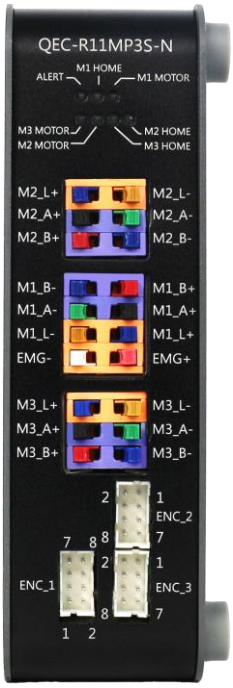


User Manual

QEC-RXXMP3S

EtherCAT Slave Stepper Motor Controller

Up to 3-axis stepper Motor control

(Revision 1.3)

REVISION

DATE	VERSION	DESCRIPTION
2023/11/02	**Version 1.0**	**New Release.**
2024/1/17	**Version 1.1**	**Update Product Data.**
2024/3/30	**Version 1.2**	**Update Homing Method.**
2024/8/14	**Version 1.3**	**Add index objects 0x50C1~0x50C6 to specify the workpiece coordinates' offset position in G-code.**

COPYRIGHT

Ver.1.3 August, 2024

TRADEMARKS ACKNOWLEDGMENT

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 +48VDC Power Input (Typ. +24VDC)
- 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.

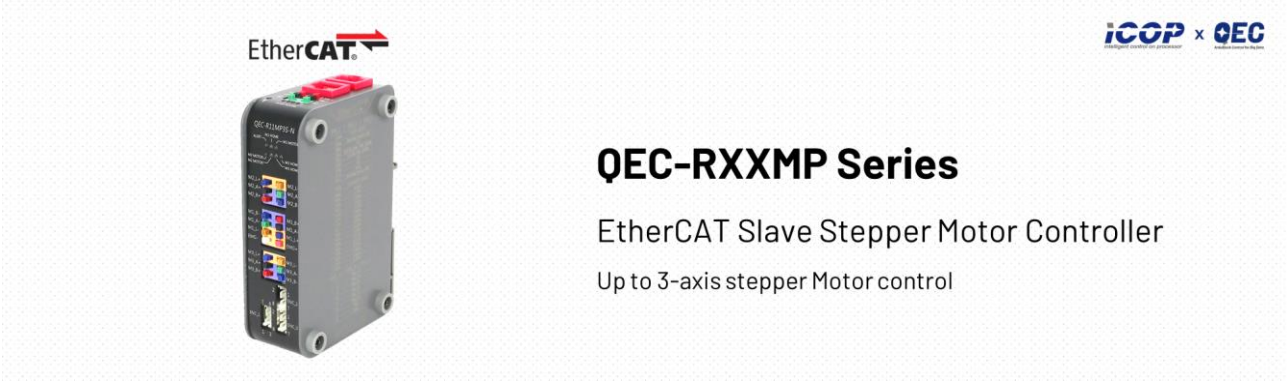
Content

Ch. 1

General Information

1.1 Introduction

The QEC-RXXMP series is an EtherCAT stepper motor open-loop controller capable of high-speed synchronization at 125μs while simultaneously driving three axes of two-phase bipolar stepper motors. Equipped with A, B, and Z encoder interfaces.



The QEC-RXXMP series has passed the verification of conformance testing tools and is suitable for various traditional industrial automation applications, such as management and precise motion control.

The QEC-RXXMP series is a standard EtherCAT slave that requires an EtherCAT master to operate the device. It supports three operating modes: Free Run, SyncManager2, and Distributed Clock (DC). It not only supports the CiA402 drive profile, a standard within EtherCAT, allowing seamless integration with a variety of control systems; But it also Equipped with G-code parser, a widely used programming language in CNC machine tools, a widely used programming language in CNC machine tools. This capability makes it adept in complex CNC machining and precise motion control domains.

This EtherCAT stepper motor controller can control three axes simultaneously with a high-speed 125μs synchronization. It provides a maximum of 4.84A drive current, suitable for demanding motors, and allows precise control adjustments using DIP switches. The controller also conserves energy and reduces wear by lowering current when the motor is idle, extending the motor's life. It simplifies wiring by using a single power source for all three axes. User can control motors with A, B, and Z encoder interfaces for accurate, interference resistant motor feedback. It supports up to 16 micro-steps for precise motor positioning and speed, ideal for high-precision tasks. Each motor has digital input channels for homing.

The QEC-RXXMP series has a compact size of 107.45 x 77.39 x 34 mm, 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

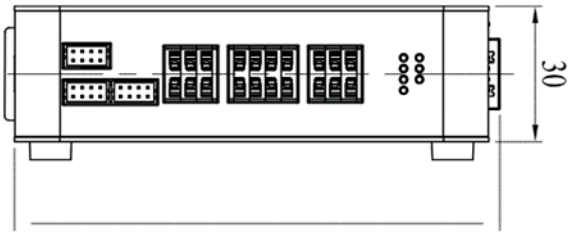
1.2.1 General Information

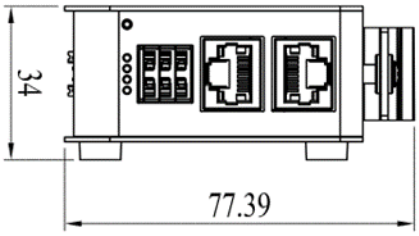
General	
Connector	Push-in Terminal (Euroblock)
Connector Color	EMG-: White, EMG+: Dark Red A+: Black, A-: Green, B+: Red, B-: Blue Limit+: Blue, Limit-: Brown
Protocol	EtherCAT (RJ-45 x2)
Ethernet Standard	IEEE 802.3
Transmission Rate	100Mbps
Power Connector	4-pin Power Input/Output & 2-pin FGND
Power Requirement	+19 to +50VDC Power Input (Typ. +24VDC@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 30mm (Without DIN-Rail)
Weight	370 g
Installation	DIN rail
Internal Monitoring	Temperature, Voltage, Current, Startup time

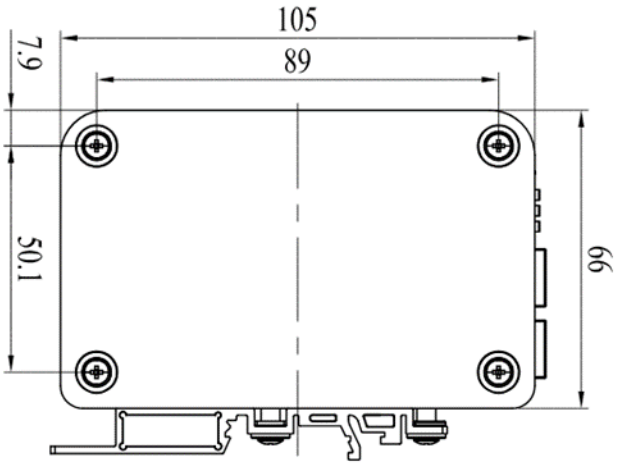
1.2.2 Stepper Driver Information

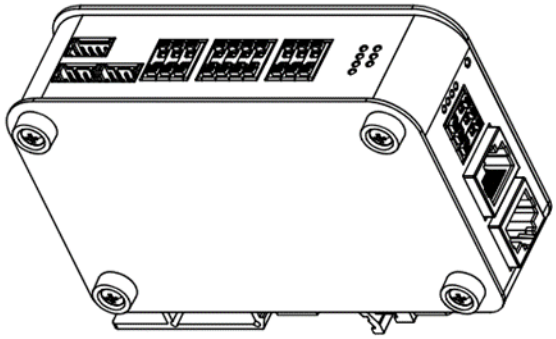
Stepper Motor	
Interface	EtherCAT
Drive Profile	CiA402, G-code
Minimum Communication Cycle	125 μs
Synchronization Mode	DC, SM2, FreeRun
Compatible Operation Mode	Profile Position (PP) Homing (HM: Support Method 19, 20, 21, 22) Cyclic Synchronous Position (CSP) Cyclic Synchronous Velocity (CSV)
Number of Motors	3 x Stepper Motors (2-phase bipolar stepper motor)
Output Current	Max. 4.84A, peak 5.0A
Voltage Requirement	+8 to +42VDC
Step frequency	200KHz
Microsteps	Max. 16 per step
Digital Input	
Digital Inputs	3 x home switch & Emergency Stop Input (+19 to +50VDC)
Isolation Voltage Protection	2500 Vrms
Encoder	
Encoder Inputs	3 x Encoder counter (A, B, Z), differential
Maximum Encoder pulse frequency	14 MHz
Encoder Power supply	5V
Encoder Type	AB Phase x4 AB Phase x2 CW/CCW x2 CW/CCW STEP/DIR x2 STEP/DIR

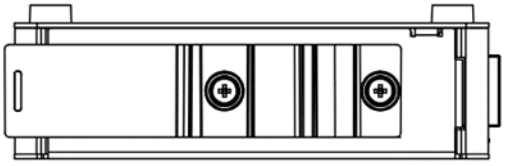
1.3 Dimension

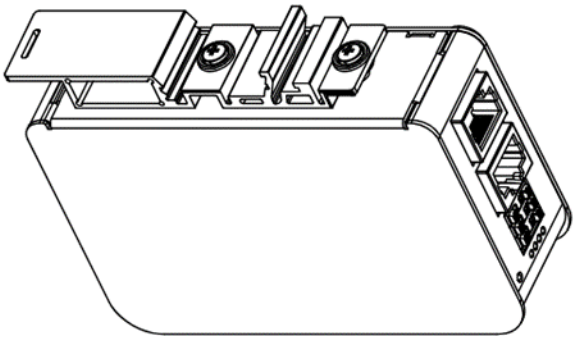










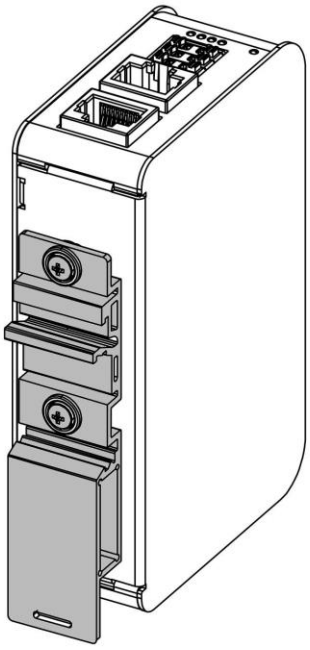


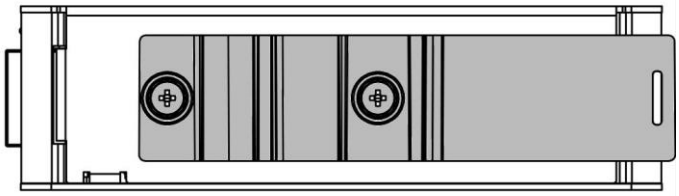
(Unit: mm)

1.4 Mounting Instruction

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

- **DIN-Rill**





1.5 Ordering Information

Type	RJ45 power source		Functions		Feature	–	Coating
	Input	Output	Stepper Motor	Functions	Standard		
QEC-R	X	X	MP	X	S		X

1. Type: Code 1~4

R: EtherCAT Slave

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

MP: Stepper Motor

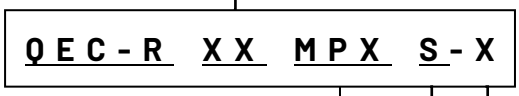
X: 1 or 3, different stepper motor axis

4. Feature: Code 10

S: Standard

5. Coating: Code 11

C: Yes / N: Normal



1.5.1 Ordering Part Number

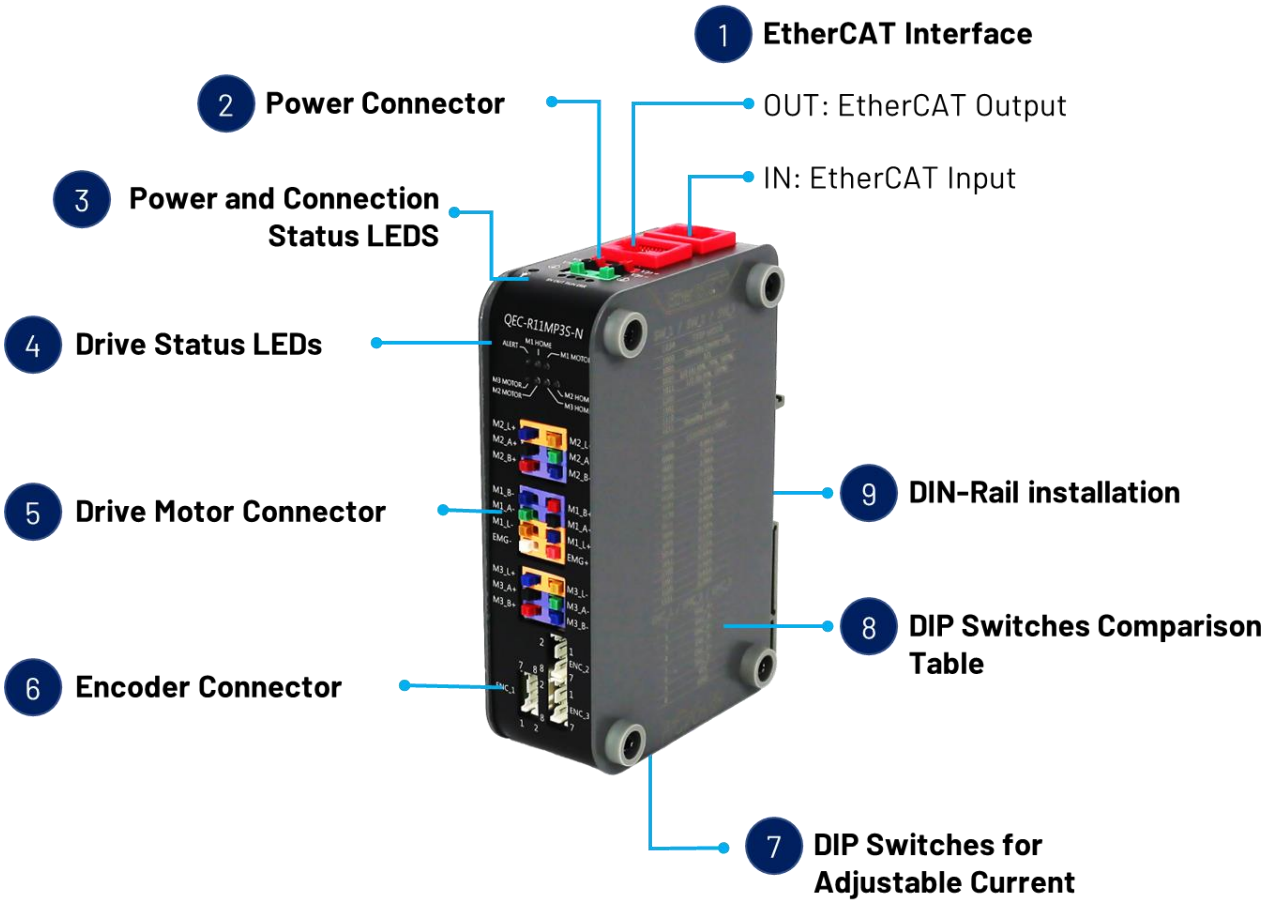
Above is the standard Part Number, please contact our sales if you need to order other part number.

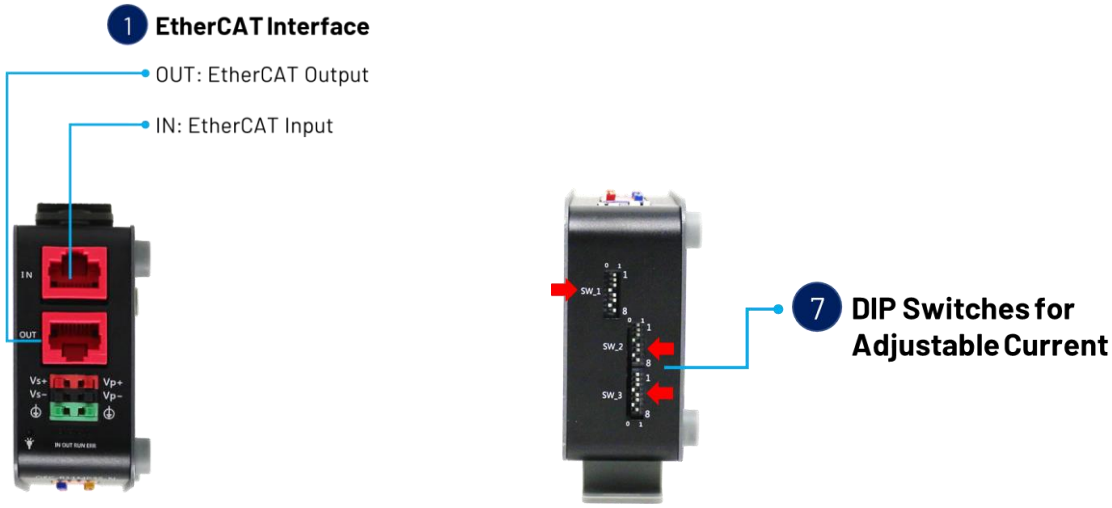
- **QEC-R00MP3S-N**: EtherCAT Slave 3 axis Stepper Motor Controller
- **QEC-R00MP3S-C**: EtherCAT Slave 3 axis Stepper Motor Controller (board with coating)
- **QEC-R11MP3S-N**: EtherCAT Slave 3 axis Stepper Motor Controller/PoE
- **QEC-R00MP1S-N**: EtherCAT Slave 1 axis Stepper Motor Controller
- **QEC-R11MP1S-N**: EtherCAT Slave 1 axis Stepper Motor Controller/PoE

Ch. 2

Hardware System

2.1 General Technical Data



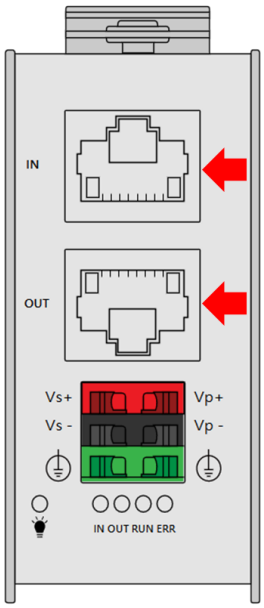


2.2 Connector Summary

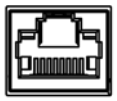
No.	Description		Type Narrative	Num #
1	EtherCAT Interface	OUT	RJ45 Connector	8-pin
		IN	(Gold finger)	8-pin
2	Power Connector		Power Socket	6-pin
3	Power and Connection Status LEDs		Status LEDs	-
4	Drive Status LEDs		Status LEDs	-
5	Drive Motor Connector		Push-in Terminal (Euroblock)	20-pin
6	Encoder Connector		-	3
7	DIP Switches for Adjustable Current		-	3
8	DIP Switches Comparison Table		-	-
9	DIN-Rail		-	-

2.2.1 EtherCAT Interface

RJ45 Connectors.



EC IN

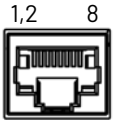


Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS-(GND)	8	VP-(GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.

* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

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)

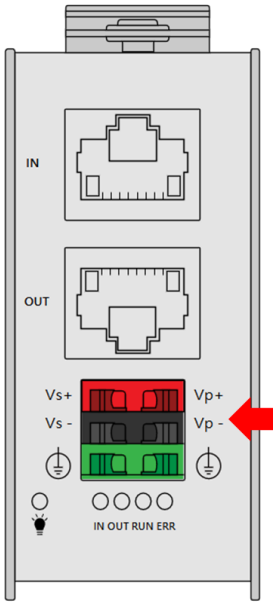
* PoE LAN with the Red Housing; Regular LAN with Black Housing.

* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

2.2.2 Power Connector

Euroblock Connectors.

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



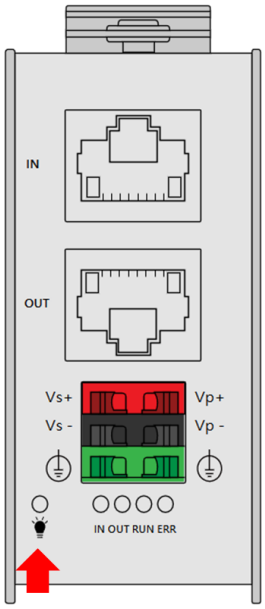
Vs for system power; Vp for peripheral power and backup power.

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

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

2.2.3 Power and Connection Status LEDs

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.

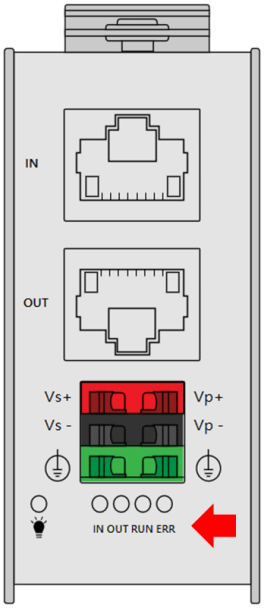
* Vs power status will be displayed first.

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 contact us.

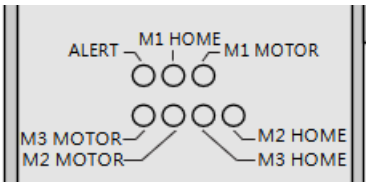
Connection Status LEDs



EtherCAT: PWR, RUN, LINK, and ERROR Status LEDs

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



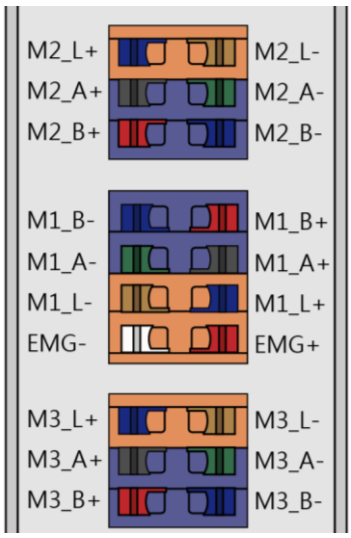
Alarm, Home, and Motor Status LEDs.

Notation	Color	Description
ALERT	Green	Alert signal
M1 HOME	Green	X-Limit (X-Home) signal
M1 MOTOR	Green	X-axis Motor action signal
M2 HOME	Green	Y-Limit (Y-Home) signal
M2 MOTOR	Green	Y-axis Motor action signal
M3 HOME	Green	Z-Limit (Z-Home) signal
M3 MOTOR	Green	Z-axis Motor action signal

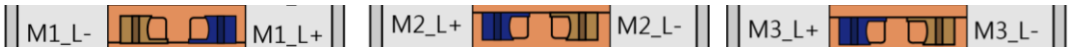
2.2.5 Drive Motor Connector

Euroblock Connectors.
3 x Stepper Motors (2-phase bipolar stepper motor) and 3 x home switch & Emergency Stop Input.

Signal Name		Signal Name
M2_L+		M2_L-
M2_A+		M2_A-
M2_B+		M2_B-
M1_B-		M1_B+
M1_A-		M1_A+
M1_L-		M1_L+
EMG-		EMG+
M3_L+		M3_L-
M3_A+		M3_A-
M3_B+		M3_B-

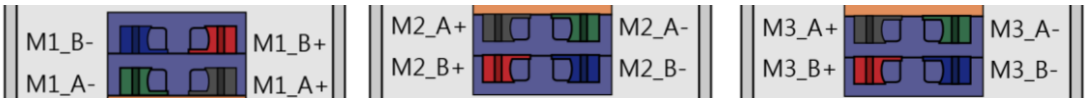


Limit Switches Description:



Name	Connector Color	Signal	Signal Description
M2_L+	Blue	+19 to +50VDC	Limit+ switch for Motor 2
M2_L-	Brown	GND	Limit- switch for Motor 2
M1_L-	Brown	GND	Limit- switch for Motor 1
M1_L+	Blue	+19 to +50VDC	Limit+ switch for Motor 1
M3_L+	Blue	+19 to +50VDC	Limit+ switch for Motor 3
M3_L-	Brown	GND	Limit- switch for Motor 3

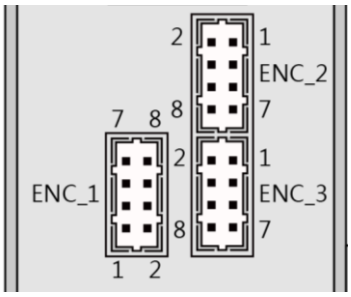
Drive Motor Pins and Digital Input Description:



Name	Connector Color	Signal	Signal Description	Default
M2_A+	Black	Input	Motor 2 winding A+	Y-Axis
M2_A-	Green	Input	Motor 2 winding A-	
M2_B+	Red	Input	Motor 2 winding B+	
M2_B-	Blue	Input	Motor 2 winding B-	
M1_B-	Blue	Input	Motor 1 winding B-	X-Axis
M1_B+	Red	Input	Motor 1 winding B+	
M1_A-	Green	Input	Motor 1 winding A-	
M1_A+	Black	Input	Motor 1 winding A+	
M3_A+	Black	Input	Motor 3 winding A+	Z-Axis
M3_A-	Green	Input	Motor 3 winding A-	
M3_B+	Red	Input	Motor 3 winding B+	
M3_B-	Blue	Input	Motor 3 winding B-	

2.2.6 Encoder Connector

TU2005 Connector Header.



3 x Encoder counter (A, B, Z), differential.

Encoder Trace width: 15mil.

Name	Signal	Signal Description	Default
ENC_1	Input	Encoder 1	X-Axis
ENC_2	Input	Encoder 2	Y-Axis
ENC_3	Input	Encoder 3	Z-Axis

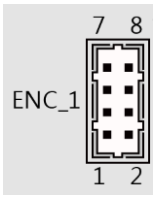
Encoder Pin assignment (ENC_1/ENC_2/ENC_3):

Pin#	Description
1	ENC_A+
2	ENC_A-
3	ENC_B+
4	ENC_B-
5	ENC_C+
6	ENC_C-
7	VCC
8	GND

Users can see in the side case of QEC-RXXMP3S.

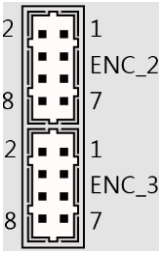
ENC_1:

Description	Pin #		Pin #	Description
X_ENCV+	7		8	X_ENCV-
ENCX-Z+	5		6	ENCX-Z-
ENCX-B+	3		4	ENCX-B-
ENCX-A+	1		2	ENCX-A-



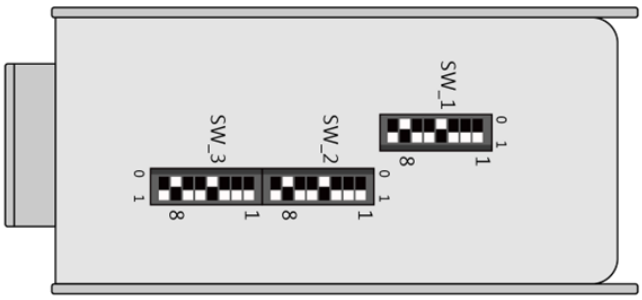
ENC_2 & ENC_3:

Description	Pin #		Pin #	Description
ENCY-A-	2		1	ENCY-A+
ENCY-B-	4		3	ENCY-B+
ENCY-Z-	6		5	ENCY-Z+
Y_ENCV-	8		7	Y_ENCV+
ENCZ-A-	2		1	ENCZ-A+
ENCZ-B-	4		3	ENCZ-B+
ENCZ-Z-	6		5	ENCZ-Z+
Z_ENCV-	8		7	Z_ENCV+



2.2.7 DIP Switches for Adjustable Current

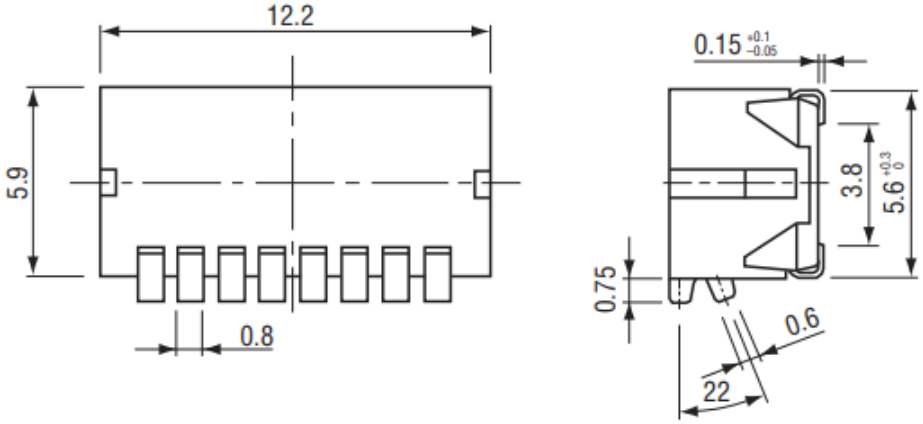
Piano switches (SW_1: CHP-080A; SW_2 and SW_3: CHP-081A).



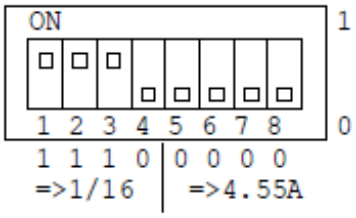
Switch Specifications:

Operating Temperature	-40 to +85°C
Humidity	–10 ~ 65 °C, Relative humidity 0 ~ 96 %
Contact resistance	50 mΩ maximum
Dimension	12.2 x 5.9 x 4.6 mm

Dimension:



Applicable current for 3 types of motors. Users can see in the side case of QEC-RXXMP3S.

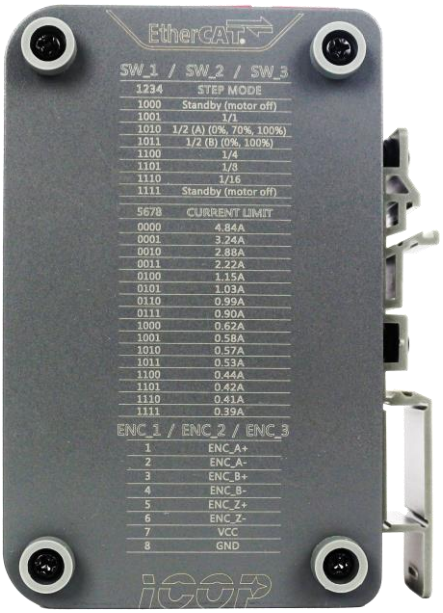


1234	STEP MODE
1000	Standby (motor off)
1001	1/1
1010	1/2 (A)(0%, 70%, 100%)
1011	7/2 (8)(0%, 100%)
1100	1/4
1101	1/8
1110	1/16
1111	Standby (motor off)

5678	Vref	Limit Current	42"	57"	86"
0000	1.59V	4.84A			
0001	1.07V	3.24A			
0010	0.95V	2.88A			V
0011	0.73V	2.22A			
0100	0.38V	1.15A			
0101	0.34V	1.03A			
0110	0.32V	0.99A		V	
0111	0.29V	0.90A			
1000	0.20V	0.62A			
1001	0.19V	0.58A			
1010	0.18V	0.57A			
1011	0.17V	0.53A			
1100	0.14V	0.44A			
1101	0.139V	0.42A	V		
1110	0.136V	0.41A			
1111	0.131V	0.39A			

* IO=(VREF*1/3)/RNf

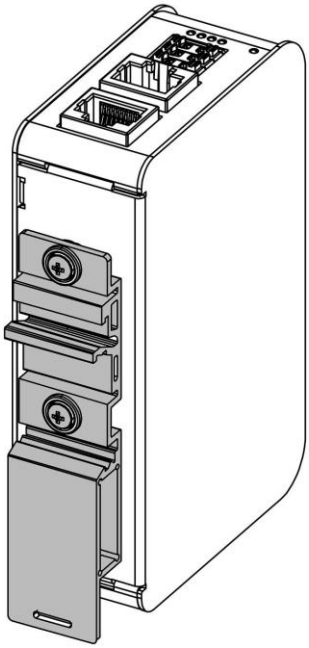
2.2.8 DIP Switches Comparison Table

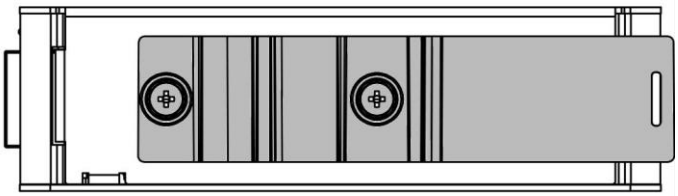


This table is for Ch.2.2.6 Encoder Connector and Ch.2.2.7 DIP Switches for Adjustable Current.

2.2.9 DIN-Rail installation

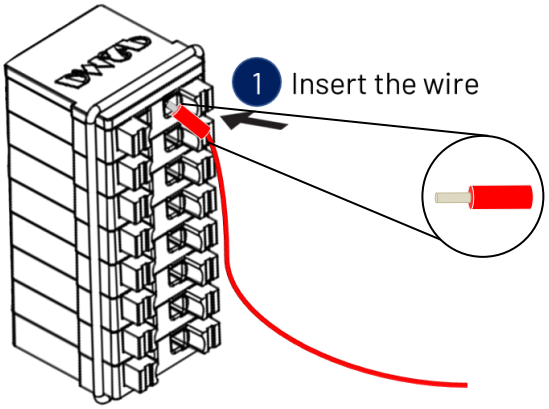
Please refer to Ch.3.1 DIN-Rail installation.

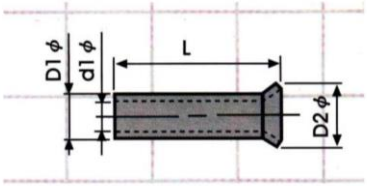




2.3 Wiring to the Connector

2.3.1 Connecting the wire to the connector

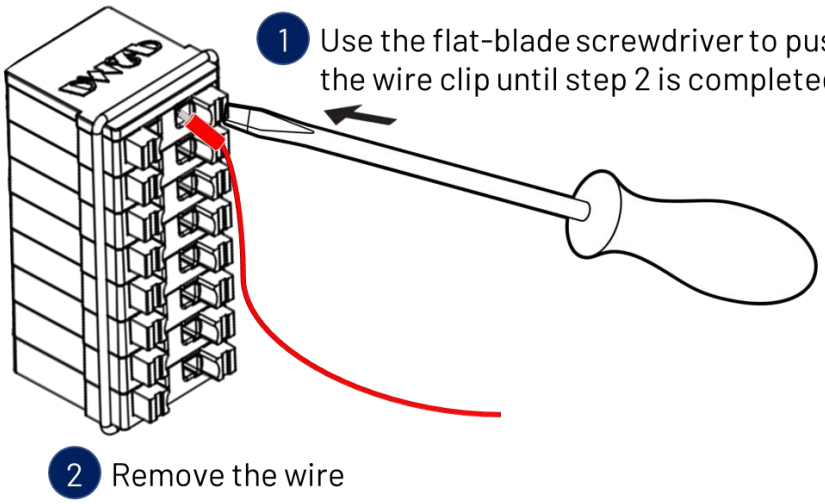




Insulated Terminals Dimensions (mm)

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

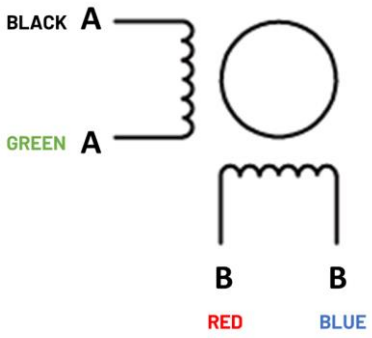
2.3.2 Removing the wire from the connector



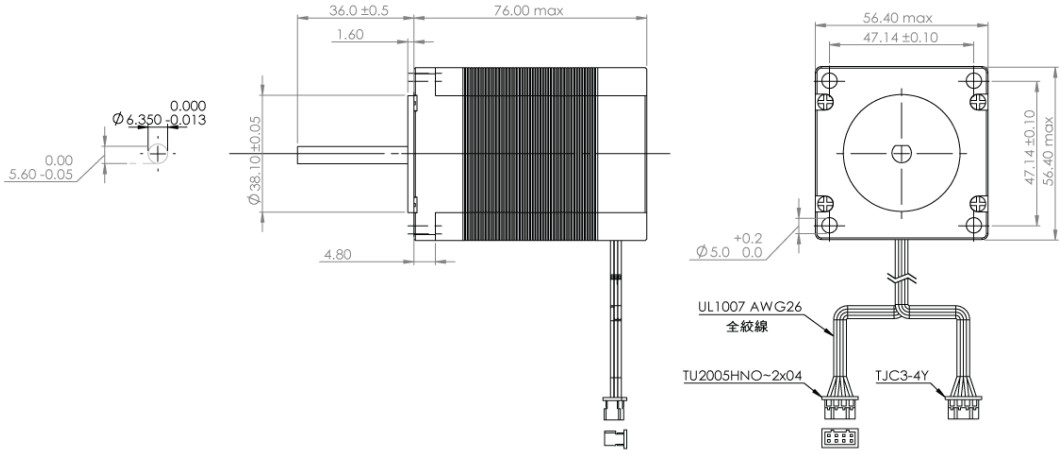
2.3.3 Stepper Motor and Encoder Wiring

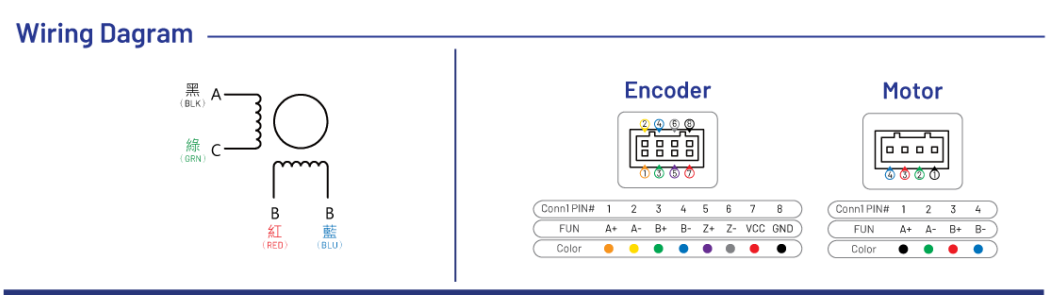
Four Lead Motor

We use 86STEP-577609 in this example, which is a four-lead, two-phase 57 stepper motor with encoder.



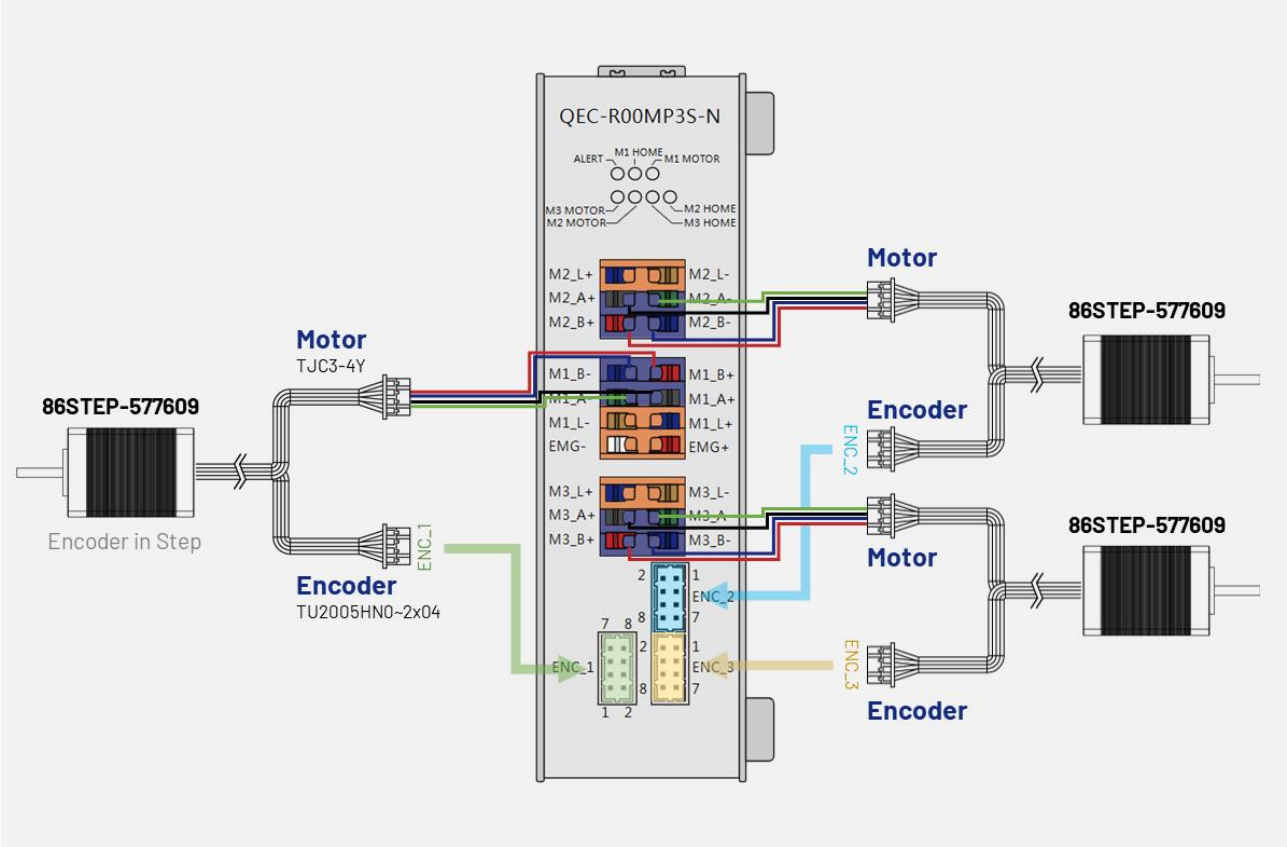
86STEP can detect lost steps in cost-effective stepper motors. The optical encoder is embedded in the design to directly monitor the motor rotor position. Through the encoder feedback position, closed-loop control is achieved, while retaining the advantages of the stepper motor and preventing out-of-step problems. For application devices that require position verification and position maintenance, it effectively improves safety and reliability.





For information about the 86STEP-577609 stepper motor, please refer to: [86STEP | 86Duino](#).

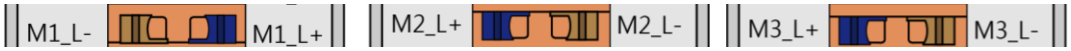
The figure below shows an example of three four-lead, two-phase motors (with encoder) connected to the QEC-RXXMP3S product.



The encoder in the QEC-RXXMP3S 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). The encoder also handles AB phase signals, which helps in precise motor positioning and speed control.

2.3.4 Limit Switches

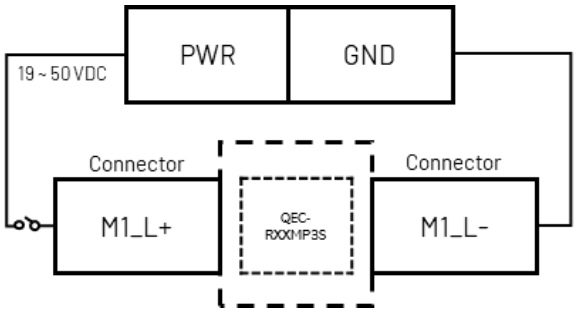
Limit Switches:

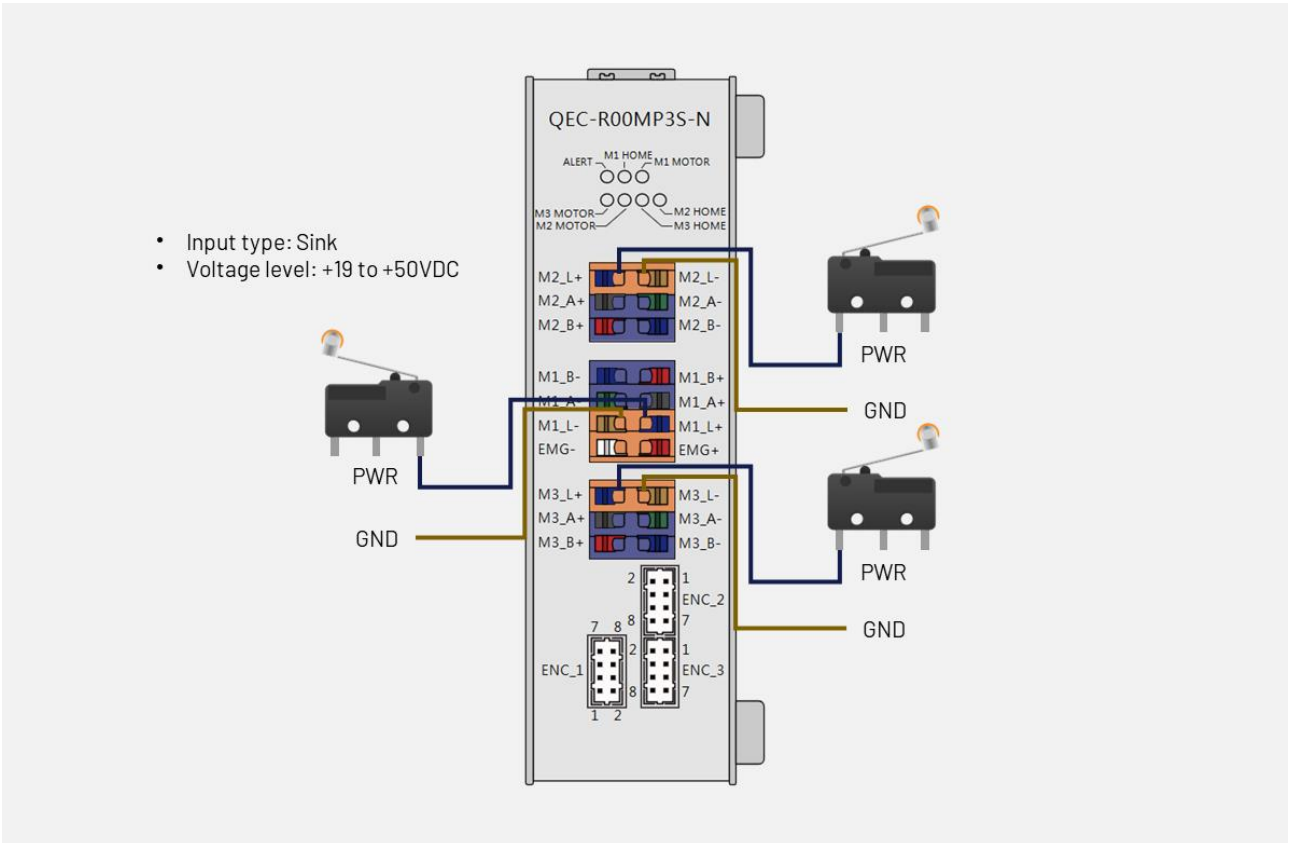


3 x home switch.

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

Application wiring:



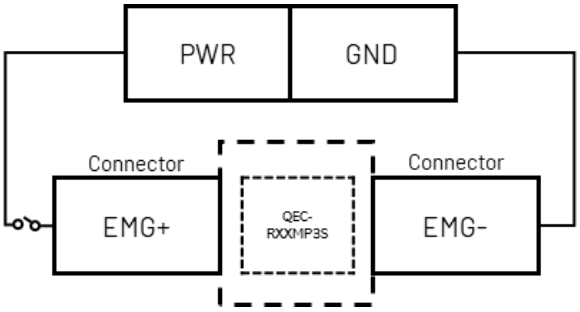


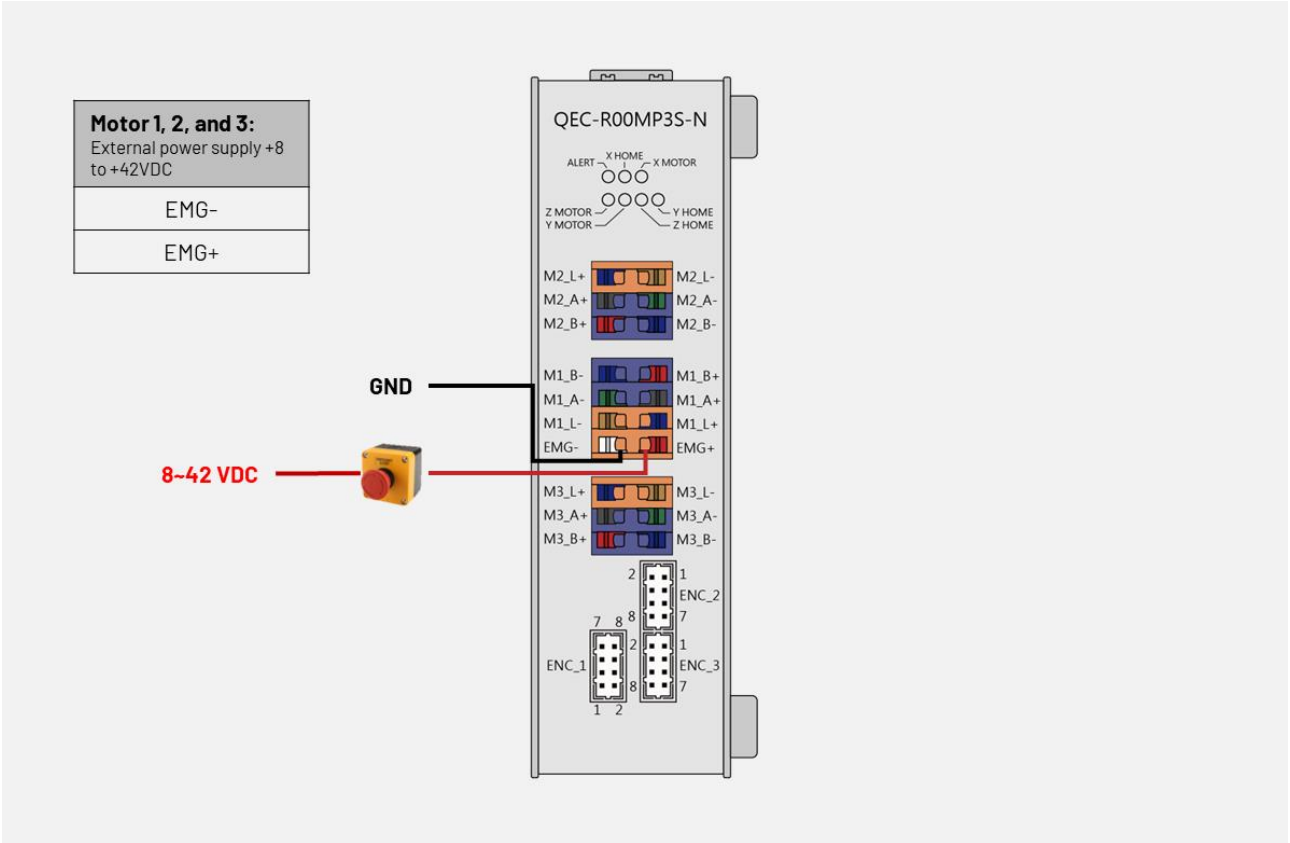
2.3.5 Emergency Stop

Drive Motor power supply Description:

Name	Connector Color	Signal	Signal Description
EMG-	White	GND	Emergency stop (E-stop) is a safety mechanism used to shut off machinery in an emergency, when it need to shut down simply and quickly via hardware switch.
EMG+	Dark Red	+8 to +42VDC	

Application Wiring:



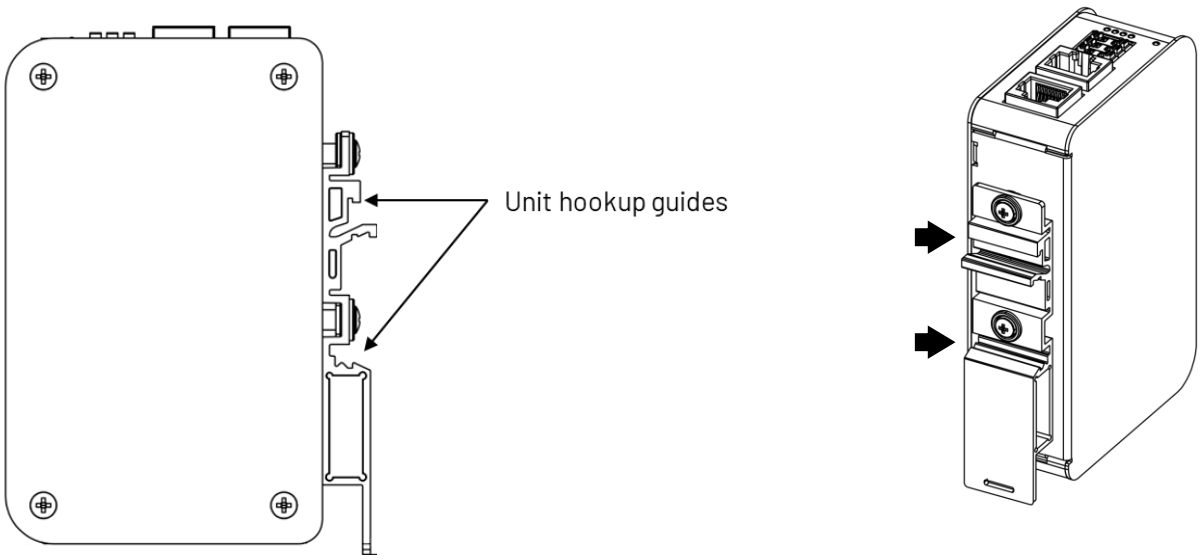


Ch. 3

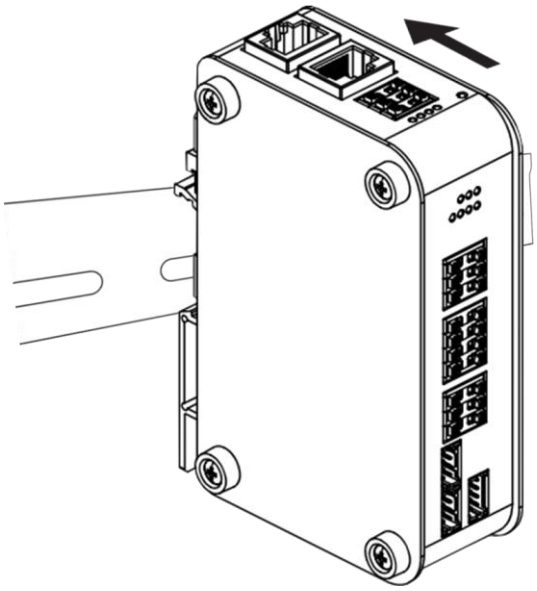
Hardware Installation

3.1 DIN-Rail installation

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



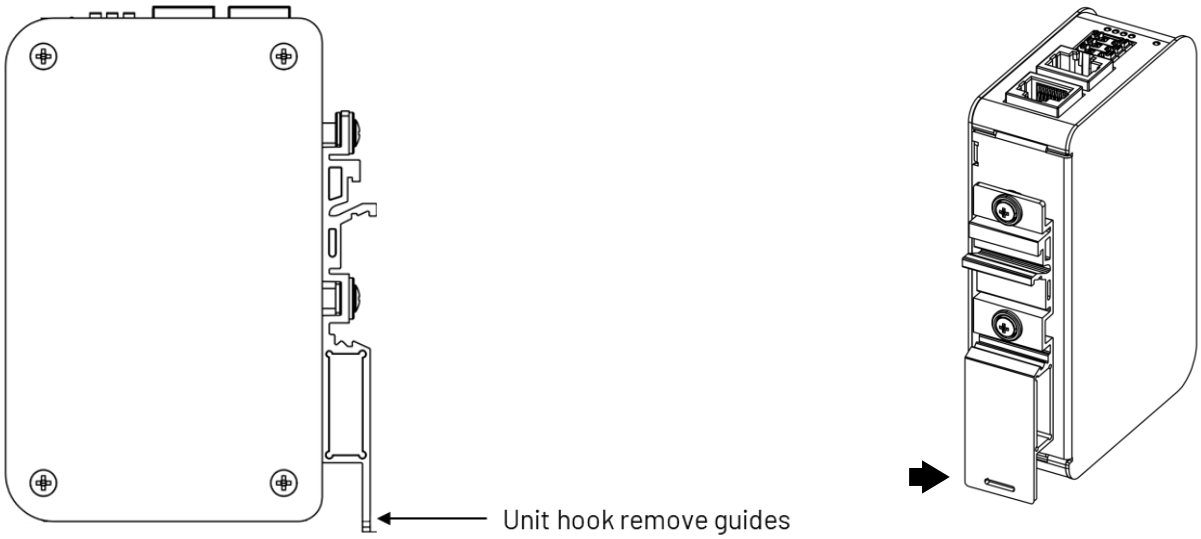
When you mount the QEC-RXXMP, releasing the DIN track mounting hook on the QEC-RXXMP is unnecessary. After you mount the QEC-RXXMP, make sure it is locked to the DIN Track.



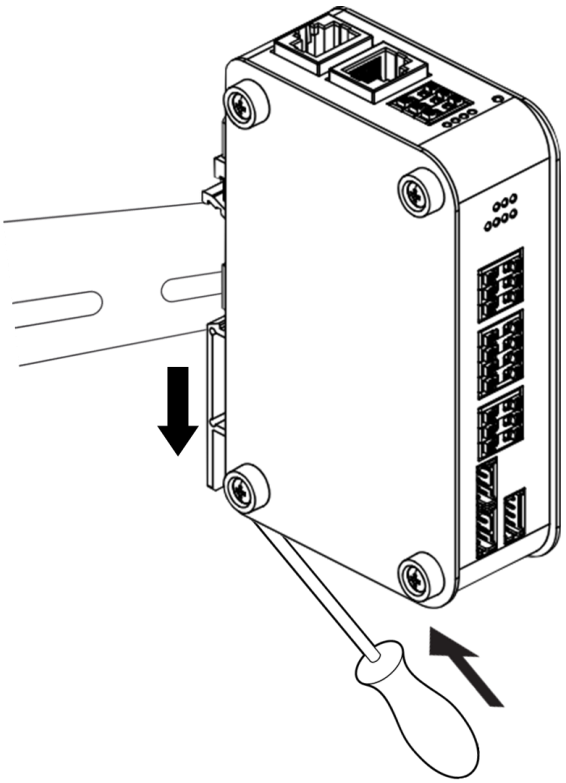
* Note: Always turn OFF the Unit power supply and I/O power supply before connecting and removing the QEC-RXXMP.

3.2 Removing QEC-RXXMP 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.

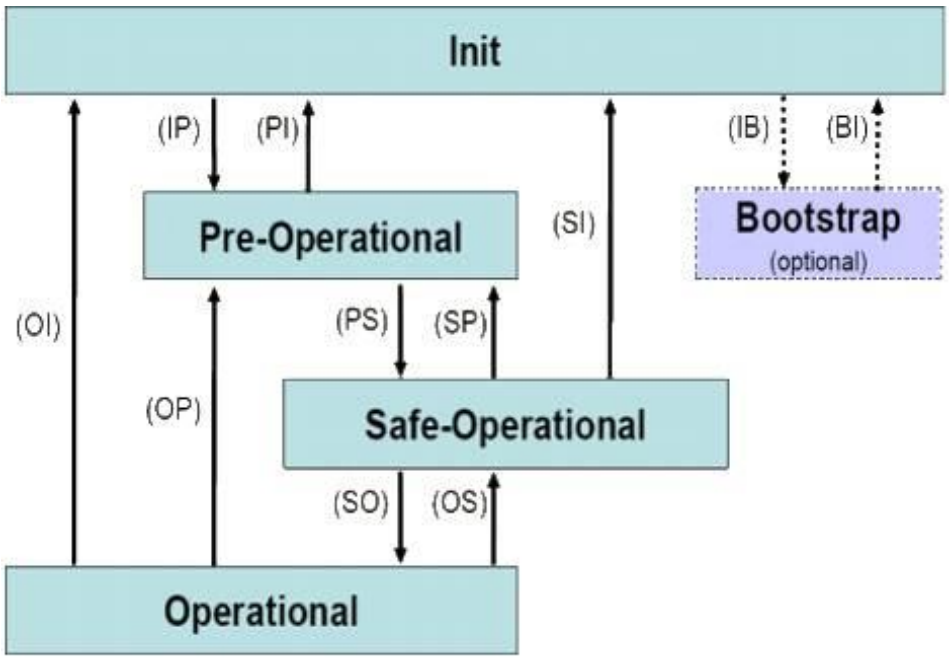
Cables and connectors

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).

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



A distinction is made between the following states:

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

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

Init

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

Pre-Operational (Pre-Op)

During the transition between Init and Pre-Op the EtherCAT slave checks whether the mailbox was initialized correctly. In Pre-Op state mailbox communication is possible, but not process data communication. The EtherCAT master initializes the sync manager channels for process data (from sync manager channel 2), the FMMU channels and, if the slave 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 slave 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 slave copies current input data into the associated DP-RAM areas of the EtherCAT slave controller (ECSC). In Safe-Op state mailbox and process data communication is possible, although the slave 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 master switches the EtherCAT slave from Safe-Op to Op it must transfer valid output data. In the Op state the slave copies the output data of the masters to its outputs. Process data and mailbox communication is possible.

Boot

In the Boot state the slave 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 Master 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 master.

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 ECAT Slave performs PDO transmission, the actions performed after receiving a network packet can be divided into three parts:

1. PDO_OutputMapping
2. ECAT_Application
3. 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 MP3S (CiA-402 mode) as an example, MP3S will start to control the 3-axis stepper 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 Master is to be set at 1A00, 1A10, and 1A20 Objects, and RxPDO Mapping information to receive a command from the Master is to be set at 1600, 1610, and 1620 Objects.

RxPDO Mapping Table:

PDO Map Object		Object Contents		
Index	**Sub**	**Object**	**Sub**	**DataType**
0x1600	X Axis RxPdoMapping0			
0x1600	1	0x6040	0x00	UINT16
0x1600	2	0x607A	0x00	INT32
0x1600	3	0x60FF	0x00	INT32
0x1600	4	0x6060	0x00	INT8
0x1610	Y Axis RxPdoMapping0			
0x1610	1	0x6840	0x00	UINT16
0x1610	2	0x687A	0x00	INT32
0x1610	3	0x68FF	0x00	INT32
0x1610	4	0x6860	0x00	INT8
0x1620	Z 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**	**DataType**	**Name**
0x1A00	X 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
0x1A00	7	0x5024	0x03	-	ENC Status
0x1A10	Y 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
0x1A10	7	0x5024	0x03	-	ENC Status
0x1A20	Z 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
0x1A20	7	0x5024	0x03	-	ENC Status

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:

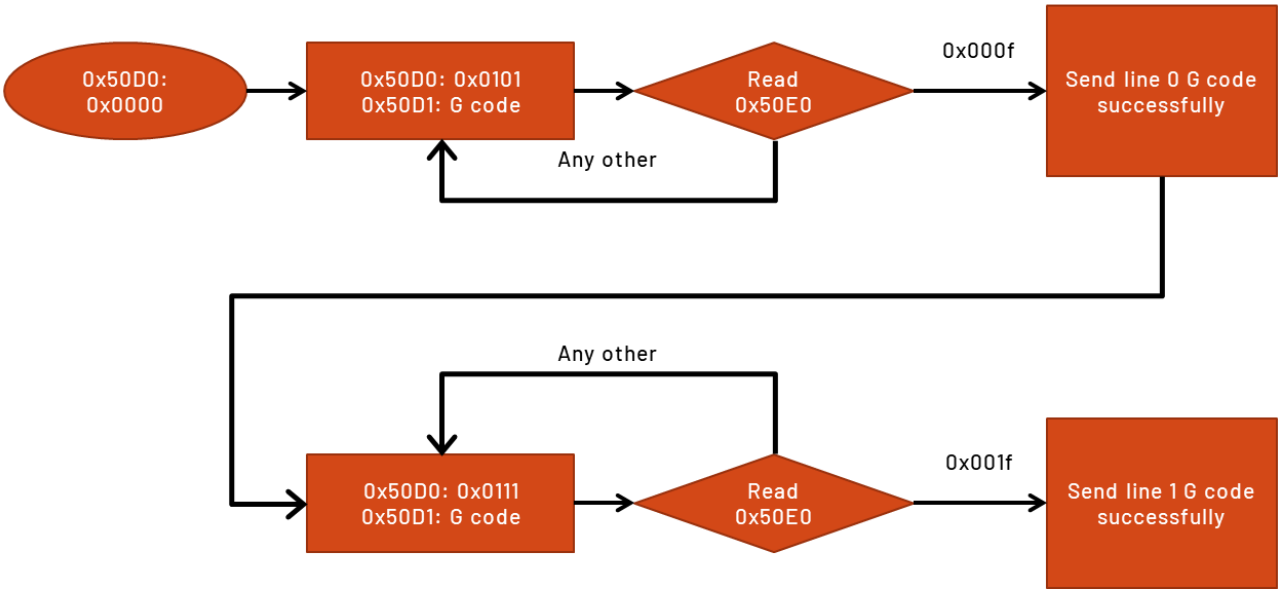
Object	Name
0x1600	Rx PDO Map0
0x1610	Rx PDO Map1
0x1620	Rx PDO Map2
...	
0x1A00	Tx PDO Map0
0x1A10	Tx PDO Map1
0x1A20	Tx PDO Map2

- 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.

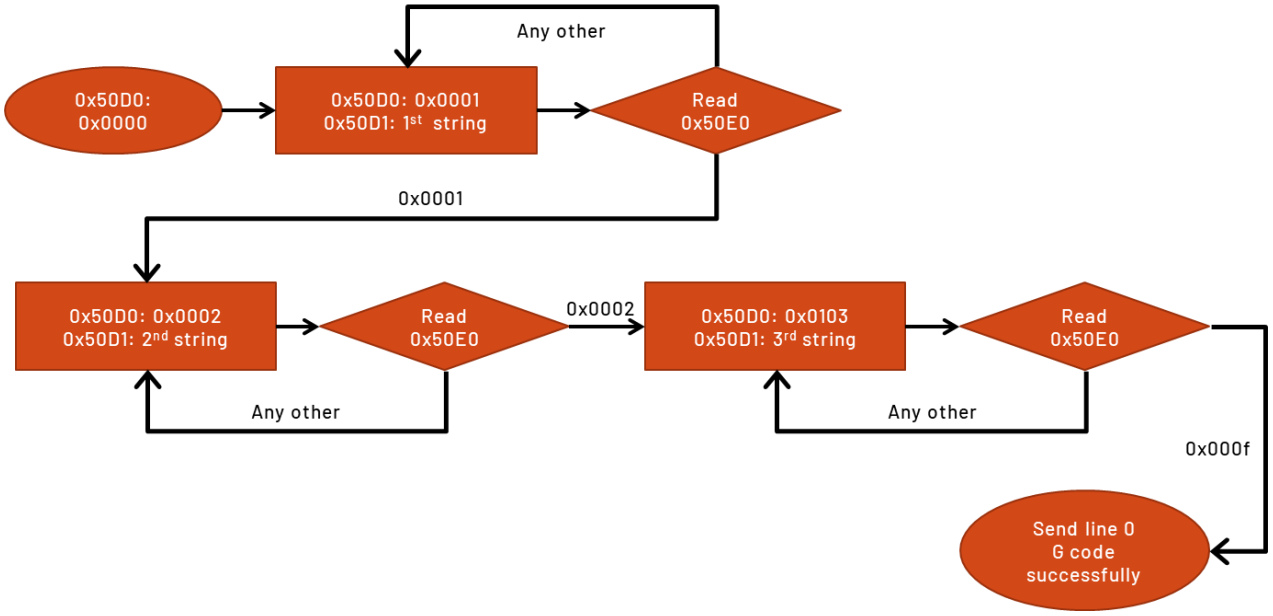
4.4.3 PDO Operation Process

For G-code Mode Operation Process:

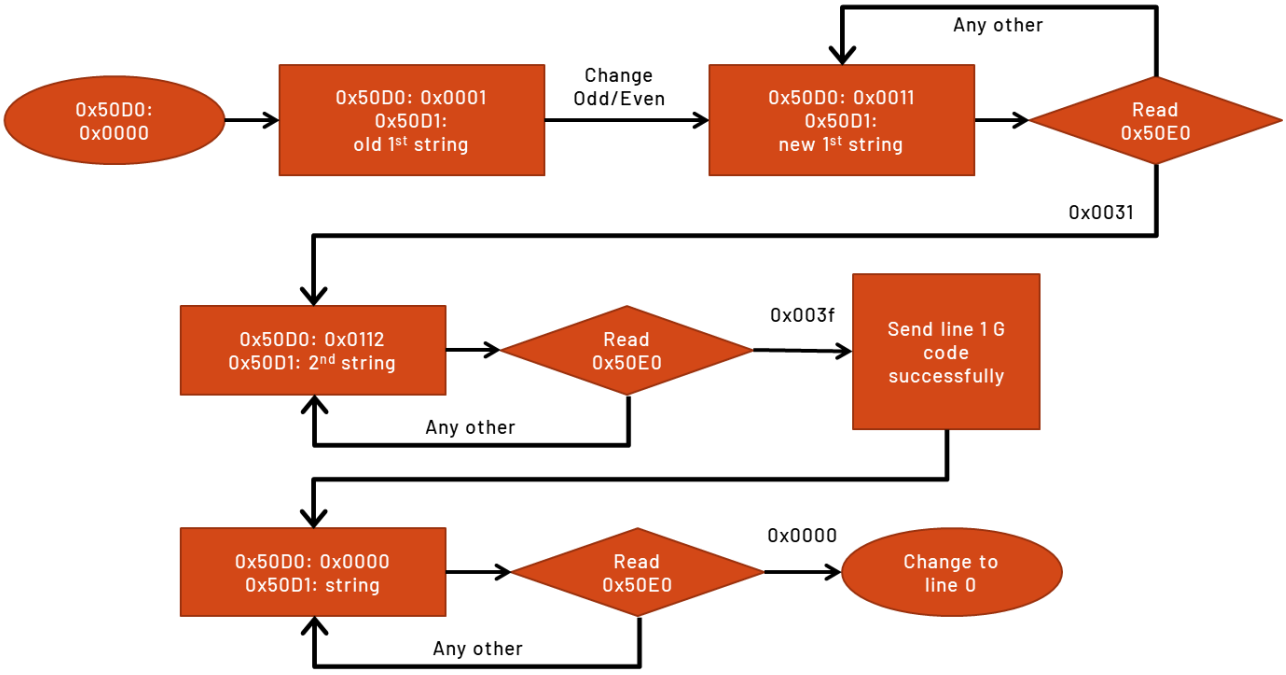
Single PDO transmission 2 lines of G code:



3 times PDO transmits 1 line of G code:



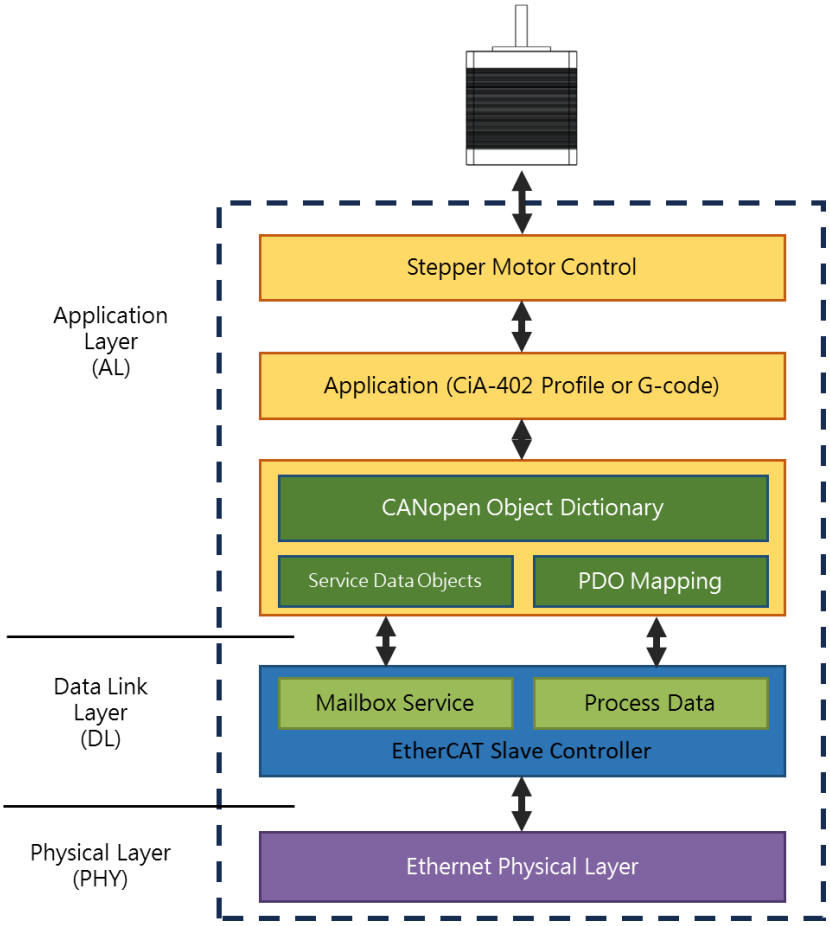
Line number reset:



4.5 CAN application protocol over EtherCAT

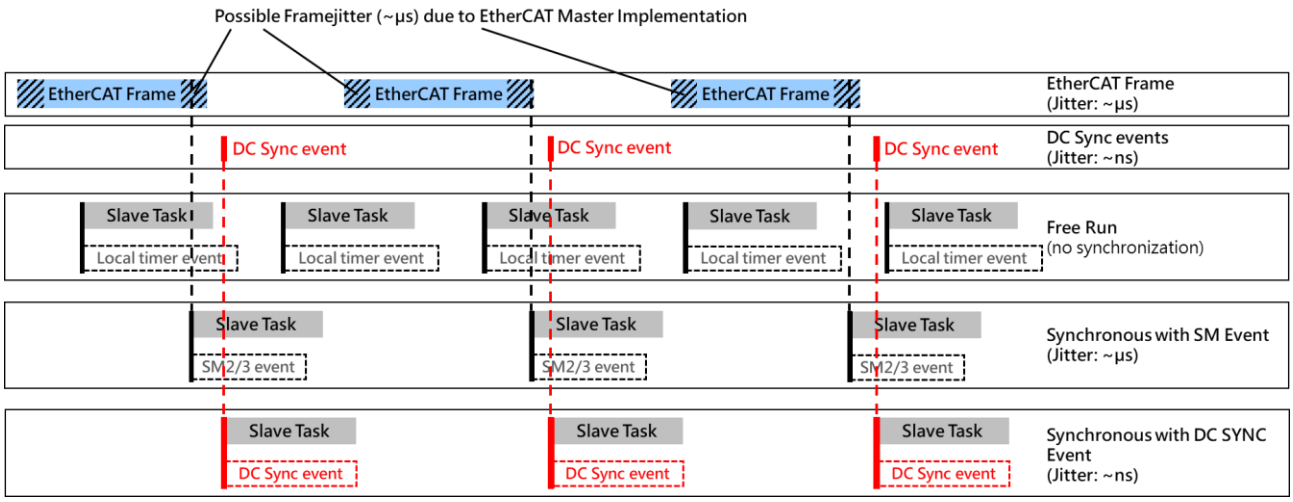
The CoE interface (CAN application protocol over EtherCAT) is used for parameter management of EtherCAT devices. EtherCAT slaves or the EtherCAT master manage fixed (read-only) or variable parameters required for operation, diagnostics, or commissioning. CoE parameters are arranged in a table hierarchy.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

QEC-RXXMP3S supports CAN application protocol over EtherCAT (CoE). EtherCAT Slave structure is as follows.



4.6 Synchronization Modes

Synchronization modes provided by QEC-RXXMP3S are as follows.



4.6.1 Free Run

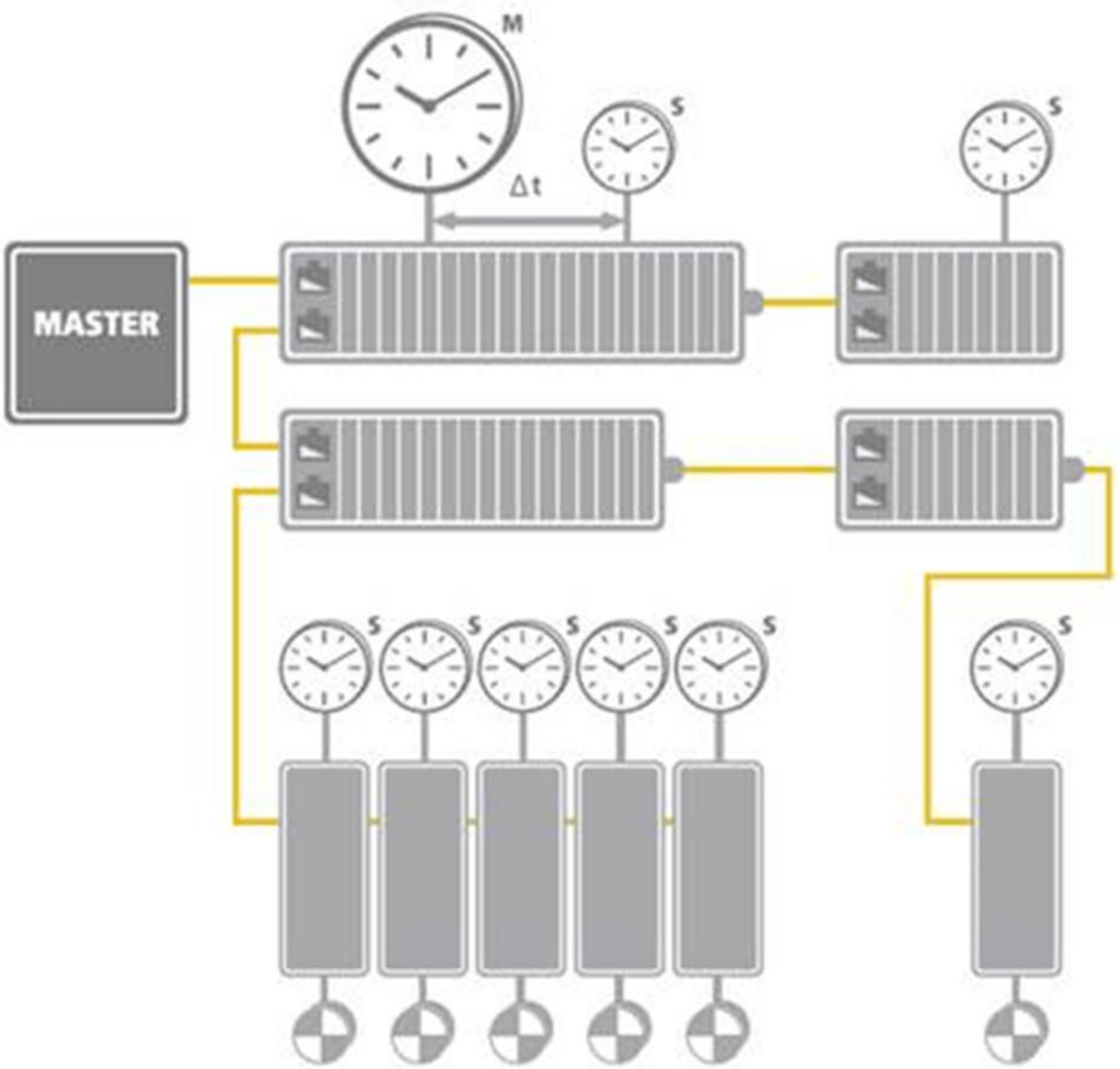
In this mode, the EtherCAT slave device operates independently of the EtherCAT master's timing. The slave does not synchronize its operations with the master's clock or any other synchronization signals in the EtherCAT network. This mode is typically used in applications where precise timing alignment 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 slave device 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 slave device aligns its tasks, such as data acquisition or actuator control, with the specific timing of the SM2 events. This mode ensures that the slave'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 slave device with the distributed clock system of the EtherCAT network. In this mode, the slave 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.



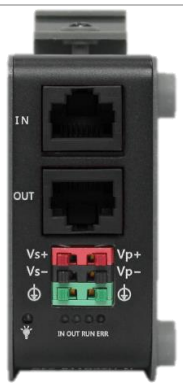
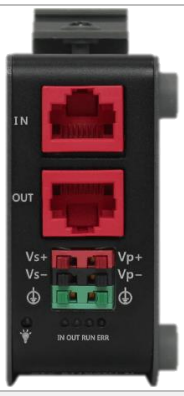
EtherCAT: Illustration of Distributed Clock (DC). (Source of information: http://www.ethercat.org/)

Ch. 5

Getting Started

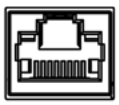
5.0 Notes: QEC's PoE (Power over Ethernet)

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:

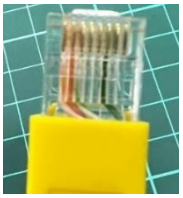


Pin #	Signal Name	Pin #	Signal Name
1	LAN1_TX+	2	LAN1_TX-
3	LAN1_RX+	4	VS+
5	VP+	6	LAN1_RX-
7	VS-(GND)	8	VP-(GND)

* PoE LAN with the Red Housing; Regular LAN with Black Housing.

* L4, L5, L7, L8 pins are option, for RJ45 Power IN/OUT.

2. 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 master connects with a third-party EtherCAT slave).



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

5.1 Introduction

Welcome to the Quick Start Guide for the QEC-RXXMP3S EtherCAT Stepper Motor Driver. This section is designed to assist you in efficiently setting up and utilizing your new stepper motor driver. To facilitate this process, we will focus on two essential software tools: TwinCAT, which operates on a PC, and the 86Duino IDE, which is compatible with the QEC Master Series.

In the following pages, we will walk you through the steps for connecting your QEC-RXXMP3S and initiating your journey toward fully integrating this stepper motor driver into your projects. Our goal is to simplify the initial setup so you can quickly begin exploring the extensive capabilities of your device.

5.2 TwinCAT (PP Mode)

If you're ready to get your QEC-RXXMP3S EtherCAT Stepper Motor Driver up and running with TwinCAT, this section is for you. We're focusing on using the CiA402 Drive Profile in Profile Position (PP) Mode, perfect for when you need your motor to hit precise positions accurately.

5.2.1 Install the ESI file

1. Obtain the ESI file:
   The ESI files for QEC-RXXMP3S device are located on our website at the following address: https://www.qec.tw/product/qec-rxxmp3s/.
   Simply download the ZIP file and extract it.

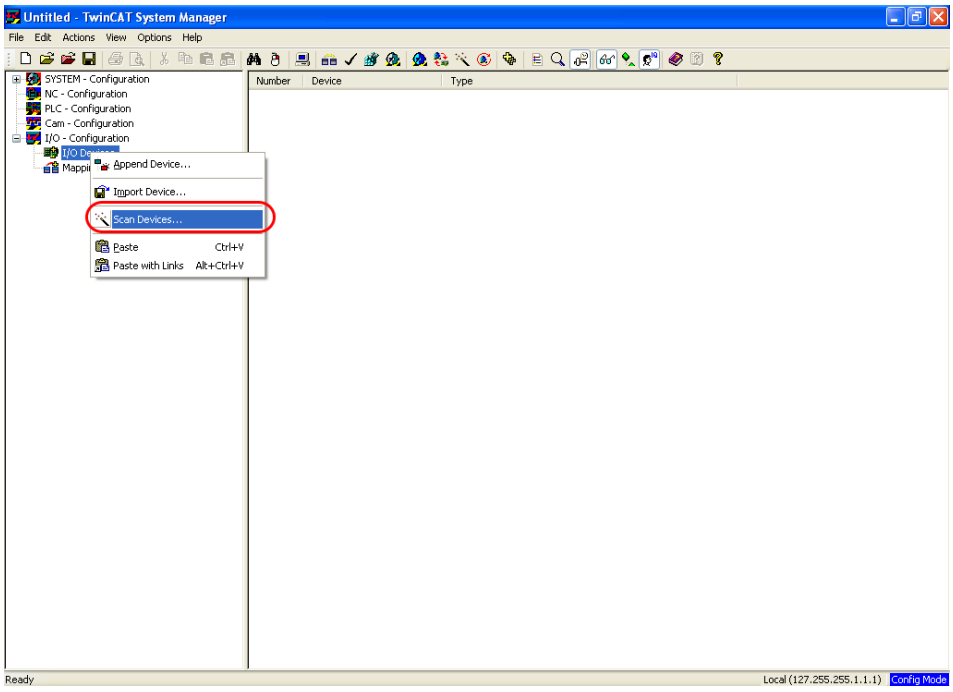
2. Install the ESI file:
   Once extracted, the xml file must be copied or moved to the appropriate system directory.

3. Restart the Programming System If Needed:
   If the TwinCAT program was running when the ESI file was copied to the appropriate system directory, you may have to restart the TwinCAT program before it will recognize the new ESI file.

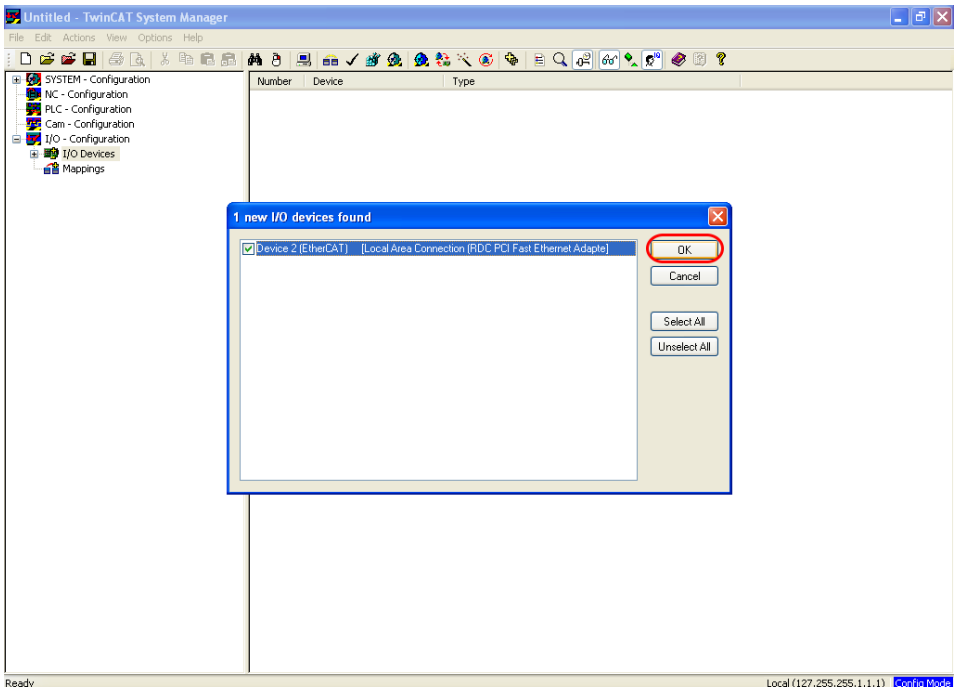
5.2.2 Add the QEC-R11MP3S to the Project

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.)

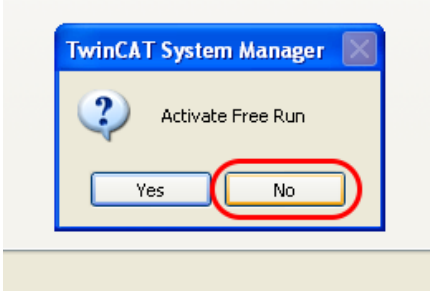


2. Choose the EtheCAT connection network.

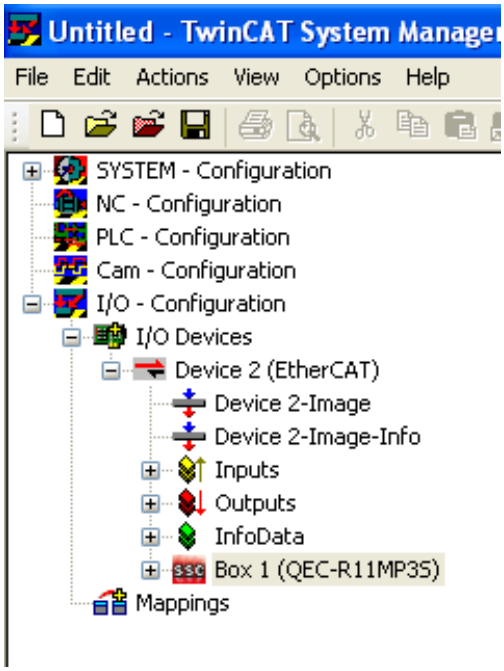


3. Confirm “Yes” to start the scan.

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

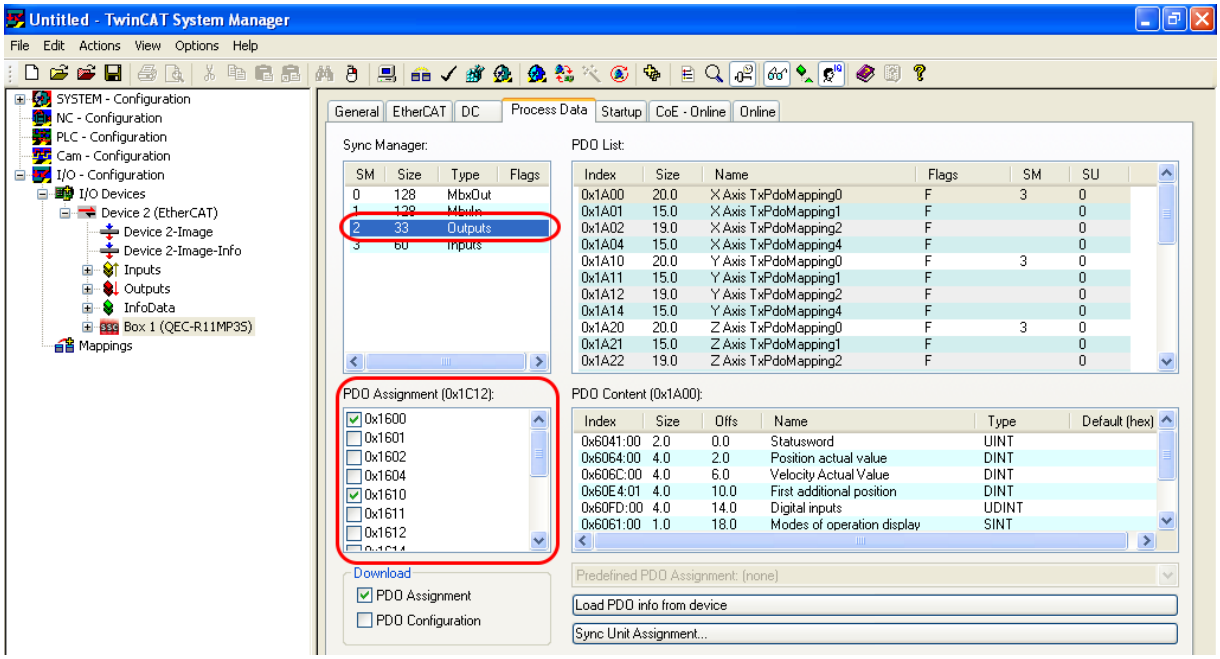


5. The QEC-R11MP3S will appear in the device tree and the name will typically begin with “Box”.

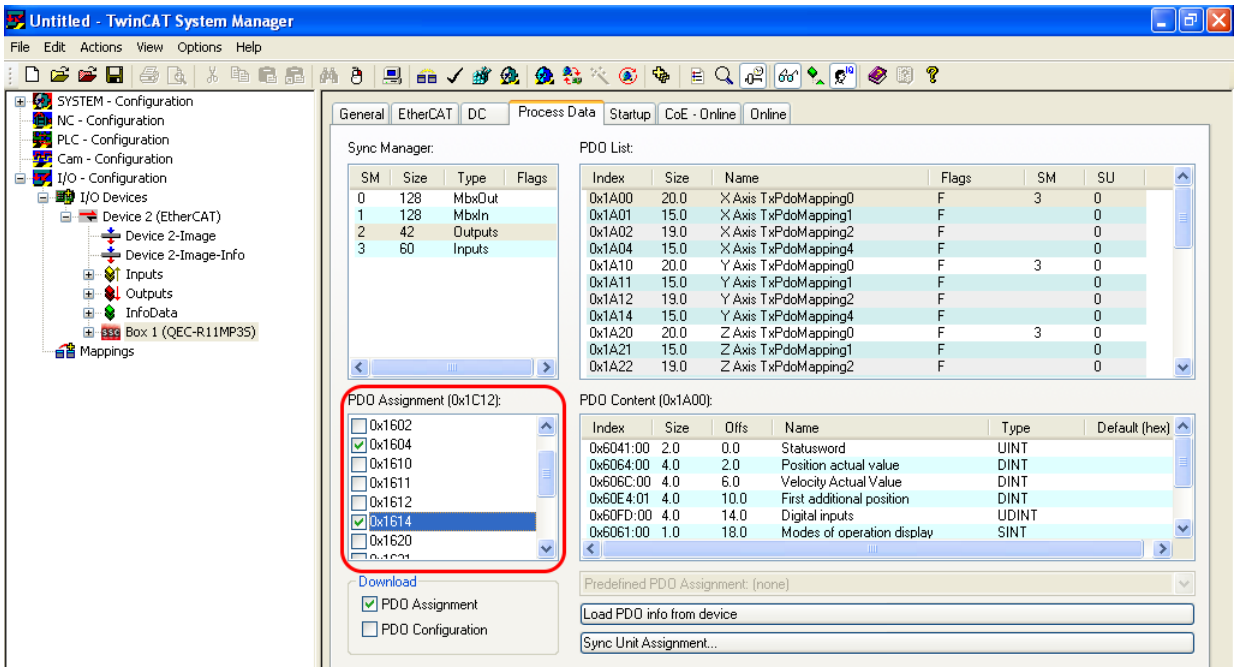


5.2.3 Start to Configure the QEC-R11MP3S

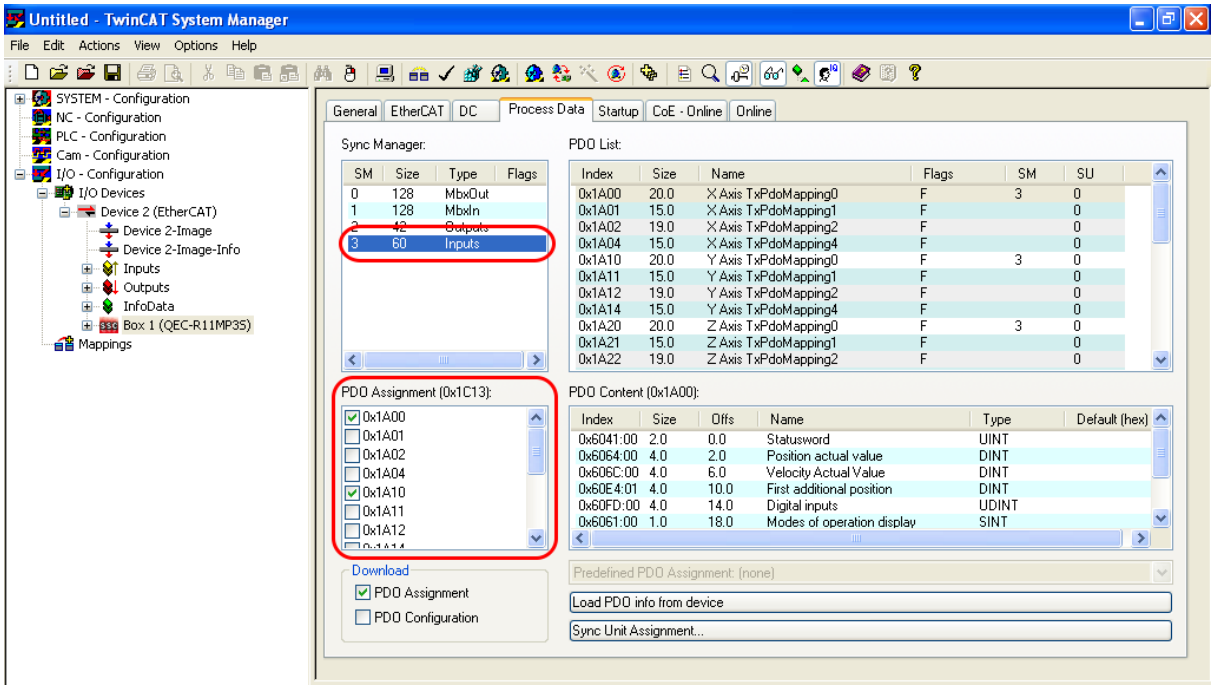
1. Click on the “Process Data” tab, and click "2 33 Outputs" in the "Sync Manager:" field to set Outputs PDO Mapping.
   The original check boxes in the "PDO Assignment (0x1C12):" field are 0x1600, 0x1610, and 0x1620. The default check settings are is used in CSP or CSV mode.



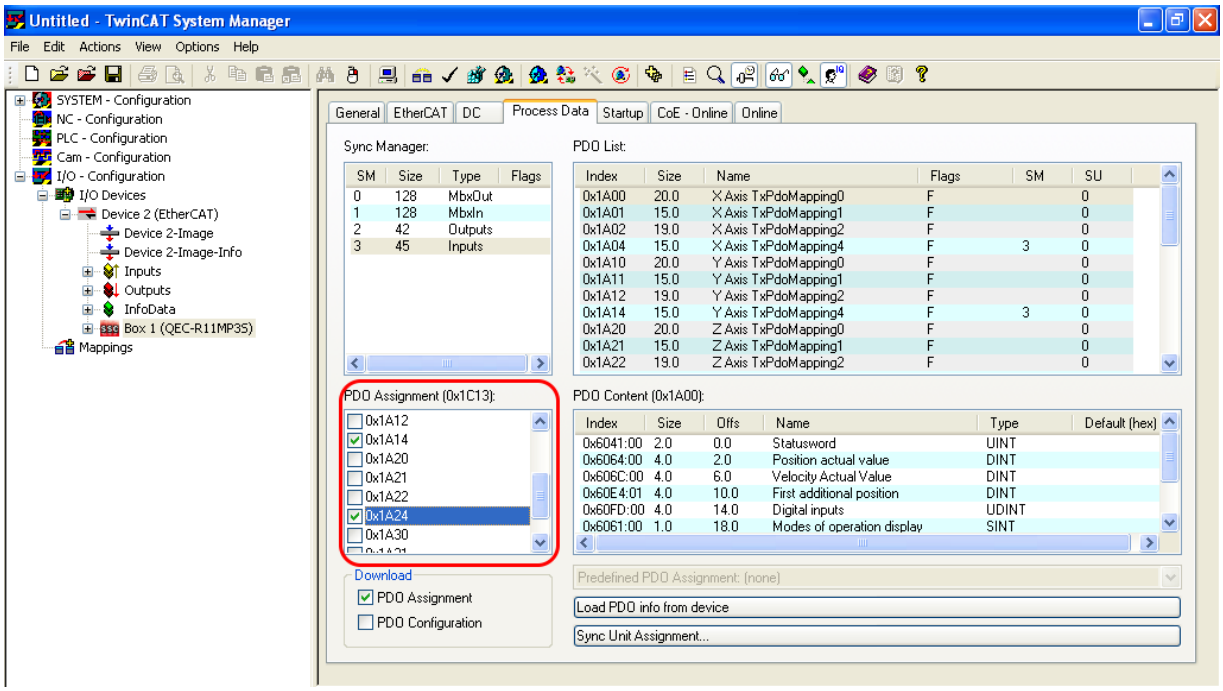
2. Change the checked items to **0x1604**, **0x1614**, and **0x1624**, which modify the Outputs PDO Mapping to suit PP mode.



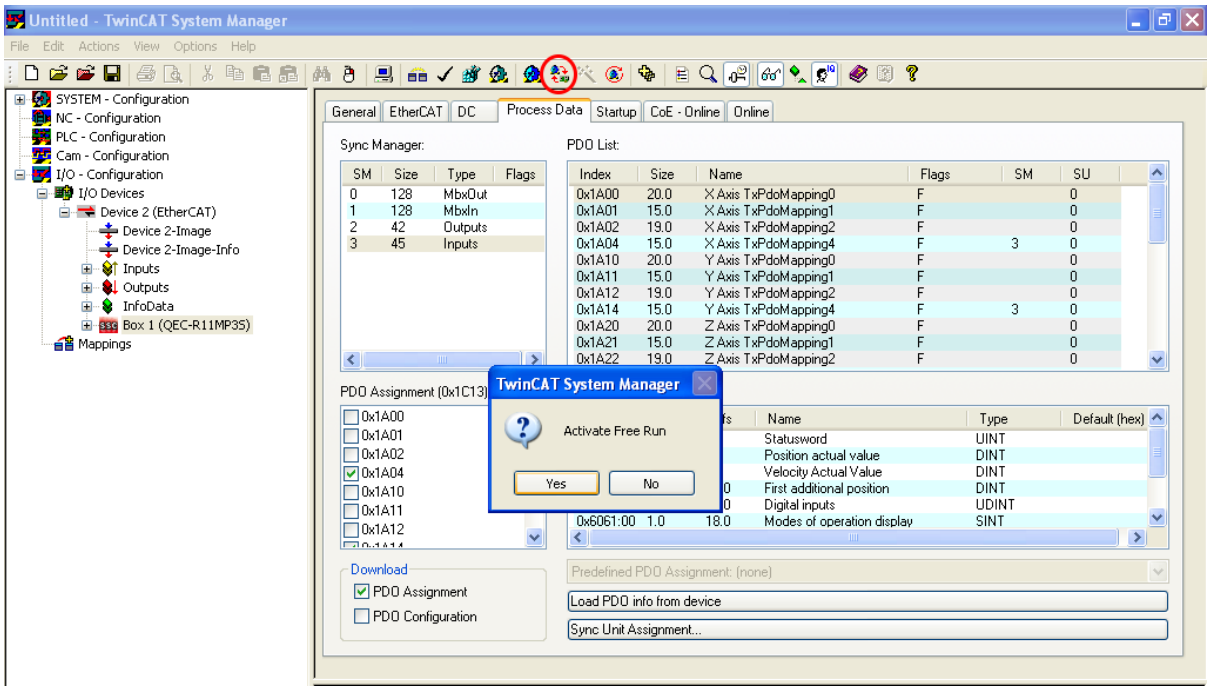
3. Click "3 60 Inputs" in the "Sync Manager:" column to set Inputs PDO Mapping. The original check in the "PDO Assignment (0x1C13):" column is 0x1A00, 0x1A10, 0x1A20, and the default check setting is used in CSP or CSV mode.



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.

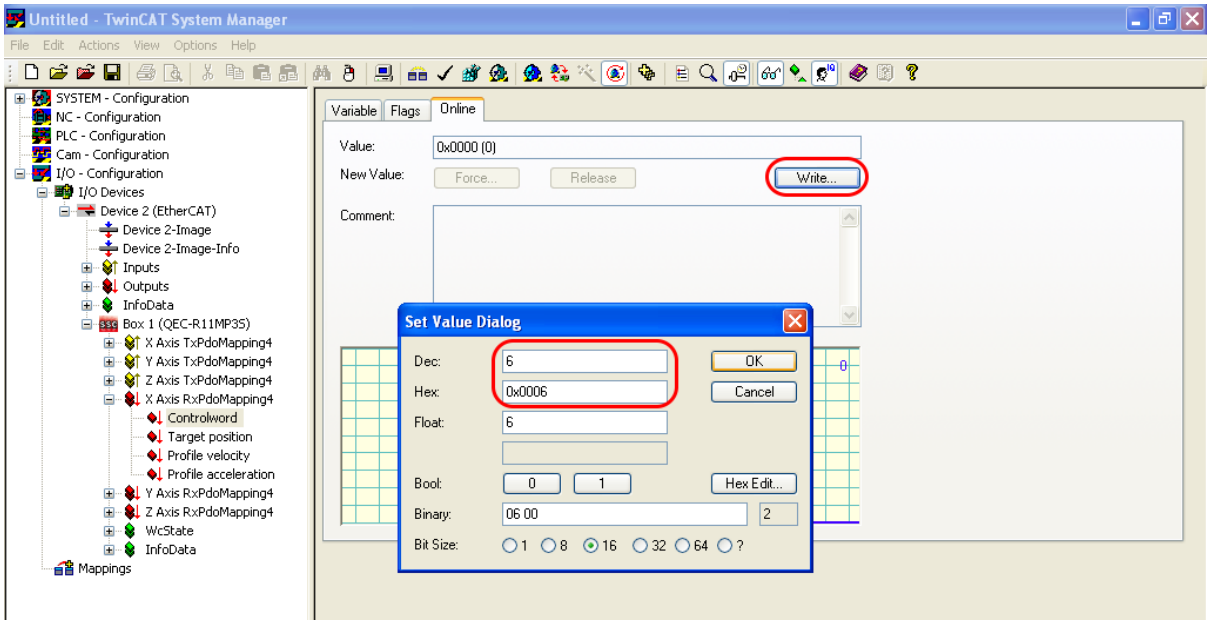


After clicking, a window will pop up asking whether to "Activate Free Run."
Please click "Yes."

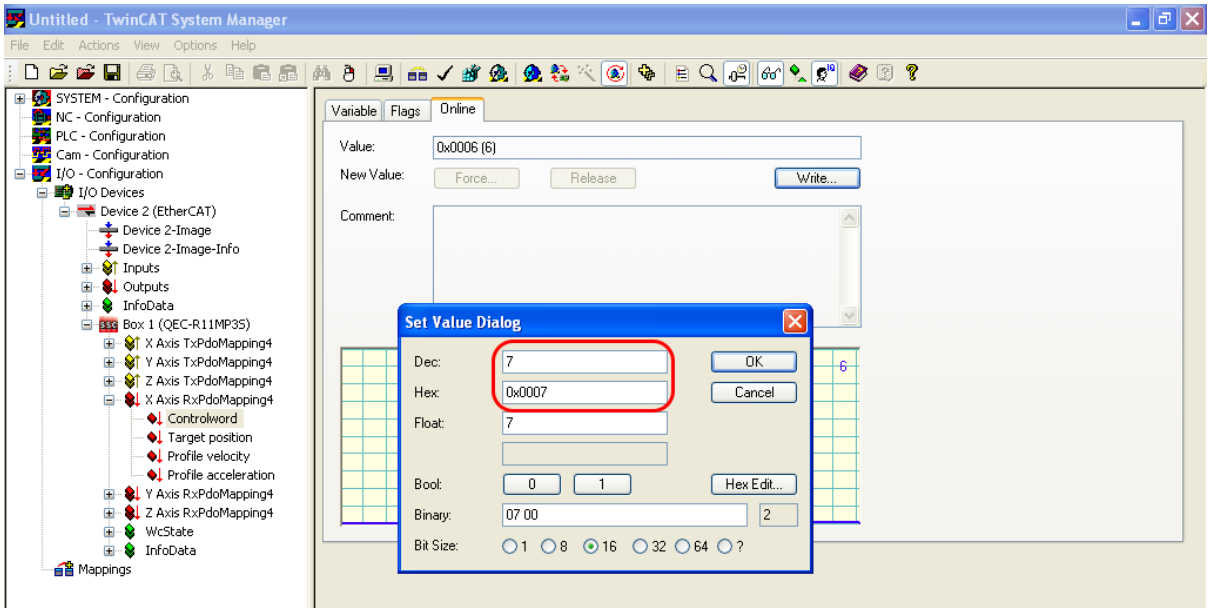
5.2.4 Control the QEC-R11MP3S

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

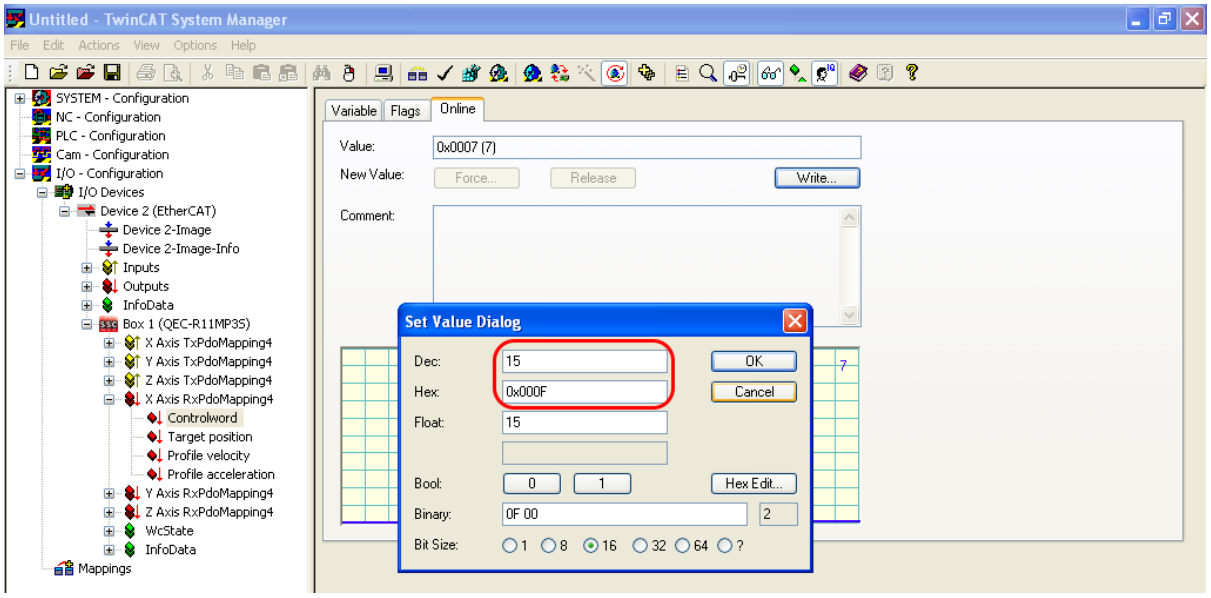
1. Select the drop-down list of QEC-R11MP3S in the left window. And choose “Controlword” in the drop-down list of the "X Asix RxPdoMapping4".
   Click the "Online" page of "Controlword" and click the "Write..." button. Enter 6 in the pop-up window to change the value of Controlword to 6.



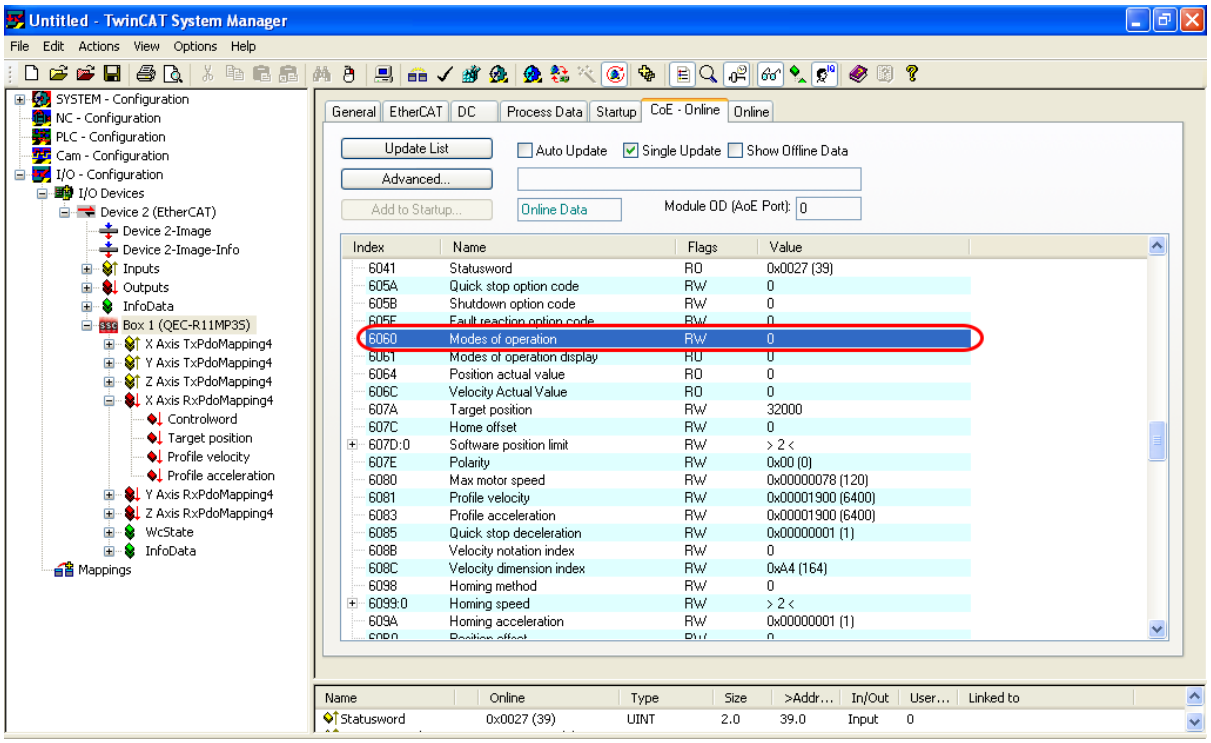
2. Then, enter 7 in the pop-up window to change the value of Controlword to 7.



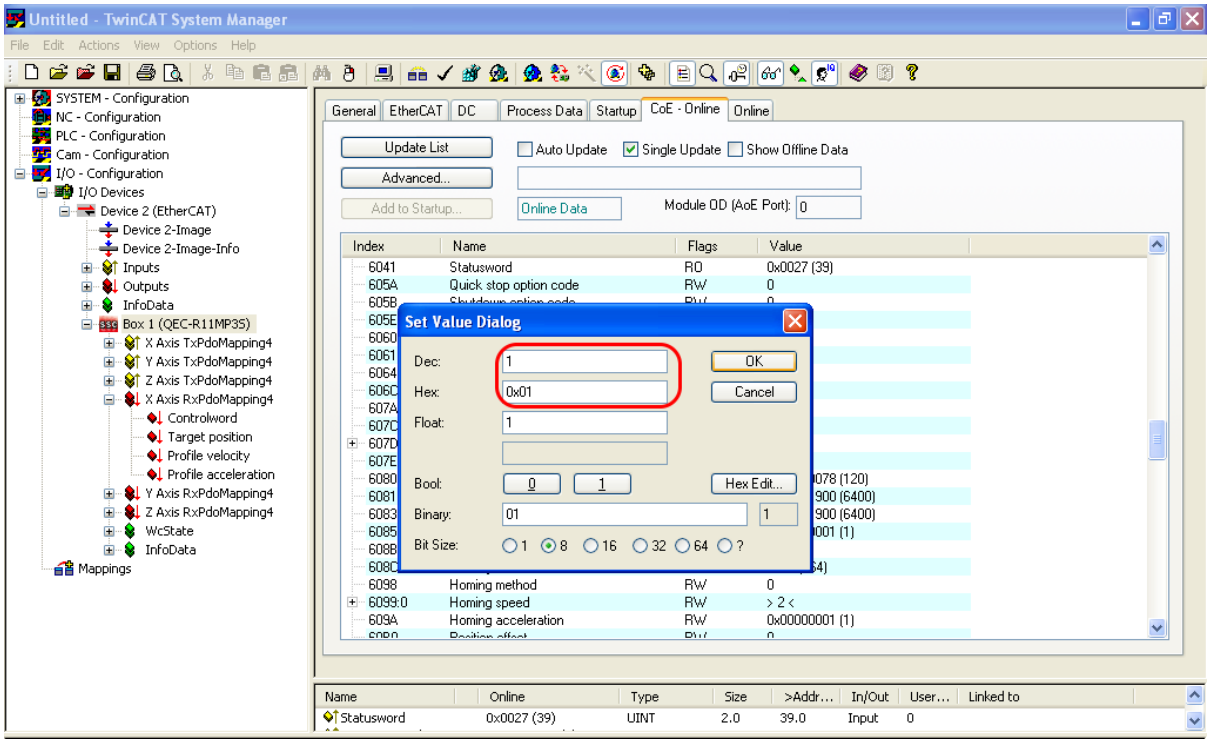
3. Again, enter 15 in the pop-up window to change the value of Controlword to 15.



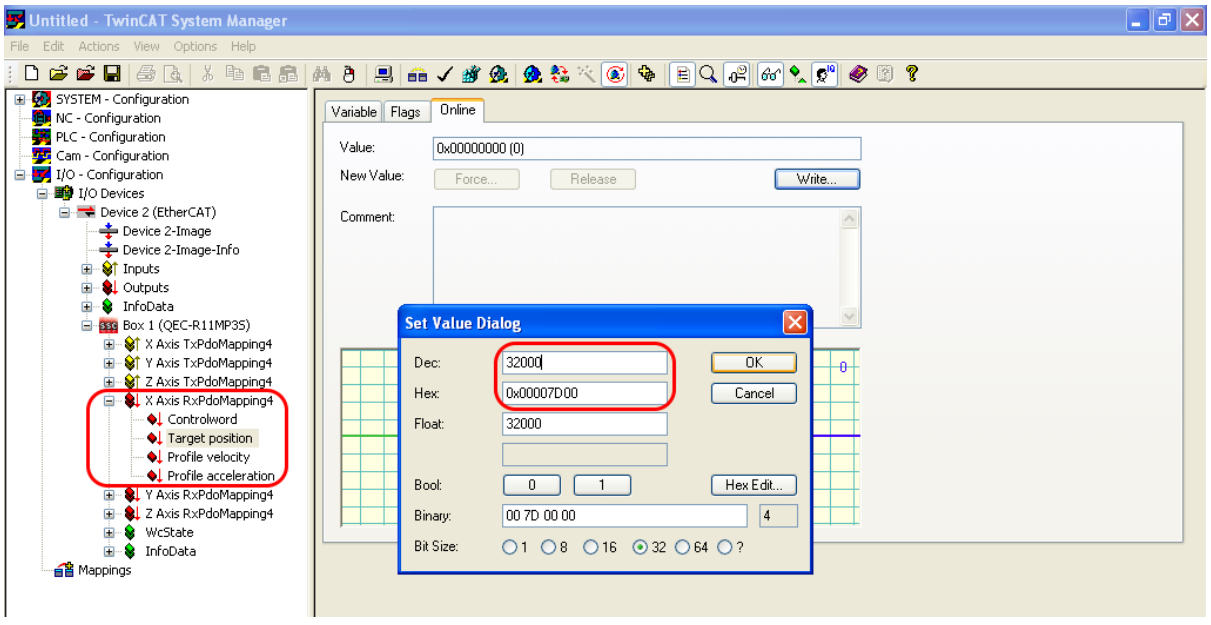
4. Next, select the BOX 1(QEC-R11MP3S) in the left window.
   Click the "CoE - Online" page to view all objects and perform Mailbox transmission. Find object **0x6060** and double-click the object with the left mouse button.



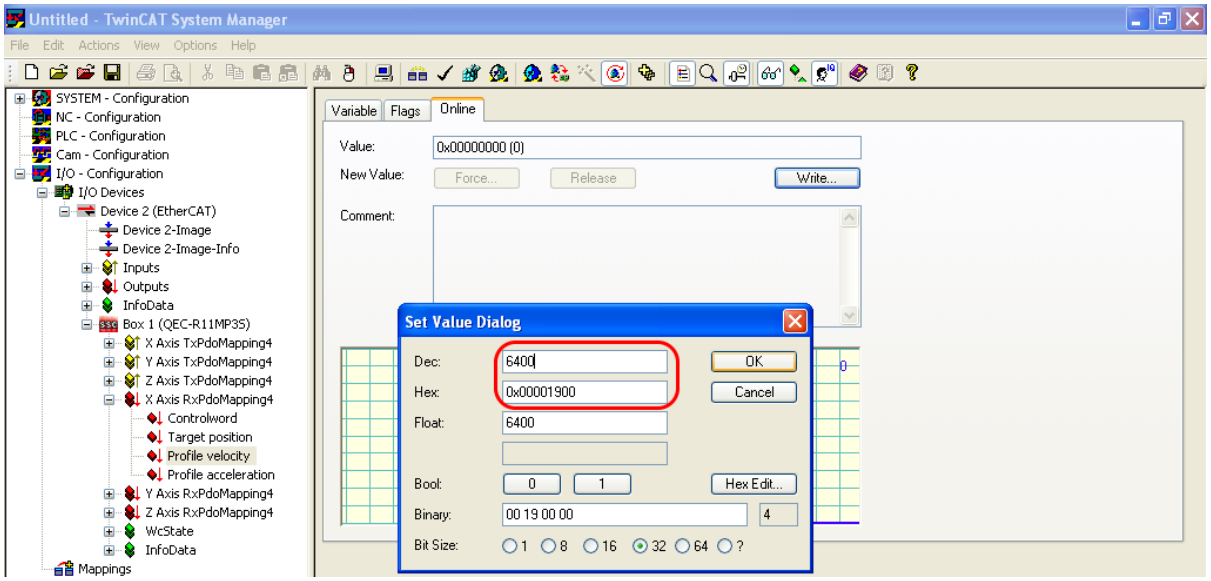
5. Enter 1 in the pop-up window to set the value of object 0x6060 to 1, and specify the operation mode to Profile Position (PP) mode.



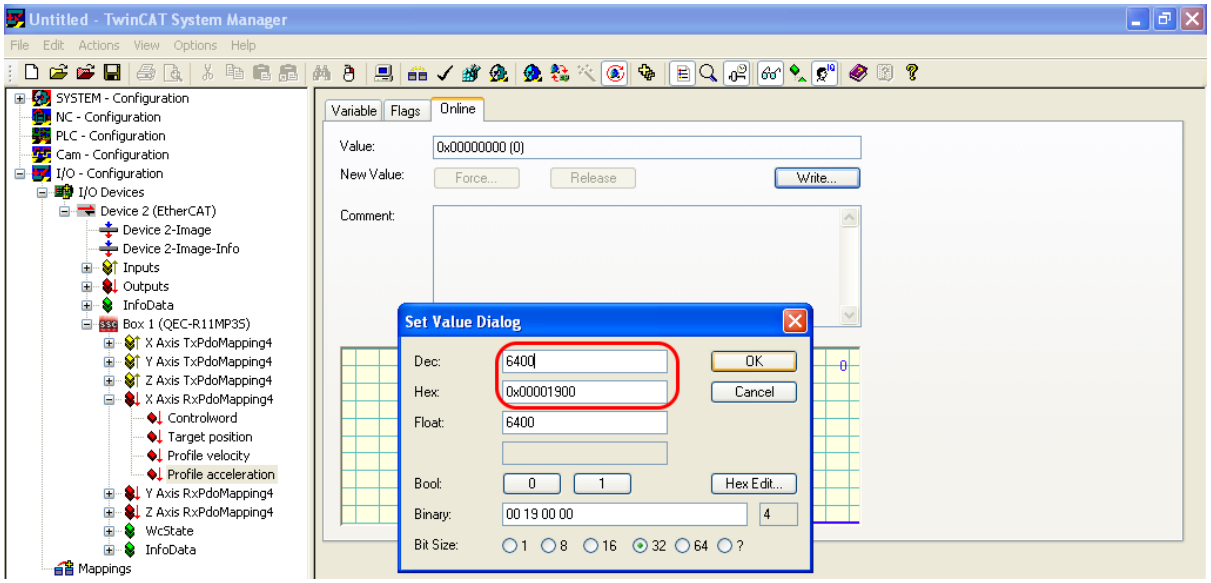
6. Expand "X Axis RxPdoMapping4" and click "Target position", click "Write..." on the "Online" page and enter the position command 32000.



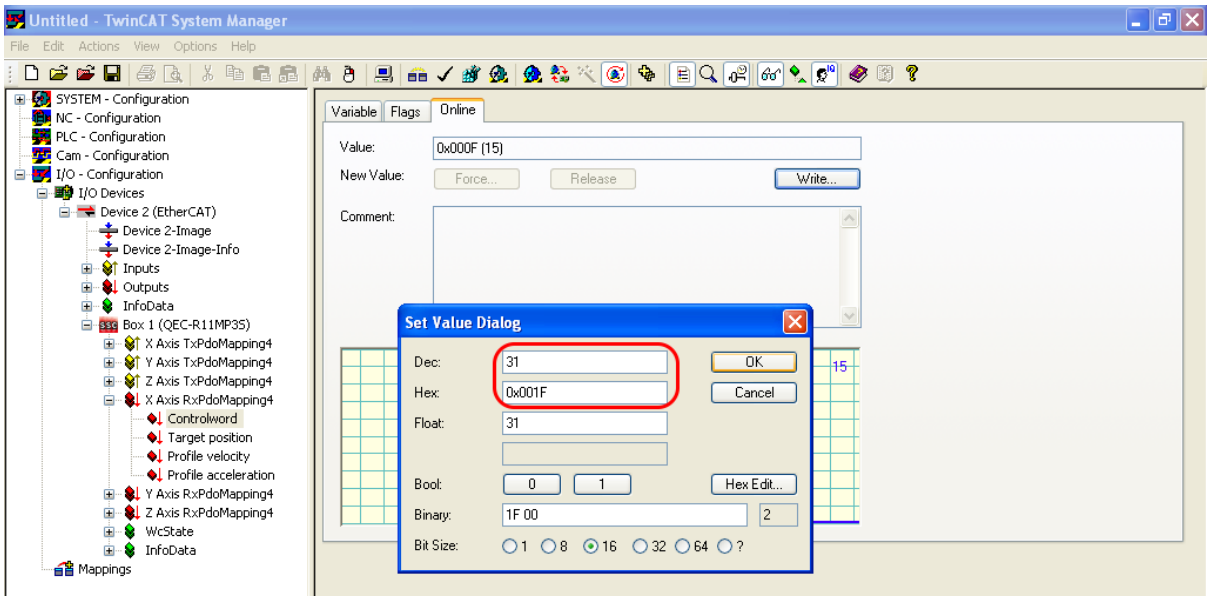
7. Set Profile velocity to 6400.



8. Set Profile acceleration to 6400.

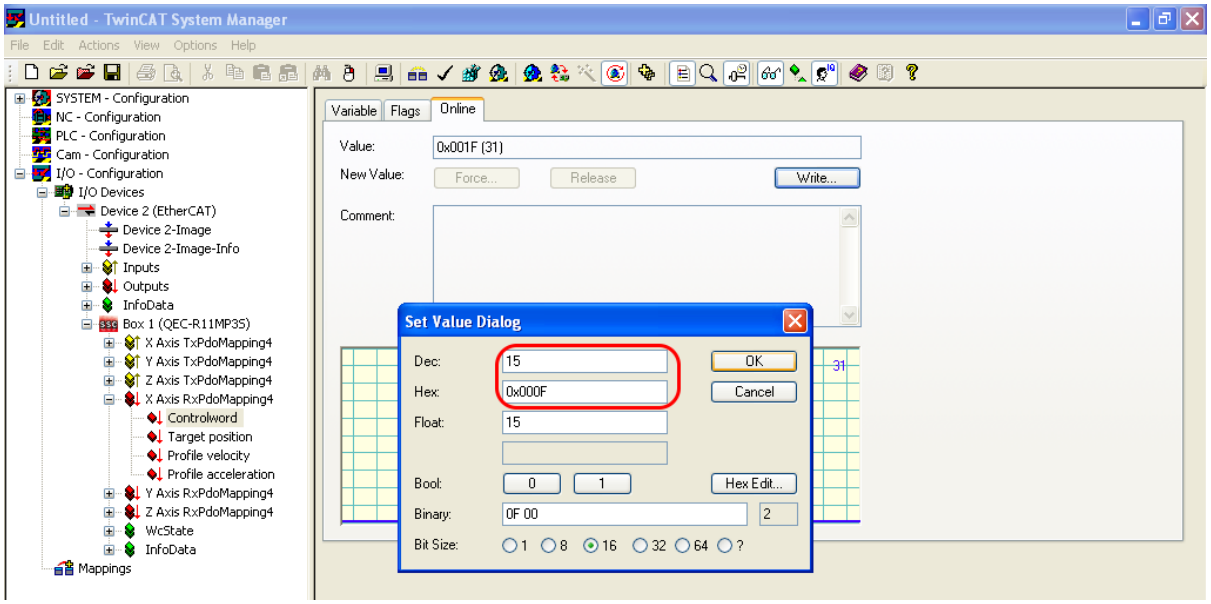


9. Set Controlword to 0x001F to make the New set-point bit 1.



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 covers the basics of using the 86Duino IDE to program and control the QEC-R11MP3S. The 86Duino Coding IDE, with its user-friendly interface and compatibility with the Arduino programming environment, offers a straightforward way to write, upload, and manage code for the QEC Master Series devices to achieve quick and easy control of the EtherCAT Network devices and Slaves, including the QEC-R11MP3S.

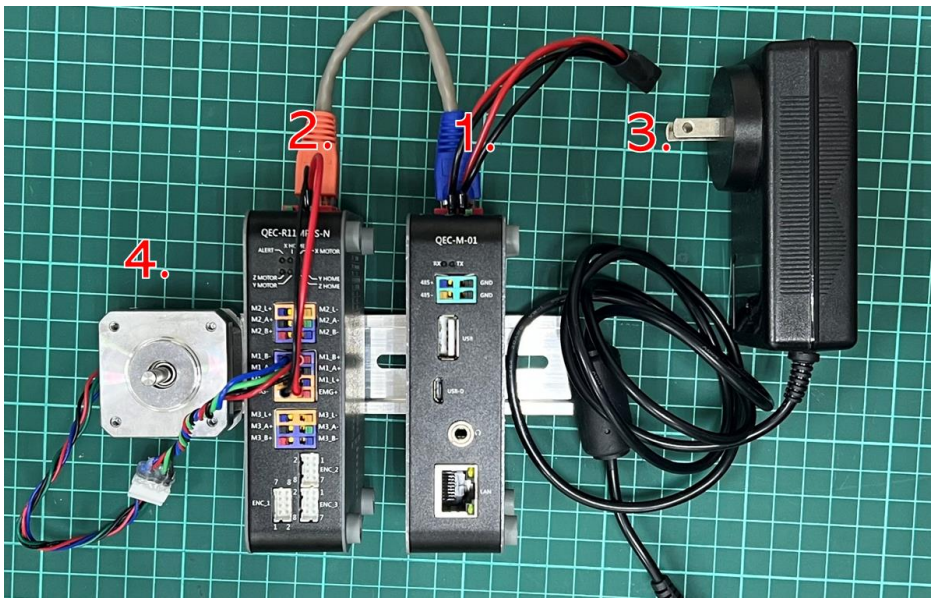
Whether you're new to programming or an experienced developer, the 86Duino IDE makes it easy to implement custom control logic and functionalities for your stepper motor driver. Here, we'll guide you through the key steps to get your QEC-R11MP3S up and running with the 86Duino IDE.

We will show you how to use the EtherCAT Master QEC-M-01P and the QEC-RXXMP3S Series (EtherCAT Slave, 3-axis Stepper Motor Controller). We will be operating CiA402 Profile Position (PP) mode.

5.3.1 Connection and wiring hardware

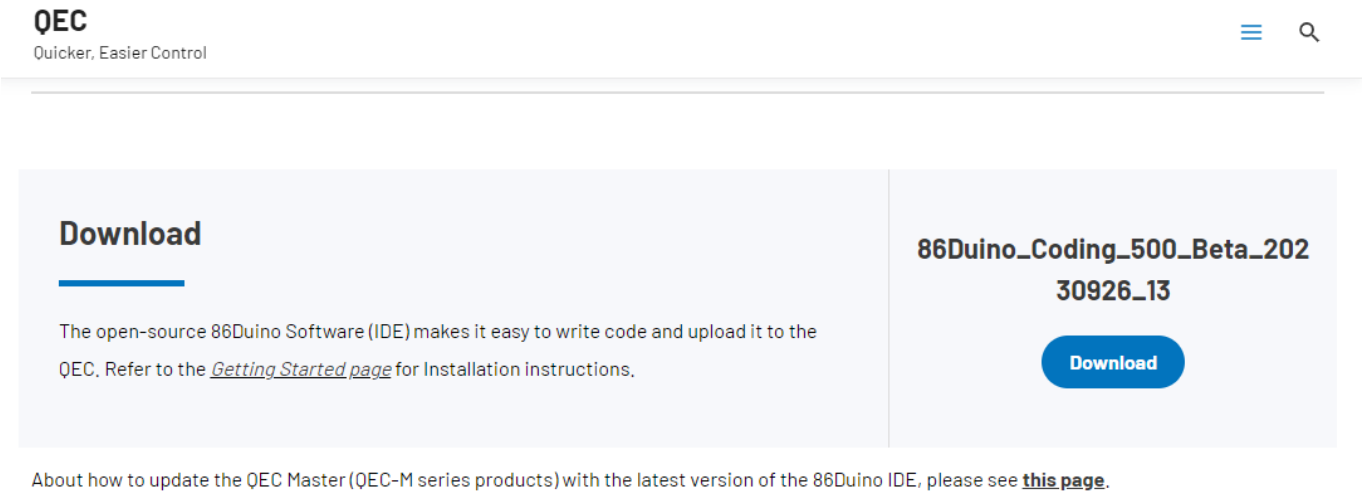
The following devices are used here:

1. QEC-M-01P（EtherCAT Master/PoE）
2. QEC-R11MP3S (EtherCAT Slave, 3-axis Stepper Motor Controller) & LAN cable
3. 24V power supply & EU-type terminal cable
4. A 4-wire two-phase bipolar 42 stepper motor (refer to 86STEP | 86Duino)



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.

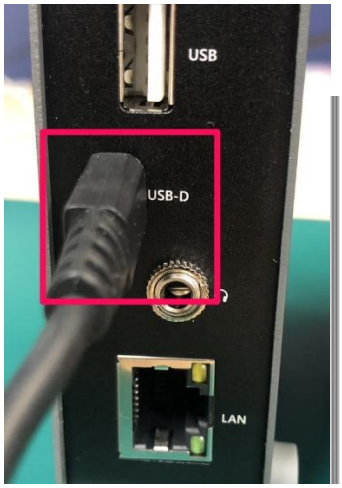
*Note: If Windows displays a warning, click Details once and then click the Continue Run button once.

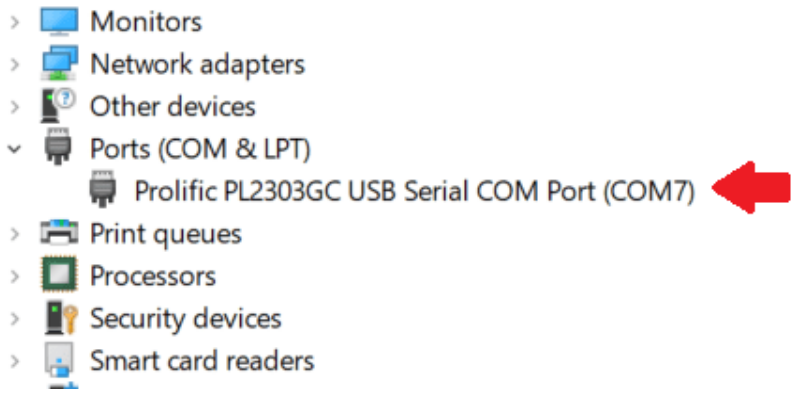
86Duino Coding IDE 500+ looks like below.

5.3.3 Connect to your PC and set up the environment

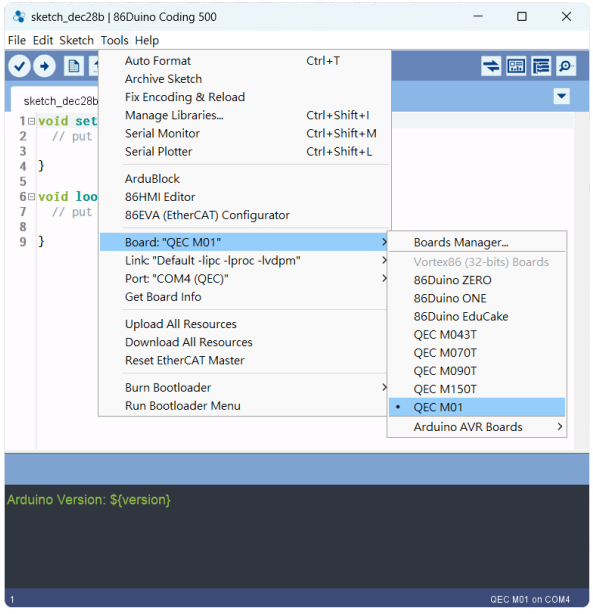
Follow the steps below to set up the environment:

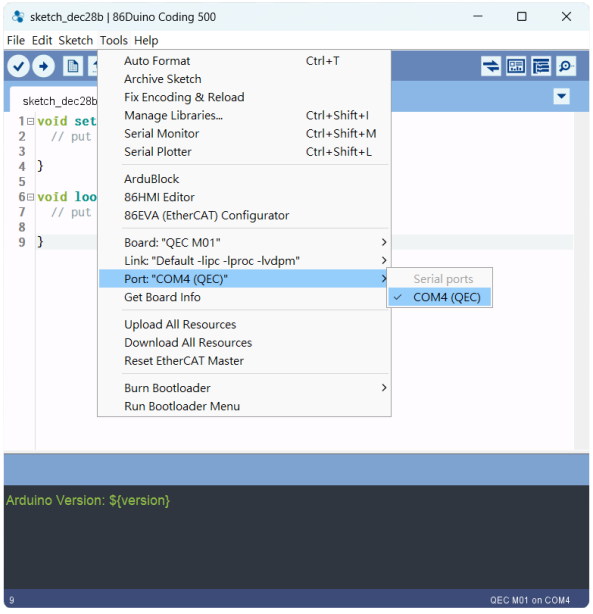
1. Connect the QEC-M-01P to your PC via a Micro USB to USB cable (86Duino IDE installed).
2. Turn on the QEC power.
3. Open "Device Manager" (select in the menu after pressing Win+X) ->" Ports (COM & LPT)" in your PC and expand the ports; you should see that the "Prolific PL2303GC USB Serial COM Port (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-M master model you use).
6. Select Port: In the IDE's menu, select Tools > Port and select the USB port to connect to the QEC-M master (in this case, COM4 (QEC)).





5.3.4 Development Method 1: Write code

In this example, we will control a single-axis stepper motor using the QEC-R11MP3S controller via the EtherCAT communication protocol. We will use the Profile Position (PP) mode from the CiA402 standard, which is a position control mode with a predefined trajectory. Through the `profilePositionBegin()` function, we will set the motor to move to a specified position of 81920, starting the motion with a speed and acceleration of 10000. Once the driver's Control word confirms receipt of the command, we will check if the motor has reached that position. Once reached, we will set the motor's target position to 0 and execute the movement with the same speed and acceleration. The motor will move back and forth between these two positions.

When using a QEC Slave, you can utilize the dedicated QEC EtherCAT Slave library. For instance, the QEC-R11MP3S can use the functionalities of the **EthercatDevice_DmpStepper** Class.

Here's an example of reference code:

```
#include "Ethercat.h"

EthercatMaster master;
EthercatDevice_QECR11MP3S slave;
EthercatDevice_CiA402 motor;
int pp_state = 0;

void setup() {
  master.begin();
  slave.attach(0, master);
  slave.cia402GetServo(1, &motor);
  motor.driveSetMode(CIA402_PP_MODE);
  master.start(1000000, ECAT_SYNC);
  motor.driveEnable();
}

void loop() {
  switch (pp_state) {
    case 0:
      if (motor.profilePositionBegin(64000, 10000, 5000) == 0)
```

```
      pp_state++;
    break;
  case 1:
    motor.profilePositionEnd();
    pp_state++;
    break;
  case 2:
    if (motor.driveIsTargetReached())
      pp_state++;
    break;
  case 3:
    if (motor.profilePositionBegin(1000, 10000, 5000) == 0)
      pp_state++;
    break;
  case 4:
    motor.profilePositionEnd();
    pp_state++;
    break;
  case 5:
    if (motor.driveIsTargetReached()) {
      pp_state = 0;
    }
    break;
 }
}
```

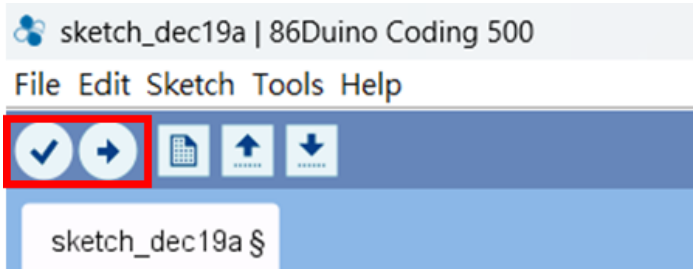
Or you can use EtherCAT CiA402 objects, you can use the dedicated EtherCAT CiA402 library.

```
#include "Ethercat.h"

EthercatMaster EcatMaster;
EthercatDevice_CiA402 motor;

int pp_state = 0;

void setup() {
  EcatMaster.begin();
  motor.attach(0, 0, EcatMaster);
  motor.driveSetMode(CIA402_PP_MODE);
  EcatMaster.start(1000000, ECAT_SYNC);
  motor.driveEnable();
}

void loop() {
  switch (pp_state) {
    case 0:
      if (motor.profilePositionBegin(64000, 10000, 5000) == 0)
        pp_state++;
      break;
    case 1:
      motor.profilePositionEnd();
      pp_state++;
      break;
    case 2:
      if (motor.driveIsTargetReached())
        pp_state++;
      break;
    case 3:
      if (motor.profilePositionBegin(1000, 10000, 5000) == 0)
        pp_state++;
      break;
    case 4:
      motor.profilePositionEnd();
      pp_state++;
      break;
    case 5:
```

```
      if (motor.driveIsTargetReached()) {
        pp_state = 0;
      }
      break;
  }
}
```

After the upload is complete, you can see the motor move to the specified position and then return to the initial position. This process will repeat continuously.

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.

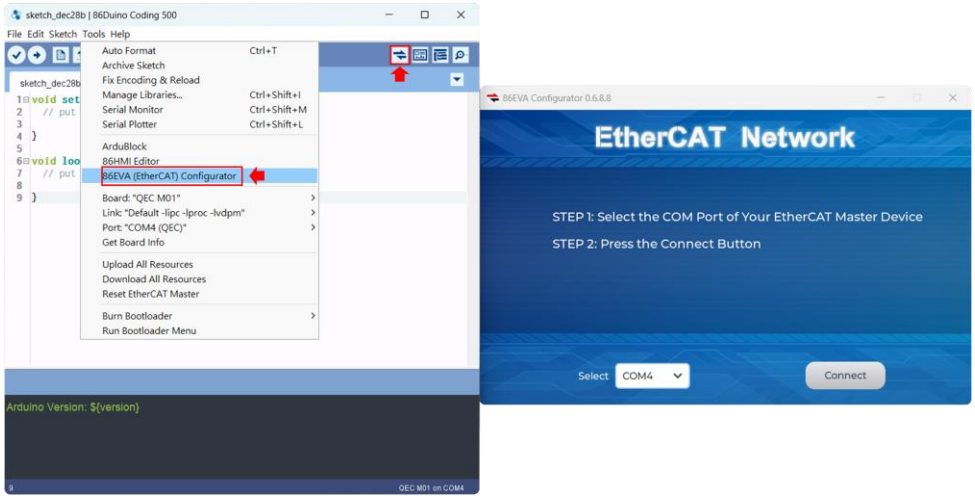


5.3.5 Development Method 2: Use 86EVA with code

86EVA is a graphical EtherCAT configurator based on the EtherCAT Library in the 86Duino IDE and is one of the development kits for 86Duino. The user can use it to configure the EtherCAT network quickly and start programming.

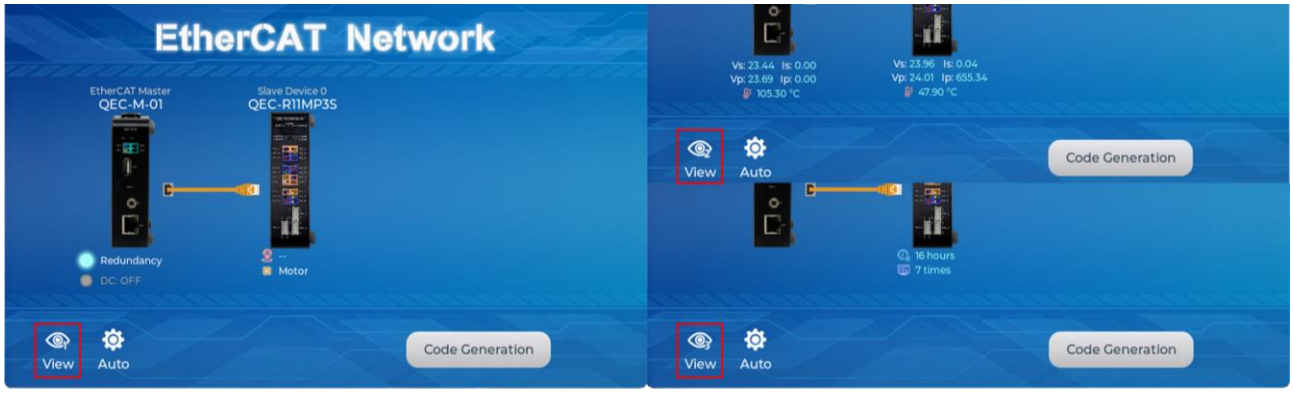
Step 1: Turn on 86EVA and scan

The 86EVA tool can be opened via the following buttons.



Once you have confirmed that the correct COM port has been selected of QEC-M-01P, press the Connect button to start scanning the EtherCAT network.

The connected devices will be displayed after the EtherCAT network has been scanned.
Press the "View" button in the lower left corner to check the device's status (Voltage, Current, and Temperature; View2) and operating time (Hours; View3).



Step 2: Set the parameters

Press twice on the scanned device image to enter the corresponding parameter setting screen.

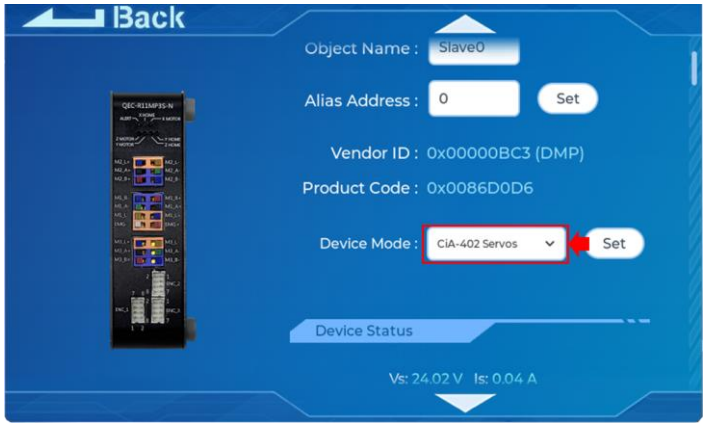
QEC-M-01

Press twice on the image of the QEC-M-01 to see the parameter settings.
This example will use the default settings and not change any settings; please click "Back" in the upper left corner to return.

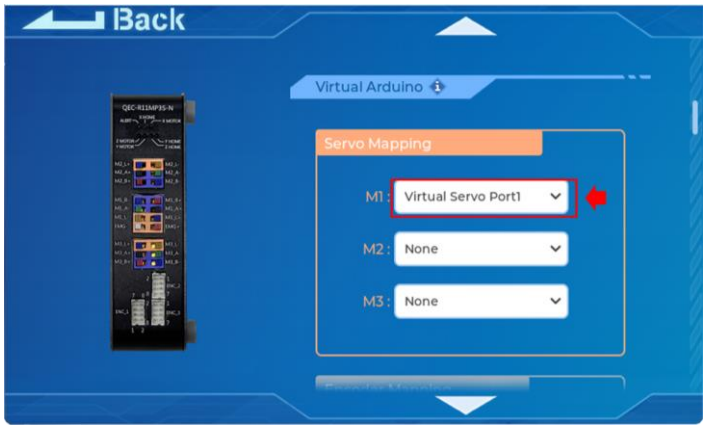
QEC-R11MP3S-N

Press twice on the image of the QEC-R11MP3S to see the parameter settings.

Please note the "Device Mode" field in "General," which has two options: CiA-402 Servos and G-code Machine. This example uses the CiA-402 Servos mode.



Continue to navigate down to the "Servo Mapping" section. Here, we select "Virtual Servo Port1" from the dropdown menu for M1 in the "Servo Mapping".



After finishing, click "Back" in the upper left corner to return.

This action sets the M1 (first-axis motor) of the QEC-R11MP3S to be the Virtual Servo Port1 of the EVA.

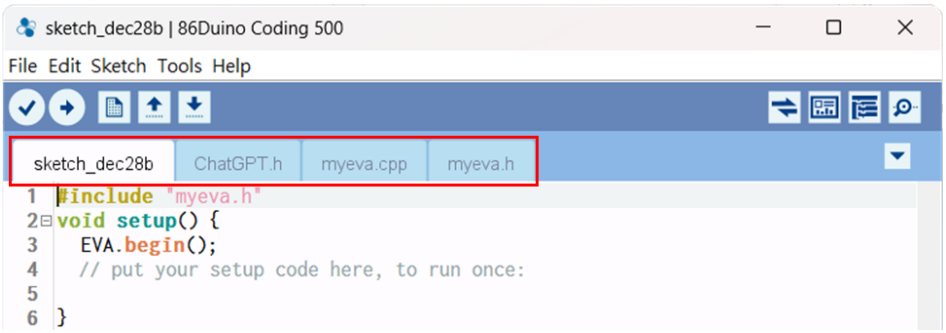
Step 3: Generate the code

Once you've set your device's parameters, go back to the home screen and press the "Code Generation" button in the bottom right corner.

When you're done, double-click the OK button to turn off 86EVA, or it will close in 10 seconds.

The generated code and files are as follows:

- sketch_dec28b: Main Project (.ino, depending on your project name)
- ChatGPT.h: Parameters to provide to ChatGPT referred
- myeva.cpp: C++ program code of 86EVA
- myeva.h: Header file of 86EVA



Additional note: After 86EVA generates code, the following code will be automatically generated in the main program (.ino), and any of them missing will cause 86EVA not to work.

1. `#include “myeva.h”` : Include EVA Header file
2. `EVA.begin() in setup()` : Initialize the EVA function

Step 4: Write the code

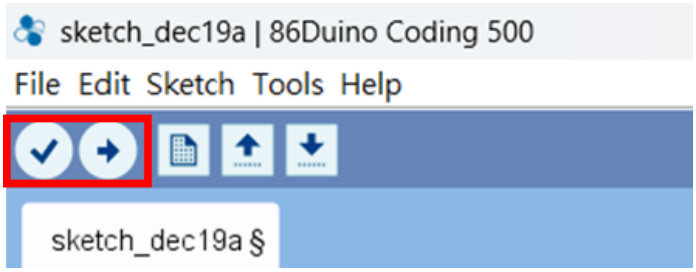
In this example, we will control a single-axis stepper motor using the QEC-R11MP3S controller via the EtherCAT communication protocol. We will use the Profile Position (PP) mode from the CiA402 standard, which is a position control mode with a predefined trajectory. Through the `profilePositionBegin()` function, we will set the motor to move to a specified position of 81920, starting the motion with a speed and acceleration of 10000. Once the driver's Control word confirms receipt of the command, we will check if the motor has reached that position. Once reached, we will set the motor's target position to 0 and execute the movement with the same speed and acceleration. The motor will move back and forth between these two positions.

```
#include "myeva.h"

int pp_state=0;

void setup() {
  EVA.begin();
  VirtualServo1.cia402GetServo()->driveEnable();
}

void loop() {
  switch (pp_state) {
    case 0:
      if (VirtualServo1.cia402GetServo()->profilePositionBegin(64000,
10000, 5000) == 0)
        pp_state++;
      break;
    case 1:
      VirtualServo1.cia402GetServo()->profilePositionEnd();
      pp_state++;
      break;
    case 2:
      if (VirtualServo1.cia402GetServo()->driveIsTargetReached())
        pp_state++;
      break;
    case 3:
      if (VirtualServo1.cia402GetServo()->profilePositionBegin(1000,
10000, 5000) == 0)
```

```
        pp_state++;
      break;
    case 4:
      motor.profilePositionEnd();
      pp_state++;
      break;
    case 5:
      if (VirtualServo1.cia402GetServo()->driveIsTargetReached()) {
        pp_state = 0;
      }
      break;
  }
}
```

After the upload is complete, you can see the motor move to the specified position and then return to the initial position. This process will repeat continuously.

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.



5.4 86Duino Coding IDE (G-code Mode)

In this section, we will show you how to use the EtherCAT Master QEC-M-01P and the QEC-RXXMP3S Series (EtherCAT Slave, 3-axis Stepper Motor Controller). We will be operating G-code mode. We use the same software and hardware with 5.3 86Duino Coding IDE (PP Mode).

5.4.1 Development Method 1: Write code

In this example, we will control a single-axis stepper motor using the QEC-R11MP3S controller via the EtherCAT communication protocol. We will use G-code mode, which is the most widely used numerical control (NC) programming language, primarily used in computer-aided manufacturing to control automated machine tools.

A G-code command `` `G1 X100 F1000` `` can be sent through the `machineGcode()` function, instructing the X-axis to move to the position of 100 with a feed rate (speed) of 1000. After the command is issued, the program pauses for 6 seconds to ensure the action is completed and the system is stable. Then, another G-code command `` `G1 X0 F10000` `` is sent, instructing the machine to return to the starting position of the X-axis (position 0) at a faster feed rate (10000). After issuing the command, the program pauses for another second to ensure the action is completed. The cycle then restarts.

When using a QEC Slave, you can utilize the dedicated QEC EtherCAT Slave library. For instance, the QEC-R11MP3S can use the functionalities of the **EthercatDevice_DmpStepper** Class.

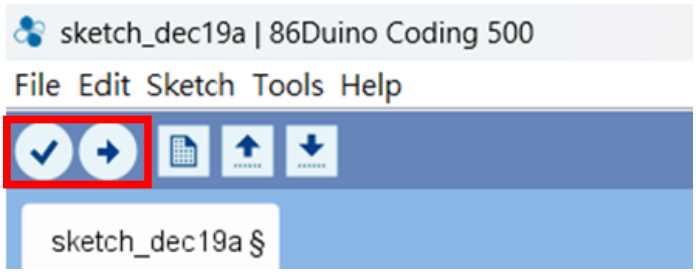
Here's an example of reference code:

```
#include "Ethercat.h"

EthercatMaster EcatMaster;
EthercatDevice_QECR11MP3S Slave1;

void myCallback() {
  Slave1.update();
}
```

```
void setup() {
  EcatMaster.begin();
  Slave1.attach(0, EcatMaster);
  EcatMaster.attachCyclicCallback(myCallback);
  EcatMaster.start(1000000, ECAT_SYNC);
  Slave1.machineServoOn();
}

void loop() {
  Slave1.machineGcode("G1 X100 F1000");
  delay(6000);
  Slave1.machineGcode("G1 X0 F10000");
  delay(1000);
}
```

After the upload is complete, you can see the X-axis motor move to position 100 at a speed of 1000, after which the program pauses for 6 seconds to ensure the action is completed. Then, the X-axis returns to position 0 at a speed of 10000, and the program pauses for 1 second to ensure the action is completed. This process will continuously repeat.

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.

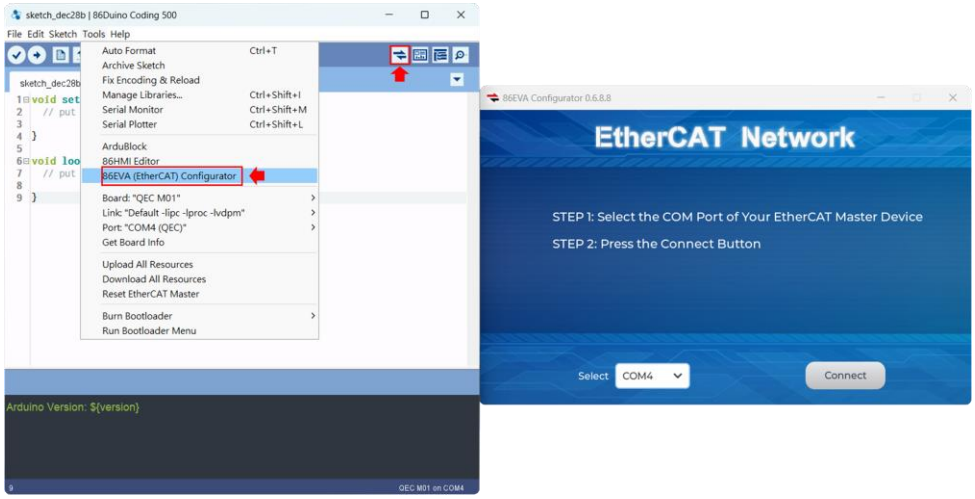


5.4.2 Development Method 2: Use 86EVA with code

86EVA is a graphical EtherCAT configurator based on the EtherCAT Library in the 86Duino IDE and is one of the development kits for 86Duino. The user can use it to configure the EtherCAT network quickly and start programming.

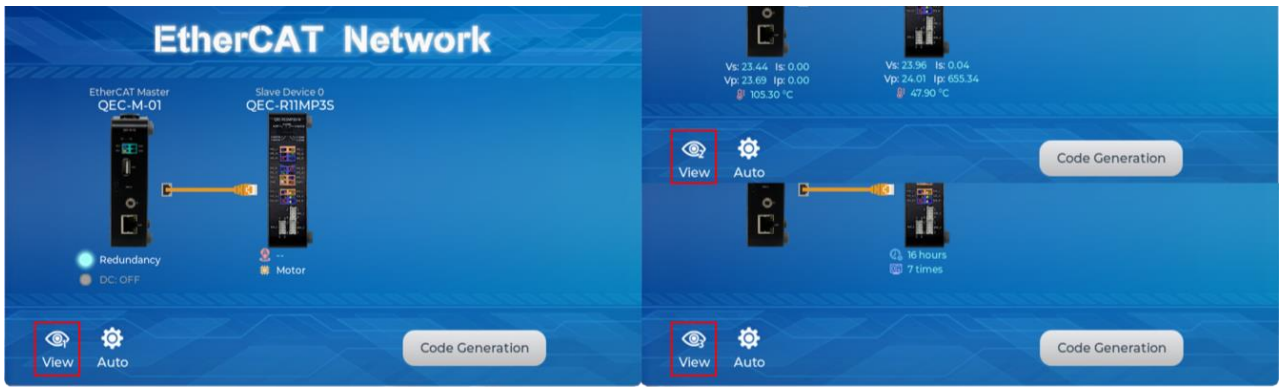
Step 1: Turn on 86EVA and scan

The 86EVA tool can be opened via the following buttons.



Once you have confirmed that the correct COM port has been selected of QEC-M-01P, press the Connect button to start scanning the EtherCAT network.

The connected devices will be displayed after the EtherCAT network has been scanned.
Press the "View" button in the lower left corner to check the device's status (Voltage, Current, and Temperature; View2) and operating time (Hours; View3).



Step 2: Set the parameters

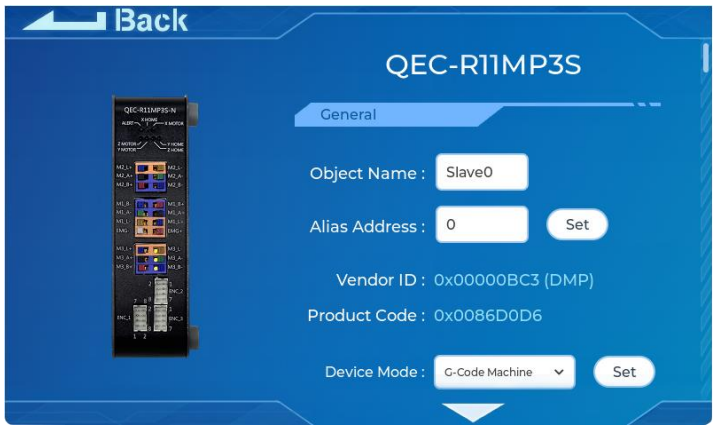
Press twice on the scanned device image to enter the corresponding parameter setting screen.

QEC-M-01

Press twice on the image of the QEC-M-01 to see the parameter settings.
This example will use the default settings and not change any settings; please click "Back" in the upper left corner to return.

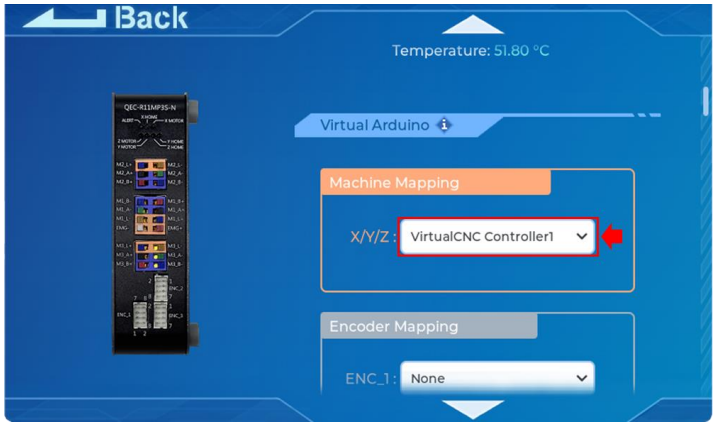
QEC-R11MP3S-N

Press twice on the image of the QEC-R11MP3S to see the parameter settings.

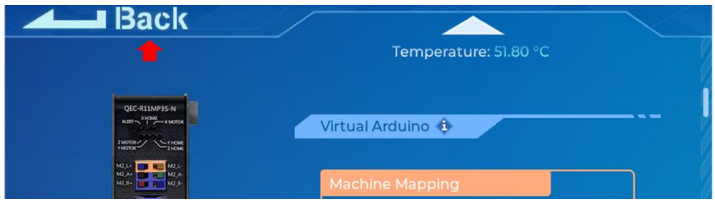


Please note the "Device Mode" field in "General," which has two options: CiA-402 Servos and G-code Machine. This example uses the G-code Machine mode.

Continue to navigate down to the "Machine Mapping" section. Here, we will select "VirtualCNC Controller1" from the dropdown menu for "X/Y/Z".



After finishing, click "Back" in the upper left corner to return.



This action sets the three-axis drive control of the QEC-R11MP3S to be the VirtualCNC Controller1 of the EVA.

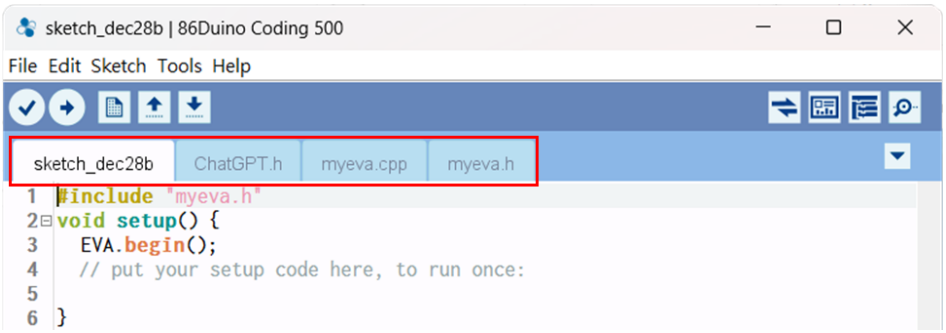
Step 3: Generate the code

Once you've set your device's parameters, go back to the home screen and press the "Code Generation" button in the bottom right corner.

When you're done, double-click the OK button to turn off 86EVA, or it will close in 10 seconds.

The generated code and files are as follows:

- sketch_dec28b: Main Project (.ino, depending on your project name)
- ChatGPT.h: Parameters to provide to ChatGPT referred
- myeva.cpp: C++ program code of 86EVA
- myeva.h: Header file of 86EVA



Additional note: After 86EVA generates code, the following code will be automatically generated in the main program (.ino), and any of them missing will cause 86EVA not to work.

3. `#include “myeva.h”` : Include EVA Header file
4. `EVA.begin() in setup()` : Initialize the EVA function

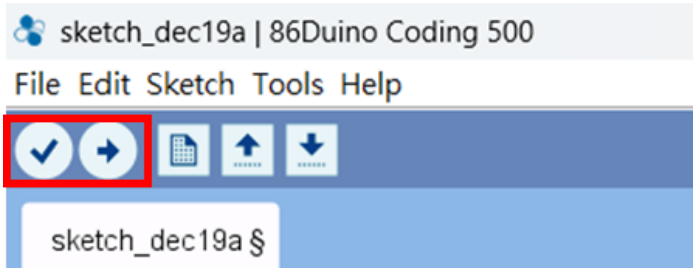
Step 4: Write the code

Here's an example of reference code:

```
#include "myeva.h"

void setup() {
  EVA.begin();
  VirtualCNC1.begin();
}

void loop() {
  VirtualCNC1.gcode("G1 X100 F1000");
  delay(6000);
  VirtualCNC1.gcode("G1 X0 F10000");
  delay(1000);
}
```

After the upload is complete, you can see the X-axis motor move to position 100 at a speed of 1000, after which the program pauses for 6 seconds to ensure the action is completed. Then, the X-axis returns to position 0 at a speed of 10000, and the program pauses for 1 second to ensure the action is completed. This process will continuously repeat.

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.



Ch. 6

Slave Information

6.1 ESI (EtherCAT Slave Information) file

The ESI files contain information unique to the EtherCAT Slave Terminals in XML format. You can load an ESI file into the Support Software to easily allocate Slave Terminal process data and other settings. The ESI files for QEC EtherCAT slaves 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 Master software. The latest version of the XML device description file can be downloaded from the QEC website.

https://www.qec.tw/

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

Index 1000 Device type

Index	Name	Data type	Flags	Default
1000	Device type	UINT32	RO	0x00040192 (262546)

Index 1001 Error register

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

Index 1008 Device name

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

Table 4-1: Device Name

Type	Device Name
Stepper Motor Controller without PoE	QEC-R00MP3S
	QEC-R00MP1S
Stepper Motor Controller with PoE	QEC-R11MP3S
	QEC-R11MP1S

Index 1009 Hardware version

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

Index 100A Software version

Index	Name	Data type	Flags	Default
100A	Software version	STRING	RO	Depends on the version of the product you have.

Index 1018 Identity

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 4-3: Product code & Revision Number

Model Name	Product code
QEC-R00MP3S	0x0086d0d9
QEC-R00MP1S	0x0086d0e1
QEC-R11MP3S	0x0086d0d6
QEC-R11MP1S	0x0086d0de

Index 10F1 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 10F8 Timestamp Object

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

6.2.2 RxPDO Mapping Objects

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

Index 1600 X Axis RX PDO Mapping0

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

Index 1601 X Axis RX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1601:0	X Axis RxPdoMapping1	UINT8	RO	> 2 <	RX PDO Mapping for CSP.
1601:01	SubIndex 001	UINT32	RO	0x6040:00, 16	Map control word.
1601:02	SubIndex 002	UINT32	RO	0x607A:00, 32	Map target position.

Index 1602 X Axis RX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1602:0	X Axis RxPdoMapping2	UINT8	RO	> 2 <	RX PDO Mapping for CSV.
1602:01	SubIndex 001	UINT32	RO	0x6040:00, 16	Map control word.
1602:02	SubIndex 002	UINT32	RO	0x60FF:00, 32	Map target velocity.

Index 1604 X Axis RX PDO Mapping4

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

Index 1610 Y Axis RX PDO Mapping0

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

Index 1611 Y Axis RX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1611:0	Y Axis RxPdoMapping1	UINT8	RO	> 2 <	RX PDO Mapping for CSP.
1611:01	SubIndex 001	UINT32	RO	0x6840:00, 16	Map control word.
1611:02	SubIndex 002	UINT32	RO	0x687A:00, 32	Map target position.

Index 1612 Y Axis RX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1612:0	Y Axis RxPdoMapping2	UINT8	RO	> 2 <	RX PDO Mapping for CSV.
1612:01	SubIndex 001	UINT32	RO	0x6840:00, 16	Map control word.
1612:02	SubIndex 002	UINT32	RO	0x68FF:00, 32	Map target velocity.

Index 1614 Y Axis RX PDO Mapping4

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

Index 1620 Z Axis RX PDO Mapping0

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

Index 1621 Z Axis RX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1621:0	Z Axis RxPdoMapping1	UINT8	RO	> 2 <	RX PDO Mapping for CSP.
1621:01	SubIndex 001	UINT32	RO	0x7040:00, 16	Map control word.
1621:02	SubIndex 002	UINT32	RO	0x707A:00, 32	Map target position.

Index 1622 Z Axis RX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1622:0	Z Axis RxPdoMapping2	UINT8	RO	> 2 <	RX PDO Mapping for CSV.
1622:01	SubIndex 001	UINT32	RO	0x7040:00, 16	Map control word.
1622:02	SubIndex 002	UINT32	RO	0x70FF:00, 32	Map target velocity.

Index 1624 Z Axis RX PDO Mapping4

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

Index 1630 G code RX PDO Mapping0

Index	Name	Data type	Flags	Default	Description
1630:0	G code RxPdoMapping0	UINT8	RO	> 6 <	RX PDO Mapping for Gcode mode.
1630:01	SubIndex 001	UINT32	RO	0x50D0:00, 16	Map G code header.
1630:02	SubIndex 002	UINT32	RO	0x50D1:00, 208	Map G code string.
1630:03	SubIndex 003	UINT32	RO	0x50D9:00, 8	Map emergency stop.
1630:04	SubIndex 004	UINT32	RO	0x50CD:01, 64	-
1630:05	SubIndex 005	UINT32	RO	0x50CD:02, 64	-
1630:06	SubIndex 006	UINT32	RO	0x50CD:03, 64	-

Index 1631 G code RX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1631:0	G code RxPdoMapping1	UINT8	RO	> 2 <	RX PDO Mapping for G-code machine objects.
1631:01	SubIndex 001	UINT32	RO	0x50D7:00, 8	Map machine servo on.
1631:02	SubIndex 002	UINT32	RO	0x50D8:00, 8	Map machine homing.

Index 1632 G code RX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1632:0	G code RxPdoMapping2	UINT8	RO	> 1 <	For RX PDO mapping number is less than 3.
1632:01	SubIndex 001	UINT32	RO	0x0000:00, 0	Empty

6.2.3 TxPDO Mapping Objects

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

Index 1A00 X Axis TX PDO Mapping0

Index	Name	Data type	Flags	Default	Description
1A00:0	X Axis TxPdoMapping0	UINT8	RO	> 7 <	TX PDO Mapping for CSP, CSV.
1A00:01	SubIndex 001	UINT32	RO	0x6041:00, 16	Map status word.
1A00:02	SubIndex 002	UINT32	RO	0x6064:00, 32	Map actual position.
1A00:03	SubIndex 003	UINT32	RO	0x606C:00, 32	Map actual velocity.
1A00:04	SubIndex 004	UINT32	RO	0x60E4:01, 32	Map additional position actual value.
1A00:05	SubIndex 005	UINT32	RO	0x60FD:00, 32	Map digital inputs.
1A00:06	SubIndex 006	UINT32	RO	0x6061:00, 8	Map mode of operation display.
1A00:07	SubIndex 007	UINT32	RO	0x5024:01, 8	ENC status.

Index 1A01 X Axis TX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1A01:0	X Axis TxPdoMapping1	UINT8	RO	> 5 <	TX PDO Mapping for CSP.
1A01:01	SubIndex 001	UINT32	RO	0x6041:00, 16	Map status word.
1A01:02	SubIndex 002	UINT32	RO	0x6064:00, 32	Map actual position.
1A01:03	SubIndex 003	UINT32	RO	0x60E4:01, 32	Map additional position actual value.
1A01:04	SubIndex 004	UINT32	RO	0x60FD:00, 32	Map digital inputs.
1A01:05	SubIndex 005	UINT32	RO	0x5024:01, 8	ENC status.

Index 1A02 X Axis TX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1A02:0	X Axis TxPdoMapping2	UINT8	RO	> 6 <	TX PDO Mapping for CSV.
1A02:01	SubIndex 001	UINT32	RO	0x6041:00, 16	Map control word.
1A02:02	SubIndex 002	UINT32	RO	0x6064:00, 32	Map actual position.
1A02:03	SubIndex 003	UINT32	RO	0x606C:00, 32	Map actual velocity.
1A02:04	SubIndex 004	UINT32	RO	0x60E4:01, 32	Map additional position actual value.
1A02:05	SubIndex 005	UINT32	RO	0x60FD:00, 32	Map digital inputs.
1A02:06	SubIndex 006	UINT32	RO	0x5024:01, 8	ENC status.

Index 1A04 X Axis TX PDO Mapping4

Index	Name	Data type	Flags	Default	Description
1A04:0	X Axis TxPdoMapping4	UINT8	RO	> 5 <	TX PDO Mapping for PP.
1A04:01	SubIndex 001	UINT32	RO	0x6041:00, 16	Map status word.
1A04:02	SubIndex 002	UINT32	RO	0x6064:00, 32	Map actual position.
1A04:03	SubIndex 003	UINT32	RO	0x60E4:01, 32	Map additional position actual value.
1A04:04	SubIndex 004	UINT32	RO	0x60FD:00, 32	Map digital inputs.
1A04:05	SubIndex 005	UINT32	RO	0x5024:01, 8	ENC status.

Index 1A10 Y Axis TX PDO Mapping0

Index	Name	Data type	Flags	Default	Description
1A10:0	Y Axis TxPdoMapping0	UINT8	RO	> 7 <	TX PDO Mapping for CSP, CSV.
1A10:01	SubIndex 001	UINT32	RO	0x6841:00, 16	Map status word.
1A10:02	SubIndex 002	UINT32	RO	0x6864:00, 32	Map actual position.
1A10:03	SubIndex 003	UINT32	RO	0x686C:00, 32	Map actual velocity.
1A10:04	SubIndex 004	UINT32	RO	0x68E4:01, 32	Map additional position actual value.
1A10:05	SubIndex 005	UINT32	RO	0x68FD:00, 32	Map digital inputs.
1A10:06	SubIndex 006	UINT32	RO	0x6861:00, 8	Map mode of operation display.
1A10:07	SubIndex 007	UINT32	RO	0x5024:02, 8	ENC status.

Index 1A11 Y Axis TX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1A11:0	Y Axis TxPdoMapping1	UINT8	RO	> 5 <	TX PDO Mapping for CSP.
1A11:01	SubIndex 001	UINT32	RO	0x6841:00, 16	Map status word.
1A11:02	SubIndex 002	UINT32	RO	0x6864:00, 32	Map actual position.
1A11:03	SubIndex 003	UINT32	RO	0x68E4:01, 32	Map additional position actual value.
1A11:04	SubIndex 004	UINT32	RO	0x68FD:00, 32	Map digital inputs.
1A11:05	SubIndex 005	UINT32	RO	0x5024:02, 8	ENC status.

Index 1A12 Y Axis TX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1A12:0	Y Axis TxPdoMapping2	UINT8	RO	> 6 <	TX PDO Mapping for CSV.
1A12:01	SubIndex 001	UINT32	RO	0x6841:00, 16	Map control word.
1A12:02	SubIndex 002	UINT32	RO	0x6864:00, 32	Map actual position.
1A12:03	SubIndex 003	UINT32	RO	0x686C:00, 32	Map actual velocity.
1A12:04	SubIndex 004	UINT32	RO	0x68E4:01, 32	Map additional position actual value.
1A12:05	SubIndex 005	UINT32	RO	0x68FD:00, 32	Map digital inputs.
1A12:06	SubIndex 006	UINT32	RO	0x5024:02, 8	ENC status.

Index 1A14 Y Axis TX PDO Mapping4

Index	Name	Data type	Flags	Default	Description
1A14:0	Y Axis TxPdoMapping4	UINT8	RO	> 5 <	TX PDO Mapping for PP.
1A14:01	SubIndex 001	UINT32	RO	0x6841:00, 16	Map status word.
1A14:02	SubIndex 002	UINT32	RO	0x6864:00, 32	Map actual position.
1A14:03	SubIndex 003	UINT32	RO	0x68E4:01, 32	Map additional position actual value.
1A14:04	SubIndex 004	UINT32	RO	0x68FD:00, 32	Map digital inputs.
1A14:05	SubIndex 005	UINT32	RO	0x5024:02, 8	ENC status.

Index 1A20 Z Axis TX PDO Mapping0

Index	Name	Data type	Flags	Default	Description
1A20:0	Z Axis TxPdoMapping0	UINT8	RO	> 7 <	TX PDO Mapping for CSP, CSV.
1A20:01	SubIndex 001	UINT32	RO	0x7041:00, 16	Map status word.
1A20:02	SubIndex 002	UINT32	RO	0x7064:00, 32	Map actual position.
1A20:03	SubIndex 003	UINT32	RO	0x706C:00, 32	Map actual velocity.
1A20:04	SubIndex 004	UINT32	RO	0x70E4:01, 32	Map additional position actual value.
1A20:05	SubIndex 005	UINT32	RO	0x70FD:00, 32	Map digital inputs.
1A20:06	SubIndex 006	UINT32	RO	0x7061:00, 8	Map mode of operation display.
1A20:07	SubIndex 007	UINT32	RO	0x5024:03, 8	ENC status.

Index 1A21 Z Axis TX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1A21:0	Z Axis TxPdoMapping1	UINT8	RO	> 5 <	TX PDO Mapping for CSP.
1A21:01	SubIndex 001	UINT32	RO	0x7041:00, 16	Map status word.
1A21:02	SubIndex 002	UINT32	RO	0x7064:00, 32	Map actual position.
1A21:03	SubIndex 003	UINT32	RO	0x70E4:01, 32	Map additional position actual value.
1A21:04	SubIndex 004	UINT32	RO	0x70FD:00, 32	Map digital inputs.
1A21:05	SubIndex 005	UINT32	RO	0x5024:03, 8	ENC status.

Index 1A22 Z Axis TX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1A22:0	Z Axis TxPdoMapping2	UINT8	RO	> 6 <	TX PDO Mapping for CSV.
1A22:01	SubIndex 001	UINT32	RO	0x7041:00, 16	Map control word.
1A22:02	SubIndex 002	UINT32	RO	0x7064:00, 32	Map actual position.
1A22:03	SubIndex 003	UINT32	RO	0x706C:00, 32	Map actual velocity.
1A22:04	SubIndex 004	UINT32	RO	0x70E4:01, 32	Map additional position actual value.
1A22:05	SubIndex 005	UINT32	RO	0x70FD:00, 32	Map digital inputs.
1A22:06	SubIndex 006	UINT32	RO	0x5024:03, 8	ENC status.

Index 1A24 Z Axis TX PDO Mapping4

Index	Name	Data type	Flags	Default	Description
1A24:0	Z Axis TxPdoMapping4	UINT8	RO	> 5 <	TX PDO Mapping for PP.
1A24:01	SubIndex 001	UINT32	RO	0x7041:00, 16	Map status word.
1A24:02	SubIndex 002	UINT32	RO	0x7064:00, 32	Map actual position.
1A24:03	SubIndex 003	UINT32	RO	0x70E4:01, 32	Map additional position actual value.
1A24:04	SubIndex 004	UINT32	RO	0x70FD:00, 32	Map digital inputs.
1A24:05	SubIndex 005	UINT32	RO	0x5024:03, 8	ENC status.

Index 1A30 G code TX PDO Mapping0

Index	Name	Data type	Flags	Default	Description
1A30:0	G code TxPdoMapping0	UINT8	RO	> 14 <	TX PDO Mapping for G-code mode.
1A30:01	SubIndex 001	UINT32	RO	0x50E0:00, 16	Map G-code statusword.
1A30:02	SubIndex 002	UINT32	RO	0x50E1:00, 8	Map machine status.
1A30:03	SubIndex 003	UINT32	RO	0x50E2:01, 64	Map actual position of X axis.
1A30:04	SubIndex 004	UINT32	RO	0x50E2:02, 64	Map actual position of Y axis.
1A30:05	SubIndex 005	UINT32	RO	0x50E2:03, 64	Map actual position of Z axis.
1A30:06	SubIndex 006	UINT32	RO	0x60E4:01, 32	Map additional position actual value of M1 axis.
1A30:07	SubIndex 007	UINT32	RO	0x68E4:01, 32	Map additional position actual value of M2 axis.
1A30:08	SubIndex 008	UINT32	RO	0x70E4:01, 32	Map additional position actual value of M3 axis.
1A30:09	SubIndex 009	UINT32	RO	0x60FD:00, 32	Map digital inputs of M1.
1A30:0A	SubIndex 010	UINT32	RO	0x68FD:00, 32	Map digital inputs of M2.
1A30:0B	SubIndex 011	UINT32	RO	0x70FD:00, 32	Map digital inputs of M3.
1A30:0C	SubIndex 012	UINT32	RO	0x5024:01, 8	Map ENC status of M1.
1A30:0D	SubIndex 013	UINT32	RO	0x5024:02, 8	Map ENC status of M2.
1A30:0E	SubIndex 014	UINT32	RO	0x5024:03, 8	Map ENC status of M3.

Index 1A31 G code TX PDO Mapping1

Index	Name	Data type	Flags	Default	Description
1A31:0	G code TxPdoMapping1	UINT8	RO	> 1 <	Additional TX PDO Mapping for G-code mode.
1A31:01	SubIndex 001	UINT32	RO	0x50E3:00, 8	Map limit inputs.

Index 1A32 G code TX PDO Mapping2

Index	Name	Data type	Flags	Default	Description
1A32:0	G code TxPdoMapping2	UINT8	RO	> 1 <	For TX PDO mapping number is less than 3.
1A32:01	SubIndex 001	UINT32	RO	0x0000:00, 0	Empty

6.2.4 Sync Manager Objects

Index 1C00 Sync manager type

Index	Name	Data type	Flags	Default
1C00:0	Sync manager type	UINT8	RO	> 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 1C12 SM2 assignment

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

Index	Name	Data type	Flags	Default	
1C12:0	RX PDO Assign	UINT8	RO	> 3 <	
				CiA-402 Mode	G-code Mode
1C12:01	SubIndex 01	UINT16	RO, wr_preop	0x1600	0x1630
1C12:02	SubIndex 02	UINT16	RO, wr_preop	0x1610	0x1632
1C12:03	SubIndex 03	UINT16	RO, wr_preop	0x1620	0x1632

Index 1C13 SM3 assignment

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

Index	Name	Data type	Flags	Default	
0x1C13:00	TX PDO Assign	UINT8	RO	> 3 <	
				CiA-402 Mode	G-code Mode
0x1C13:01	SubIndex 01	UINT16	RO, wr_preop	0x1A00	0x1A30
0x1C13:02	SubIndex 02	UINT16	RO, wr_preop	0x1A10	0x1A31
0x1C13:03	SubIndex 03	UINT16	RO, wr_preop	0x1A20	0x1A32

Index 1C32 SM Output Parameters

Index	Name	Data type	Flags	Default
1C32:0	SM output parameter	UINT8	RO	> 3 <
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	0x0001E848 (125000)
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 1C33 SM input parameter

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	0x0001E848 (125000)
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

6.2.5 Manufacturer Objects

Index 0x5xxn Manufacturer 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

6.2.6 Motor Objects

Index 0x5010 Pulse Period

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

Index 0x5011 Motor Pulse

Position distance of motor full rotation. If set 16 micro-steps per full step and use stepper motor features a 1.8° step angle, Motor Pulse should be set to 360/1.8*16 = 3200.

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

MP3S 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)

List of objects per group for saving:

Common	CiA-402		G-code
0x5020	0x607C	0x6880	0x50D3
0x5021	0x607D	0x688C	0x50D4
0x5022	0x607E	0x707C	0x50D5
0x5023	0x6080	0x707D	0x50FD
	0x608C	0x707E	0x50FE
	0x687C	0x7080	0x50DB
	0x687D	0x708C	0x50DC
	0x687E	0x5011	

Index 0x5013 Set Actual POS

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

Index 0x5020 ENC Mode

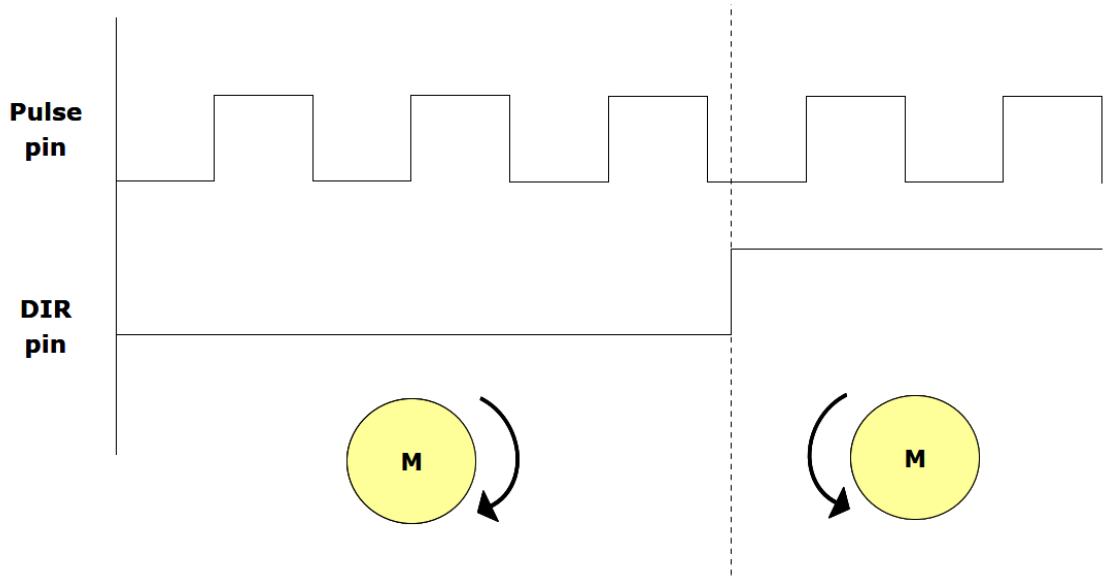
Additional position encoder value (0x60E4/0x68E4 /0x70E4) is updated via encoder interface (ENC_1/ ENC_2/ENC_3). Input signal mode of encoder interface can be set by ENC Mode 0x5020.

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)

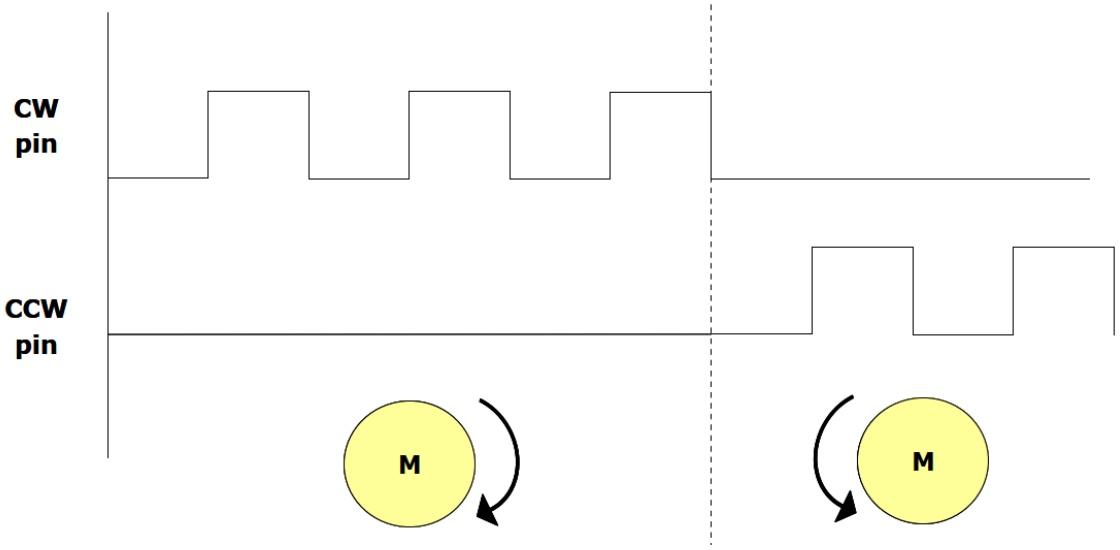
Input signal waveforms:

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

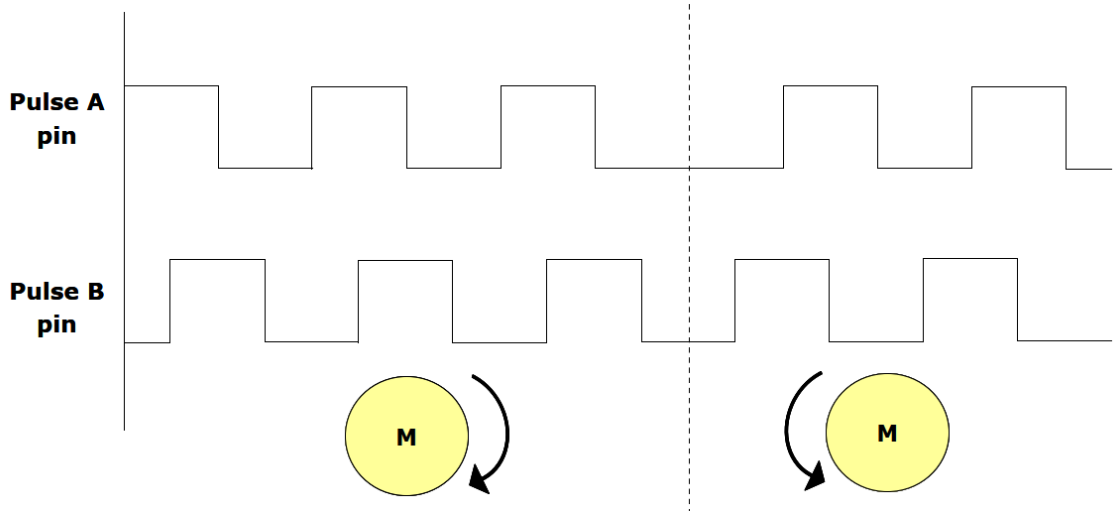
MODE_STEP_DIR



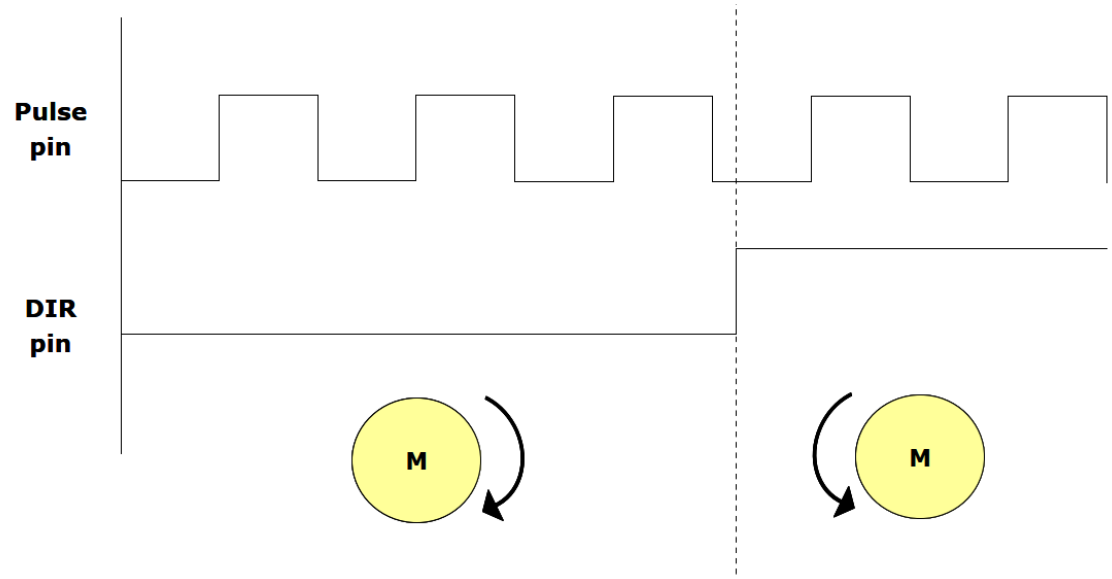
MODE_CWCCW



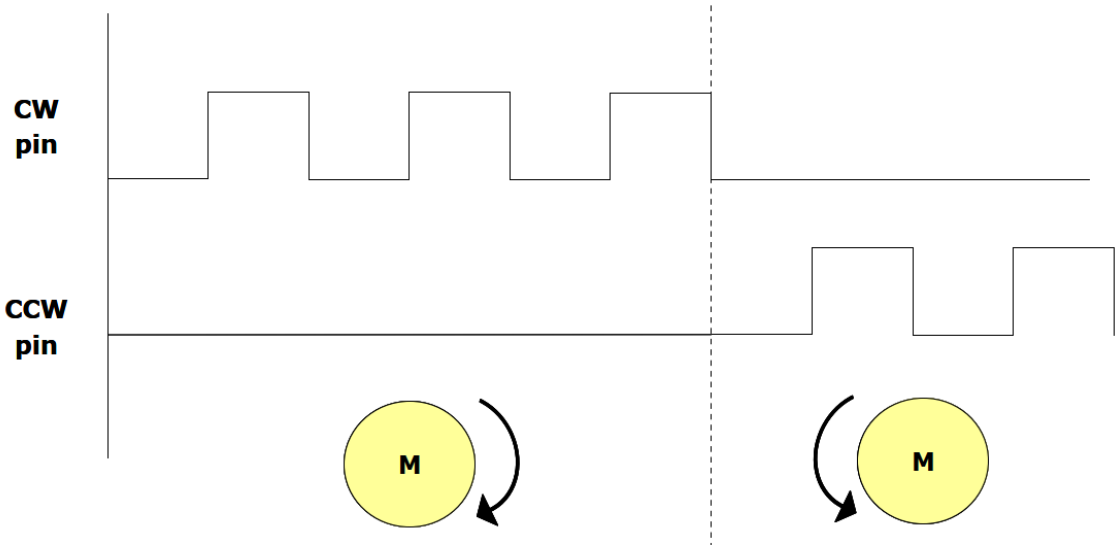
MODE_AB_PHASE



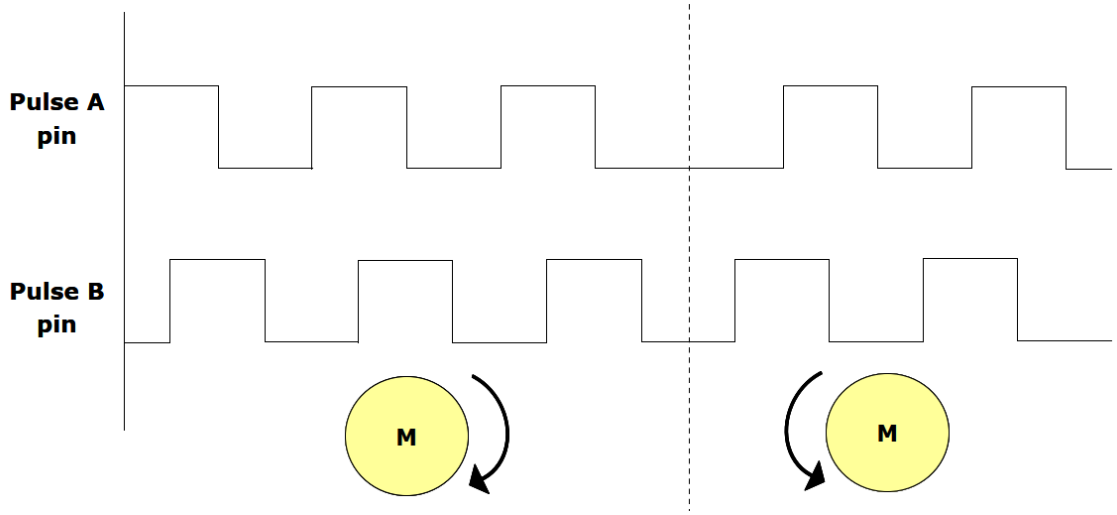
MODE_STEP_DIR_x2



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

Set signal polarity for pin A, B and Z, such as: Set Pol A to 1, which configure the system to interpret the low level for pin A as signal 0 and high level as 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		
		Pol Z	Pol B	Pol A

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

Index 0x5023 ENC Range

Set an upper bound value of encoder counter. When the counter value increase 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

- When device reset encoder value to 0 due to zero index, IDX-Reset bit would be set to 1.
- When the encoder value increase 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.

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

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

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 0x5030 Home Direction

Set homing direction.

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

Bit 0, 1, and 2 set the direction of finding the Home Limit.

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 0x50C1 G54 Work Offset

Read/Write work offset of G54.

- 0x50C1.1: G54 X-axis coordinate.
- 0x50C1.2: G54 Y-axis coordinate.
- 0x50C1.3: G54 Z-axis coordinate.

Index	Name	Data type	Flags	Default
50C1:0	G54_WorkOffset	-	-	-
50C1:01	X	REAL64	RW	0
50C1:02	Y	REAL64	RW	0
50C1:03	Z	REAL64	RW	0

Index 0x50C2 G55 Work Offset

Read/Write work offset of G55.

- 0x50C2.1: G55 X-axis coordinate.
- 0x50C2.2: G55 Y-axis coordinate.
- 0x50C2.3: G55 Z-axis coordinate.

Index	Name	Data type	Flags	Default
50C2:0	G55_WorkOffset	-	-	-
50C2:01	X	REAL64	RW	0
50C2:02	Y	REAL64	RW	0
50C2:03	Z	REAL64	RW	0

Index 0x50C3 G56 Work Offset

Read/Write work offset of G56.

- 0x50C3.1: G56 X-axis coordinate.
- 0x50C3.2: G56 Y-axis coordinate.
- 0x50C3.3: G56 Z-axis coordinate.

Index	Name	Data type	Flags	Default
50C3:0	G56_WorkOffset	-	-	-
50C3:01	X	REAL64	RW	0
50C3:02	Y	REAL64	RW	0
50C3:03	Z	REAL64	RW	0

Index 0x50C4 G57 Work Offset

Read/Write work offset of G57.

- 0x50C4.1: G57 X-axis coordinate.
- 0x50C4.2: G57 Y-axis coordinate.
- 0x50C4.3: G57 Z-axis coordinate.

Index	Name	Data type	Flags	Default
50C4:0	G57_WorkOffset	-	-	-
50C4:01	X	REAL64	RW	0
50C4:02	Y	REAL64	RW	0
50C4:03	Z	REAL64	RW	0

Index 0x50C5 G58 Work Offset

Read/Write work offset of G58.

- 0x50C5.1: G58 X-axis coordinate.
- 0x50C5.2: G58 Y-axis coordinate.
- 0x50C5.3: G58 Z-axis coordinate.

Index	Name	Data type	Flags	Default
50C5:0	G58_WorkOffset	-	-	-
50C5:01	X	REAL64	RW	0
50C5:02	Y	REAL64	RW	0
50C5:03	Z	REAL64	RW	0

Index 0x50C6 G59 Work Offset

Read/Write work offset of G59.

- 0x50C6.1: G59 X-axis coordinate.
- 0x50C6.2: G59 Y-axis coordinate.
- 0x50C6.3: G59 Z-axis coordinate.

Index	Name	Data type	Flags	Default
50C6:0	G59_WorkOffset	-	-	-
50C6:01	X	REAL64	RW	0
50C6:02	Y	REAL64	RW	0
50C6:03	Z	REAL64	RW	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.

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	Flags	Default
50D1	G Code string	ARRAY [0..25] OF BYTE	RW	-

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

Set maximum value of feed-rate. Velocity unit is mm per second.

- 0x50D3.1: X axis, object type is REAL64
- 0x50D3.2: Y axis
- 0x50D3.3: Z axis

Index	Name	Data type	Flags	Default
50D3:0	Max Velocity	-	-	-
50D3:01	X axis	REAL64	RW	---
50D3:02	Y axis	REAL64	RW	---
50D3:03	Z axis	REAL64	RW	---

Index 0x50D4 Max Acceleration

Set maximum acceleration of step motor. The XYZ axis applies the same value, object unit is MPS (mm per second) per second.

Index	Name	Data type	Flags	Default
50D4	Max Acceleration	REAL64	RW	-

Index 0x50D5 Position Limit

If position limit (see 0x50DD) is enabled, when motor reaches position limit, motor will stay at limit position.

Index	Name	Data type	Flags	Default
50D5:0	Position Limit	-	-	-
50D5:01	Min of X axis	REAL64	RW	---
50D5:02	Max of X axis	REAL64	RW	---
50D5:03	Min of Y axis	REAL64	RW	---
50D5:04	Max of Y axis	REAL64	RW	---
50D5:05	Min of Z axis	REAL64	RW	---
50D5:06	Max of Z axis	REAL64	RW	---

Index 0x50D6 Homing Speed

Set homing speed of XYZ axes. Unit of speed is mm per minute.

Index	Name	Data type	Flags	Default
50D6:0	Homing Speed	UINT8	RO	> 0 <
50D6:01	X axis	REAL64	RW	---
50D6:02	Y axis	REAL64	RW	---
50D6:03	Z axis	REAL64	RW	---

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	RO, wr_op	0x00 (0)

Object used to control the state of G Code Controller. The object type is UINT8, 0: Servo Off, 1: Servo On. When Servo Off, it cannot receive G code and cannot execute and return to Home.

[7 - 4]	3	2	1	0
Reserved	Reserved	Reserved	Reserved	Servo On/Off

Index 0x50D8 Home

Set 0x50D8 will execute Homing. Motor will rotate in the negative direction 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	RO, wr_op	0x00 (0)

When the Home bit switches from 0 to 1, Home will be executed. If the synchronization mode of ECAT is Free-RUN, Emergency Stop cannot be received during Homing. Emergency Stop can be received normally in SyncManager and DC Mode.

[7 - 4]	3	2	1	0
Reserved	Reserved	Reserved	Home	Reserved

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	RO, wr_op	0x00 (0)

Object used to control the status of G Code Controller. The object type is UINT8. Changing the object to 4 will execute the software EmgStop. Restoring the object to 0 will not clear EmgStop. Clear EmgStop must be executed.

[7 - 4]	3	2	1	0
Reserved	Reserved	Emergency Stop	Reserved	Reserved

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	WO	-

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	Flags	Default
50DB	Default Feedrate	REAL64	RW	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. Unit of 0x50DC is mm per minute.

Index	Name	Data type	Flags	Default
50DC:0	Default Homing SPD	UINT8	RO	> 0 <
50DC:01	X axis	REAL64	RW	---
50DC:02	Y axis	REAL64	RW	---
50DC:03	Z axis	REAL64	RW	---

Index 0x50DD Enable Position Limit

If position limit is enabled, when motor reaches position limit(0x50D5), motor will stay at limit position.

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

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

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

- **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 ocde:
  - 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 Master, 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 When 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

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

[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	Definition	Description
0	Servo On	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 EmergencyStop state.

When the controller is not in Emergency Stop, the user can switch the machine status of the controller through objects 0x50D7 ~ 0x50D9 and changing the hardware EmgStop input. The following table shows the machines corresponding to objects 0x50D7 ~ 0x50D9 and the hardware EmgStop input pins. Status, the X in the table indicates that the object can be any value:

Machine State	EmgStop Input	0x50D9	0x50D8	0x50D7
Hardware Emergency Stop	0	X	X	X
Software Emergency Stop	1	4	X	X
Servo Off	1	0	X	0
Servo On	1	0	0	1
Homing	1	0	2	1

Index 0x50E2 Actual Position

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

Index	Name	Data type	Flags	Default
50E2:0	Actual Position	UINT8	RO	> 0 <
50E2:01	X axis	REAL64	RO	---
50E2:02	Y axis	REAL64	RO	---
50E2:03	Z axis	REAL64	RO	---

Index 0x50E3 Limit Inputs

Display the pin status of XYZ three-axis Limit inputs.

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

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

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	RO, wr_preop	0x00 (0)

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, MP3S receives G-code and interpolates the 3-axis motor rotation.
If the value of 0x50FC is not 1, MP3S is a 3-axis CiA-402 device.

Index	Name	Data type	Flags	Default
50FC	Device Profile	UINT8	RO, wr_preop	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

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

G-code distance unit of MP3S is mm, and motor rotation unit is motor microstep. Object 0x50FE is the ratio that converts mm to microsteps. For example, if 0x50FE:01 is 33.6, "G1 X13.0" will cause X-axis motor to rotate 436 microsteps.

Index	Name	Data type	Flags	Default
50FE:0	Pulse of per millimeter	-	-	-
50FE:01	X axis	REAL64	RO, wr_preop	---
50FE:02	Y axis	REAL64	RO, wr_preop	---
50FE:03	Z axis	REAL64	RO, wr_preop	---

Index 0x50FF G Code command

QEC-RXXMP3S supports receiving G-code via mailbox.
Writing G-code string to object 0x50FF instructs QEC-RXXMP3S 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)	RO, wr_op	-

6.2.7 Especial Objects (0x6000-0xFFFF)

For especial objects description.

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

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

Index 0x603F Error Code

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

Index 0x6040 Controlword

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

Index 0x6041 Statusword

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

Index 0x605A Quick stop option code

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

Index 0x605B Shutdown option code

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

Index 0x605E Fault reaction option code

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

Index 0x6060 Modes of operation

Mode Write.

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

Index 0x6061 Modes of operation display

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

Index 0x6064 Position actual value

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

Index 0x606C Velocity Actual Value

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

Index 0x607A Target position

Mandatory if CSP is supported.

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

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 CSP 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

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

Can’t be zero.

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

Index 0x6083 Profile acceleration

Can’t be zero.

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

Index 0x6085 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

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

Index 0x608C Velocity dimension index

Save to EEPROM.

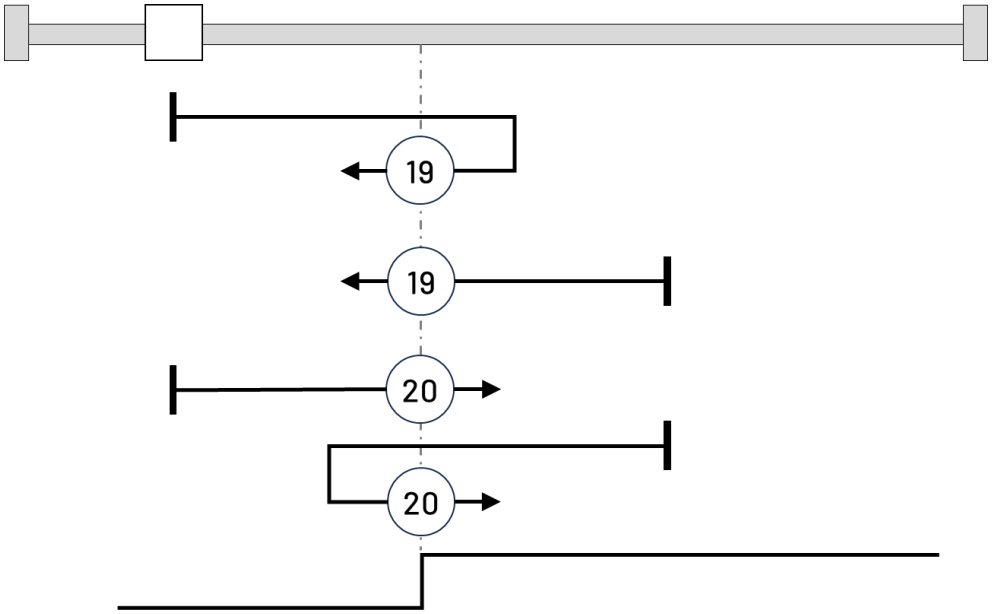
Index	Name	Data type	Flags	Default
608C	Velocity dimension index	UINT8	RW	0xA4(164)

Index 0x6098 Homing method

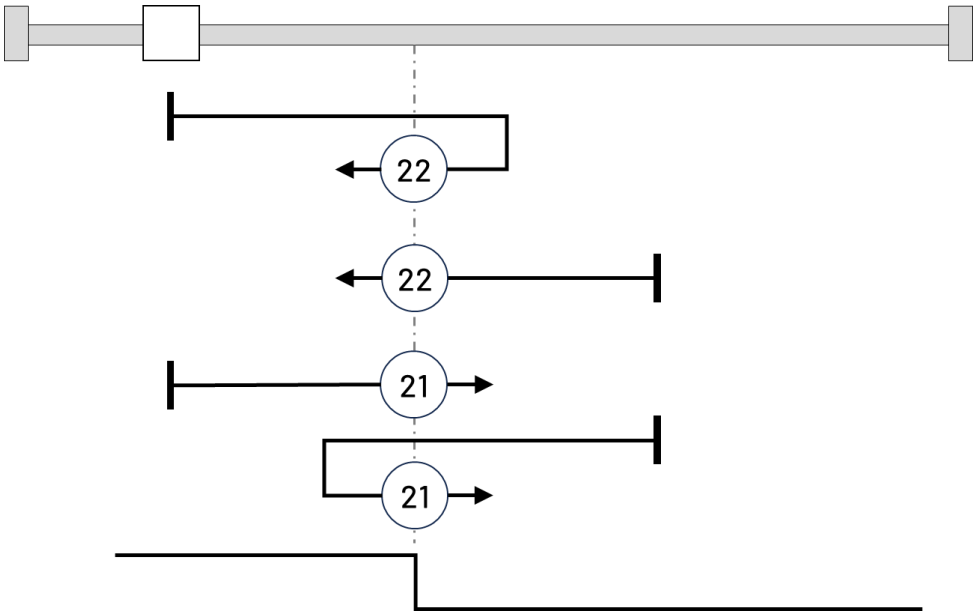
Specify the return-to-origin method.
Possible values are: 19, 20, 21, 22. Set other values and perform a return to origin.
If this occurs, a homing error occurs, and 1 is returned to Bit13: Homing error of statusword.

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:



Index 0x6099 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

Can't be zero.

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

Index 0x60B0 Position offset

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

Index 0x60B1 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

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

Index 0x60BA 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

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

Index 0x60C2 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

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

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

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

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

Index 0x60FD Digital inputs

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

Index 0x60FF Target velocity

Mandatory if PV or CSV is supported.

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

Index 0x6502 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 Control word

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

Index 0x6841 Status word

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

Index 0x6860 Modes of operation

Mode write.

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

Index 0x6861 Modes of operation display

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

Index 0x6864 Position actual value

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

Index 0x686C Velocity Actual Value

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

Index 0x687A Target position

Mandatory if CSP is supported.

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

Index 0x687C Home offset

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

Index 0x687D Software position limit

Set Position Limit. Recommended if 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

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

Can't be zero.

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

Index 0x6883 Profile acceleration

Can't be zero.

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

Index 0x688B Velocity notation index

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

Index 0x688C Velocity dimension index

Save to EEPROM.

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

Index 0x6898 Homing method

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

Index 0x6899 Homing 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

Can't be zero.

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

Index 0x68B0 Position offset

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

Index 0x68B1 Velocity offset

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

Index 0x68B8 Touch probe function

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

Index 0x68B9 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

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

Index 0x68BB 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

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

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

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

Index 0x68E4 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

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

Index 0x68FD Digital inputs

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

Index 0x68FF Target velocity

Mandatory if PV or CSV is supported.

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

Index 0x6D02 Supported drive modes

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

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

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

Index 0x7040 Control word

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

Index 0x7041 Status word

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

Index 0x7060 Modes of operation

Mode Write.

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

Index 0x7061 Modes of operation display

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

Index 0x7064 Position actual value

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

Index 0x706C Velocity Actual Value

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

Index 0x707A Target position

Mandatory if CSP 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 CSP 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

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

Can't be zero.

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

Index 0x7083 Profile acceleration

Can't be zero.

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

Index 0x708B Velocity notation index

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

Index 0x708C Velocity dimension index

Save to EEPROM.

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

Index 0x7098 Homing method

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

Index 0x7099 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

Can't be zero.

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

Index 0x70B0 Position offset

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

Index 0x70B1 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

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

Index 0x70BA 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

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

Index 0x70C2 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

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

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

Index 0x70E4 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

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

Index 0x70FD Digital inputs

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

Index 0x70FF Target velocity

Mandatory if PV or CSB is supported.

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

Index 0x7502 Supported drive modes

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

Index 0xAxxx Device Object (0xA000 – 0xAFFF)

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

Index 0xA000 FoE Transmission Status

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

Index 0xFxxx Device Object (0xF000 – 0xFFFF)

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

Index 0xF000 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.