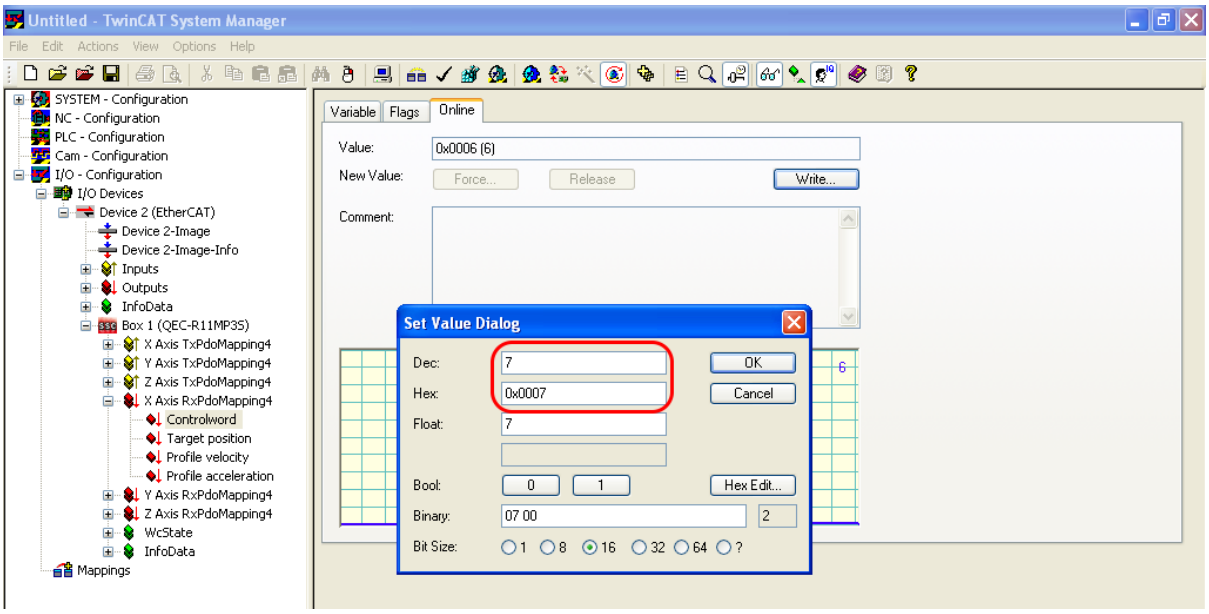
5.2.4 Control the QEC-R11MP3S

Next, please change the Controlword in order to let the CiA-402 state machine enter the Operation Enable state.

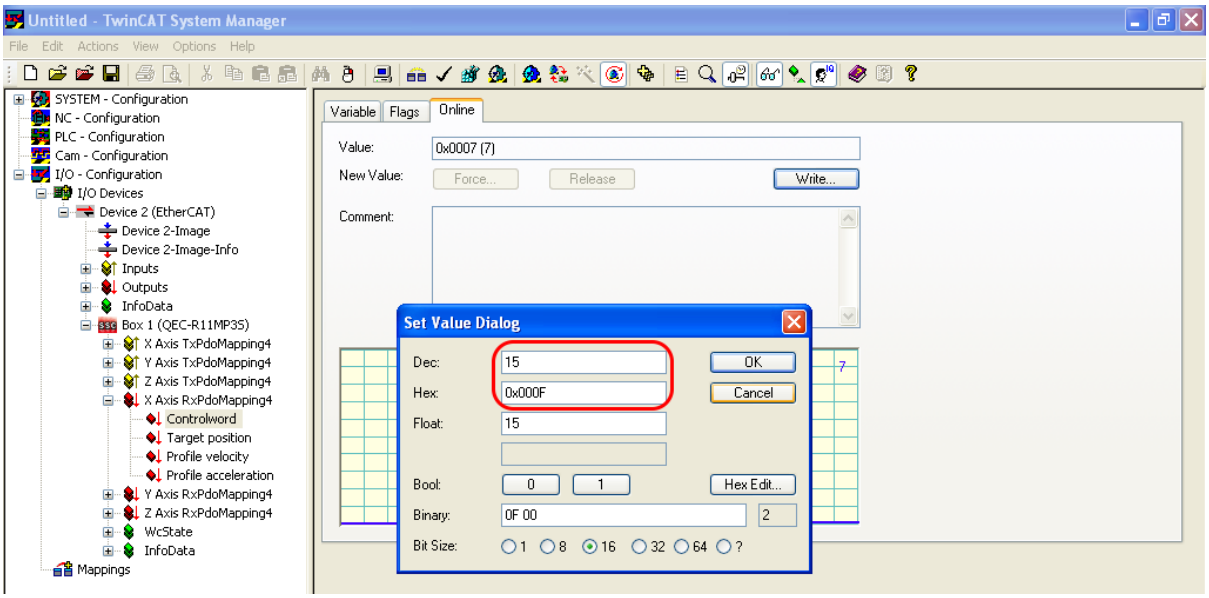
1. Select the drop-down list of QEC-R11MP3S in the left window. And choose “**Controlword**” in the drop-down list of the "**M1 Axis RxPdoMapping4**".
   Click the "**Online**" page of "**Controlword**" and click the "**Write...**" button. Enter 6 in the pop-up window to change the value of Controlword to 6.



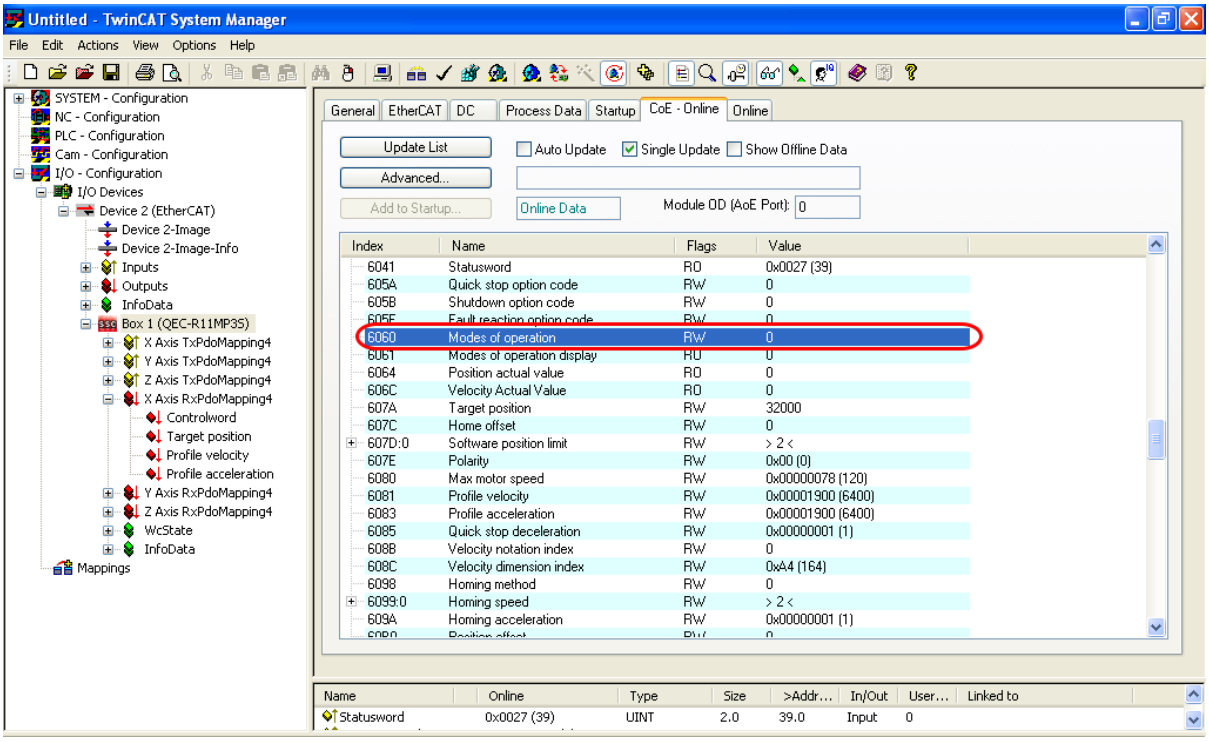
2. Then, enter 7 in the pop-up window to change the value of Controlword to 7.



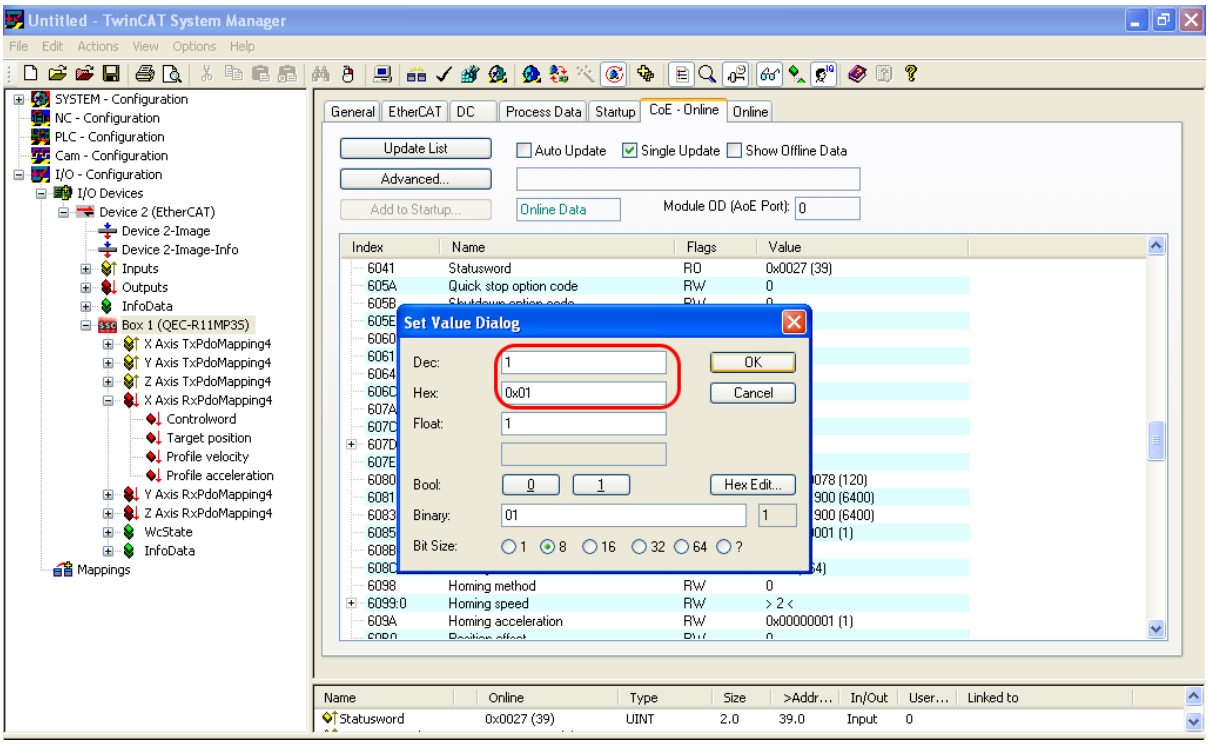
3. Again, enter 15 in the pop-up window to change the value of Controlword to 15.



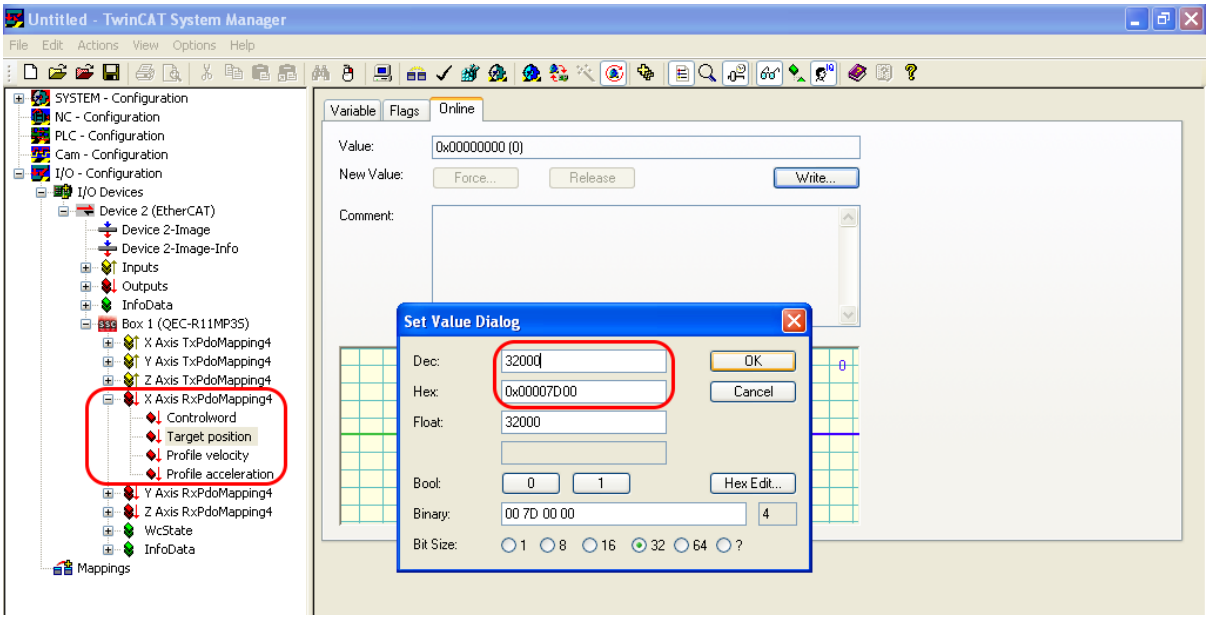
4. Next, select the BOX 1(QEC-R11MP3S) in the left window.
   Click the "**CoE - Online**" page to view all objects and perform Mailbox transmission. Find object **0x6060** and double-click the object with the left mouse button.



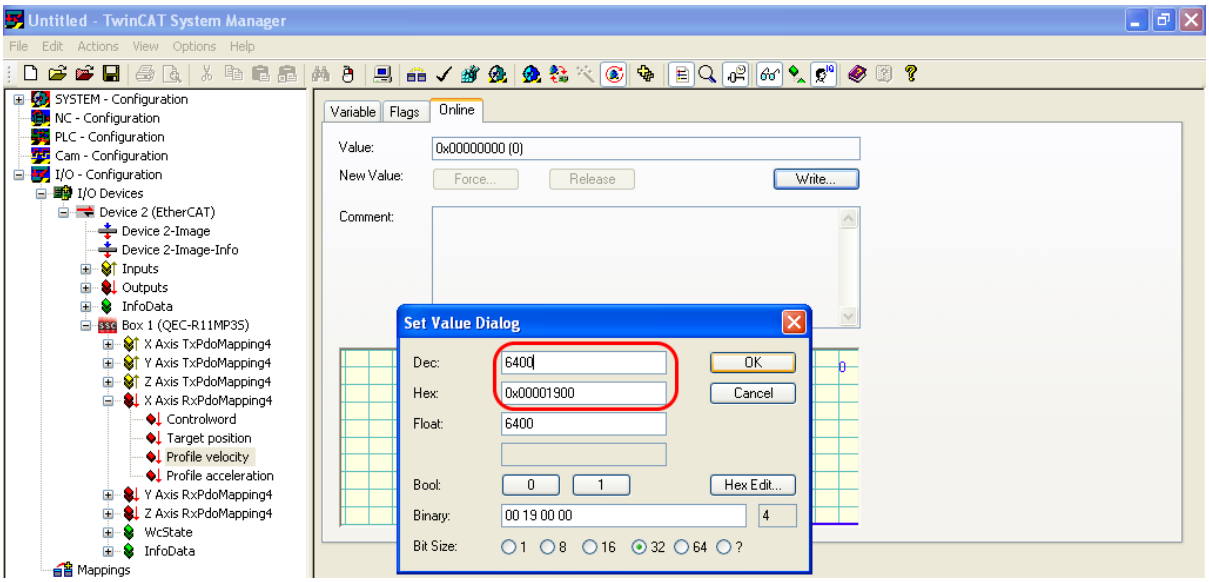
5. Enter 1 in the pop-up window to set the value of object 0x6060 to 1, and specify the operation mode to Profile Position (PP) mode.



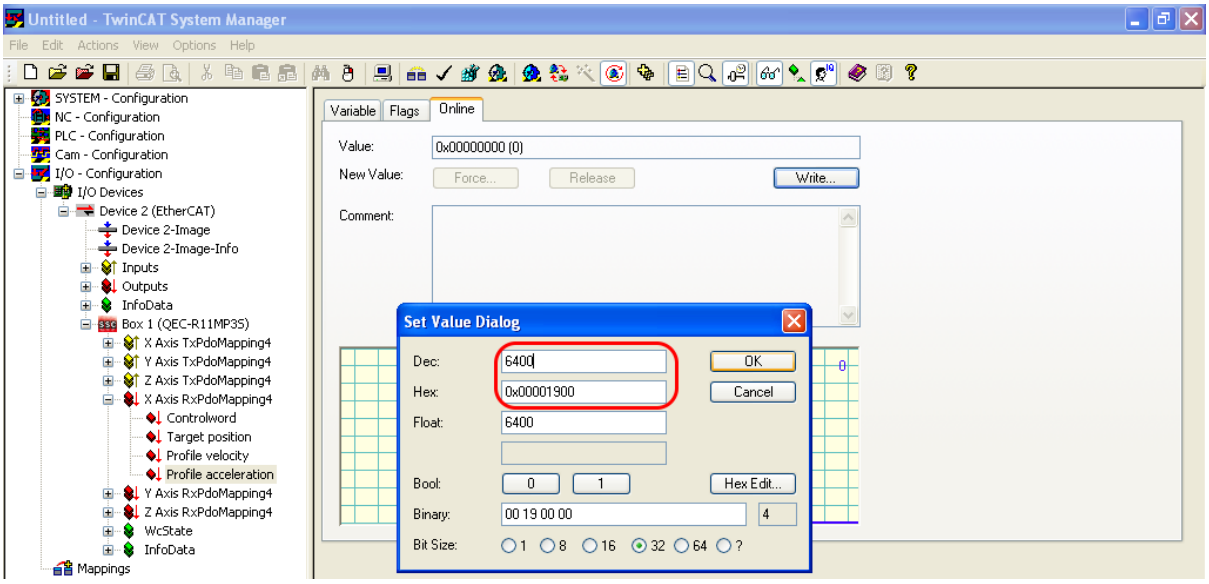
6. Expand "**M1 Axis RxPdoMapping4**" and click "**Target position**", click "**Write...**" on the "Online" page and enter the position command 32000.



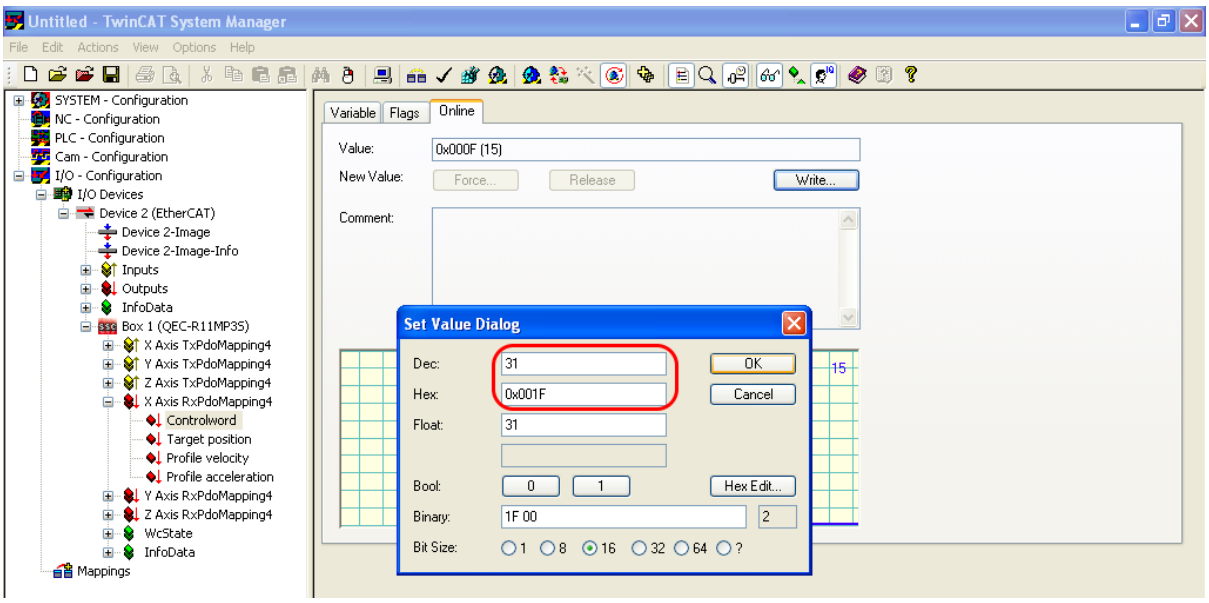
7. Set Profile velocity to 6400.



8. Set Profile acceleration to 6400.

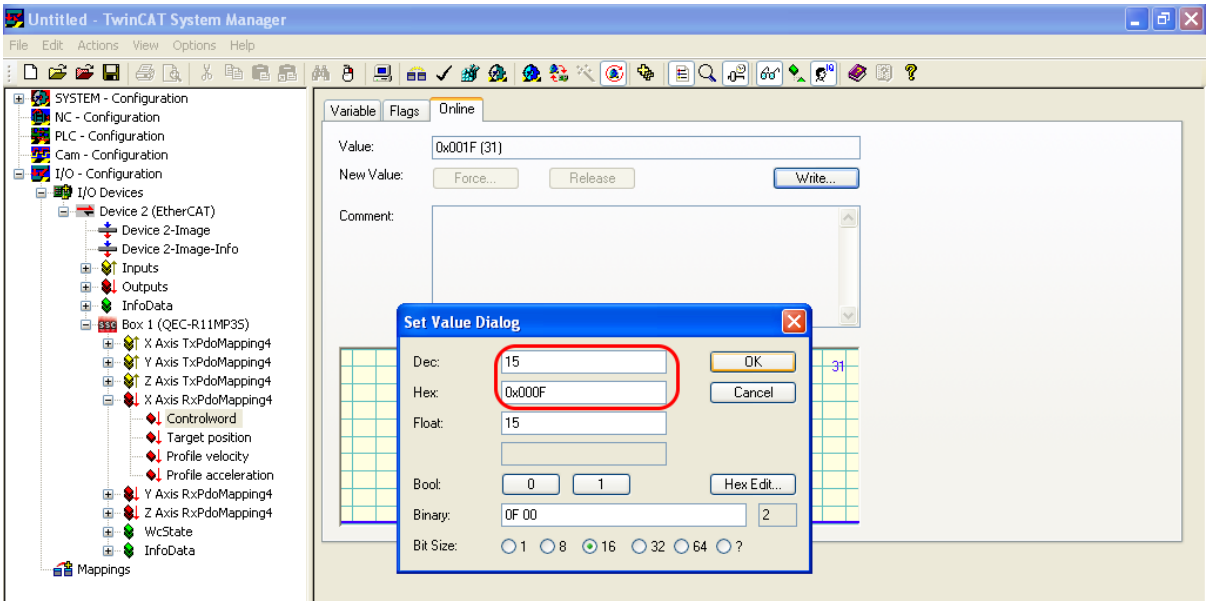


9. Set Controlword to 0x001F to make the New set-point bit 1.



Then, your QEC-R11MP3S starts to drive the motor to rotate for 32,000 steps with a velocity of 6,400 and an acceleration of 6,400.

10. When the New set-point bit changes from 0 to 1, MP3S will receive the new Target position, Profile velocity, and Profile acceleration commands. Therefore, when the motor rotates to the target position, the Controlword must be set to 0x000F to make the New set-point Bit is 0.



5.3 86Duino Coding IDE (PP Mode)

This section introduces how to use the 86Duino IDE to develop an application on the **QEC-M Series** EtherCAT master (MDevice) and control the QEC-RXXMV3S EtherCAT Pulse Output Module (SubDevice) in CiA 402 Profile Position (PP) mode.
With an Arduino-compatible programming environment, the 86Duino IDE provides a straightforward workflow to write, build, and deploy control logic to the QEC-M controller, enabling quick validation and integration of EtherCAT devices.

In this chapter, you will learn how to compile and upload a sample project to the QEC-M, establish EtherCAT communication, and execute basic position commands in PP mode. The examples also demonstrate how to read status information and handle common initialization steps required for motion operation.

Example setup used in this section:

- MDevice: **QEC-M-01**
- SubDevice: **QEC-RXXMV3S**
- Operation mode: CiA 402 Profile Position (PP)

***Note**: The programming workflow and PP-mode control sequence remain the same; please use the ESI/firmware and sample project that match your actual device model.

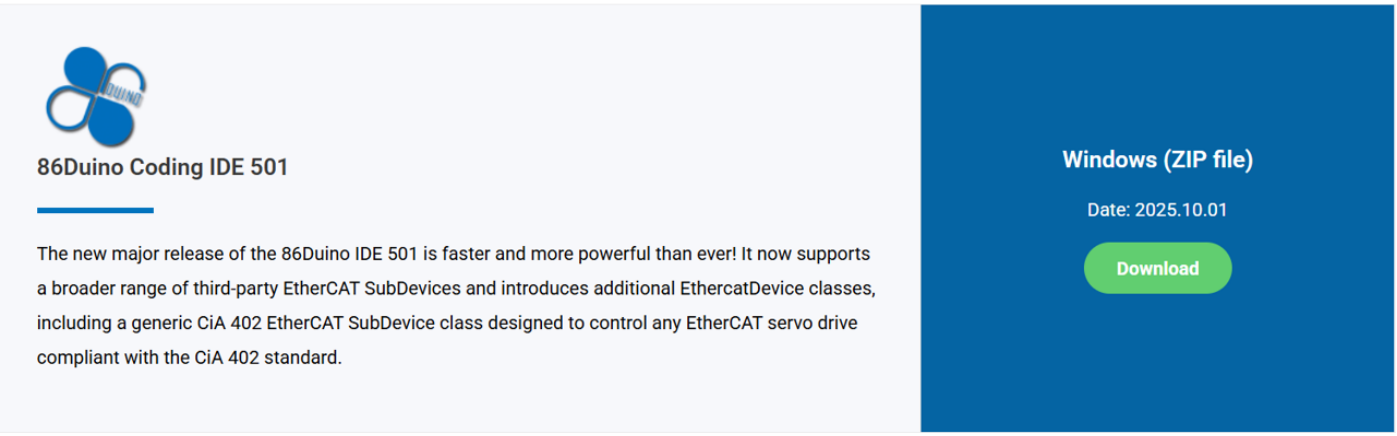
5.3.1 Connection and wiring hardware

The following devices are used here:

1. QEC-M-01（EtherCAT MDevice /PoE）
2. QEC-R11MV3S (EtherCAT 3-axis Pulse Output module)
3. 24V power supply & EU-type terminal cable & LAN cable

5.3.2 Download Software

Download 86duino IDE from https://www.qec.tw/software/.



After downloading, please unzip the downloaded zip file, no additional software installation is required, just double-click 86duino.exe to start the IDE.

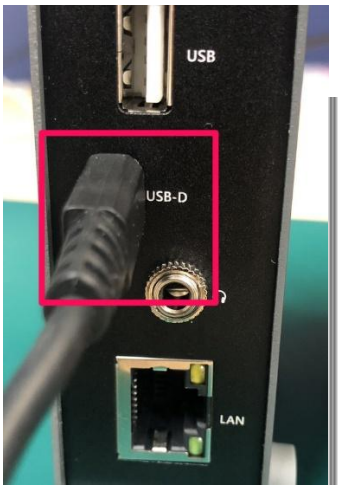
* **Note**: If Windows displays a warning, click Details once and then click the Continue Run button once.

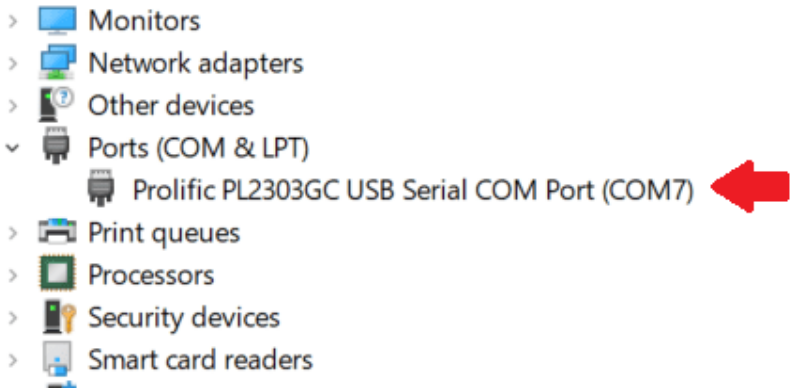
86Duino Coding IDE 501+ looks like below.

5.3.3 Connect to your PC and set up the environment

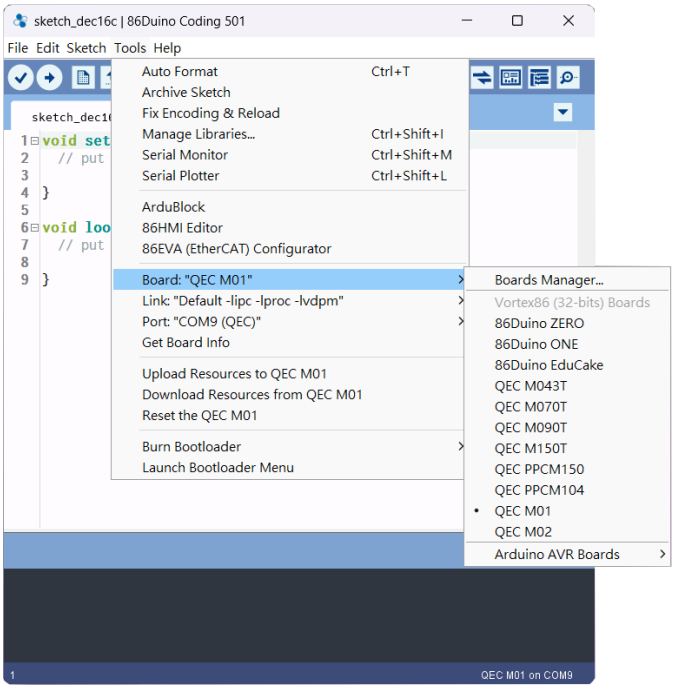
Follow the steps below to set up the environment:

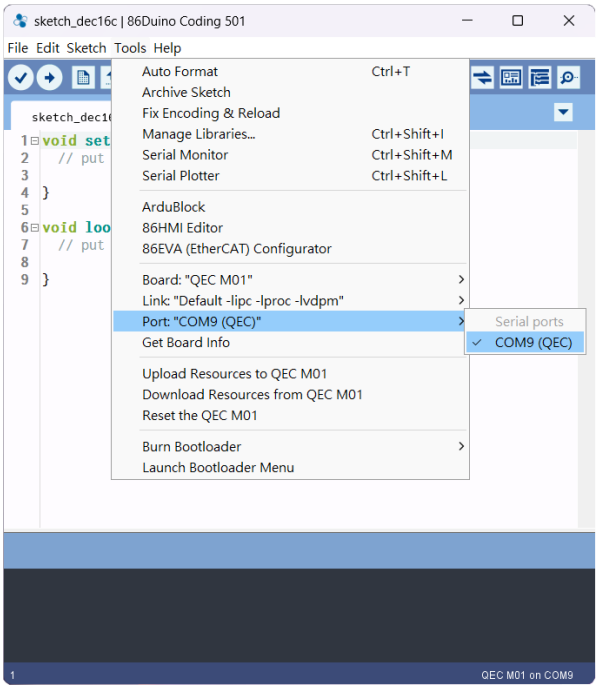
1. Connect the QEC-M-01 to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open "**Device Manager**" (select in the menu after pressing Win+X) ->" **Ports (COM & LPT)**" in your PC and expand the ports; you should see that the "**Prolific PL2303GC USB Serial COM Port (COMx)**" is detected; if not, you will need to install the required drivers.
   (For Windows PL2303 driver, you can download here)





4. Open the 86Duino IDE.
5. Select the correct board: In the IDE's menu, select **Tools**> **Board** > **QEC-M-01** (or the QEC MDevice model you use).
6. Select Port: In the IDE's menu, select **Tools** > **Port** and select the USB port to connect to the QEC MDevice (in this case, COM9 (QEC)).





5.3.4 Write code

The EtherCAT MDevice (QEC-M-01) and the QEC-RXXMV3S (EtherCAT pulse output module) can be configured and programmed via the EtherCAT library in the 86Duino IDE.
The Arduino development environment has two main parts: `setup()` and `loop()`, which correspond to initialization and main programs. Before operating the EtherCAT network, you must configure it once. The process should be from Pre-OP to OP mode in EtherCAT devices.
In the next sections, we provide step-by-step examples showing how to control one axis and three axes using CiA 402 Profile Position (PP) mode

5.3.4.1 Single Axis

In this section, we will control a single axis on the QEC-R11MV3S. We will read the current position of the motor, continuously printing it to the Serial Monitor, and use a simple state machine to move the motor back and forth between two target positions.

The following program sets the QEC-R11MV3S into CiA402 Profile Position (PP) mode:

- EtherCAT Cycle Time: 1 millisecond.
- EtherCAT Mode: ECAT_SYNC.

The EthercatMaster object ("master") represents the QEC-M-01, while the EthercatDevice_QECR11MV3S object ("slave0") represents the QEC-R11MV3S, and the EthercatDevice_CiA402 object ("motor") represents a CiA402 identifier to utilize the CiA402 Library for the specified motor driver.

A. In Setup Function:

In the `setup()` function, the program initializes communication, starts the EtherCAT MDevice, and configures Motor 1 of the QEC-R11MV3S for CiA402 Profile Position (PP) mode.
Follow the steps below:

1. Start serial communication at a baud rate of 115200.
2. Start the EtherCAT MDevice
   - Use the `begin()` function to initialize the EtherCAT MDevice.
3. Attach the QEC-R11MV3S EtherCAT SubDevice
   - Use the `attach()` function to attach the QEC-R11MV3S as slave0 to the EtherCAT.
4. Get the CiA402 Motor Instance
   - Call `cia402GetServo(1)` to obtain a pointer to Motor 1 on the QEC-R11MV3S.
5. Set the CiA402 Mode to Profile Position (PP) Mode
   - Configure the motor to PP mode using `motor->setCiA402Mode(CIA402_PP_MODE)`
6. Start the EtherCAT MDevice
   - Use the `start(1000000, ECAT_SYNC)` function to switch the EtherCAT state machine to the OPERATIONAL state.
   - The cycle time is set to 1,000,000 ns (1 ms) with ECAT_SYNC mode.
7. Enable the Motor
   - Call `motor->enable()` to enable Motor 1 and transition it to the CIA402_OPERATION_ENABLED state.
8. Configure Profile Parameters
   - Set the motion profile parameters for Profile Position mode:
     - Motion Profile Type:
       Linear ramp using `motor->pp_SetMotionProfileType(0)`
     - Profile Velocity: 10,000 using `motor->pp_SetVelocity(10000)`
     - Acceleration: 10,000 using `motor->pp_SetAcceleration(10000)`
     - Deceleration: 10,000 using `motor->pp_SetDeceleration(10000)`

B. In Loop Function:

In the `loop()` function, the current position of the motor is continuously printed on the Serial Monitor, and a simple state machine is used to move the motor back and forth between two target positions.

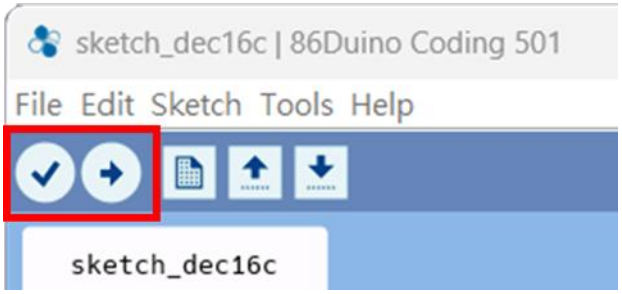
1. State Machine Logic
   - case 0:
     - Call `motor->pp_Run(100000)` to move Motor 1 to the positive target position (+100,000 units).
     - When `pp_Run()` returns 0 (command accepted), advance to case 1.
   - case 1:
     - Wait for the motor to reach the target position.
     - Use `motor->pp_IsTargetReached()` to check whether the target position has been reached.
     - Once the function returns true, advance to case 2.
   - case 2:
     - Call `motor->pp_Run(-100000)` to move Motor 1 to the negative target position (−100,000 units).
     - When `pp_Run()` returns 0, advance to case 3.
   - case 3:
     - Again use `motor->pp_IsTargetReached()` to wait until the motor reaches the negative target position.
     - When the target is reached, reset pp_state to 0 to repeat the movement cycle.
2. Code Logic Summary
   - The current position is displayed using `motor->getPositionActualValue()` and printed as "Pos: <value>" on the Serial Monitor.
   - `pp_Run()` is used to send a new Profile Position command to the motor.
   - `pp_IsTargetReached()` is used to confirm when the commanded position has been reached.
   - The state machine starts at case 0 and continuously cycles through case 0–case 3 to alternately move the motor between +100,000 and −100,000 position units.

The example code is as follows:

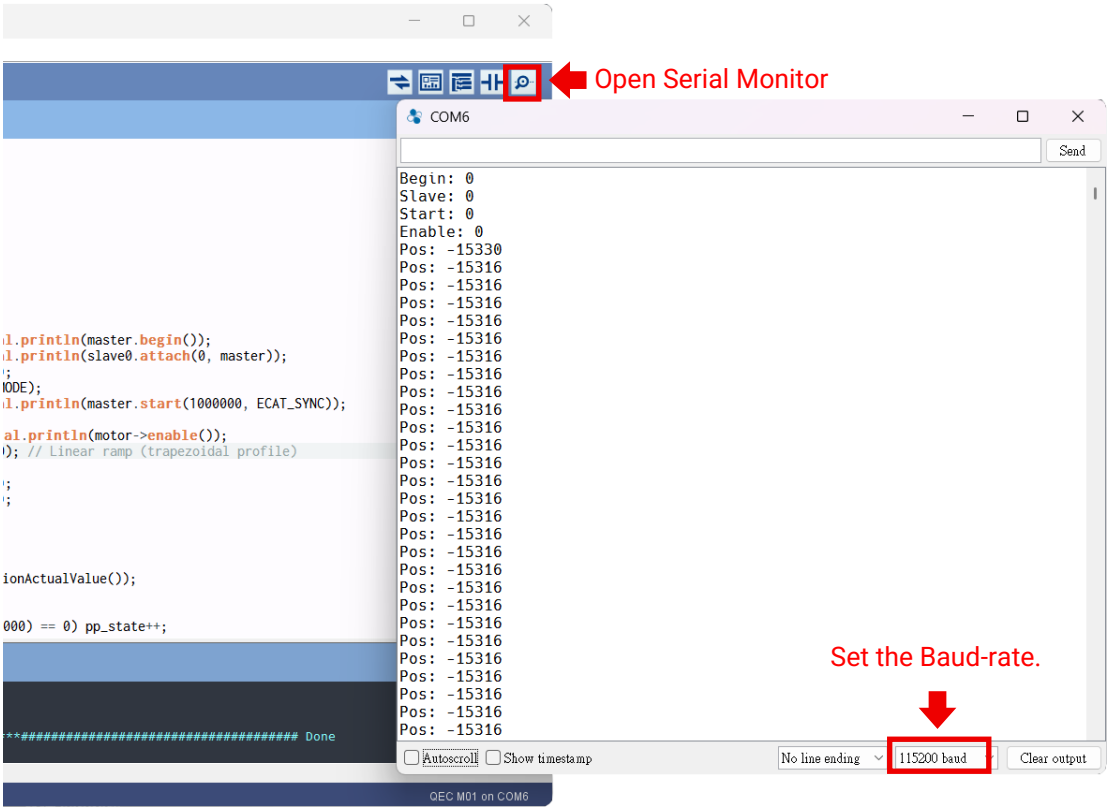
```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_QECR11MV3S slave0;
EthercatDevice_CiA402 *motor;

int pp_state = 0;

void setup() {
  Serial.begin(115200);
```

```
  while (!Serial);

  Serial.print("Begin: "); Serial.println(master.begin());
  Serial.print("Slave: "); Serial.println(slave0.attach(0, master));
  motor = slave0.cia402GetServo(1);
  motor->setCiA402Mode(CIA402_PP_MODE);
  Serial.print("Start: "); Serial.println(master.start(1000000, ECAT_SYNC));

  Serial.print("Enable: "); Serial.println(motor->enable());
  motor->pp_SetMotionProfileType(0); // Linear ramp (trapezoidal profile)
  motor->pp_SetVelocity(10000);
  motor->pp_SetAcceleration(10000);
  motor->pp_SetDeceleration(10000);
}

void loop() {
  Serial.print("Pos: ");
  Serial.println(motor->getPositionActualValue());

  switch (pp_state) {
    case 0: if (motor->pp_Run(100000) == 0) pp_state++;
      break;
    case 1: if (motor->pp_IsTargetReached()) pp_state++;
      break;
    case 2: if (motor->pp_Run(-100000) == 0) pp_state++;
      break;
    case 3: if (motor->pp_IsTargetReached()) pp_state = 0;
      break;
  }
}
```

***Note:** Once the code is written, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload. The program will run when the upload is complete.



After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on 86Duino IDE. Please check the Serial baud rate is same as your setting.



If the EtherCAT communication config successful, Serial Monitor will print "**0**" of each EtherCAT State, and "**Enable: 0**" for the motor state.

```
Begin: 0
Slave: 0
Start: 0
Enable: 0
```

It will print the motor's current position to the serial monitor.

```
Pos: -15330
Pos: -15316
Pos: -15316
Pos: -15316
Pos: -15316
```

5.3.4.2 Three Axes

In this section, we will control three axes on the QEC-R11MV3S. For each axis, we read the current position, continuously print it to the Serial Monitor, and use a simple state machine to move the motors back and forth between two target positions.

The following program sets the QEC-R11MV3S into CiA402 Profile Position (PP) mode:

- EtherCAT Cycle Time: 1 millisecond
- EtherCAT Mode: ECAT_SYNC

The EthercatMaster object ("master") represents the QEC-M-01,
the EthercatDevice_QECR11MV3S object ("slave0") represents the QEC-R11MV3S,
and the EthercatDevice_CiA402 *motor[MOTORS] array stores three CiA402 motor instances for Motor 1, Motor 2, and Motor 3.

A. In Setup Function

In the `setup()` function, the program initializes communication, starts the EtherCAT MDevice, and configures all three motors of the QEC-R11MV3S for CiA402 Profile Position (PP) mode.
Follow the steps below:

1. Start serial communication at a baud rate of 115200.
2. Start the EtherCAT MDevice
   - Use the `begin()` function to initialize the EtherCAT MDevice.
3. Attach the QEC-R11MV3S EtherCAT SubDevice
   - Use the `attach(0, master)` function to attach the QEC-R11MV3S as slave0 to the EtherCAT network.
4. Get the CiA402 Motor Instances
   - Call `cia402GetServo(1)`, `cia402GetServo(2)`, and `cia402GetServo(3)` and store the returned pointers in the motor[0], motor[1], and motor[2] of the motor array.
5. Set the CiA402 Mode to Profile Position (PP) for All Axes
   - For each index i from 0 to MOTORS - 1, call `setCiA402Mode(CIA402_PP_MODE)`
   - This configures all three axes to operate in Profile Position mode.
6. Start the EtherCAT MDevice
   - Use `start(1000000, ECAT_SYNC)` to switch the EtherCAT state machine to the OPERATIONAL state.
   - The cycle time is set to 1,000,000 ns (1 ms) with ECAT_SYNC mode.
   - Print the return value of `start()` to confirm that the transition to OPERATIONAL is successful.
7. Enable All Three Motors
   - For each axis i, call `motor[i]->enable()` and print the result to the Serial Monitor.
   - Each motor is now in the CIA402_OPERATION_ENABLED state.

8. Configure Profile Parameters (Common Settings for All Axes)
    - For each axis i, set the motion profile parameters for Profile Position mode:
        - Motion Profile Type:
          linear ramp using `motor[i]->pp_SetMotionProfileType(0)`
        - Profile Velocity: `motor[i]->pp_SetVelocity(10000)`
        - Acceleration: `motor[i]->pp_SetAcceleration(10000)`
        - Deceleration: `motor[i]->pp_SetDeceleration(10000)`

B. In Loop Function

In the `loop()` function, the current position of each motor is continuously printed on the Serial Monitor, and a per-axis state machine is used to move all motors between +100,000 and −100,000 position units.

1. Position Monitoring
    - For each axis i, the program reads the actual position with
      `motor[i]->getPositionActualValue()`
      and prints it in the format “Pos: i : <value>” on the Serial Monitor.
2. Per-Axis State Machine Logic
   For each axis i, the pp_state[i] variable controls a simple four-state motion cycle:
    - case 0:
        - Call `motor[i]->pp_Run(100000)` to move the motor to the positive target position (+100,000 units).
        - When `pp_Run()` returns 0 (command accepted), increment pp_state[i] to move to case 1.
    - case 1:
        - Use `motor[i]->pp_IsTargetReached()` to check whether the target has been reached.
        - When the function returns true, increment pp_state[i] to advance to case 2.
    - case 2:
        - Call `motor[i]->pp_Run(-100000)` to move the motor to the negative target position (−100,000 units).
        - When `pp_Run()` returns 0, increment pp_state[i] to advance to case 3.
    - case 3:
        - Use `motor[i]->pp_IsTargetReached()` to check whether the negative target position has been reached.
        - When the target has been reached, pp_state[i] is reset to 0 so the motion cycle can repeat.
        - The pp_done counter is incremented to indicate that this axis has completed one full back-and-forth cycle.

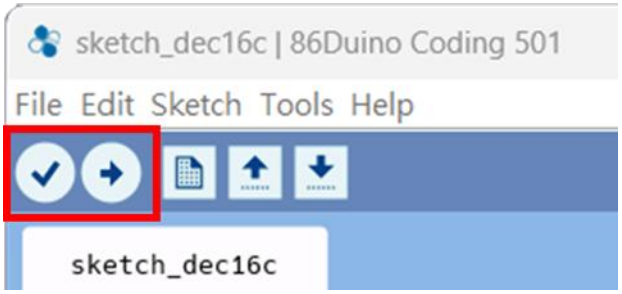
3. Cycle Synchronization
   - After all axes are processed, the program checks whether pp_done equals MOTORS.
   - When all three axes have completed one cycle, the program prints a newline, clears pp_done, and resets all pp_state[i] values to 0.
   - This causes all three axes to start the next motion cycle from the same state.
4. Code Logic Summary
   - Three axes are controlled using a common array of CiA402 motor pointers and individual state variables.
   - `pp_Run()` issues Profile Position commands for each axis.
   - pp_IsTargetReached() is used to confirm that the commanded positions have been reached.
   - The three motors repeatedly move between +100,000 and −100,000 position units, while their positions are continuously displayed on the Serial Monitor.

The example code is as follows:

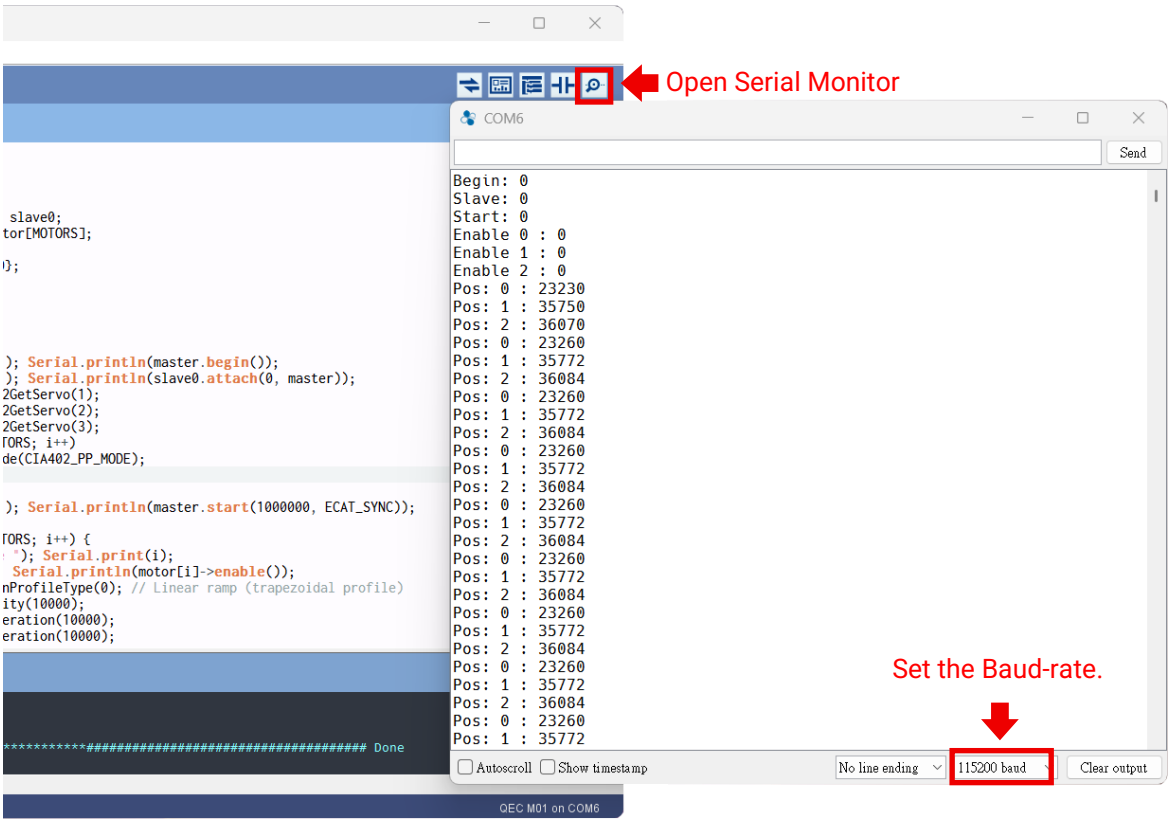
```
#include "Ethercat.h"
#define MOTORS     (3)

EthercatMaster master;
EthercatDevice_QECR11MV3S slave0;
EthercatDevice_CiA402 *motor[MOTORS];

int pp_state[MOTORS] = {0};
int pp_done = 0;

void setup() {
  Serial.begin(115200);

  Serial.print("Begin: "); Serial.println(master.begin());
  Serial.print("Slave: "); Serial.println(slave0.attach(0, master));
  motor[0] = slave0.cia402GetServo(1);
  motor[1] = slave0.cia402GetServo(2);
  motor[2] = slave0.cia402GetServo(3);
  for (int i = 0; i < MOTORS; i++)
    motor[i]->setCiA402Mode(CIA402_PP_MODE);


  Serial.print("Start: "); Serial.println(master.start(1000000, ECAT_SYNC));

  for (int i = 0; i < MOTORS; i++) {
    Serial.print("Enable "); Serial.print(i);
    Serial.print(" : "); Serial.println(motor[i]->enable());
```

```
    motor[i]->pp_SetMotionProfileType(0); // Linear ramp (trapezoidal profile)
    motor[i]->pp_SetVelocity(10000);
    motor[i]->pp_SetAcceleration(10000);
    motor[i]->pp_SetDeceleration(10000);
  }
}

void loop() {
  for (int i = 0; i < MOTORS; i++) {
    Serial.print("Pos: "); Serial.print(i);
    Serial.print(" : "); Serial.println(motor[i]->getPositionActualValue());

    switch (pp_state[i]) {
      case 0: if (motor[i]->pp_Run(100000) == 0) pp_state[i]++;
        break;
      case 1: if (motor[i]->pp_IsTargetReached()) pp_state[i]++;
        break;
      case 2: if (motor[i]->pp_Run(-100000) == 0) pp_state[i]++;
        break;
      case 3: if (motor[i]->pp_IsTargetReached()) pp_state[i] = 0; pp_done++;
        break;
    }
  }

  if (pp_done == MOTORS) {
    pp_done = 0;
    for (int i = 0; i < MOTORS; i++)
      pp_state[i] = 0;
  }
}
```

***Note:** Once the code is written, click on the toolbar to compile, and to confirm that the compilation is complete and error-free, you can click to upload. The program will run when the upload is complete.



After you successfully upload the program to the QEC-M-01, you can open the Serial Monitor on 86Duino IDE. Please check the Serial baud rate is same as your setting.



If the EtherCAT communication config successful, Serial Monitor will print "**0**" of each EtherCAT State, and "**Enable: 0**" for the motor state.

```
Begin: 0
Slave: 0
Start: 0
Enable 0 : 0
Enable 1 : 0
Enable 2 : 0
```

It will print the motor's current position to the serial monitor.

```
Pos: 0 : 23230
Pos: 1 : 35750
Pos: 2 : 36070
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
Pos: 2 : 36084
Pos: 0 : 23260
Pos: 1 : 35772
```

Ch. 6

SubDevice Information

6.1 ESI (EtherCAT SubDevice Information) file

The ESI files contain information unique to the EtherCAT SubDevice Terminals in XML format. You can load an ESI file into the Support Software to easily allocate SubDevice Terminal process data and other settings. The ESI files for QEC EtherCAT SubDevices are already installed in the Support Software.

*** Note: Ensuring Up-to-date Installation of the XML Device Description File (ESI)**

To ensure smooth functioning, it is important to install the latest version of the XML device description file in the EtherCAT MDevice software. The latest version of the XML device description file can be downloaded from the QEC website.

https://www.qec.tw/product/qec-rxxmv3s/

6.2 Object Dictionary

The object dictionary defined here shall be used complementary with ETG.5001 and ETG.1000.

- Device Profile: 402 Device Profile
- Modul Profile: 2 Servo

Usage Notes:

- The PDO mapping object and SyncManager assignment object doesn't need to be defined. In that case they are created automatically.
- The following objects are fixed included in the SSC and shall not be defined in the file: 0x1000, 0x1001, 0x1008, 0x1009, 0x100a, 0x1010, 0x1011, 0x1018, 0x10F0, 0x10F1, 0x10F3, 0x1c00, 0x1c32, 0x1c33
- Entries less or equal one 8Bit shall not overlap byte borders.
- Entries greater 8Bit shall always start at an exact word border.

6.2.1 Standard Objects (0x1000-0x1FFF)

Index 0x1000 Device type

Device type.

Index	Name	Data type	Flags	Default
1000	Device type	UINT32	ro	0x00020192 (131474)

Index 0x1001 Error register

Error register.

Index	Name	Data type	Flags	Default
1001	Error register	UINT8	ro	0x00 (0)

Index 0x1008 Device name

Device name.

Index	Name	Data type	Flags	Default
1008	Device name	STRING	ro	Refer to following table.

Table 6-1: Device Name

Type	Device Name
Pulse Output module without PoE	QEC-R00MV3S
Pulse Output module with PoE	QEC-R11MV3S

Index 0x1009 Hardware version

Hardware version.

Index	Name	Data type	Flags	Default
1009	Hardware version	STRING	ro	Depends on the version of the product you have.

Index 0x100A Software version

Software version.

Index	Name	Data type	Flags	Default
100A	Software version	STRING	ro	1.08

* **Note:** This document is written based on **software version 1.08**.

Index 0x1018 Identity

Identity objects.

Index	Name	Data type	Flags	Default
1018:0	Identity	UINT8	ro	> 4 <
1018:01	Vendor ID	UINT32	ro	0x00000BC3 (3011)
1018:02	Product code	UINT32	ro	Refer to following table.
1018:03	Revision	UINT32	ro	Depending by model.
1018:04	Serial number	UINT32	ro	0x00000001 (1)

Table 6-2: Product code

Model Name	Product code	Revision Number
QEC-R11MV3S	0x0086d0e4	0x20240530
QEC-R00MV3S	0x0086d0e5	0x20240530

Index 0x10F1 Error Settings

Error settings.

Index	Name	Data type	Flags	Default
10F1:0	Error Settings	UINT8	ro	> 2 <
10F1:01	Local Error Reaction	UINT32	rw	0x00000001 (1)
10F1:02	Sync Error Counter Limit	UINT32	rw	0x0004 (4)

Index 0x10F8 Timestamp Object

Timestamp object.

Index	Name	Data type	Flags	Default
10F8	Timestamp Object	UINT64	rw P	9E 04 CA F3 20 00 00 00

6.2.1.1 RxPDO Mapping Objects

RxPDO Mapping (0x1600 - 0x17FF).
If no RxPDO mapping object is defined they will be created automatically.

Index 0x1600 M1 Axis RX PDO Mapping0

RX PDO Mapping for Cyclic Synchronous Position (CSP) and Cyclic Synchronous Velocity (CSV) modes.

Index	Name	Data type	Flags	Default	Description
1600:0	M1 Axis RxPdoMapping0	UINT8	rw	> 4 <	-
1600:01	SubIndex 001	UINT32	rw	0x6040:00, 16	Map control word.
1600:02	SubIndex 002	UINT32	rw	0x607A:00, 32	Map target position.
1600:03	SubIndex 003	UINT32	rw	0x60FF:00, 32	Map target velocity.
1600:04	SubIndex 004	UINT32	rw	0x6060:00, 8	Map mode of operation.

Index 0x1604 M1 Axis RX PDO Mapping4

RX PDO Mapping for Profile Position (PP) mode.

Index	Name	Data type	Flags	Default	Description
1604:0	M1 Axis RxPdoMapping4	UINT8	rw	> 4 <	-
1604:01	SubIndex 001	UINT32	rw	0x6040:00, 16	Map control word.
1604:02	SubIndex 002	UINT32	rw	0x607A:00, 32	Map target position.
1604:03	SubIndex 003	UINT32	rw	0x6081:00, 32	Map profile velocity.
1604:04	SubIndex 004	UINT32	rw	0x6083:00, 32	Map profile acceleration.

Index 0x1610 M2 Axis RX PDO Mapping0

RX PDO Mapping for Cyclic Synchronous Position (CSP) and Cyclic Synchronous Velocity (CSV) modes.

Index	Name	Data type	Flags	Default	Description
1610:0	M2 Axis RxPdoMapping0	UINT8	rw	> 4 <	-
1610:01	SubIndex 001	UINT32	rw	0x6840:00, 16	Map control word.
1610:02	SubIndex 002	UINT32	rw	0x687A:00, 32	Map target position.
1610:03	SubIndex 003	UINT32	rw	0x68FF:00, 32	Map target velocity.
1610:04	SubIndex 004	UINT32	rw	0x6860:00, 8	Map mode of operation.

Index 0x1614 M2 Axis RX PDO Mapping4

RX PDO Mapping for Profile Position (PP) mode.

Index	Name	Data type	Flags	Default	Description
1614:0	M2 Axis RxPdoMapping4	UINT8	rw	> 4 <	-
1614:01	SubIndex 001	UINT32	rw	0x6840:00, 16	Map control word.
1614:02	SubIndex 002	UINT32	rw	0x687A:00, 32	Map target position.
1614:03	SubIndex 003	UINT32	rw	0x6881:00, 32	Map profile velocity.
1614:04	SubIndex 004	UINT32	rw	0x6883:00, 32	Map profile acceleration.

Index 0x1620 M3 Axis RX PDO Mapping0

RX PDO Mapping for Cyclic Synchronous Position (CSP) and Cyclic Synchronous Velocity (CSV) modes.

Index	Name	Data type	Flags	Default	Description
1620:0	M3 Axis RxPdoMapping0	UINT8	rw	> 4 <	-
1620:01	SubIndex 001	UINT32	rw	0x7040:00, 16	Map control word.
1620:02	SubIndex 002	UINT32	rw	0x707A:00, 32	Map target position.
1620:03	SubIndex 003	UINT32	rw	0x70FF:00, 32	Map target velocity.
1620:04	SubIndex 004	UINT32	rw	0x7060:00, 8	Map mode of operation.

Index 0x1624 M3 Axis RX PDO Mapping4

RX PDO Mapping for Profile Position (PP) mode.

Index	Name	Data type	Flags	Default	Description
1624:0	M3 Axis RxPdoMapping4	UINT8	rw	> 4 <	-
1624:01	SubIndex 001	UINT32	rw	0x7040:00, 16	Map control word.
1624:02	SubIndex 002	UINT32	rw	0x707A:00, 32	Map target position.
1624:03	SubIndex 003	UINT32	rw	0x7081:00, 32	Map profile velocity.
1624:04	SubIndex 004	UINT32	rw	0x7083:00, 32	Map profile acceleration.

Index 0x1680 G code RX PDO Mapping0

RX PDO Mapping for G-code mode.

Index	Name	Data type	Flags	Default	Description
1680:0	G code RxPdoMapping0	UINT8	rw	> 6 <	-
1680:01	SubIndex 001	UINT32	rw	0x50D0:00, 16	Map G code header.
1680:02	SubIndex 002	UINT32	rw	0x50D1:00, 208	Map G code string.
1680:03	SubIndex 003	UINT32	rw	0x50D9:00, 8	Map emergency stop.
1680:04	SubIndex 004	UINT32	rw	0x50CD:01, 64	Map X axis Jog Command.
1680:05	SubIndex 005	UINT32	rw	0x50CD:02, 64	Map Y axis Jog Command.
1680:06	SubIndex 006	UINT32	rw	0x50CD:03, 64	Map Z axis Jog Command.

Index 0x1681 G code RX PDO Mapping1

RX PDO Mapping for G-code machine objects.

Index	Name	Data type	Flags	Default	Description
1681:0	G code RxPdoMapping1	UINT8	rw	> 2 <	-
1681:01	SubIndex 001	UINT32	rw	0x50D7:00, 8	Map machine servo on.
1681:02	SubIndex 002	UINT32	rw	0x50D8:00, 8	Map machine homing.

Index 0x1700: Empty RX PDO Mapping0

0x1700: Empty RX PDO Mapping0.

Index	Name	Data type	Flags	Default	Description
1700:0	Empty RxPdoMapping	UINT8	rw	> 1 <	-
1700:01	SubIndex 001	UINT32	rw	0x0000:00, 0	Empty

If user only need to operate the M1 axis, object 0x1C12 could be set as:

- Set 0x1C12:02 to 0x1700;
- Set 0x1C12:03 to 0x1700;

PDO mapping for M2 and M3 will be replaced with zero-length PDO mappings. The total byte length of output PDO Mapping will change from 33 to 11.

In G-code mode, default value of object 0x1C12:

- 0x1C12:01 is 0x1680;
- 0x1C12:02 is 0x1700;
- 0x1C12:03 is 0x1700;

PDO mapping 0x1700 is a padding for PDO mapping slots 2 and 3.

6.2.1.2 TxPDO Mapping Objects

TxPDO Mapping (0x1A00 - 0x1BFF).
If no TxPDO mapping object is defined they will be created automatically.

Index 0x1A00 M1 Axis TX PDO Mapping0

TX PDO Mapping for Cyclic Synchronous Position (CSP) and Cyclic Synchronous Velocity (CSV) modes.

Index	Name	Data type	Flags	Default	Description
1A00:0	M1 Axis TxPdoMapping0	UINT8	rw	> 6 <	-
1A00:01	SubIndex 001	UINT32	rw	0x6041:00, 16	Map status word.
1A00:02	SubIndex 002	UINT32	rw	0x6064:00, 32	Map actual position.
1A00:03	SubIndex 003	UINT32	rw	0x606C:00, 32	Map actual velocity.
1A00:04	SubIndex 004	UINT32	rw	0x60E4:01, 32	Map additional position actual value.
1A00:05	SubIndex 005	UINT32	rw	0x60FD:00, 32	Map digital inputs.
1A00:06	SubIndex 006	UINT32	rw	0x6061:00, 8	Map mode of operation display.

Index 0x1A04 M1 Axis TX PDO Mapping4

TX PDO Mapping for Profile Position (PP) mode.

Index	Name	Data type	Flags	Default	Description
1A04:0	M1 Axis TxPdoMapping4	UINT8	rw	> 4 <	-
1A04:01	SubIndex 001	UINT32	rw	0x6041:00, 16	Map status word.
1A04:02	SubIndex 002	UINT32	rw	0x6064:00, 32	Map actual position.
1A04:03	SubIndex 003	UINT32	rw	0x60E4:01, 32	Map additional position actual value.
1A04:04	SubIndex 004	UINT32	rw	0x60FD:00, 32	Map digital inputs.

Index 0x1A10 M2 Axis TX PDO Mapping0

TX PDO Mapping for Cyclic Synchronous Position (CSP) and Cyclic Synchronous Velocity (CSV) modes.

Index	Name	Data type	Flags	Default	Description
1A10:0	M2 Axis TxPdoMapping0	UINT8	rw	> 6 <	-
1A10:01	SubIndex 001	UINT32	rw	0x6841:00, 16	Map status word.
1A10:02	SubIndex 002	UINT32	rw	0x6864:00, 32	Map actual position.
1A10:03	SubIndex 003	UINT32	rw	0x686C:00, 32	Map actual velocity.
1A10:04	SubIndex 004	UINT32	rw	0x68E4:01, 32	Map additional position actual value.
1A10:05	SubIndex 005	UINT32	rw	0x68FD:00, 32	Map digital inputs.
1A10:06	SubIndex 006	UINT32	rw	0x6861:00, 8	Map mode of operation display.

Index 0x1A14 M2 Axis TX PDO Mapping4

TX PDO Mapping for Profile Position (PP) mode.

Index	Name	Data type	Flags	Default	Description
1A14:0	M2 Axis TxPdoMapping4	UINT8	rw	> 4 <	-
1A14:01	SubIndex 001	UINT32	rw	0x6841:00, 16	Map status word.
1A14:02	SubIndex 002	UINT32	rw	0x6864:00, 32	Map actual position.
1A14:03	SubIndex 003	UINT32	rw	0x68E4:01, 32	Map additional position actual value.
1A14:04	SubIndex 004	UINT32	rw	0x68FD:00, 32	Map digital inputs.

Index 0x1A20 M3 Axis TX PDO Mapping0

TX PDO Mapping for Cyclic Synchronous Position (CSP) and Cyclic Synchronous Velocity (CSV) modes.

Index	Name	Data type	Flags	Default	Description
1A20:0	M3 Axis TxPdoMapping0	UINT8	rw	> 6 <	-
1A20:01	SubIndex 001	UINT32	rw	0x7041:00, 16	Map status word.
1A20:02	SubIndex 002	UINT32	rw	0x7064:00, 32	Map actual position.
1A20:03	SubIndex 003	UINT32	rw	0x706C:00, 32	Map actual velocity.
1A20:04	SubIndex 004	UINT32	rw	0x70E4:01, 32	Map additional position actual value.
1A20:05	SubIndex 005	UINT32	rw	0x70FD:00, 32	Map digital inputs.
1A20:06	SubIndex 006	UINT32	rw	0x7061:00, 8	Map mode of operation display.

Index 0x1A24 M3 Axis TX PDO Mapping4

TX PDO Mapping for Profile Position (PP) mode.

Index	Name	Data type	Flags	Default	Description
1A24:0	M3 Axis TxPdoMapping4	UINT8	rw	> 4 <	-
1A24:01	SubIndex 001	UINT32	rw	0x7041:00, 16	Map status word.
1A24:02	SubIndex 002	UINT32	rw	0x7064:00, 32	Map actual position.
1A24:03	SubIndex 003	UINT32	rw	0x70E4:01, 32	Map additional position actual value.
1A24:04	SubIndex 004	UINT32	rw	0x70FD:00, 32	Map digital inputs.

Index 0x1A80 G code TX PDO Mapping0

TX PDO Mapping for G-code mode.

Index	Name	Data type	Flags	Default	Description
1A80:00	G code TxPdoMapping0	UINT8	rw	> 6 <	-
1A80:01	SubIndex 01	UINT32	rw	0x50E0:00, 16	Map G-code statusword.
1A80:02	SubIndex 02	UINT32	rw	0x50E1:00, 8	Map machine status.
1A80:03	SubIndex 03	UINT32	rw	0x50E2:01, 64	Map actual position of X axis.
1A80:04	SubIndex 04	UINT32	rw	0x50E2:02, 64	Map actual position of Y axis.
1A80:05	SubIndex 05	UINT32	rw	0x50E2:03, 64	Map actual position of Z axis.
1A80:06	SubIndex 06	UINT32	rw	0x50E3:00, 8	Map Home Inputs.

Index 0x1A81 G code TX PDO Mapping1

Additional TX PDO Mapping1 for G-code mode.

Index	Name	Data type	Flags	Default	Description
1A81:00	G code TxPdoMapping1	UINT8	rw	> 3 <	-
1A81:01	SubIndex 01	UINT32	rw	0x60FD:00, 32	Digital Inputs for M1 and ENC_1
1A81:02	SubIndex 02	UINT32	rw	0x68FD:00, 32	Digital Inputs for M2 and ENC_2
1A81:03	SubIndex 03	UINT32	rw	0x70FD:00, 32	Digital Inputs for M3 and ENC_3

Index 0x1B00 ENC TxPdoMapping0

TX PDO Mapping1 for Encoder.

Index	Name	Data type	Flags	Default	Description
1B00:00	ENC TxPdoMapping0	UINT8	rw	> 6 <	-
1B00:01	SubIndex 01	UINT32	rw	0x5024:01, 08	ENC Status.
1B00:02	SubIndex 02	UINT32	rw	0x5024:02, 08	ENC Status.
1B00:03	SubIndex 03	UINT32	rw	0x5024:03, 08	ENC Status.
1B00:04	SubIndex 04	UINT32	rw	0x502A:01, 32	ENC Counts.
1B00:05	SubIndex 05	UINT32	rw	0x502A:02, 32	ENC Counts.
1B00:06	SubIndex 06	UINT32	rw	0x502A:03, 32	ENC Counts.

Index 0x1B01: Empty TX PDO Mapping0

Empty TX PDO Mapping0.

Index	Name	Data type	Flags	Default	Description
1B01:00	Empty TxPdoMapping	UINT8	rw	1	
1B01:01	SubIndex 01	UINT32	rw	0x0000:00, 0	

If user only need to operate the M1 axis, object 0x1C13 could be set as:

- Set 0x1C13:02 to 0x1B01;
- Set 0x1C13:03 to 0x1B01;

PDO mapping for M2 and M3 will be replaced with zero-length PDO mappings. The total byte length of output PDO Mapping will change from 72 to 34.

In G-code mode, default value of object 0x1C13:

- 0x1C13:01 is 0x1A80;
- 0x1C13:02 is 0x1A81;
- 0x1C13:03 is 0x1B00;
- 0x1C13:04 is 0x1B01;

PDO mapping 0x1B01 is a padding for PDO mapping slot 4. It can also be used to replace mappings 0x1A81 and 0x1B00.

6.2.1.3 Sync Manager Objects

Sync Manager Objects.

Index 0x1C00 Sync manager type

SyncManager type.

Index	Name	Data type	Flags	Default
1C00:0	Sync manager type	UINT8	rw	> 4 <
1C00:01	SubIndex 001	UINT8	ro	0x01 (1)
1C00:02	SubIndex 002	UINT8	ro	0x02 (2)
1C00:03	SubIndex 003	UINT8	ro	0x03 (3)
1C00:04	SubIndex 004	UINT8	ro	0x04 (4)

Index 0x1C12 SM2 assignment

SyncManager 2 Assignment. RX PDO Assign.
If this object is not defined it will be created automatically.

Index	Name	Data type	Flags	Default
1C12:0	RX PDO Assign	UINT8	rw	> 4 <
1C12:01	SubIndex 01	UINT16	ro, wr_preop	0x1600
1C12:02	SubIndex 02	UINT16	ro, wr_preop	0x1610
1C12:03	SubIndex 03	UINT16	ro, wr_preop	0x1620
1C12:04	SubIndex 04	UINT16	ro, wr_preop	0x1700

Index 0x1C13 SM3 assignment

SyncManager 3 Assignment. TX PDO Assign.
If this object is not defined it will be created automatically.

Index	Name	Data type	Flags	Default
1C13:00	TX PDO Assign	UINT8	ro	> 4 <
1C13:01	SubIndex 01	UINT16	ro	0x1A00
1C13:02	SubIndex 02	UINT16	ro	0x1A10
1C13:03	SubIndex 03	UINT16	ro	0x1A20
1C13:04	SubIndex 04	UINT16	ro	0x1B00

Index 0x1C32 SM Output Parameters

SM output parameters.

Index	Name	Data type	Flags	Default
1C32:0	SM output parameter	UINT8	ro	> 32 <
1C32:01	Synchronization Type	UINT16	rw	0x0000 (0)
1C32:02	Cycle Time	UINT32	ro	0x00000000 (0)
1C32:03	SubIndex 003	-	-	-
1C32:04	Synchronization Types supported	UINT16	ro	0x001F (31)
1C32:05	Minimum Cycle Time	UINT32	ro	0x0003D090 (250000)
1C32:06	Calc and Copy Time	UINT32	ro	0x00004E20 (20000)
1C32:07	SubIndex 007	-	-	-
1C32:08	Get Cycle Time	UINT16	rw	0x0000 (0)
1C32:09	Delay Time	UINT32	ro	0x00000000 (0)
1C32:0A	Sync0 Cycle Time	UINT32	rw	0x00000000 (0)
1C32:0B	SM-Event Missed	UINT16	ro	0x0000 (0)
1C32:0C	Cycle Time Too Small	UINT16	ro	0x0000 (0)
1C32:0D	Shift Time Too Short	-	-	-
1C32:0E	SubIndex 014	-	-	-
1C32:0F	SubIndex 015	-	-	-
1C32:10	SubIndex 016	-	-	-
1C32:11	SubIndex 017	-	-	-
1C32:12	SubIndex 018	-	-	-
1C32:20	Sync Error	BOOL	ro	FALSE

Index 0x1C33 SM input parameters

SM input parameters.

Index	Name	Data type	Flags	Default
1C33:0	SM input parameter	UINT8	ro	> 32 <
1C33:01	Synchronization Type	UINT16	rw	0x0000 (0)
1C33:02	Cycle Time	UINT32	ro	0x00000000 (0)
1C33:03	SubIndex 003	-	-	-
1C33:04	Synchronization Types supported	UINT16	ro	0x001F (31)
1C33:05	Minimum Cycle Time	UINT32	ro	0x0003D090 (250000)
1C33:06	Calc and Copy Time	UINT32	ro	0x000088B8 (35000)
1C33:07	SubIndex 007	-	-	-
1C33:08	Get Cycle Time	UINT16	rw	0x0000 (0)
1C33:09	Delay Time	UINT32	ro	0x00000000 (0)
1C33:0A	Sync0 Cycle Time	UINT32	rw	0x00000000 (0)
1C33:0B	SM-Event Missed	UINT16	ro	0x0000 (0)
1C33:0C	Cycle Time Too Small	UINT32	ro	0x0000 (0)
1C33:0D	Shift Time Too Short	-	-	-
1C33:0E	SubIndex 014	-	-	-
1C33:0F	SubIndex 015	-	-	-
1C33:10	SubIndex 016	-	-	-
1C33:11	SubIndex 017	-	-	-
1C33:12	SubIndex 018	-	-	-
1C33:20	Sync Error	BOOL	ro	FALSE

6.2.2 Manufacturer Objects (0x5000-0x5FFF)

Manufacturer Objects (0x5000-0x5FFF).

Index 0x5xxn Manufacturer Objects

Device Information Objects.

Index	Name	Data type	Flags	Default	Description
5000	SP_Voltage	UINT16	rw	0	Read SP Voltage
5001	SP_Current	UINT16	rw	0	Read SP Current
5002	PP_Voltage	UINT16	rw	0	Read PP Voltage
5003	PP_Current	UINT16	rw	0	Read PP Current
5004	Temperature	INT16	rw	0	Read Temperature
5005	BoxStatus	UINT8	ro	0	-
5006:0	OrderInformation			-	Read Order Information
5006:01	Customer	STRING(6)	ro	7878787878	Customer
5006:02	OrderNo	STRING(8)	ro	78787878787878	Order No
5006:03	InvNo	STRING(11)	ro	78787878787878787878	Inv No
5006:04	DelyDate	STRING(4)	ro	787878	Dely Date
5007:0	MTBF			> 2 <	MTBF
5007:01	WorkingHours	INT32	ro	-	Working Hours
5007:02	BootTimes	INT32	ro	-	Boot Times

6.2.2.1 Motor Objects

Motor objects.

Index 0x5010 Pulse Period

Pulse period.

Index	Name	Data type	Flags	Default
5010	PulsePeriod	UINT32	ro	0x00000010 (16)

Index 0x5011 Motor Pulse

Position distance of motor full rotation.

Index	Name	Data type	Flags	Default
5011:0	Motor Pulse	UINT8	ro	> 3 <
5011:01	M1	UINT32	ro, wr_preop	0x00000C80 (3200)
5011:02	M2	UINT32	ro, wr_preop	0x00000C80 (3200)
5011:03	M3	UINT32	ro, wr_preop	0x00000C80 (3200)

Index 0x5012 Initialize EEPROM

MV3S can restore object setting to EEPROM. Write corresponding value to object 0x5012 will initialize EEPROM:

- 0x1108: initialize common objects
- 0x1104: initialize objects of CiA-402
- 0x1100: initialize objects of G-code mode

Index	Name	Data type	Flags	Default
5012	Initialize EEPROM	UINT16	rw	0x0000 (0)

Storable Objects: The item data for the items in the following list will be stored in the device's SPIROM and will not need to be reset after a power restart.

Common	CiA-402		G-code
0x5011	0x607C	0x707C	0x5030
0x5014	0x607D	0x707D	0x50D3
0x5020	0x607E	0x707E	0x50D4
0x5021	0x6080	0x7080	0x50D5
0x5022	0x687C		0x50FD
0x5023	0x687D		0x50FE
0x5027	0x687E		0x50DB
0x5028	0x6880		0x50DC
0x5029			0x5031
0x50FA			0x50CA

Index 0x5013 Set Actual POS

Set motor actual position value.

Index	Name	Data type	Flags	Default
5013:0	Set Actual POS	UINT16	rw	> 3 <
5013:01	M1	INT32	rw	0x0000 (0)
5013:02	M2	INT32	rw	0x0000 (0)
5013:03	M3	INT32	rw	0x0000 (0)

Index 0x5014 Velocity Unit

Vel unit is a configurable speed unit. If 0x5014:01 is 0, it means vel unit = pulse counts, and 1 means vel unit = rpm.
M1 represents the setting of axis 0 (6000h to 67FFh) in CiA-402 mode. M2 represents the setting of axis 1 (6800h to 6FFFh). M3 represents the setting of axis 2 (7000h to 77ffh).

Index	Name	Data Type	Flags	Default
5014:00	Subindex 00	UINT8	ro	> 3 <
5014:01	M1	UINT8	rw	0
5014:02	M2	UINT8	rw	0
5014:03	M3	UINT8	rw	0

Index 0x5015 PulseFrequency

PulseFrequency sets the maximum output pulse frequency of each axis in CiA-402 mode. M1 represents the setting of axis 0 (6000h to 67FFh) in CiA-402 mode. M2 represents the setting of axis 1 (6800h to 6FFFh). M3 represents the setting of axis 2 (7000h to 77FFh).

Three setting values are supported:

- 1 = default, maximum pulse frequency 100 kHz;
- 5 = maximum pulse frequency 500 kHz;
- 40 = maximum pulse frequency 4 MHz.
- Writing any other value sets it back to the default value (1).

After changing this object, the power of the QEC-RXXMV3S must be restarted for the new setting to take effect.

Index	Name	Data Type	Flags	Default
5015:00	Subindex 00	UINT8	ro	> 3 <
5015:01	M1	UINT8	rw	1
5015:02	M2	UINT8	rw	1
5015:03	M3	UINT8	rw	1

Index 0x5020 ENC Mode

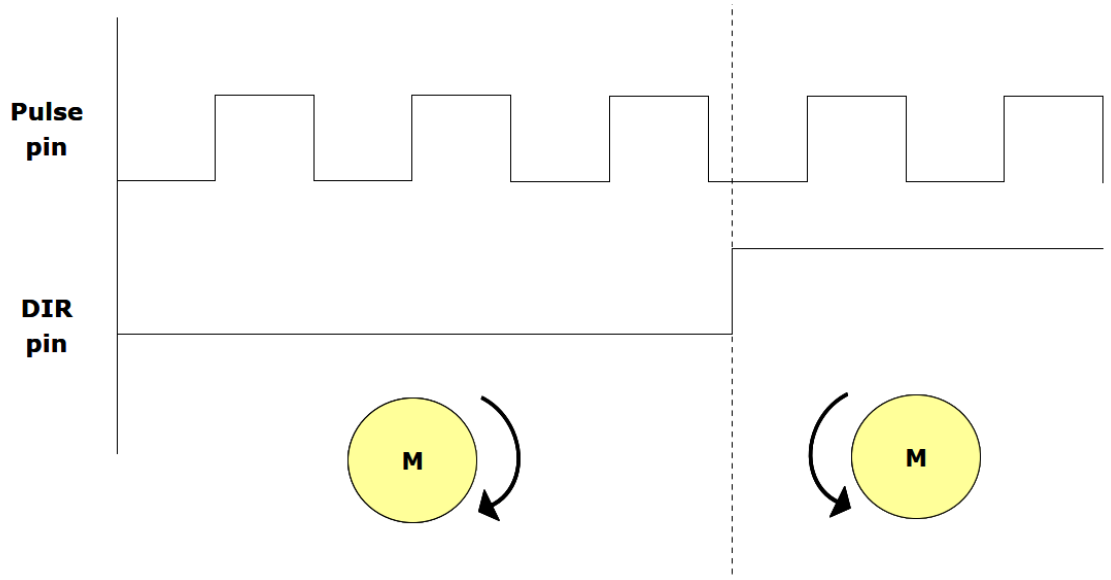
MV3S includes 3 encoder input interfaces. The encoder input interface allows user to set the received signal format. Object 0x5020 has 3 sub-objects, used to set the signal format of the 3 encoder input interfaces.

Index	Name	Data type	Flags	Default
5020:0	ENC Mode	UINT8	ro	> 3 <
5020:01	ENC_1	UINT8	rw	0x07 (7)
5020:02	ENC_2	UINT8	rw	0x07 (7)
5020:03	ENC_3	UINT8	rw	0x07 (7)

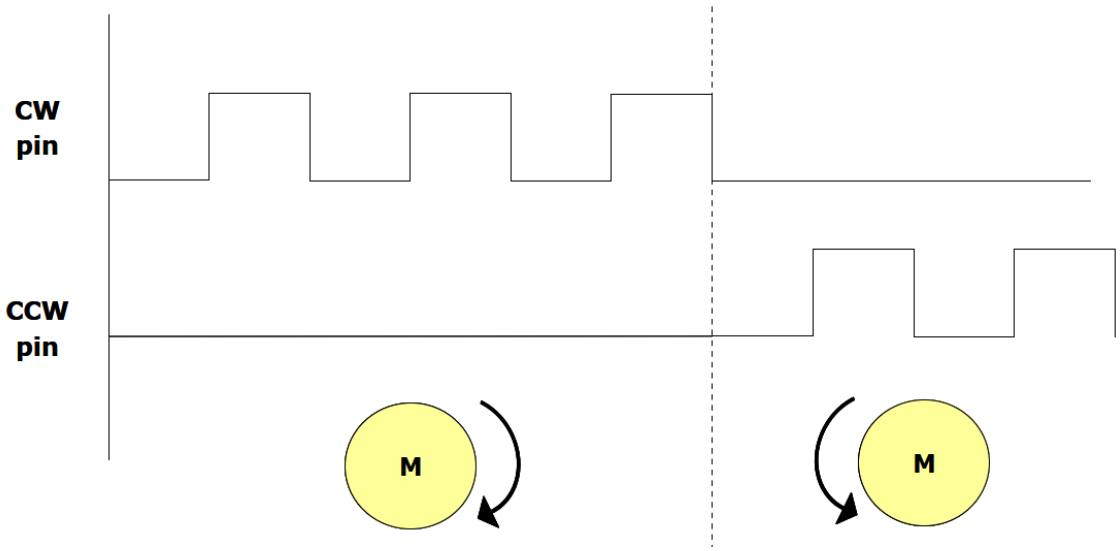
Input signal waveforms:

Mode	Code
MODE_STEP_DIR	0
MODE_CWCCW	1
MODE_AB_PHASE	2
MODE_STEP_DIR_x2	5
MODE_CWCCW_x2	6
MODE_AB_PHASE_x2	7
MODE_AB_PHASE_x2	Any other

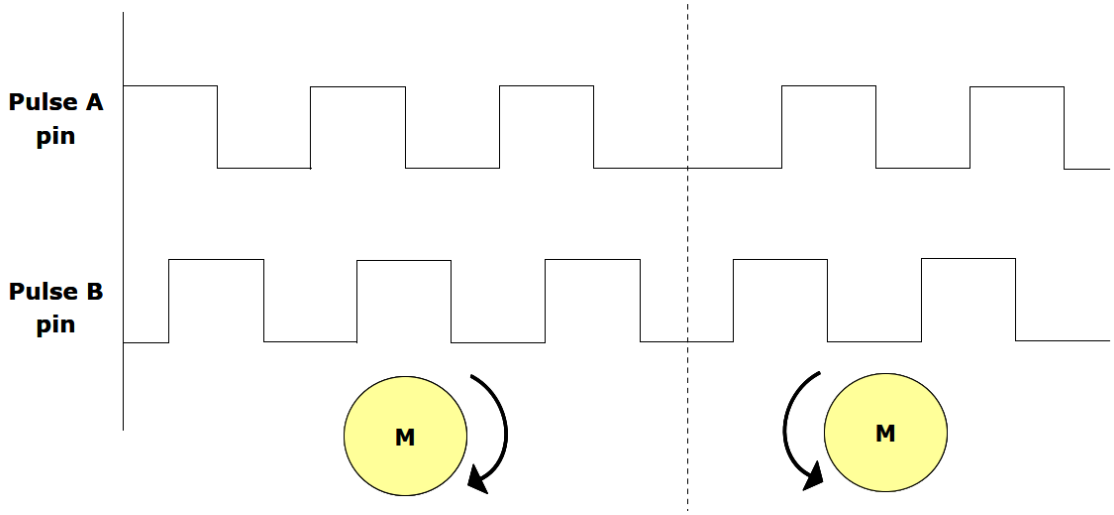
MODE_STEP_DIR



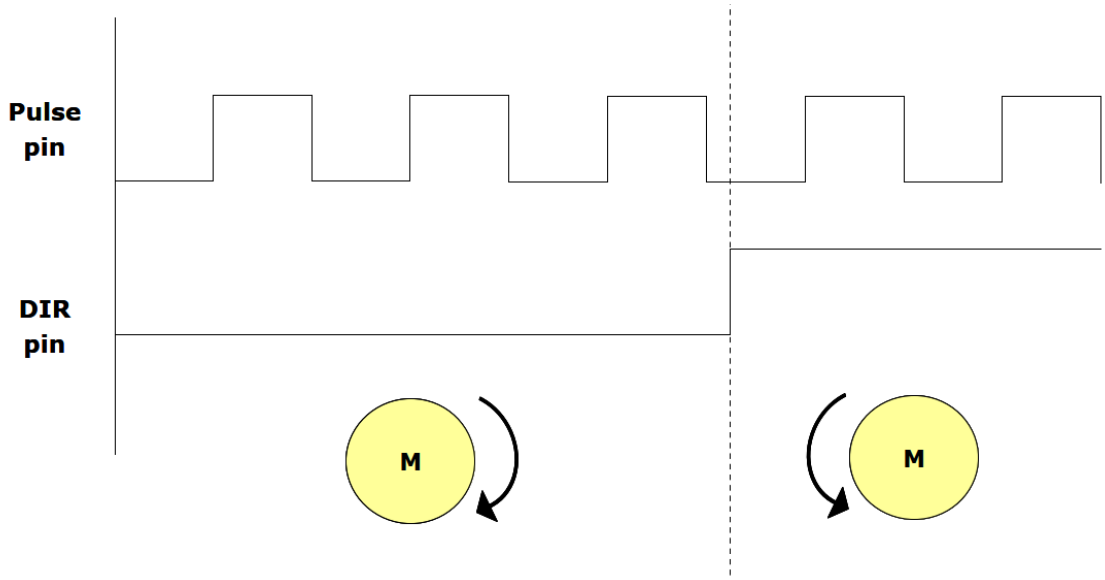
MODE_CWCCW



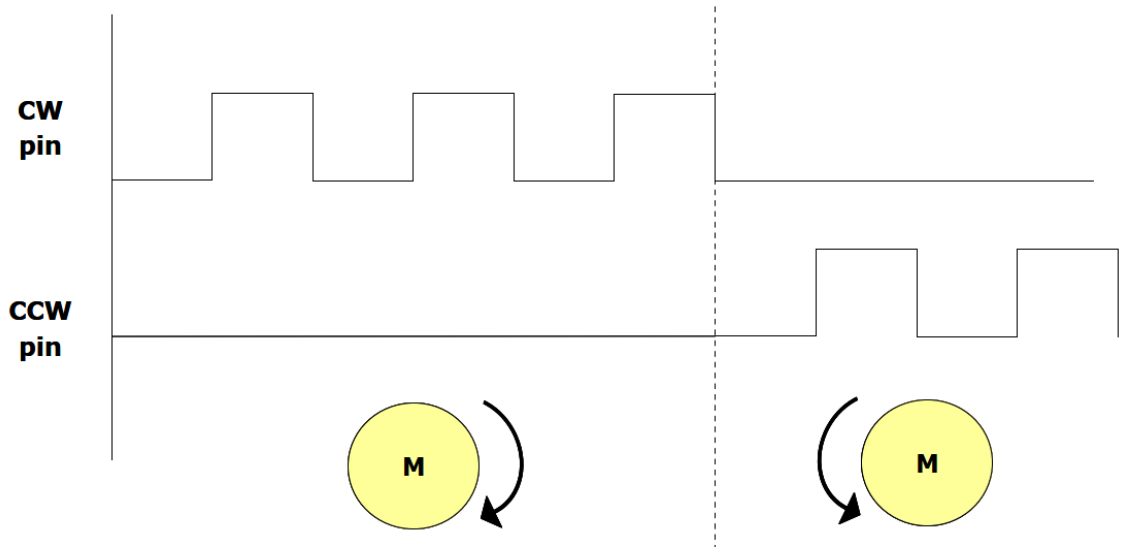
MODE_AB_PHASE



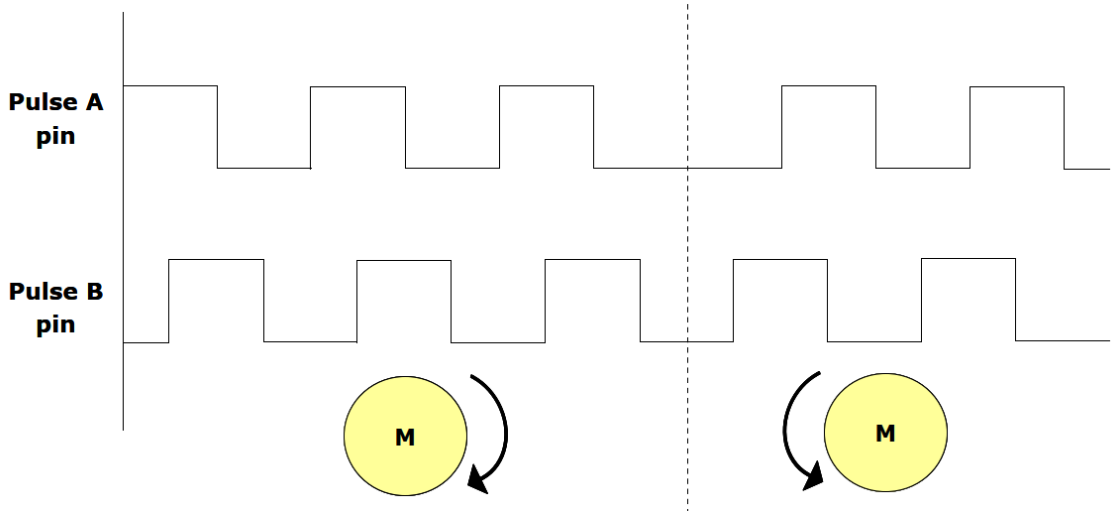
MODE_STEP_DIR_x2



MODE_CWCCW_x2



MODE_AB_PHASE_x2



Index 0x5021 ENC Digital Filter

Unit of ENC Digital Filter is 10 nanoseconds. If set ENC Digital Filter to 100, encoder interface will delay sampling for 1000 ns after rising/falling edge trigger.

Index	Name	Data type	Flags	Default
5021:0	ENC Digital Filter	UINT8	ro	> 3 <
5021:01	ENC_1	UINT32	rw	0x00000064 (100)
5021:02	ENC_2	UINT32	rw	0x00000064 (100)
5021:03	ENC_3	UINT32	rw	0x00000064 (100)

Index 0x5022 ENC Control Byte

Encoder Control Byte.

Index	Name	Data type	Flags	Default
5022:0	ENC Control Byte	UINT8	ro	> 3 <
5022:01	ENC_1	UINT8	rw	0x07
5022:02	ENC_2	UINT8	rw	0x07
5022:03	ENC_3	UINT8	rw	0x07

Set signal polarity for pin A, B and Z. If set Input Polarity to 1, low level is considered signal 0 and high level is considered signal 1.
If set ENC Index Reset to 1, encoder value will be reset to 0 when zero index triggers.

[7 – 4]	3	2	1	0
Reserved	ENC Index Reset	ENC Input Polarity		
		Input Z Polarity	Input B Polarity	Input A Polarity

Index 0x5023 ENC Range

Set an upper bound value of encoder counter. When the counter value increases up to the upper bound, then back to zero. Otherwise, When the counter value decrease to the lower bound, then back to max value.

Index	Name	Data type	Flags	Default
5023:0	ENC Range	UINT8	ro	> 3 <
5023:01	ENC_1	UINT32	rw	0x7FFFFFFF (2147483647)
5023:02	ENC_2	UINT32	rw	0x7FFFFFFF (2147483647)
5023:03	ENC_3	UINT32	rw	0x7FFFFFFF (2147483647)

Index 0x5024 ENC Status

Encoder Status.

Index	Name	Data type	Flags	Default
5024:0	ENC Status	UINT8	ro	> 3 <
5024:01	ENC_1	UINT8	ro	0
5024:02	ENC_2	UINT8	ro	0
5024:03	ENC_3	UINT8	ro	0

When device reset encoder value to 0 due to zero index, IDX-Reset bit would be set to 1.
When the encoder value increases up to the upper bound, then back to zero, PCNT-OV will be set 1.
When the encoder value decrease to 0, then back to max value, PCNT-UV will be set to 1.
All bits are automatically cleared after one cycle.

7	[6 – 2]	1	0
IDX-Reset	Reserved	PCNT-OV	PCNT-UV

Index 0x5025 ENC Write

Writing object 0x5025 would set encoder value to writing value. Additional position encoder value (0x60E4/0x68E4/ 0x70E4) would be updated to the same value.

Index	Name	Data type	Flags	Default
5025:0	ENC Write	UINT8	ro	> 3 <
5025:01	ENC_1	INT32	WO	0
5025:02	ENC_2	INT32	WO	0
5025:03	ENC_3	INT32	WO	0

Index 0x5026 Step Loss Compensation

Step Loss Compensation settings of X/Y/Z.
0 indicates disabled, 1 indicate enabled.

Index	Name	Data Type	Flags	Default
5026:00	Subindex 00	UINT8	ro	> 3 <
5026:01	X	UINT8	rw	0
5026:02	Y	UINT8	rw	0
5026:03	Z	UINT8	rw	0

Step Loss Compensation has the following features: Compare the POS_FB of the M1/M2/M3 axis with the Actual Position (0x50E2) of the X/Y/Z axis.
When all axes stop moving, if the error of any comparison is greater than one motor step, the G-code machine will execute following steps:

1. Correct the Actual Position to POS_FB;
2. Use the previous Actual Position (expected G-code position) as the target position;
3. Move to the target position at half the speed set by Self Starting SPD (0x50CA).

The value of POS_FB is equal to the following function:

- *POS_FB = ENC_Counts × FB_Scale + FB_Offset*
- ENC_Counts: Refer to object ENC_Counts (0x502A).
- FB_Scale: Refer to object FB_Scale (0x5028).
- FB_Offset: Refer to object FB_Offset (0x5029).

Index 0x5027 MAX POS Error

When Step Loss Compensation is enabled, if the error between POS_FB and Actual Position exceeds MAX POS Error, a software emergency stop is initiated.

Index	Name	Data Type	Access	Unit	Default
5027:00	Subindex 00	UINT8	ro		> 3 <
5027:01	X	REAL64	rw	mm	0.0 *1
5027:02	Y	REAL64	rw	mm	0.0 *1
5027:03	Z	REAL64	rw	mm	0.0 *1

*1 Factory setting is 0, but setting would be stored in EEPROM after writing.

Index 0x5028 FB_Scale

Scale parameter used to convert encoder counts to POS_FB.

POS_FB = ENC_counts × FB_Scale + FB_Offset

Index	Name	Data Type	Access	Unit	Default
5028:00	Subindex 00	UINT8	ro		> 3 <
5028:01	M1	REAL64	rw	mm/counts	1.0 *1
5028:02	M2	REAL64	rw	mm/counts	1.0 *1
5028:03	M3	REAL64	rw	mm/counts	1.0 *1

*1 Factory setting is 1.0, but setting would be stored in EEPROM after writing.

In CiA-402 mode, FB_Scale used to convert encoder counts to Additional position encoder value(60E4h, 68E4h, 70E4h).

APEV = ENC_counts × FB_Scale + FB_Offset

Index	Name	Data Type	Access	Unit	Default
5028:00	Subindex 00	UINT8	ro		> 3 <
5028:01	M1	REAL64	rw	counts/counts	1.0 *1
5028:02	M2	REAL64	rw	counts/counts	1.0 *1
5028:03	M3	REAL64	rw	counts/counts	1.0 *1

*1 Factory setting is 1.0, but setting would be stored in EEPROM after writing.

Index 0x5029 FB_Offset

Offset parameter used to convert encoder counts to POS_FB.

POS_FB = ENC_counts × FB_Scale + FB_Offset

Index	Name	Data Type	Access	Unit	Default
5029:00	Subindex 00	UINT8	ro		> 3 <
5029:01	M1	REAL64	rw	counts	0.0 *1
5029:02	M2	REAL64	rw	counts	0.0 *1
5029:03	M3	REAL64	rw	counts	0.0 *1

*1 Factory setting is 0, but setting would be stored in EEPROM after writing.

Index 0x502A ENC_Counts

Encoder counts of encoder interface ENC_1, ENC_2 and ENC_3.

Index	Name	Data Type	Access	Unit	Default
502A:00	Subindex 00	UINT8	ro		> 3 <
502A:01	ENC_1	INT32	ro	enc counts	0
502A:02	ENC_2	INT32	ro	enc counts	0
502A:03	ENC_3	INT32	ro	enc counts	0

Index 0x5030 Home Direction

Set homing direction.

Index	Name	Data type	Flags	Default
5030	Home Direction	UINT8	rw	0

Bits 0, 1, and 2 of the object's Home Direction (0x5030) are set in G-code mode to indicate the direction for finding the mechanical origin.

7-3	2	1	0
Reserved	Z direction	Y direction	X direction
	0: negative 1: positive	0: negative 1: positive	0: negative 1: positive

Index 0x5031 Home Priority

When set to Homing, the execution order of each axis starts with Homing from the axis with the highest set value.

- 0x5031.1: Order number of X-axis;
- 0x5031.2: Order number of Y-axis;
- 0x5031.3: Order number of Z-axis.

Index	Name	Data type	Flags	Default
5031:0	Home Priority	-		
5031:1	X	UINT8	rw	0x03
5031:2	Y	UINT8	rw	0x02
5031:3	Z	UINT8	rw	0x01

Index 0x50C0 Machine Mode

When MV3S is running in G-code mode, it can be switched to JOG mode via Machine Mode (0x50C0):

- 0: Normal G-code machine;
- 1: JOG Mode.

Index	Name	Data type	Flags	Default
50C0:00	MachineMode	UINT8	rw	0x00

6.2.2.2 G code Objects

G code machine support G code G54 ~ G59.

Index 0x50C1 G54 Work Offset

User can use object 0x50C1 to set the offset of work coordinate G54 relative to the Machine Home.

- 0x50C1.1: Offset of G54 X-axis relative to Machine Home;
- 0x50C1.2: Offset of G54 Y-axis relative to Machine Home;
- 0x50C1.3: Offset of G54 Z-axis relative to Machine Home.

Index	Name	Data Type	Access	Unit	Default
50C1:00	Subindex 00	UINT8	ro		> 3 <
50C1:01	X	REAL64	rw	mm	0.0
50C1:02	Y	REAL64	rw	mm	0.0
50C1:03	Z	REAL64	rw	mm	0.0

Index 0x50C2 G55 Work Offset

User can use object 0x50C2 to set the offset of work coordinate G55 relative to the Machine Home.

- 0x50C2.1: Offset of G55 X-axis relative to Machine Home;
- 0x50C2.2: Offset of G55 Y-axis relative to Machine Home;
- 0x50C2.3: Offset of G55 Z-axis relative to Machine Home.

Index	Name	Data Type	Access	Unit	Default
50C2:00	Subindex 00	UINT8	ro		> 3 <
50C2:01	X	REAL64	rw	mm	0.0
50C2:02	Y	REAL64	rw	mm	0.0
50C2:03	Z	REAL64	rw	mm	0.0

Index 0x50C3 G56 Work Offset

User can use object 0x50C3 to set the offset of work coordinate G56 relative to the Machine Home.

- 0x50C3.1: Offset of G56 X-axis relative to Machine Home;
- 0x50C3.2: Offset of G56 Y-axis relative to Machine Home;
- 0x50C3.3: Offset of G56 Z-axis relative to Machine Home.

Index	Name	Data Type	Access	Unit	Default
50C3:00	Subindex 00	UINT8	ro		> 3 <
50C3:01	X	REAL64	rw	mm	0.0
50C3:02	Y	REAL64	rw	mm	0.0
50C3:03	Z	REAL64	rw	mm	0.0

Index 0x50C4 G57 Work Offset

User can use object 0x50C4 to set the offset of work coordinate G57 relative to the Machine Home.

- 0x50C4.1: Offset of G57 X-axis relative to Machine Home;
- 0x50C4.2: Offset of G57 Y-axis relative to Machine Home;
- 0x50C4.3: Offset of G57 Z-axis relative to Machine Home.

Index	Name	Data Type	Access	Unit	Default
50C4:00	Subindex 00	UINT8	ro		> 3 <
50C4:01	X	REAL64	rw	mm	0.0
50C4:02	Y	REAL64	rw	mm	0.0
50C4:03	Z	REAL64	rw	mm	0.0

Index 0x50C5 G58 Work Offset

User can use object 0x50C5 to set the offset of work coordinate G58 relative to the Machine Home.

- 0x50C5.1: Offset of G58 X-axis relative to Machine Home;
- 0x50C5.2: Offset of G58 Y-axis relative to Machine Home;
- 0x50C5.3: Offset of G58 Z-axis relative to Machine Home.

Index	Name	Data Type	Access	Unit	Default
50C5:00	Subindex 00	UINT8	ro		> 3 <
50C5:01	X	REAL64	rw	mm	0.0
50C5:02	Y	REAL64	rw	mm	0.0
50C5:03	Z	REAL64	rw	mm	0.0

Index 0x50C6 G59 Work Offset

User can use object 0x50C6 to set the offset of work coordinate G59 relative to the Machine Home.

- 0x50C6.1: Offset of G59 X-axis relative to Machine Home;
- 0x50C6.2: Offset of G59 Y-axis relative to Machine Home;
- 0x50C6.3: Offset of G59 Z-axis relative to Machine Home.

Index	Name	Data Type	Access	Unit	Default
50C6:00	Subindex 00	UINT8	ro		> 3 <
50C6:01	X	REAL64	rw	mm	0.0
50C6:02	Y	REAL64	rw	mm	0.0
50C6:03	Z	REAL64	rw	mm	0.0

Index 0x50CA Self Starting SPD

This parameter sets the speed the motor can reach directly without acceleration. If the motor manufacturer provides this parameter, the user can use it directly; otherwise, the user must set it manually.

- 0x50CA.1: Stepper motor auto-start speed for M1;
- 0x50CA.2: Stepper motor auto-start speed for M2;
- 0x50CA.3: Stepper motor auto-start speed for M3.

Index	Name	Data Type	Access	Unit	Default
50CA:00	Subindex 00	UINT8	ro		> 3 <
50CA:01	M1	REAL64	rw	rpm	60.0*[1]
50CA:02	M2	REAL64	rw	rpm	60.0*[1]
50CA:03	M3	REAL64	rw	rpm	60.0*[1]

*1 Factory setting is 60, but setting would be stored in EEPROM after writing.

Index 0x50CB Jog Increment

If the jog is in jog mode, writing a jog increment to an object via the mailbox will move that axis one distance, specified by the jog increment.

- 0x50CB.1: X-axis jog increment;
- 0x50CB.2: Y-axis jog increment;
- 0x50CB.3: Z-axis jog increment;

Index	Name	Data Type	Access	Unit	Default
50CB:00	Subindex 00	UINT8	ro		> 3 <
50CB:01	X	REAL64	rw	mm	0.0
50CB:02	Y	REAL64	rw	mm	0.0
50CB:03	Z	REAL64	rw	mm	0.0

* **Notes** about jog mode: The movement distance in jog mode is still limited by the Position Limit (0x50D5) unless the Enable Position Limit (0x50DD) is turned off to 0; After receiving a jog increment in jog mode, a new jog increment cannot be sent before reaching the specified position, and the Mailbox will reply with Abort Codes 0x19.

Index 0x50CC Jog Function

Set the Jog execution mode: 0 for jog mode, 1 for continuous mode;

- 0x50CC.1: Jog execution mode for the X-axis;
- 0x50CC.2: Jog execution mode for the Y-axis;
- 0x50CC.3: Jog execution mode for the Z-axis.

Index	Name	Data Type	Access	Default
50CC:00	Subindex 00	UINT8	ro	> 3 <
50CC:01	X	UINT8	rw	0
50CC:02	Y	UINT8	rw	0
50CC:03	Z	UINT8	rw	0

Index 0x50CD Jog Command

In continuous mode, if the value is greater than zero, it will continuously move at a constant speed in the positive direction at half the Self Starting SPD (0x50CA); if the value is less than zero, it will continuously move at a constant speed in the negative direction at half the Self Starting SPD (0x50CA).

- 0x50CD.1: Jog Command for the X-axis;
- 0x50CD.2: Jog Command for the Y-axis;
- 0x50CD.3: Jog Command for the Z-axis.

Index	Name	Data Type	Access	Default
50CD:00	Subindex 00	UINT8	ro	> 3 <
50CD:01	X	REAL64	rw	0.0
50CD:02	Y	REAL64	rw	0.0
50CD:03	Z	REAL64	rw	0.0

Index 0x50D0 G Code Header

Transmit G-code via PDO: 0x50D0 and 0x50D1 are designed to transmit G-code via PDO.

Index	Name	Data type	Flags	Default
50D0	G Code Header	UINT16	rw	0x0000 (0)

The header of the G code string data during PDO transmission. The functions of each bit in the header are as follows:

[15 – 9]	8	[7 - 5]	4	[3 – 0]
Reserved	Finish	Reserved	Odd Even	String Segment Number

- **Finish**: If the string is the last string of G code in the line, Finish must be set to 1.
  After receiving Finish as 1, the Controller will assemble the string and start parsing the G code.
- **Odd Even**: Specify whether the number of G code lines in the transmission is an odd or even number. Changing the parity bit is equivalent to ending the transmission of the G code in that line. If Finish has not been executed, all previously transmitted strings will be discarded.
- **String Segment Number**: String Segment Number valid values are 1~14.

* Transmit G-code via PDO: 0x50D0 and 0x50D1 are designed to transmit G-code via PDO.

Index 0x50D1 G Code string

Transmit G-code via PDO: 0x50D0 and 0x50D1 are designed to transmit G-code via PDO.
Object for PDO transmission, the object type is string (26 bytes).

Index	Name	Data Type	Access	Default
50D1	G Code string	STRING(26)	rw	

* Transmit G-code via PDO: 0x50D0 and 0x50D1 are designed to transmit G-code via PDO.

Examples of PDO transmitting G-code:
If string length is less than 26 (25 characters + '\0'), G-code can transmit completely in one PDO.

PDO Cycle	0x50D0	0x50D1	0x50E0
0	0x0000		0x0000
1	0x0101	G1 X100.0 F1536.0	0x0000
2	0x0101	G1 X100.0 F1536.0	0x000F
3	0x0111	G1 Y50.0 F768.0	0x000F
4	0x0111	G1 Y50.0 F768.0	0x001F
5	0x0101	G1 X0.0 Y0.0 F1536.0	0x001F
6	0x0101	G1 X0.0 Y0.0 F1536.0	0x000F

If string length is over than 26 (G2 X2000000.0 Y2000000.0 Z100.0 I2000000.0 J0.0 F1536.0), G-code must be divided into multiple segments and transmitted in multiple PDO cycles.

PDO Cycle	0x50D0	0x50D1	0x50E0
0	0x0000		0x0000
1	0x0001	"G2 X2000000.0 Y2000000.0 "	0x0000
2	0x0001	"G2 X2000000.0 Y2000000.0 "	0x0001
3	0x0002	“Z100.0 I2000000.0 J0.0 F1"	0x0001
4	0x0002	“Z100.0 I2000000.0 J0.0 F1"	0x0002
5	0x0103	“536.0”	0x0002
6	0x0103	“536.0”	0x000F

Index 0x50D3 Max Velocity

The components of the feed rate (e.g. G0 F600.0) on the X, Y, and Z axes are Fx, Fy, and Fz. The relationship between them is:
*Feed-rate = sqrt(Fx*Fx + Fy*Fy + Fz*Fz)*

In G-code mode, in order to avoid the X-Y-Z motors losing step due to too high feed-rate, the user can set the maximum feed-rate of the X-Y-Z motors through Max Velocity.

Index	Name	Data Type	Access	Unit	Default
50D3:00	Max Velocity	UINT8	ro		> 3 <
50D3:01	X axis	REAL64	rw	mm/s	24000.0
50D3:02	Y axis	REAL64	rw	mm/s	24000.0
50D3:03	Z axis	REAL64	rw	mm/s	24000.0

The Max Velocity of the X-axis is Fxmax. When Fx exceeds Fxmax, the feed-rate will be recalculated according to the following formula:

- *R = Fxmax / Fx*
- *feed-ratenew = sqrt(Fxmax * Fxmax + Fy*R * Fy*R + Fz*R * Fz*R)*

Index 0x50D4 Max Acceleration

Set the acceleration to the feed-rate and the deceleration to stop moving.

Index	Name	Data Type	Access	Unit	Default
50D4	Max Acceleration	REAL64	rw	mm/s^2	180000.0

Index 0x50D5 Position Limit

The command position specified by the G-code cannot exceed the range set by Position Limit. For example, if Xmin is 0.0 and "G1 X-10.0 Y5.0 Z 5.0" is executed, the actual position moved to is (0.0, 5.0, 5.0).

Index	Name	Data Type	Access	Unit	Default
50D5:00	Position Limit	UINT8	ro		> 6 <
50D5:01	Min of X axis	REAL64	rw	mm	-1000.0
50D5:02	Max of X axis	REAL64	rw	mm	1000.0
50D5:03	Min of Y axis	REAL64	rw	mm	-1000.0
50D5:04	Max of Y axis	REAL64	rw	mm	1000.0
50D5:05	Min of Z axis	REAL64	rw	mm	-1000.0
50D5:06	Max of Z axis	REAL64	rw	mm	1000.0

Index 0x50D6 Homing Speed

Set the moving speed of the X, Y and Z axes when searching for the machine zero point.

Index	Name	Data Type	Access	Unit	Default
50D6:00	Homing Speed	UINT8	ro		> 3 <
50D6:01	X axis	REAL64	rw	mm / min	-
50D6:02	Y axis	REAL64	rw	mm / min	-
50D6:03	Z axis	REAL64	rw	mm / min	-

Index 0x50D7 Servo On

Change G-code machine to Servo-On or Servo-Off. When Servo-Off, G-code cannot be received and Homing cannot be executed.

Definition of 0x50D7:

- 0: Servo-Off
- 1: Servo-On

Index	Name	Data type	Flags	Default
50D7	Servo On	UINT8	rw	0x00 (0)

Index 0x50D8 Home

Set 0x50D8 will execute searching machine zero point. Motor will rotate in the direction specified by object 0x5030 until home switch is triggered.

Definition of 0x50D8:

- 0 → 2: Start homing
- 2 → 0: Stop homing

Index	Name	Data type	Flags	Default
50D8	Home	UINT8	rw	0x00 (0)

Index 0x50D9 Emergency Stop

Set 0x50D9 will execute Emergency Stop. After entering Emergency Stop, motor will stop outputting torque and discard all G-code commands. In order to exit Emergency Stop, Clear EMG Stop should be executed.

Definition of 0x50D9:

- 0: Allow Clear EMG Stop to be executed
- 4: Enter or stay in Emergency Stop state

Index	Name	Data type	Flags	Default
50D9	Emergency Stop	UINT8	rw	0x00 (0)

Index 0x50DA Clear EMG Stop

When object 0x50D9 equals 0, set 0x50DA will execute Clear EMG Stop.

Definition of 0x50DA:

- 0: Nothing
- 1: Execute Clear EMG Stop

Index	Name	Data type	Flags	Default
50DA	Clear EMG Stop	UINT8	rw	-

When the Controller enters Emergency Stop, it will stop acting and will no longer receive any instructions other than Clear EmgStop. To leave Emergency Stop, Clear EmgStop must be executed; And when the hardware EmgStop Input pin is high and the object Emergency Stop (0x50D9) is also 0, changing the content of the object Clear EmgStop (0x50DA) from 0 to 1 will execute Clear EmgStop to leave the Emergency Stop.

Index 0x50DB Default Feedrate

If feed-rate is never specified, feed-rate is equal to the value of 0x50DB.
If first G-code is “G1 X10.0” and the value of 0x50DB is 1000, X-axis motor will rotate with “G1 X10.0 F1000.0”.

Index	Name	Data Type	Access	Unit	Default
50DB	Default feed rate	REAL64	rw	mm / min	0

When parsing the G Code string, if the G Code string does not contain Feed-rate, and the previously sent G Code string also does not contain Feed-rate, G Code Controller will use the setting of the object Default Feedrate as Feed-rate. ;
The unit of Default Feedrate is mm per minute, and the object data type is REAL64.

Index 0x50DC Default Homing SPD

Object 0x50DC provide a default value of Homing Speed(0x50D6). The value of 0x50D6 will be set to the value of 0x50DC after G-code machine boots.

Index	Name	Data Type	Access	Unit	Default
50DC:00	Subindex 00	UINT8	ro		> 3 <
50DC:01	X axis	REAL64	rw	mm / min	0
50DC:02	Y axis	REAL64	rw	mm / min	0
50DC:03	Z axis	REAL64	rw	mm / min	0

Index 0x50DD Enable Position Limit

When Enable Position Limit is enabled, the target position of G-code will be limited by the position limit (0x50D5).

Setting object 0x50DD can enable/disable position limit:

- 0: disable position limit
- 1: enable position limit

Index	Name	Data type	Flags	Default
50DD	Enable Position Limit	UINT8	rw	0x01 (1)

Index 0x50E0 G code Statusword

Object 0x50E0 provide a G-code Statusword.

Index	Name	Data type	Flags	Default
50E0	G code Statusword	UINT16	rw	0x0000 (0)

Bit Definition:

[15-8]	5	4	[3 – 0]
Error Code	Discard	Odd Even	String Segment Counter

Bit Description:

- **Error Code**: After changing the Finish bit of G Code Header to 1, the Error Code of G Code Statusword will be refreshed to display the processing results of G Code:
  - 0: No error, G code has been received successfully
  - 1: The trajectory planning buffer is full, keep Finish at 1 and wait for the Error Code to clear to 0, or change Odd Even to reset the string transmission
  - 2: Finish received in Servo Off state, G code reception failed
  - 3: Finish received in Emergency Stop state, G code reception failed

  Failure to receive the G code will not clear the previously transmitted string. After the error state is resolved, directly change the Finish bit to 1 to try processing the G code again.
- **Discard**: If Finish is not executed successfully, if the odd or even number of G code lines changes, the transmitted string will be discarded and Discard will be changed to 1.
- **Odd Even**: Displays whether the current number of rows is an odd or even number. This bit reflects the odd or even number of rows specified by the MDevice, and the controller will not change it by itself.
- **String Segment Counter**: String Segment Counter represents the number of G code strings that have been written.

Using PDO to transmit a line of G code usually requires multiple transmissions. String Segment is used to allow the user to control the progress of transmitting multiple G code strings and the Controller to return the number of G code strings received. When the Controller replies, the String Segment Counter is 0x0F, it means that the transmitted single line of G code has been successfully processed.

If the String Segment Counter is n (0~13), the String Segment Number must be n+1 for the Controller to receive the new string.

String Segment Number	Sting Segment Counter	Action
n + 1	n	Receive string to FIFO
Any other		Skip

Discard Bit of 0x50E0:

If Odd Even bit of 0x50D0 changed before finish bit of 0x50D0 is set to 1, discard bit of 0x50E0 would be set to 1.

PDO Cycle	0x50D0	0x50D1	0x50E0
0	0x0000		0x0000
1	0x0001	"G2 X2000000.0 Y2000000.0 "	0x0000
2	0x0001	"G2 X2000000.0 Y2000000.0 "	0x0001
3	0x0011	"G1 X2000000.0 Y2000000.0 "	0x0001
4	0x0011	"G1 X2000000.0 Y2000000.0 "	0x0031
5	0x0112	“F1500.0”	0x0011
6	0x0112	“F1500.0”	0x001F

Index 0x50E1 Machine Status

Object 0x50E1 provide a Machine Status.

Index	Name	Data type	Flags	Default
50E1	Machine Status	UINT8	rw	0x00 (0)

Bit Definition:

[7 – 4]	3	2	1	0
Reserved	Emergency Stop	Is Moving	Homing attained	Servo On/Off
	If this bit is 1, it means that the EmgStop of the software or hardware has been pressed down. To clear EmgStop, Clear EmgStop must be executed and the EmgStop pin of the hardware must be pulled up.	If Is Moving is 1, it means that any axis motor is in a moving state, and 0 means that each XYZ axis is in a stationary state.	After Homing is completed, this bit changes to 1. After Home (0x50D8) is cleared to 0, this bit is cleared to 0.	0: Servo Off, 1: Servo On.

Bit Description:

Bit	Definition	Description
0	Servo On/Off	0: Servo Off, 1: Servo On.
1	Homing attained	0: Homing has not been executed yet or the motors have not reached the limit switch. 1: The motors reach the limit switch.
2	Is Moving	0: The motors of XYZ axes stop rotating. 1: The motors of XYZ are rotating.
3	Emergency Stop	0: Machine is in normal state. 1: Machine is in Emergency Stop state.
4	X POS Error Exceeds	When this bit is 1, it indicates that an X-axis event has occurred Position Error Exceeds ◦
5	Y POS Error Exceeds	When this bit is 1, it indicates that an X-axis event has occurred Position Error Exceeds ◦
6	Z POS Error Exceeds	When this bit is 1, it indicates that an X-axis event has occurred Position Error Exceeds ◦

Bits - POS Error Exceeds:

Conditions for setting POS Error Exceeds to 1: The G-code Machine is in the isMoving state, and the error between POS_FB and Actual Position exceeds MAX POS Error;

Conditions for setting POS Error Exceeds to 0: Clear EMG Stop is executed.

Index 0x50E2 Actual Position

Read-only file, returns the position of the three axes of XYZ in mm.

Index	Name	Data Type	Access	Unit	Default
50E2:00	Subindex 00	UINT8	ro		> 3 <
50E2:01	X axis	REAL64	ro	mm	0
50E2:02	Y axis	REAL64	ro	mm	0
50E2:03	Z axis	REAL64	ro	mm	0

Index 0x50E3 Limit Inputs

The input status of the XYZ three-axis Home Switch is 0, indicating that the Home Switch is not triggered, and 1, indicating that it is triggered. The signal input source of the XYZ Home Switch is determined according to the setting of the object MotorMapping (0x50FB).

Index	Name	Data type	Flags	Default
50E3	Limit Inputs	UINT8	ro	-

Bit Definition:

[7 – 3]	2	1	0
Reserved	Z limit	Y limit	X limit

Index 0x50FA Motor Feedback Source

Set the source for calculating ENC_Counts in the POS_FB of axes M1, M2, and M3. 1 represents selecting the ENC_1 interface, 2 represents selecting the ENC_2 interface, and 3 represents selecting the ENC_3 interface. If set to a value other than 1/2/3, objects 0x60E4/0x68E4/0x70E4 will no longer be updated. In G-code mode, because POS_FB is no longer updated, enabling the Step Loss Compensation function and executing G-code may cause Max POS Exceeds.

Index	Name	Data Type	Access	Default
50FA:00	Subindex 00	UINT8	ro	> 3 <
50FA:01	M1	UINT8	rw	1 *1
50FA:02	M2	UINT8	rw	2 *1
50FA:03	M3	UINT8	rw	3 *1

*1 Factory setting is 1/2/3, but setting would be stored in EEPROM after writing.

Index 0x50FB Motor Mapping

By default, X-axis corresponds to the M1 motor interface, Y-axis corresponds to M2, Z-axis corresponds to M3. Writing object 0x50FB and then rebooting device will swap the motor mappings.

Index	Name	Data type	Flags	Default
50FB	Motor Mapping	UINT8	rw	0x00 (0)

Value Description:

0x50FB	X axis	Y axis	Z axis
0	M1	M2	M3
1	M1	M3	M2
2	M2	M1	M3
3	M2	M3	M1
4	M3	M1	M2
5	M3	M2	M1

When Servo is powered on, it will read the SPI ROM through the EEPROM Lib and set the initial value of the Motor Mapping. Subsequent writing to the Motor Mapping will only change the storage content of the SPI ROM. The Motor Mapping and reading results will not change.

Index 0x50FC Device Profile

When the value of object 0x50FC is 1, MV3S receives G-code and interpolates the 3-axis motor rotation.
If the value of 0x50FC is not 1, MV3S is a 3-axis CiA-402 device.

Index	Name	Data type	Flags	Default
50FC	Device Profile	UINT8	rw	0x00 (0)

When Servo is powered on, it will read the SPI ROM through the EEPROM Lib and set the initial value of Device Profile (0x50FC). 1 represents G code Controller mode, 0 (except 1) represents CiA-402 Servos mode;

Writing to the Device Profile (0x50FC) after booting will only change the address content of the SPI ROM where the Device Profile is stored. The results of Servo mode and reading the Device Profile will not change.

Index 0x50FD Reverse Direction

In G-code mode, the forward and reverse rotation of the motor can be defined. If the value of object 0x50FD is 0, the motor direction is the default direction. If 1, the motor direction is reversed. This object can only be set when Servo-Off.

Index	Name	Data type	Flags	Default
50FD:0	Reverse Direction	UINT8	ro	> 3 <
50FD:01	X axis	UINT8	rw	0x00 (0)
50FD:02	Y axis	UINT8	rw	0x00 (0)
50FD:03	Z axis	UINT8	rw	0x00 (0)

Index 0x50FE Pulse of per millimeter

Position unit of G-code command is mm, and one pulse count will cause the motor to rotate one micro-step. Object 0x50FE is the ratio that converts mm to pulse counts. For example, if 0x50FE:01 is 33.6, ”G1 X13.0” will cause X-axis motor to rotate 436 micro-steps.

Index	Name	Data Type	Access	Unit	Default
50FE:00	Subindex 00	UINT8	ro		3
50FE:01	X axis	REAL64	rw	counts	1.0 *[1]
50FE:02	Y axis	REAL64	rw	counts	1.0 *[1]
50FE:03	Z axis	REAL64	rw	counts	1.0 *[1]

*1 Factory setting is 1.0, but setting would be stored in EEPROM after writing.

Index 0x50FF G Code command

MV3S supports receiving G-code via mailbox. Writing G-code string to object 0x50FF instructs MV3S to execute this G-code when Servo-On.
String length should be less than 63 characters.

Index	Name	Data type	Flags	Default
50FF	G Code command	STRING (64)	rw	-

6.2.3 Especial Objects (0x6000-0xFFFF)

For especial objects description.

6.2.3.1 CiA-402 Objects

For the definitions of Velocity Unit and Counts, please refer to the description of object Velocity Unit 0x5014.

Index	Name	Data Type	Access	Unit	Default
603F	Error Code	UINT16	ro		0
6040	Controlword	UINT16	rw		0
6041	Statusword	UINT16	ro		0
6060	Modes of operation	INT8	rw		0
6061	Modes of operation display	INT8	ro		0
6064	Position actual value	INT32	ro	counts	0
606C	Velocity Actual Value	INT32	ro	vel unit	0
607A	Target position	INT32	rw	counts	0
607B	Position range limit				
607B:01	Min	INT32	rw	counts	-2147483648
607B:02	Max	INT32	rw	counts	2147483647
607C	Home offset	INT32	rw	counts	0
607D	Software position limit				
607D:01	Min position limit	INT32	rw	counts	-2000000000
607D:02	Max position limit	INT32	rw	counts	2000000000
607E	Polarity	UINT8	rw		0
6080	Max motor speed	UINT32	rw	RPM	300
6081	Profile velocity	UINT32	rw	counts / s	1
6083	Profile acceleration	UINT32	rw	counts / s^2	1
6098	Homing method	INT32	rw		0
6099	Homing speed				
6099:01	Speed for searching switch	UINT32	rw	vel unit	1
6099:02	Speed for searching zero	UINT32	rw	vel unit	1
609A	Homing acceleration	UINT32	rw	vel unit / s	1
60B0	Position offset	INT32	rw	counts	0
60B1	Velocity offset	INT32	rw	vel unit	0
60B8	Touch probe function	UINT16	rw		0
60B9	Touch probe status	UINT16	ro		0
60BA	Touch probe position 1 positive value	INT32	ro	enc counts	0
60BB	Touch probe position 1 negative value	INT32	ro	enc counts	0
60C2	Interpolation time period				

60C2:01	Interpolation time period value	UINT8	rw		0
60C2:02	Interpolation time index	INT8	rw		0
60D5	Touch probe 1 positive edge counter	UINT16	ro		0
60D6	Touch probe 1 negative edge counter	UINT16	ro		0
60E3	Supported homing methods				
60E3:01		INT8	ro		19
60E3:02		INT8	ro		20
60E3:03		INT8	ro		21
60E3:04		INT8	ro		22
60E3:05		INT8	ro		37
60E4	Additional position encoder value				
60E4:01	First additional position	INT32	ro	enc counts	0
60EF	Motor resolution	UINT32	ro		3200
60FD	Digital inputs	UINT32	ro		0
60FF	Target velocity	INT32	rw	vel unit	0
6502	Supported drive modes	UINT32			0x000001A1

6.2.3.2 Index 0x6nnx Input Data of the Module (0x6000 - 0x6FFF)

For Index 0x6000 - 0x6FFF, the input data of the EtherCAT SubDevice module.

Index 0x603F Error Code

Error code.

Index	Name	Data type	Flags	Default
603F	Error Code	UINT16	ro	0x0000 (0)

Index 0x6040 Controlword

Control word.

Index	Name	Data type	Flags	Default
6040	Controlword	UINT16	rw	0x0000 (0)

Index 0x6041 Statusword

Status word.

Index	Name	Data type	Flags	Default
6041	Statusword	UINT16	ro	0x0000 (0)

Index 0x605A Quick stop option code

Quick stop option code.

Index	Name	Data type	Flags	Default
605A	Quick stop option code	INT16	rw	0x0000 (0)

Index 0x605B Shutdown option code

Shutdown option code.

Index	Name	Data type	Flags	Default
605B	Shutdown option code	INT16	rw	0x0000 (0)

Index 0x605E Fault reaction option code

Fault reaction option code.

Index	Name	Data type	Flags	Default
605E	Fault reaction option code	INT16	rw	0x0000 (0)

Index 0x6060 Modes of operation

Write the modes of operation.

Index	Name	Data type	Flags	Default
6060	Modes of operation	INT8	rw	0

Index 0x6061 Modes of operation display

Modes of operation display.

Index	Name	Data type	Flags	Default
6061	Modes of operation display	INT8	ro	0x00

Index 0x6064 Position actual value

Position actual value.

Index	Name	Data type	Flags	Default
6064	Position actual value	INT32	ro	0x00000000

Index 0x606C Velocity Actual Value

Velocity actual value.

Index	Name	Data type	Flags	Default
606C	Velocity Actual Value	INT32	ro	0x00000000

Index 0x607A Target position

Mandatory if Cyclic-Synchronous Position (CSP) Mode is supported.

Index	Name	Data type	Flags	Default
607A	Target position	INT32	rw	0x00000000

Index 0x607B Position Range Limit

Position Range Limit applies to rotating outputs with a disk as the output end. In speed mode, when the Position actual value exceeds the maximum value of Position Range Limit, the Position actual value will overflow from the minimum value of Position Range Limit and continue to increase in the direction of speed.

Index	Name	Data type	Flags	Default
607B:0	Position range limit	-	ro	> 2 <
607B:01	Min	INT32	rw	-2147483648
607B:02	Max	INT32	rw	2147483648

If the Position Range Limit is 0~359, and the previous cycle's actual Position value was 350, while the motor performed 12 counts during the previous cycle, then the current cycle's actual Position value will be 2. Conversely, if the previous cycle's actual Position value was 1, while the motor performed -2 counts during the previous cycle, then the current cycle's actual Position value will be 359.

Index 0x607C Home offset

Set Home Position.

Index	Name	Data type	Flags	Default
607C	Home offset	INT32	rw	0x00

Index 0x607D Software position limit

Set Position Limit. Recommended if Cyclic-Synchronous Position (CSP) Mode is supported.

Index	Name	Data type	Flags	Default
607D:0	Software position limit	-	-	-
607D:01	Min position limit	INT32	ro, wr_preop	0x88CA6C00
607D:02	Max position limit	INT32	ro, wr_preop	0x77359400

Index 0x607E Polarity

Polarity, save to EEPROM.

Index	Name	Data type	Flags	Default
607E	Polarity	UINT8	rw	0x00 (0)

Index 0x6080 Max motor speed

Set Max motor speed.

Index	Name	Data type	Flags	Default
6080	Max motor speed	UINT32	rw	0x0000012C (300)

Index 0x6081 Profile velocity

Profile velocity. Can’t be zero.

Index	Name	Data type	Flags	Default
6081	Profile velocity	UINT32	rw	0x00000001 (1)

Index 0x6083 Profile acceleration

Profile acceleration. Can’t be zero.

Index	Name	Data type	Flags	Default
6083	Profile acceleration	UINT32	rw	0x00000001 (1)

Index 0x6085 Quick stop deceleration

Quick stop deceleration. Can’t be zero.

Index	Name	Data type	Flags	Default
6085	Quick stop deceleration	UINT32	rw	0x00000001 (1)

Index 0x608B Velocity notation index

Velocity notation index.

Index	Name	Data type	Flags	Default
608B	Velocity notation index	INT8	rw	0

Index 0x608C Velocity dimension index

Velocity dimension index.

Index	Name	Data type	Flags	Default
608C	Velocity dimension index	UINT8	rw	0xA4 (164)

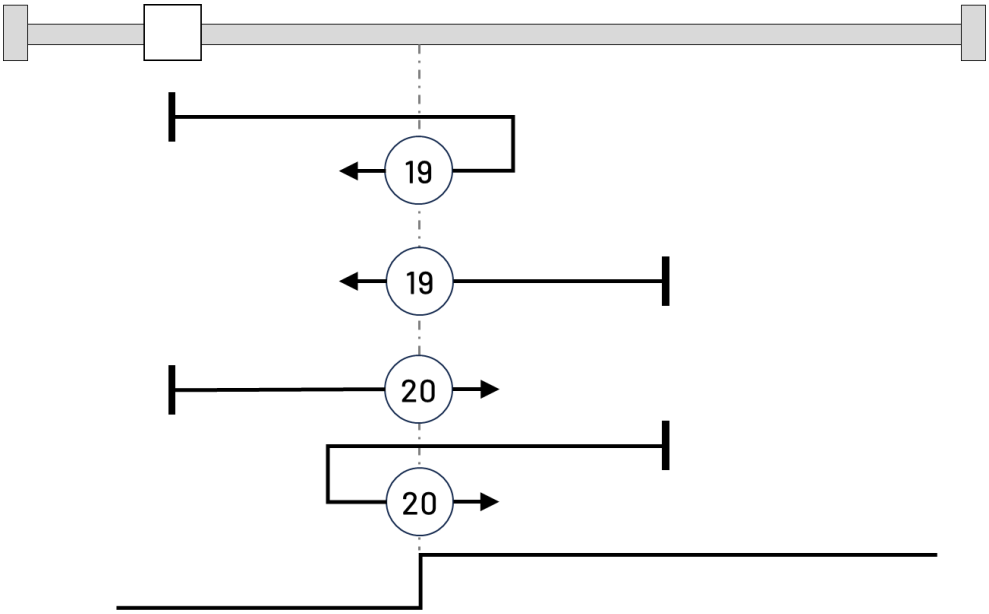
Index 0x6098 Homing method

Specify the return-to-origin method.
Possible values are: 19, 20, 21, 22, and 37. Set other values and perform a return to origin.
If this occurs, a homing error occurs, and 1 is returned to Bit13: Homing error of statusword.

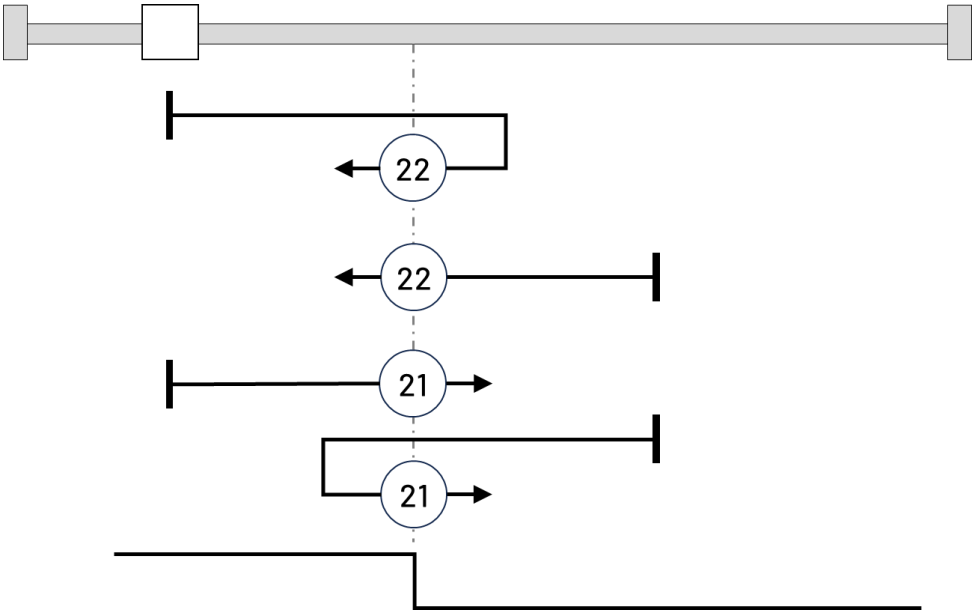
Index	Name	Data type	Flags	Default
6098	Homing method	INT8	rw	-

Reference to the switching edge of the home switch without the index pulse.

With **methods 19** and **20** (equivalent to methods 3 and 4), the left switching edge of the home switch is used as reference:



With **methods 21** and **22** (equivalent to methods 5 and 6), the right switching edge of the home switch is used as reference:



With **method 37**, homing on the current position.
This method is mandatory if homing mode is supported.
In this method, the position sensor information (converted into user-defined position units) shall be taken to be the home position. This method does not require the drive device to be in an operational enabled state.

At the home position (i.e., after the homing process), the position actual value (6064h) is calculated as follows: *Position actual value (6064h) = Home offset (607Ch)*.

For more details, please refer to the CiA402 specification by the "CAN in Automation" organization, which has changed it.

Index 0x6099 Homing speed

Homing speed. Can't be zero.

Index	Name	Data type	Flags	Default
6099:0	Homing speed	UINT8	ro	> 2 <
6099:01	Speed for searching switch	UINT32	rw	0x00000001 (1)
6099:02	Speed for searching zero	UINT32	rw	0x00000001 (1)

Index 0x609A Homing acceleration

Homing acceleration. Can't be zero.

Index	Name	Data type	Flags	Default
609A	Homing acceleration	UINT32	rw	0x00000001 (1)

Index 0x60B0 Position offset

Position offset.

Index	Name	Data type	Flags	Default
60B0	Position offset	INT32	rw	0

Index 0x60B1 Velocity offset

Velocity offset.

Index	Name	Data type	Flags	Default
60B1	Velocity offset	INT32	rw	0

Index 0x60B8 Touch probe function

Set Touch probe.

Index	Name	Data type	Flags	Default
60B8	Touch probe function	UINT16	rw	0x0000 (0)

Index 0x60B9 Touch probe status

Touch probe status.

Index	Name	Data type	Flags	Default
60B9	Touch probe status	UINT16	rw	0x0000 (0)

Index 0x60BA Touch probe position 1 positive value

Touch probe position 1 positive value.

Index	Name	Data type	Flags	Default
60BA	Touch probe position 1 positive value	INT32	ro	0

Index 0x60BB Touch probe position 1 negative value

Touch probe position 1 negative value.

Index	Name	Data type	Flags	Default
60BB	Touch probe position 1 negative value	INT32	ro	0

Index 0x60C2 Interpolation time period

Interpolation time period.

Index	Name	Data type	Flags	Default
60C2:0	Interpolation time period	-	-	-
60C2:01	Interpolation time period value	UINT8	rw	0x00 (0)
60C2:02	Interpolation time index	INT8	rw	0

Index 0x60D5 Touch probe 1 positive edge counter

Touch probe 1 positive edge counter.

Index	Name	Data type	Flags	Default
60D5	Touch probe 1 positive edge counter	UINT16	ro	0x0000 (0)

Index 0x60D6 Touch probe 1 negative edge counter

Touch probe 1 negative edge counter.

Index	Name	Data type	Flags	Default
60D6	Touch probe 1 negative edge counter	UINT16	ro	0x0000 (0)

Index 0x60E3 Supported homing methods

List of the homing methods supported by the drive.

Index	Name	Data type	Flags	Default
60E3:0	Supported homing methods	-	-	-
60E3:01	-	INT8	ro	19
60E3:02	-	INT8	ro	20
60E3:03	-	INT8	ro	21
60E3:04	-	INT8	ro	22
60E3:05	-	INT8	ro	37

Index 0x60E4 Additional position encoder value

G-code is an object shared with CiA-402 mode and returns the encoder input of the M1 axis, M2 axis, and M3 axis.

Index	Name	Data type	Flags	Default
60E4:0	Additional position encoder value	-	-	-
60E4:01	First additional position	INT32	ro	---

Index 0x60EF Motor resolution

Motor resolution.

Index	Name	Data type	Flags	Default
60EF	Motor resolution	UINT32	ro	0x00000C80 (3200)

Index 0x60FD Digital inputs

Digital Input 0x60FD/0x68FD/0x70FD are objects defined by the CiA-402 specification. Bits 16-31 are manufacturer-defined functions. In QEC-RXXMV3S, these custom bits are used to indicate the input status of the encoder interface.

Index	Name	Data type	Flags	Default
60FD	Digital inputs	UINT32	ro	0x00000000 (0)

Among them:

- Bits 16-19 in object 0x60FD respond to the input status of encoder interface ENC_1
- Bits 16-19 in object 0x68FD respond to the input status of encoder interface ENC_2
- Bits 16-19 in object 0x70FD respond to the input status of encoder interface ENC_3

Bit Description:

[3-0]	CiA-402 bits	
16	ENC pin A	The encoder pin A input state is 0 for low potential and 1 for high potential.
17	ENC pin B	The encoder pin B input state is 0 for low potential and 1 for high potential.
18	ENC pin Z	The encoder pin Z input state is 0 for low potential and 1 for high potential.
19	ENC Input Dir	The current direction of the encoder signal count: a positive bit is 0 and a negative bit is 1. A positive direction means the last signal pulse was +1, and a negative direction means the last signal pulse was -1.
20	Alarm	The level of the alarm digital input: 0 indicates low level, and 1 indicates high level.

Index 0x60FF Target velocity

Mandatory if Profile Velocity (PV) or Cyclic Synchronous Velocity (CSV) Modes are supported.

Index	Name	Data type	Flags	Default
60FF	Target velocity	INT32	rw	0

Index 0x6502 Supported drive modes

Supported drive modes. Bit7 (csp) and Bit8 (csv) are set.

Index	Name	Data type	Flags	Default
6502	Supported drive modes	UINT32	ro	0x000001A1 (417)

Index 0x6840 Controlword

Control word.

Index	Name	Data type	Flags	Default
6840	Controlword	UINT16	rw	0x0000 (0)

Index 0x6841 Statusword

Status word.

Index	Name	Data type	Flags	Default
6841	Statusword	UINT16	ro	0x0000 (0)

Index 0x6860 Modes of operation

Write the modes of operation.

Index	Name	Data type	Flags	Default
6860	Modes of operation	INT8	rw	0

Index 0x6861 Modes of operation display

Modes of operation display.

Index	Name	Data type	Flags	Default
6861	Modes of operation display	INT8	ro	0

Index 0x6864 Position actual value

Position actual value.

Index	Name	Data type	Flags	Default
6864	Position actual value	INT32	ro	0

Index 0x686C Velocity Actual Value

Velocity actual value.

Index	Name	Data type	Flags	Default
686C	Velocity Actual Value	INT32	ro	0

Index 0x687A Target position

Mandatory if Cyclic Synchronous Position (CSP) Mode is supported.

Index	Name	Data type	Flags	Default
687A	Target position	INT32	rw	0

Index 0x687C Home offset

Home offset.

Index	Name	Data type	Flags	Default
687C	Home offset	INT32	rw	0

Index 0x687D Software position limit

Set Position Limit. Recommended if Cyclic Synchronous Position (CSP) is supported.

Index	Name	Data type	Flags	Default
687D:0	Software position limit	-	-	-
687D:01	Min position limit	INT32	ro, wr_preop	0x88CA6C00
687D:02	Max position limit	INT32	ro, wr_preop	0x77359400

Index 0x687E Polarity

Polarity. Save to EEPROM.

Index	Name	Data type	Flags	Default
687E	Polarity	UINT8	rw	0x00 (0)

Index 0x6880 Max motor speed

Set Max motor speed.

Index	Name	Data type	Flags	Default
6880	Max motor speed	UINT32	rw	0x0000012C (300)

Index 0x6881 Profile velocity

Profile velocity. Can’t be zero.

Index	Name	Data type	Flags	Default
6881	Profile velocity	UINT32	rw	0x00000001 (1)

Index 0x6883 Profile acceleration

Profile acceleration. Can’t be zero.

Index	Name	Data type	Flags	Default
6883	Profile acceleration	UINT32	rw	0x00000001 (1)

Index 0x688B Velocity notation index

Velocity notation index.

Index	Name	Data type	Flags	Default
688B	Velocity notation index	INT8	rw	0

Index 0x688C Velocity dimension index

Velocity dimension index.

Index	Name	Data type	Flags	Default
688C	Velocity dimension index	UINT8	rw	0xA4 (164)

Index 0x6898 Homing method

Same as Object 0x6098 usage, please refer to Index 0x6098 Homing method.

Index	Name	Data type	Flags	Default
6898	Homing method	INT8	rw	0

Index 0x6899 Homing speed

Home speed. Can't be zero.

Index	Name	Data type	Flags	Default
6899:0	Homing speed	-	-	-
6899:01	Speed for searching switch	UINT32	rw	0x00000001 (1)
6899:02	Speed for searching zero	UINT32	rw	0x00000001 (1)

Index 0x689A Homing acceleration

Homing acceleration. Can't be zero.

Index	Name	Data type	Flags	Default
689A	Homing acceleration	UINT32	rw	0x00000001 (1)

Index 0x68B0 Position offset

Position offset.

Index	Name	Data type	Flags	Default
68B0	Position offset	INT32	rw	0

Index 0x68B1 Velocity offset

Velocity offset.

Index	Name	Data type	Flags	Default
68B1	Velocity offset	INT32	rw	0

Index 0x68B8 Touch probe function

Touch probe function.

Index	Name	Data type	Flags	Default
68B8	Touch probe function	UINT16	rw	0x0000 (0)

Index 0x68B9 Touch probe status

Touch probe status.

Index	Name	Data type	Flags	Default
68B9	Touch probe status	UINT16	rw	0x0000 (0)

Index 0x68BA Touch probe position 1 positive value

Touch probe position 1 positive value.

Index	Name	Data type	Flags	Default
68BA	Touch probe position 1 positive value	INT32	ro	0

Index 0x68BB Touch probe position 1 negative value

Touch probe position 1 negative value.

Index	Name	Data type	Flags	Default
68BB	Touch probe position 1 negative value	INT32	ro	0

Index 0x68C2 Interpolation time period

Interpolation time period.

Index	Name	Data type	Flags	Default
68C2:0	Interpolation time period	-	-	-
68C2:01	Interpolation time period value	UINT8	rw	0x00 (0)
68C2:02	Interpolation time index	UINT8	rw	0

Index 0x68D5 Touch probe 1 positive edge counter

Touch probe 1 positive edge counter.

Index	Name	Data type	Flags	Default
68D5	Touch probe 1 positive edge counter	UINT16	ro	0x0000 (0)

Index 0x68D6 Touch probe 1 negative edge counter

Touch probe 1 negative edge counter.

Index	Name	Data type	Flags	Default
68D6	Touch probe 1 negative edge counter	UINT16	ro	0x0000 (0)

Index 0x68E3 Supported homing methods

List of the homing methods supported by the drive.

Index	Name	Data type	Flags	Default
68E3:0	Supported homing methods	-	-	-
68E3:01	-	INT8	ro	19
68E3:02	-	INT8	ro	20
68E3:03	-	INT8	ro	21
68E3:04	-	INT8	ro	22
68E3:05	-	INT8	ro	37

Index 0x68E4 Additional position encoder value

Additional position encoder value.

Index	Name	Data type	Flags	Default
68E4:0	Additional position encoder value	-	-	-
68E4:01	First additional position	INT32	ro	---

Index 0x68EF Motor resolution

Motor resolution.

Index	Name	Data type	Flags	Default
68EF	Motor resolution	UINT32	ro	0x00000C80 (3200)

Index 0x68FD Digital inputs

Same as index 0x60FD. Please refer to Index 0x60FD Digital Inputs.

Index	Name	Data type	Flags	Default
68FD	Digital inputs	UINT32	ro	0x00000000 (0)

Index 0x68FF Target velocity

Mandatory if Profile Velocity (PV) or Cyclic Synchronous Velocity (CSV) is supported.

Index	Name	Data type	Flags	Default
68FF	Target velocity	INT32	rw	0

Index 0x6D02 Supported drive modes

Supported drive modes.

Index	Name	Data type	Flags	Default
6D02	Supported drive modes	UINT32	ro	0x000001A1 (417)

6.2.3.3 Index 0x7nnx Output Data of the Module (0x7000 - 0x7FFF)

For Index 0x7000 - 0x7FFF, the output data of the EtherCAT SubDevice module.

Index 0x7040 Controlword

Control word.

Index	Name	Data type	Flags	Default
7040	Controlword	UINT16	rw	0x0000 (0)

Index 0x7041 Statusword

Status word.

Index	Name	Data type	Flags	Default
7041	Statusword	UINT16	ro	0x0000 (0)

Index 0x7060 Modes of operation

Write the modes of operation.

Index	Name	Data type	Flags	Default
7060	Modes of operation	INT8	rw	0

Index 0x7061 Modes of operation display

Modes of operation display.

Index	Name	Data type	Flags	Default
7061	Modes of operation display	INT8	ro	0

Index 0x7064 Position actual value

Position actual value.

Index	Name	Data type	Flags	Default
7064	Position actual value	INT32	ro	0

Index 0x706C Velocity Actual Value

Velocity actual value.

Index	Name	Data type	Flags	Default
706C	Velocity Actual Value	INT32	ro	0

Index 0x707A Target position

Mandatory if Cyclic Synchronous Position (CSP) Mode is supported.

Index	Name	Data type	Flags	Default
707A	Target position	INT32	rw	0

Index 0x707C Home offset

Set Home Position.

Index	Name	Data type	Flags	Default
707C	Home offset	INT32	rw	0

Index 0x707D Software position limit

Set Position Limit. Recommended if Cyclic Synchronous Position (CSP) Mode is supported.

Index	Name	Data type	Flags	Default
707D:0	Software position limit	-	-	-
707D:01	Min position limit	INT32	ro, wr_preop	0x88CA6C00
707D:02	Max position limit	INT32	ro, wr_preop	0x77359400

Index 0x707E Polarity

Polarity. Save to EEPROM.

Index	Name	Data type	Flags	Default
707E	Polarity	UINT8	rw	0x00 (0)

Index 0x7080 Max motor speed

Set Max motor speed.

Index	Name	Data type	Flags	Default
7080	Max motor speed	UINT32	rw	0x0000012C (300)

Index 0x7081 Profile velocity

Profile velocity. Can’t be zero.

Index	Name	Data type	Flags	Default
7081	Profile velocity	UINT32	rw	0x00000001 (1)

Index 0x7083 Profile acceleration

Profile acceleration. Can’t be zero.

Index	Name	Data type	Flags	Default
7083	Profile acceleration	UINT32	rw	0x00000001 (1)

Index 0x708B Velocity notation index

Velocity notation index.

Index	Name	Data type	Flags	Default
708B	Velocity notation index	INT8	rw	0

Index 0x708C Velocity dimension index

Velocity dimension index.

Index	Name	Data type	Flags	Default
708C	Velocity dimension index	UINT8	rw	0xA4 (164)

Index 0x7098 Homing method

Same as Object 0x6098 usage, please refer to Index 0x6098 Homing method.

Index	Name	Data type	Flags	Default
7098	Homing method	INT8	rw	0

Index 0x7099 Homing speed

Homing speed. Can't be zero.

Index	Name	Data type	Flags	Default
7099:0	Homing speed	-	-	-
7099:01	Speed for searching switch	UINT32	rw	0x00000001 (1)
7099:02	Speed for searching zero	UINT32	rw	0x00000001 (1)

Index 0x709A Homing acceleration

Homing acceleration. Can't be zero.

Index	Name	Data type	Flags	Default
709A	Homing acceleration	UINT32	rw	0x00000001 (1)

Index 0x70B0 Position offset

Position offset.

Index	Name	Data type	Flags	Default
70B0	Position offset	INT32	rw	0

Index 0x70B1 Velocity offset

Velocity offset.

Index	Name	Data type	Flags	Default
70B1	Velocity offset	INT32	rw	0

Index 0x70B8 Touch probe function

Set Touch probe.

Index	Name	Data type	Flags	Default
70B8	Touch probe function	UINT16	rw	0x0000 (0)

Index 0x70B9 Touch probe status

Touch probe status.

Index	Name	Data type	Flags	Default
70B9	Touch probe status	UINT16	rw	0x0000 (0)

Index 0x70BA Touch probe position 1 positive value

Touch probe position 1 positive value.

Index	Name	Data type	Flags	Default
70BA	Touch probe position 1 positive value	INT32	ro	0

Index 0x70BB Touch probe position 1 negative value

Touch probe position 1 negative value.

Index	Name	Data type	Flags	Default
70BB	Touch probe position 1 negative value	INT32	ro	0

Index 0x70C2 Interpolation time period

Interpolation time period.

Index	Name	Data type	Flags	Default
70C2:0	Interpolation time period	-	-	-
70C2:01	Interpolation time period value	UINT8	rw	0x00 (0)
70C2:02	Interpolation time index	INT8	rw	0

Index 0x70D5 Touch probe 1 positive edge counter

Touch probe 1 positive edge counter.

Index	Name	Data type	Flags	Default
70D5	Touch probe 1 positive edge counter	UINT16	ro	0x0000 (0)

Index 0x70D6 Touch probe 1 negative edge counter

Touch probe 1 negative edge counter.

Index	Name	Data type	Flags	Default
70D6	Touch probe 1 negative edge counter	UINT16	ro	0x0000 (0)

Index 0x70E3 Supported homing methods

List of the homing methods supported by the drive.

Index	Name	Data type	Flags	Default
70E3:0	Supported homing methods	-	-	-
70E3:01	-	INT8	ro	19
70E3:02	-	INT8	ro	20
70E3:03	-	INT8	ro	21
70E3:04	-	INT8	ro	22
70E3:05	-	INT8	ro	37

Index 0x70E4 Additional position encoder value

Additional position encoder value.

Index	Name	Data type	Flags	Default
70E4:0	Additional position encoder value	-	-	-
70E4:01	First additional position	INT32	ro	---

Index 0x70EF Motor resolution

Motor resolution.

Index	Name	Data type	Flags	Default
70EF	Motor resolution	UINT32	ro	0x00000C80 (3200)

Index 0x70FD Digital inputs

Please refer to Index 0x60FD Digital Inputs.

Index	Name	Data type	Flags	Default
70FD	Digital inputs	UINT32	ro	0x00000000 (0)

Index 0x70FF Target velocity

Mandatory if Profile Velocity (PV) or Cyclic Synchronous Velocity (CSV) is supported.

Index	Name	Data type	Flags	Default
70FF	Target velocity	INT32	rw	0

Index 0x7502 Supported drive modes

Supported drive modes.

Index	Name	Data type	Flags	Default
7502	Supported drive modes	UINT32	ro	0x000001A1 (417)

6.2.3.4 Index 0xAnnx Device Object (0xA000 – 0xAFFF)

For Index 0xA000 – 0xAFFF, the device object of the EtherCAT SubDevice module.

Index 0xA000 FoE Transmission Status

FoE transmission status.

Index	Name	Data type	Flags	Default
A000	FoE_TransmissionStatus	UINT16	ro	0x0000 (0)

6.2.3.5 Index 0xFnnx Device Object (0xF000 – 0xFFFF)

For Index 0xF000 – 0xFFFF, the device object of the EtherCAT SubDevice module.

Index 0xF000 Modular Device Profile

Modular device profile.

Index	Name	Data type	Flags	Default
F000:0	Modular Device Profile	-	-	-
F000:01	Index distance	UINT16	ro	0x0010
F000:02	Maximum number of modules	UINT16	ro	0x00FF

Warranty

This product is warranted to be in good working order for a period of one year from the date of purchase. Should this product fail to be in good working order at any time during this period, we will, at our option, replace or repair it at no additional charge except as set forth in the following terms. This warranty does not apply to products damaged by misuse, modifications, accident or disaster. Vendor assumes no liability for any damages, lost profits, lost savings or any other incidental or consequential damage resulting from the use, misuse of, originality to use this product. Vendor will not be liable for any claim made by any other related party. Return authorization must be obtained from the vendor before returned merchandise will be accepted. Authorization can be obtained by calling or faxing the vendor and requesting a Return Merchandise Authorization (RMA) number. Returned goods should always be accompanied by a clear problem description.

All Trademarks appearing in this manuscript are registered trademark of their respective owners.
All Specifications are subject to change without notice.
©ICOP Technology Inc. 2026