OMRON

CK3C-Series

Programmable Multi-Axis Controller

User Manual



O054-E1-03

Note

- All rights reserved. No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form, or by any means, mechanical, electronic, photocopying, recording, or otherwise, without the prior written permission of OMRON.
- No patent liability is assumed with respect to the use of the information contained herein. Moreover, because OMRON is constantly striving to improve its high-quality products, the information contained in this manual is subject to change without notice.
- Every precaution has been taken in the preparation of this manual. Nevertheless, OMRON assumes no responsibility for errors or omissions.
  Neither is any liability assumed for damages resulting from the use of the information contained in this publication.

Trademarks

- Microsoft, Windows, Excel, and Visual Basic are either registered trademarks or trademarks of Microsoft Corporation in the United States and other countries.

  Other company names and product names in this document are the trademarks or registered trademarks of their respective companies.

Copyrights

- Microsoft product screen shots reprinted with permission from Microsoft Corporation.

Introduction

Thank you for purchasing the CK3C controller.

This manual contains the necessary information for proper mounting, installation, wiring, configuration, and troubleshooting of the CK3C-series.

Intended Audience

This manual is intended for the following personnel, who must also have knowledge of electrical systems, such as an electrical engineer or equivalent:

- Personnel in charge of introducing FA systems.
- Personnel in charge of designing FA systems.
- Personnel in charge of installing and maintaining FA systems.
- Personnel in charge of managing FA systems and facilities.

Notice

This User's Manual contains information that is critical for the correct use of the CK3C controller.

Read this Manual in its entirety to gain full understanding of the proper method to use the CK3C controller. Keep this manual in a convenient place for quick reference, and make sure that it is provided to the end-user.

Applicable Products

The applicable part numbers for this manual are:

- CK3C-AX□□□□□□



Additional Information

Please see Part Number Designation section for full list of options

Technical Support

Phone: Call 1-800-556-6766, and follow the prompts for technical support on PMAC products.

Email: **ODT-Support@omron.com**

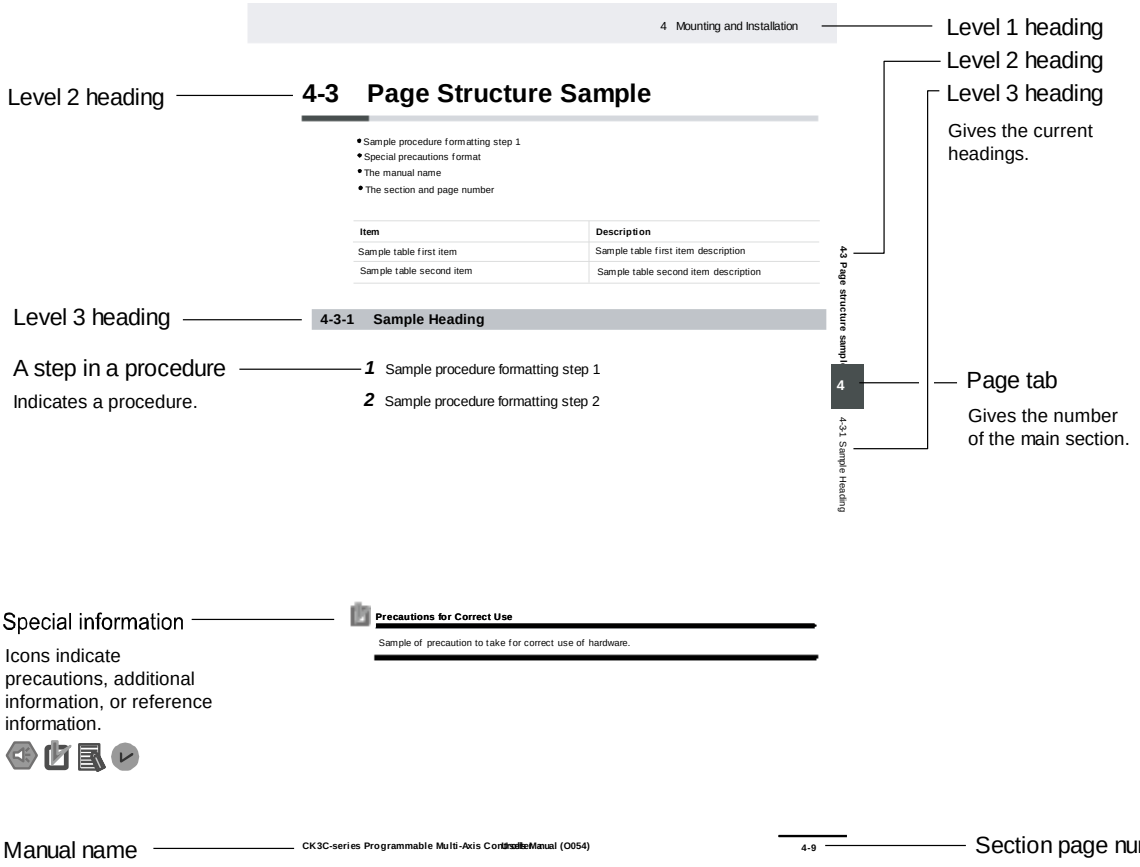
Alternate Email: **ia.techsupport@omron.com**

Type "Delta Tau Support" or "PMAC Support" in the subject line.

Manual Structure

Page Structure

The following page structure is used in this manual.



Note this page illustration is only a sample. Its content is not information material, and it does not appear in the manual.

PMAC Script

This manual contains PMAC (language) script samples that can be used in the IDE software. These script samples are enclosed in the following format.

```
GLOBAL MyVar1         // Global variable 1
GLOBAL MyVar2 = 0     // Global variable 2
```

Special Information

Special information in this manual is classified as follow:



Precautions for Safe Use

Precautions on what to do and what not to do to ensure safe usage of the product.



Precautions for Correct Use

Precautions on what to do and what not to do to ensure proper operation and performance.



Additional Information

Additional information to read as required. This information is provided to increase understanding or make operation easier.

Sections in this Manual

1 Introduction to the CK3C
2 Specifications
3 Mounting and Installation
4 Wiring
5 Software Configuration
6 Troubleshooting
A Appendices

CONTENTS

Introduction **1**
Intended Audience 1
Notice 1
Applicable Products 1
Manual Structure **2**
Page Structure 2
PMAC Script 2
Special Information 3
Sections in this Manual **5**
CONTENTS **6**
Terms and Conditions Agreement **10**
Warranty, Limitations of Liability 10
Application Considerations 11
Disclaimers 12
Safety Precautions **13**
Definition of Precautionary Information 13
Definition of Symbols 13
Precautionary Information 14
Warnings **15**
Cautions **18**
Precautions for Safe Use **19**
Precautions for Correct Use **21**
Items to Check after Unpacking **22**
Location of Nameplate 22
Accessories 23
Regulations and Standards **24**
Conformance to EU/CE Directives 24
Conformance to UL Directives 24
Conformance to UKCA Standards 24
Conformance to KC Standards 25
Versions **26**
Hardware Version 26
Firmware Version 26
Related Manuals **27**
Terminology **29**
Manual Revision History **30**

Section 1 Introduction to the CK3C 1-1

1-1 Outline **1-2**
1-1-1 Controller Features 1-2
1-1-2 Typical Configuration 1-4
1-1-3 Part Names, Locations, and Functions 1-5

1-2 Operating Procedure 1-6
1-2-1 Preparation 1-6
1-2-2 Safety 1-6
1-2-3 Mounting and Wiring 1-6
1-2-4 Safety Test and Power Up 1-6
1-2-5 Controller Settings 1-7
1-2-6 Test Run 1-7

Section 2 Specifications 2-1

2-1 Controller Specifications 2-2
2-1-1 General/Mechanical 2-2
2-1-2 Environmental 2-4
2-1-3 Electrical 2-5

2-2 Part Number Designation 2-6
2-2-1 Controller Part Number 2-6
2-2-2 Stackable Accessory Part Numbers 2-7

Section 3 Mounting and Installation 3-1

3-1 External and Mounting Dimensions 3-2
3-1-1 Isometric View 3-2
3-1-2 CK3C Dimensions and Mounting 3-3

3-2 Installation 3-4
3-2-1 Ventilation 3-5
3-2-2 Panel Clearance/Mounting Direction 3-5
3-2-3 Installation Conditions 3-5
3-2-4 Keeping Foreign Objects Out of Units 3-6

Section 4 Wiring 4-1

4-1 DIP Switch configurations 4-3
4-1-1 Serial Encoder Interface Port Selection 4-3
4-1-2 General Purpose I/O Buffer Direction Selection 4-4

4-2 Power Supply Input 4-5
4-2-1 Connector Pinout 4-5
4-2-2 Wiring the Terminal 4-6

4-3 Machine Connector (J3) 4-9
4-3-1 Circuit Specifications 4-9
4-3-2 Connector Pinout 4-13
4-3-3 Wiring a Digital Quadrature Encoder 4-16
4-3-4 Wiring an Analog Input Servo Drive 4-17
4-3-5 Wiring the Analog Inputs 4-18

4-4 Machine Connector (J4) 4-19
4-4-1 Circuit Specifications 4-19
4-4-2 Connector Pinout 4-22
4-4-3 Wiring a Stepper Drive 4-24
4-4-4 Wiring the PMAC Reset Input 4-25
4-4-5 Wiring the Limits and Flags 4-25
4-4-6 Wiring the Compare Equal Outputs 4-26
4-4-7 Wiring an External Safety Circuit 4-26

4-5 Machine Connector (J7) 4-27

4-5-1 Circuit Specifications 4-27
4-5-2 Connector Pinout 4-30
4-5-3 Wiring a Hall Sensor 4-31
4-5-4 Wiring a Serial Encoder 4-32
4-5-5 Wiring the Brake Outputs 4-33
4-5-6 Wiring the Analog Inputs 4-33

4-6 GPIO Port (J8, J9) 4-34

4-6-1 Circuit Specifications 4-34
4-6-2 Connector Pinout (J8) 4-36
4-6-3 Connector Pinout (J9) 4-37
4-6-4 Wiring a General Purpose Input 4-39
4-6-5 Wiring a General Purpose Output 4-39

4-7 Handwheel/Pulse-Dir Port (J10) 4-40

4-7-1 Circuit Specifications 4-40
4-7-2 Connector Pinout 4-43
4-7-3 Wiring a Stepper Drive 4-44
4-7-4 Wiring a Quadrature Encoder 4-45
4-7-5 Wiring the Analog Output 4-45

4-8 Diagnostic Port 4-46

4-9 EtherCAT™/Ethernet Port 4-47

4-10 Ethernet Communications Port 4-48

4-11 USB Host Port 4-49

4-11-1 Circuit Specifications 4-49

4-12 Serial Port 4-50

4-13 Wiring Conforming to EMC/EU Directives 4-51

4-14 Selecting Peripheral Components 4-52

4-14-1 Recommended Power Supply 4-52
4-14-2 Alternate Power Supply Solution 4-52

Section 5 Software Configuration 5-1

5-1 Outline of Software Configuration 5-2

5-1-1 Summary 5-2
5-1-2 Intended Users 5-3
5-1-3 Intended Hardware 5-3
5-1-4 Gate3 Alias Name 5-3

5-2 CK3C Basic Configuration 5-4

5-2-1 System and Gate3 Settings 5-4
5-2-2 Quadrature Encoder Configuration 5-5
5-2-3 Handwheel Encoder Configuration 5-6
5-2-4 Serial Encoder Configuration 5-7
5-2-5 Setting up the Analog (ADC) Inputs 5-34
5-2-6 Limit, Flag, and Brake [Channel 0-3] Structure Elements 5-36
5-2-7 General Purpose IO (GPIO) Configuration 5-37

Section 6 Troubleshooting 6-1

6-1 LED Indicators 6-2

6-1-1 Controller Operation Status Indicators 6-2
6-1-2 Ethernet / EtherCAT Indicators 6-2

6-2 Error Processing 6-3
6-2-1 Classification of Errors 6-3
6-2-2 Fatal Errors 6-3

Appendices A-1

A-1 CK3C Project Management Example A-2
A-1-1 Creating an IDE Project A-2
A-1-2 Basic Optimization, System and Gate Settings A-6
A-1-3 Encoder On-going Position, Verification A-7
A-1-4 Motor Setup and Commissioning A-11
A-1-5 Reversing Motor Operation Direction A-36

A-2 Sample Motor Setup Examples A-37
A-2-1 Filtered PWM Output with Quadrature Encoder A-38
A-2-2 Direct PWM Output with Quadrature Encoder (ACC-8FS) A-40
A-2-3 Direct PWM Output with Sinusoidal Encoder (ACC-8FS, ACC-51S) A-42
A-2-4 Direct PWM Output with BiSS Encoder (ACC-8FS, ACC-84S) A-44

Terms and Conditions Agreement

Warranty, Limitations of Liability

Warranties

- **Exclusive Warranty**

  Omron's exclusive warranty is that the Products will be free from defects in materials and workmanship for a period of twelve months from the date of sale by Omron (or such other period expressed in writing by Omron). Omron disclaims all other warranties, express or implied.

- **Limitations**

  OMRON MAKES NO WARRANTY OR REPRESENTATION, EXPRESS OR IMPLIED, ABOUT NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE OF THE PRODUCTS. BUYER ACKNOWLEDGES THAT IT ALONE HAS DETERMINED THAT THE PRODUCTS WILL SUITABLY MEET THE REQUIREMENTS OF THEIR INTENDED USE.
  Omron further disclaims all warranties and responsibility of any type for claims or expenses based on infringement by the Products or otherwise of any intellectual property right.

- **Buyer Remedy**

  Omron's sole obligation hereunder shall be, at Omron's election, to (i) replace (in the form originally shipped with Buyer responsible for labor charges for removal or replacement thereof) the non-complying Product, (ii) repair the non-complying Product, or (iii) repay or credit Buyer an amount equal to the purchase price of the non-complying Product; provided that in no event shall Omron be responsible for warranty, repair, indemnity or any other claims or expenses regarding the Products unless Omron's analysis confirms that the Products were properly handled, stored, installed and maintained and not subject to contamination, abuse, misuse or inappropriate modification. Return of any Products by Buyer must be approved in writing by Omron before shipment. Omron Companies shall not be liable for the suitability or unsuitability or the results from the use of Products in combination with any electrical or electronic components, circuits, system assemblies or any other materials or substances or environments. Any advice, recommendations or information given orally or in writing, are not to be construed as an amendment or addition to the above warranty.

  See http://www.omron.com/global/ or contact your Omron representative for published information.

Limitation on Liability; Etc

OMRON COMPANIES SHALL NOT BE LIABLE FOR SPECIAL, INDIRECT, INCIDENTAL, OR CONSEQUENTIAL DAMAGES, LOSS OF PROFITS OR PRODUCTION OR COMMERCIAL LOSS IN ANY WAY CONNECTED WITH THE PRODUCTS, WHETHER SUCH CLAIM IS BASED IN CONTRACT, WARRANTY, NEGLIGENCE OR STRICT LIABILITY.

Further, in no event shall liability of Omron Companies exceed the individual price of the Product on which liability is asserted.

Application Considerations

Suitability of Use

Omron Companies shall not be responsible for conformity with any standards, codes or regulations which apply to the combination of the Product in the Buyer's application or use of the Product. At Buyer's request, Omron will provide applicable third party certification documents identifying ratings and limitations of use which apply to the Product. This information by itself is not sufficient for a complete determination of the suitability of the Product in combination with the end product, machine, system, or other application or use. Buyer shall be solely responsible for determining appropriateness of the particular Product with respect to Buyer's application, product or system. Buyer shall take application responsibility in all cases.

NEVER USE THE PRODUCT FOR AN APPLICATION INVOLVING SERIOUS RISK TO LIFE OR PROPERTY WITHOUT ENSURING THAT THE SYSTEM AS A WHOLE HAS BEEN DESIGNED TO ADDRESS THE RISKS, AND THAT THE OMRON PRODUCT(S) IS PROPERLY RATED AND INSTALLED FOR THE INTENDED USE WITHIN THE OVERALL EQUIPMENT OR SYSTEM.

Programmable Products

Omron Companies shall not be responsible for the user's programming of a programmable Product, or any consequence thereof.

Security Measures

Anti-virus protection

Install the latest commercial-quality antivirus software on the computer connected to the control system and maintain to keep the software up to date.

Security measures to prevent unauthorized access

Take the following measures to prevent unauthorized access to our products.

- Install physical controls so that only authorized personnel can access control systems and equipment.
- Reduce connections to control systems and equipment via networks to prevent access from untrusted devices.
- Install firewalls to shut down unused communications ports and limit communications hosts and isolate control systems and equipment from the IT network.
- Use a virtual private network (VPN) for remote access to control systems and equipment.
- Adopt multifactor authentication to devices with remote access to control systems and equipment.
- Set strong passwords and change them frequently.
- Scan virus to ensure safety of USB drives or other external storages before connecting them to control systems and equipment.

Data input and output protection

Validate backups and ranges to cope with unintentional modification of input/output data to control systems and equipment.

- Checking the scope of data
- Checking validity of backups and preparing data for restore in case of falsification and abnormalities
- Safety design, such as emergency shutdown and fail-soft operation in case of data tampering and abnormalities

Data recovery

Backup data and keep the data up-to-date periodically to prepare for data loss.
When using an intranet environment through a global address, connecting to an unauthorized terminal such as a SCADA, HMI or to an unauthorized server may result in network security issues such as spoofing and tampering. You must take sufficient measures such as restricting access to the terminal, using a terminal equipped with a secure function, and locking the installation area by yourself.

When constructing an intranet, communication failure may occur due to cable disconnection or the influence of unauthorized network equipment. Take adequate measures, such as restricting physical access to network devices, by means such as locking the installation area.

When using a device equipped with the SD Memory Card function, there is a security risk that a third party may acquire, alter, or replace the files and data in the removable media by removing the removable media or unmounting the removable media. Please take sufficient measures, such as restricting physical access to the Controller or taking appropriate management measures for removable media, by means of locking the installation area, entrance management, etc., by yourself.

Disclaimers

Performance Data

Data presented in Omron Company websites, catalogs and other materials is provided as a guide for the user in determining suitability and does not constitute a warranty. It may represent the result of Omron’s test conditions, and the user must correlate it to actual application requirements. Actual performance is subject to the Omron’s Warranty and Limitations of Liability.

Change in Specifications

Product specifications and accessories may be changed at any time based on improvements and other reasons. It is our practice to change part numbers when published ratings or features are changed, or when significant construction changes are made. However, some specifications of the Product may be changed without any notice. When in doubt, special part numbers may be assigned to fix or establish key specifications for your application. Please consult with your Omron’s representative at any time to confirm actual specifications of purchased Product.

Errors and Omissions

Information presented by Omron Companies has been checked and is believed to be accurate; however, no responsibility is assumed for clerical, typographical or proofreading errors or omissions.

Safety Precautions

To ensure safe and correct use of the CK3C controller, the Safety Precautions section must be read carefully and fully before installation and wiring. This User's Manual must be provided to the end-user, and kept in a readily accessible location for reference.

Definition of Precautionary Information

The safety precautions described in this manual are extremely important for safety. Always read and heed the information provided in all safety precautions.

The following notation is used to provide precautions required to ensure safe usage of the CK3C:

WARNING	Indicates a potentially hazardous situation that, if not avoided, could result in death or serious injury. Additionally, there may be severe property damage.

Definition of Symbols

This manual contains the following symbols:

	The circle and slash symbol indicates operations that you must not do. The specific operation is shown in the circle and explained in text. This example indicates that disassembly is prohibited.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for electric shock.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for fire.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a precaution for burning.
	The triangle symbol indicates precautions (including warnings). The specific operation is shown in the triangle and explained in text. This example indicates a general precaution.
	The filled circle symbol indicates operations that you must do. The specific operation is shown in the circle and explained in text. This example shows a general precaution for something that you must do.

Precautionary Information

Illustrations contained in this manual sometimes depict conditions without covers and safety shields for the purpose of showing the details. When you use this product, be sure to install the covers and shields as specified and use the product according to this manual.

Warnings

Transfer

⚠ WARNING

Always confirm safety at the destination before you transfer a user program, configuration data, or setup data from the Power PMAC IDE.
The devices or machines may perform unexpected operation regardless of the operating mode of the Controller.

After you transfer the user program, the Controller is restarted and communications with the EtherCAT slaves are cut off. During that period, the slave outputs behave according to the slave specifications. The time that communications are cut off depends on the EtherCAT network configuration. Before you transfer the user program, confirm that the system will not be adversely affected.

Installation, Wiring and Maintenance

⚠ WARNING

Observe warnings for proper ground bonding of equipment in the system.

Sensitive signals may require the use of shielded wire or other specifically rated wiring for the application type. Refer to the CK3C User Manual when selecting wire types.

Before applying power after installation, verify that there is no contamination developed in the product. Contamination may cause unpredictable operation of the product.

Machine integrator is responsible to verify that the encoders are programmed to operate in the correct direction prior to engaging motors to spin.

Provide warning in CK3C User's Manual that machine integrator is responsible to provide safe machine shutdown and user isolation in case of watchdog or power cycle condition.

WARNING- THIS MACHINE STARTS AND STOPS AUTOMATICALLY
Sign is expected to be placed on the machine containing this control equipment.

Provide warning in CK3C User's Manual that E-stop and safety interlocks must be verified for operation before commissioning machine and during periodic maintenance.

Operators, maintenance operators and machine integrators must read the Product Manuals and Safety Precaution sheet prior to operating the controller.	
Use of Lockout/Tagout procedures are expected when product is accessed in control cabinets of machinery.	
If a fuse in the product is replaceable, it must be replaced with a fuse of the same rating. Not all fuses in the product are user replaceable.	

Use

⚠ WARNING

Machine integrator is responsible to prepare machinery so that safety zones and physical stops are implemented in a responsible manner. Refer to "HOW TO MAKE YOUR APPLICATION SAFE" in the software reference manual.	
Machine integrator is responsible to insure that external controlling programs (HMIs) are implemented in a responsible manner. Machinery must stop in a manner similar to an E-Stop when external control programs cease to communicate with the controller. Refer to "HOW TO MAKE YOUR APPLICATION SAFE" in the software reference manual.	
Machine integrator is responsible to insure that devices connected to the EtherCAT interfaces are implemented in a responsible manner. EtherCAT endpoints must be configured to shut down in a manner similar to an E-Stop when EtherCAT comms ceases to communicate correctly with the controller. Refer to "EtherCAT Configuration" in the software reference manual.	
Do not attempt to take any unit apart.	
Provide safety measures in external circuits to ensure safety in the system if an abnormality occurs due to malfunction of the system or due to other external factors affecting operation. Not doing so may result in serious accidents due to incorrect operation.	
Emergency stop circuits, interlock circuits, limit circuits, and similar safety measures must be provided in external control circuits.	
Check the user program, servo algorithm, data, and parameter settings for proper execution before you use them for actual operation.	
You must take fail-safe measures to ensure safety in the event of incorrect, missing, or abnormal signals caused by broken signal lines, momentary power interruptions, or other causes.	

Before you start a test run, make sure that the operation parameters are set correctly.	
To ensure safe use of the Controller, correctly make the limit settings for the position, speed, acceleration, jerk, current, and following error, as well as the encoder loss detection.	

Cautions

Installation, Wiring and Maintenance

⚠CAUTION

Rack system must be mounted with sufficient clearance for cooling. Refer to installation instruction in the CK3C User's Manual for specified clearances.



STATIC SENSITIVE DEVICE - HANDLE WITH CARE
1. When removing from Anti-static bag, touch bag to the equipment where the product is being installed.
2. Hold the product by the edges and avoid touching the solder or the components on the board as it is removed from the anti-static bag.
3. Do not place the product on a counter while moving it from the Anti-static bag to the equipment.
4. Electrostatic wrist strap is recommended for handling the product.



Use

⚠CAUTION

To control the motor safely and correctly, the servo algorithm design and gain setting work must be performed by engineers who understand control theories and the specifications of this product.



When you perform a test run, take fail-safe measures and run the motor at a sufficiently low speed to ensure safety.



Before you download a project written in C language, execute the re-initialization command ($$$***).



If you download a validated program to a different product, check the operation of the program again on the product because it may have different settings.



Be careful when disabling or lowering PMAC's built-in safety features such as: fatal following error limit or over travel limit input.



Precautions for Safe Use

If the equipment is used in a manner not specified by Delta Tau Data Systems, the protection provided by the equipment may be impaired.

Installation, Wiring and Maintenance

If the equipment is used in a manner not specified by Delta Tau Data Systems, the protection provided by the equipment may be impaired.

- Do not drop the Controller or expose it to abnormal vibration or shock. Doing so may result in the controller malfunctioning or burning.
- Follow the instructions in the relevant manuals to correctly perform terminal block and connector wiring and insertion. Double-check all wiring and connector insertion before turning ON the power supply.
- Care should be exercised by machine integrator when implementing a mating connector to the product. Consult the CK3C User's Manual for the correct components to use for external interconnects to this product.
- Prior to applying power, machine integrator should verify that the loading of the I/O interfaces will not exceed the parameters published in the CK3C User's Manual.
- Be mindful of a laceration hazard that can occur with the controller and associated system units. PCBs have sharp edges on the faceplates, sharp components installed and exposed circuit card edges.
- STATIC SENSITIVE DEVICE
- Observe safe static handling practives when making contact with the wiring and electronics of this product.
- Maintenance operator should have a machine diagram and plan developed by the machine integrator for proper service of the machine.
- Do not use corrosive chemicals to clean the Controller. Doing so may result in a failure or malfunction of the Controller.

Use

- The controller and its associated system units may not be removed or inserted (Hot Plugged) while power is applied to the system. Damage to the product may result.
- Do not remove power during a save operation. Memory corruption may occur causing a system error.
- NOR flash has a write limitation of 100K cycles and 20 years data retention. Avoid excessive write cycles in application.
- Do not attempt to disassemble, repair, or modify the Controller. Doing so may result in malfunction or fire.
- Always use power supply wires with sufficient wire diameters and current capacity to prevent voltage drop, burning, and excessive heat generation. When cross-wiring terminals, the total current for all terminals will flow in the wire. When wiring cross-overs, ensure the current capacity of all wires are not exceeded.
- Do not pull on the cables or bend the cables beyond their natural limit.
- Do not place heavy objects on top of the cables or other wiring lines. Doing so may break the cables.
- When connecting the CK3C and stacked accessories together, the power to the Unit should be OFF. Stacked accessories should be mounted securely in place using standoffs and screws.
- The number of stacked accessories connected to the CK3C must be within the specified range.

- Prior to applying power, ensure that Acc-51S and Acc-84S are not stacked on the controller at the same time. This stack configuration is not supported.
- If the external power supply to a digital output or a slave has polarity, connect it with the correct polarity. If the polarity is reversed, current may flow in the reverse direction and damage the connected devices regardless of the operation of the Controller.
- Do not turn OFF the power supply to the Controller while the built-in flash memory is being written. Data may become corrupted, and the Controller may not operate correctly.
- In the system, only use a power supply within the rated supply capacity range specified in the relevant manuals.
- Always turn OFF the power supply to the system before you attempt any of the following.
  - Mounting or removing the Units
  - Assembling the Units
  - Setting DIP switches
  - Connecting cables or wiring the system
  - Connecting or disconnecting the terminal blocks or connectors
- The motor is stopped if communications are interrupted between the Power PMAC IDE and the Controller during a Test Run. Connect the communications cable securely and confirm that the system will not be adversely affected before you perform a Test Run.
- When you create a servo algorithm, take fail-safe measures in the user program which includes the servo algorithm.
- When replacing the Controller or externally connected devices, ensure the transfer of required data such as the user program, configurations, settings and variables is completed before restarting operations.
- When upgrading a project file to a newer version of the Power PMAC IDE, perform a test run before use to check that the project file was upgraded correctly.
- It takes approximately several tens of seconds to enter RUN mode after the power supply is turned ON. During that time, outputs will be OFF, while the values and outputs will behave according to settings and/or specifications in the Unit and/or slaves. Also, external communications cannot perform. Implement fail-safe circuits so that external devices do not operate incorrectly.

Disposal

- Please dispose of this product as E-Waste. Product is marked for WEEE.

Precautions for Correct Use

If the equipment is used in a manner not specified by Delta Tau Data Systems, the protection provided by the equipment may be impaired.

Installation, Wiring and Maintenance

- Ensure the standoffs mounting stacked boards are tight and mounting standoffs on CK3C are connected to CHGND in customer control panel
- Refer to CK3C User's Manual for minimum wire sizes used on connectors.
- Refer to the CK3C User's Manual for proper instructions on how to remove power from the controller, what cleaners can be used, and how to re-apply power to the controller after cleaning is complete.
- Refer to the CK3C User's Manual before using any cleaning solution on the controller.

Transfer

- Do not operate or store the Controller in the following locations. Doing so may result in a malfunction or halt operations.
  - Locations subject to direct sunlight.
  - Locations subject to temperatures or humidity outside the range specified in the specifications.
  - Locations subject to condensation as a result of severe changes in temperature.
  - Locations subject to corrosive or flammable gases.
  - Locations subject to dust (especially iron dust) or salts.
  - Locations subject to exposure to water, oil, or chemicals.
  - Locations subject to shock or vibration.
- Take appropriate and sufficient countermeasures when installing the Controller in the following locations.
  - Locations subject to strong, high-frequency noise.
  - Locations subject to static electricity or other forms of noise.
  - Locations subject to strong electromagnetic fields.
  - Locations subject to possible exposure to radioactivity.
  - Locations close to power lines.
- Install the Controller away from sources of heat and ensure appropriate ventilation. Not doing so may result in a malfunction, operations halting, or burning.

Use

- Do not disconnect the communications cable while the system is running. Doing so may result in a failure or malfunction of the system.

Storage

- Do not drop the Controller or expose it to abnormal vibration or shock. Doing so may result in the controller malfunctioning or burning.

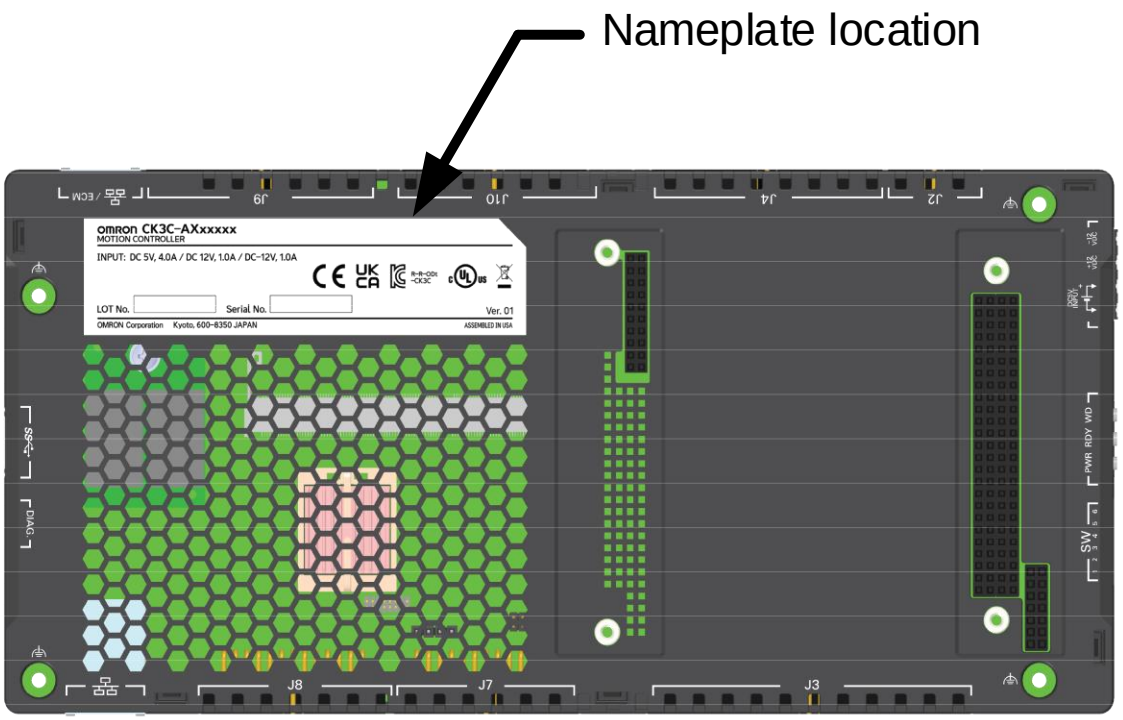
Items to Check after Unpacking

After you unpack the product, check the following items:

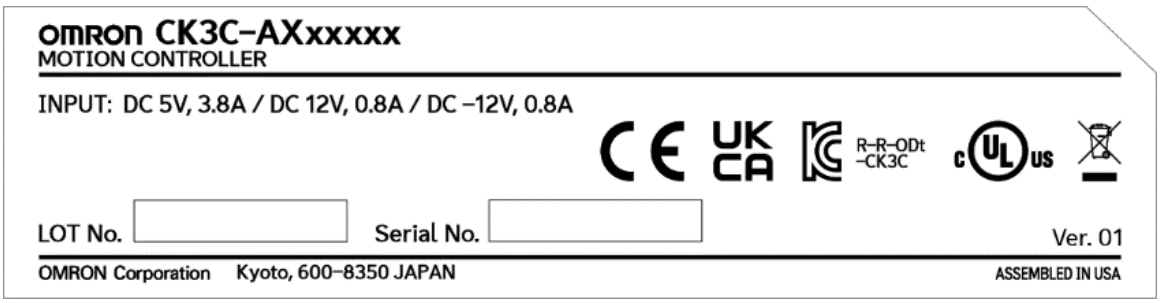
- Is this the model you ordered?
- Was there any damage sustained during shipment?

Location of Nameplate

The CK3C bears a nameplate label at the following location.



Information on Nameplate



Note: Nameplate example for CK3C-Series.

Accessories

The CK3C comes with the following accessories:

- Instruction Manual x 1 copy
- Warning label × 1 sheet
- General Compliance Information and instructions for EU × 1 copy
- Attached connectors (See table below)

When UL/CSA certification is required, attach the warning label to a place around the controller.

Connectors, mounting screws, mounting brackets, and other accessories than those in the table below are not supplied. They must be procured by the customer.

Model	DAC Output Power Cable	Control Power Cable	Encoder Connector	PWM Connector/Cable
-	NOT Included	NOT Included	NOT Included	NOT Included

If any item is missing or a problem is found such as controller damage, contact your local OMRON dealer or sales office where the product was purchased.

Regulations and Standards

This section describes the standards applicable to the CK3C.

Conformance to EU/CE Directives

Item	Standard
EU/CE	EN 61326-1:2013 (IEC 61326-1:2012)
EMC	EN55011:2009, +A1:2010 Class A Group 1 EN 61000-4-2:2009 (IEC 61000-4-2:2008) EN 61000-4-3:2007+A1: 2008+A2:2011 (IEC 61000-4-3:2006+A1: 2007+A2:2010) EN 61000-4-4: 2013 (IEC 61000-4-4: 2012) EN 61000-4-5:2014 (IEC 61000-4-5:2013) EN 61000-4-6:2014 (IEC 61000-4-6:2013) EN 61000-4-8:2010 (IEC 61000-4-8:2009) EN 61000-4-11:2010 (IEC 61000-4-11:2009)

Conformance to UL Directives

Item	Standard
UL	UL 61010-1 CAN/CSA C22.2 No. 1010.1-92

Conformance to UKCA Standards

Item	Standard
UKCA	2016 No. 1091

Conformance to KC Standards

Item	Standard
KC	EMI: KS C 9610-6-4:2017 EMS: KS C 9610-6-2:2019

When you use the CK3C in South Korea, observe the following precautions.

사 용 자 안 내 문
이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다.

한국 EMC적용제품 준수사항
본 제품은 전파법(KC 규정)을 준수합니다. 제품을 사용하려면 다음 사항에 유의하십시오. 이 기기는 업무용 환경에서 사용할 목적으로 적합성평가를 받은 기기로서 가정용 환경에서 사용하는 경우 전파간섭의 우려가 있습니다. 입력에 EMC 필터, 서지 보호기, 페라이트 코어 또는 1차측의 케이블에 노이즈 필터를 입력으로 사용하십시오.

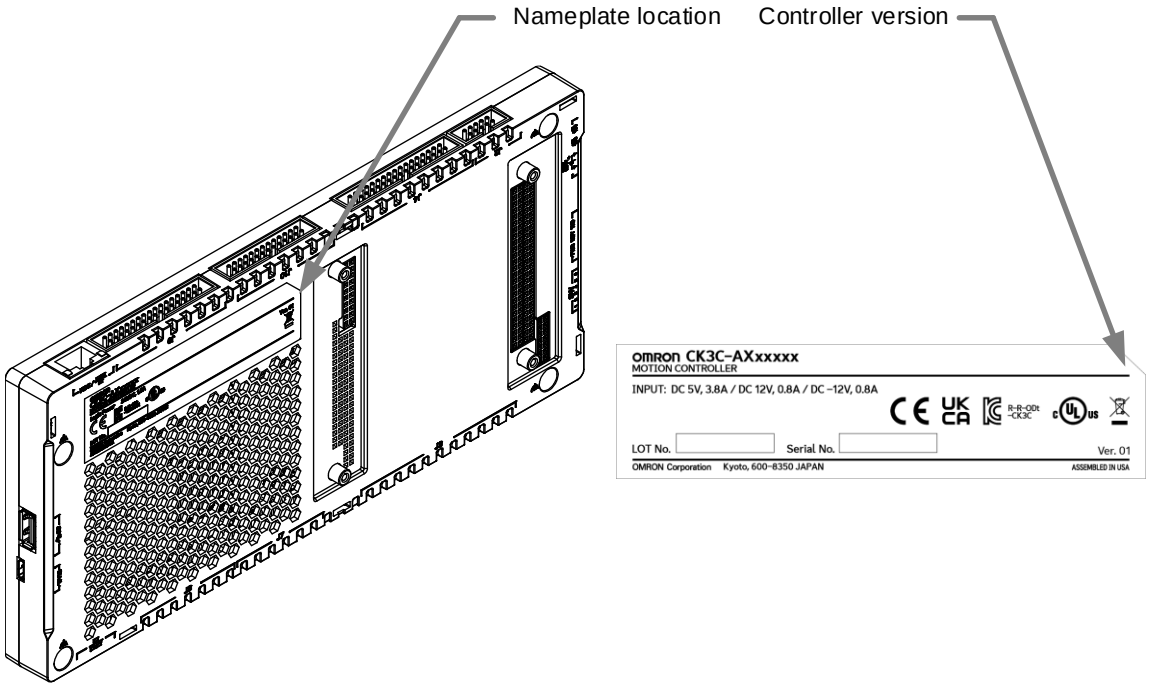
This product meets the electromagnetic compatibility requirements for business use. There is a risk of radio interference when this product is used in home.

Versions

The CK3C uses unit version. Unit versions are useful to manage differences in product modifications, upgrades, etc..

Hardware Version

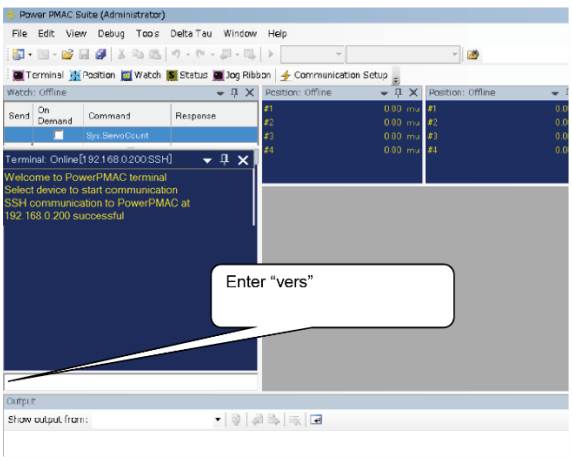
The hardware unit version of the CK3C controller is displayed on the nameplate.



Firmware Version

The controller firmware version can be retrieved by typing the following command in the terminal:

Command	Description	Example response
vers	The **vers** command causes Power PMAC to report the version number string of the firmware it is using.	2.8.1.1



Related Manuals

The following are related manuals for reference. Contact your local OMRON representative for procuring them.

Manual Name	Cat. No.	Application	Description
Power PMAC Software Reference Manual	O015	Learning the command set and structure elements of the Power PMAC Controller.	• Power PMAC Data structure • List and description of all commands • List and description of all ASIC, Coordinate System and Motor structure elements, including CK3C, CK3M and UMAC
Power PMAC User's Manual	O036	Learning the features and usage examples of the Power PMAC Controller.	• Parameter settings relevant to the Controller • Motor basic functions • Encoder configuration examples • Motor setup examples • Power PMAC programming examples
Power PMAC IDE User Manual	O016	Learning how to use the integrated development environment IDE of the Power PMAC Controller.	• Operating procedures of the Power PMAC IDE software • Configuration of the Direct PWM Controller using system setup
ACC-8ES Hardware Reference Manual	N/A	Learning the basic specifications of the CK3C accessory, ACC-8ES, including introductory information, design, installation, maintenance. The ACC-8AS has the same functionality as the ACC-8ES except the DAC output resolution, which is 16-bit in the former vs 18-bit in the later.	• Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring
ACC-8FS Hardware Reference Manual	N/A	Learning the basic specifications of the CK3C accessory, ACC-8FS, including introductory information, design, installation, maintenance.	• Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring
ACC-24S3 Hardware Reference Manual	N/A	Learning the basic specifications of the CK3C accessory, ACC-24S3, including introductory information, design, installation, maintenance.	• Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring

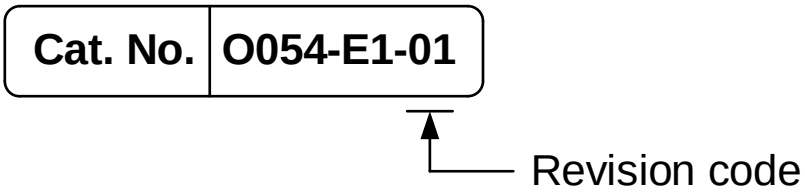
Manual Name	Cat. No.	Application	Description
ACC-51S Hardware Reference Manual	N/A	Learning the basic specifications of the CK3C accessory, ACC-51S, including introductory information, design, installation, maintenance.	• Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring
ACC-84S Hardware Reference Manual	N/A	Learning the basic specifications of the CK3C accessory, ACC-84S, including introductory information, design, installation, maintenance.	• Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring

Terminology

TERM	DESCRIPTION
Power PMAC	• PMAC is the acronym for Programmable Multi-Axis Controller • Power refers to the platform at the time of creation of this manual
Gate3	Delta Tau ASIC in the CK3C board
Controller	In this manual, used as a short for the Power PMAC Controller w/ Gate3 ASIC
IDE	• Acronym for Integrated Development Environment • The primary software that is used to program the Power PMAC Controller
FPGA	Field Programmable Gate Arrays
ADC	Analog to Digital Converter
ADC Strobe Word	Refers to Power PMAC structure element AdcAmpStrobe
PWM	Acronym for Pulse Width Modulation
PFM	Acronym for Pulse Frequency Modulation
Logic power	Refers to the control power supply
PLC	• Acronym for Programmable Logic Controller • In this manual, referring to a Power PMAC PLC program
I2T	Thermal protection model, can refer to Controller built-in I2T or PMAC configurable I2T
ECT	Acronym for Encoder Conversion Table
LSB	Acronym for “Least Significant Bit”
MSB	Acronym for “Most Significant Bit”
CH (e.g. CH0-3)	“CH” is an acronym for channel

Manual Revision History

A manual revision code appears as a suffix to the catalog number on the front and back covers of the manual.



Revision code	Date	Revised Content
01	May 2022	Original production
02	August 2023	Added General Purpose I/O software configuration
03	August 2025	Updated front matter and spec tables

1

Introduction to the CK3C

This section describes the features, and basic configuration of the CK3C. We may refer to this device at times as “Controller” in parts of this manual for simplicity.

1-1 Outline 1-2
1-1-1 Controller Features 1-2
1-1-2 Typical Configuration 1-4
1-1-3 Part Names, Locations, and Functions 1-5
1-2 Operating Procedure 1-6
1-2-1 Preparation 1-6
1-2-2 Safety 1-6
1-2-3 Mounting and Wiring 1-6
1-2-4 Safety Test and Power Up 1-6
1-2-5 Controller Settings 1-7
1-2-6 Test Run 1-7

1-1 Outline

This section describes the features, and basic configuration of the CK3C.

1-1-1 Controller Features

About

The CK3C is a multi-axis controller with 4 local axis and has the intelligence and capability of a Power PMAC ARM CPU. The primary advantage of this controller is that it comes with an integrated interface at a very suitable size. The CK3C is typically used as an interface with various controller/drive, and I/O technology. In addition, the controller can support virtually any type of feedback device.

The CK3C system is considered for use as a building block module.
Attaching a unit to the system stack can expand the function or increase the number of I/O's.
The building block arrangement is flexible and easy to expand the system.

Flexibility

The controller has the ability to interface with various encoder feedback:

Item	Specification
Servo Interface (standard)	• Quadrature (differential, with index) • UVW digital hall sensor • Serial encoder interface (software configurable): - SSI - EnDat 2.1 / 2.2 (2.1-compatible features only) with delay comp. - Hiperface - Yaskawa Sigma I / II / III / V (no position reset or fault clear) - Tamagawa FA-Coder (no servo clock output) - Panasonic (no servo clock output) - Mitutoyo - Kawasaki • Pulse & direction output • Position compare (EQU) output (5 V TTL) • Input flags (home, plus limit, minus limit, user)

Additionally, the CK3C provides the support of the following types of Amplifier output signals:

Item	Specification
Amplifier Output	• Analog voltage ±10V (Filtered PWM) • Analog voltage ±10V (True DAC) • Direct PWM • Pulse & direction (PFM)

Expandability

If additional motors are required, there are multiple methods to expand the functionality of the CK3C:

Item	Method
Axis Expansion	• Adding an axis device such as an ACC-24S3, the system can support an additional 4 local axes • Using EtherCAT to communicate with up to 8 slave axes

If additional encoder protocols and features are required, the ACC-84S and ACC-51S can provide:

Item	Specification
Servo Interface (with ACC-84S)	• SSI (no additional capability over Gate3 built-in interface) • EnDat 2.2 with additional information, no delay compensation • Yaskawa Sigma II/III/V with position reset and fault clear • Tamagawa FA-Coder with servo clock output • Panasonic (no additional capability over Gate3 built-in interface) • Mitutoyo (no additional capability over Gate3 built-in interface) • BiSS-B/C • Mitsubishi • Omron1S • Table Based Position Compare • XY2-100
Servo Interface (with ACC-51S)	• Sinusoidal feedback x 4096

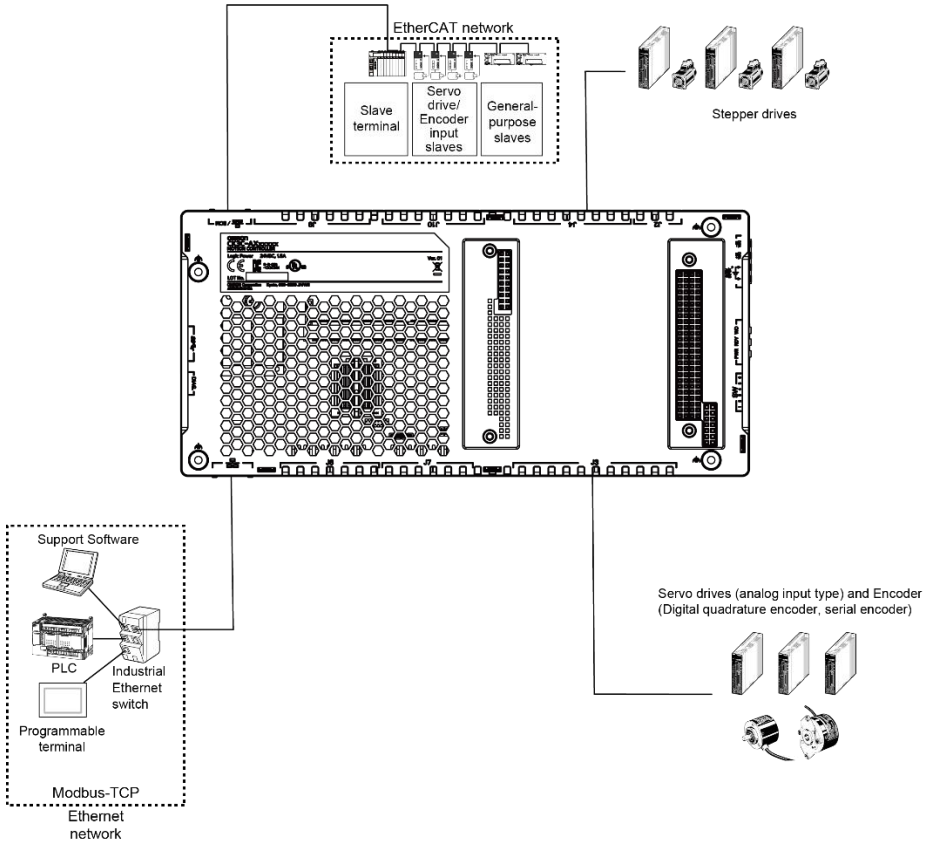
Features

- Fast Multi-Axis Control
- Stackable motion controller for enhanced system compactness

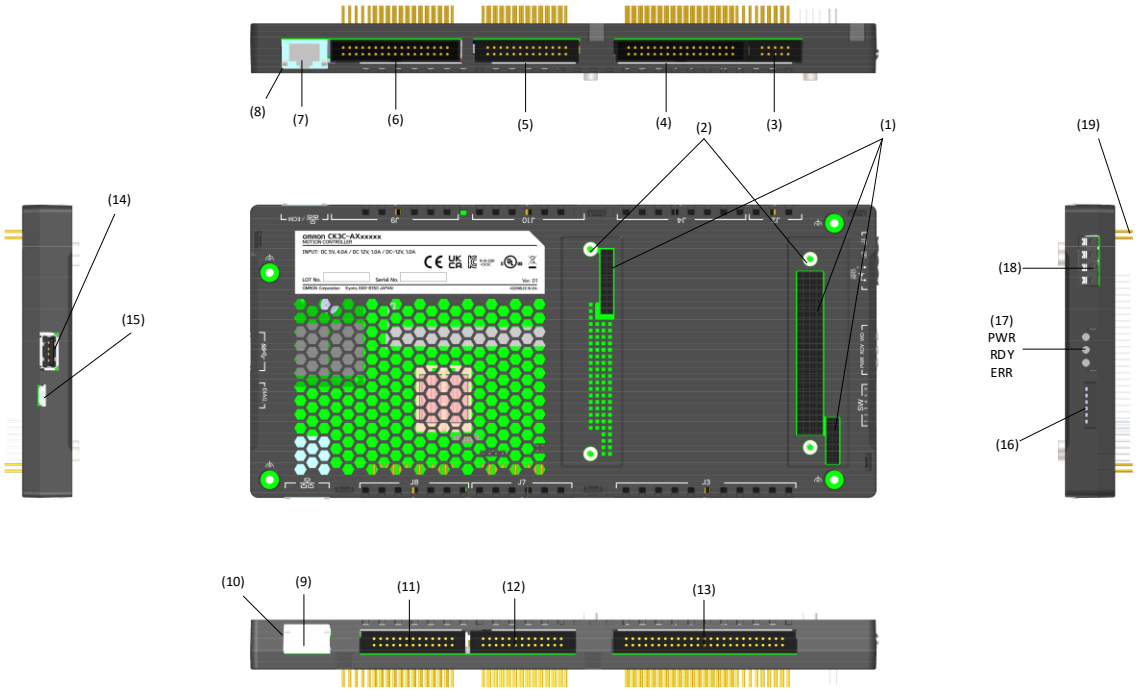
1-1-2 Typical Configuration

The CK3C can connect to the following types of devices illustrated below.

- The controller setup software is typically done in the Power PMAC IDE and can be done in the NC environment as well
- The Motor connects directly to a third party drive.
- The encoder connects directly to the CK3C.
- There are digital inputs and outputs on the controller that can communicate with I/O devices
- The CK3C can communicate with other hardware devices over Modbus/TCP or EtherCAT such as an OMRON NJ/NX coupler or EtherCAT slave devices.



1-1-3 Part Names, Locations, and Functions



Number	Name
1	CK3C Stack Connectors
2	Stack Standoffs
3	J2
4	J4
5	J10
6	J9
7	EtherCAT communications connector
8	EtherCAT communications port operation indicators
9	Ethernet communications connector
10	Ethernet communications port operation indicators
11	J8
12	J7
13	J3
14	USB 3.0 Host Port
15	Diagnostic Port
16	DIP Switch
17	Indicators
18	Power Supply Input
19	Embedded Pin

1-2 Operating Procedure

This section describes the necessary steps to integrate the CK3C into a system.

1-2-1 Preparation

Item	Reference
Logic power supply	Must be within the specifications stated in this manual
Analog I/O power supply	Must be within the specifications stated in this manual
Dimensions, panel size and location	Must be within conditions and specifications stated in this manual

1-2-2 Safety

Item	Reference
Overall Safety	Determining safety measures based on risk assessment
Warnings and Cautions Review	Notifications stated in Safety Precautions section of this manual
Design	Design safety measures based on overall machine safety, and warnings

1-2-3 Mounting and Wiring

Item	Reference
Mounting	Follow mounting guidelines described in this manual • Ergonomic and environmental location • Mounting direction • Single-unit clearance
Wiring	Use wiring diagrams and guidelines described in this manual • Safety measures • Correct pinout • Wire gauges • Ergonomic cable management

1-2-4 Safety Test and Power Up

Item	Reference
Precautions for Safe Use	Review Precautions for Safe Use section of this manual

1-2-5 Controller Settings



Precautions for Safe Use

This step should be performed with the main circuit power OFF.

Item	Reference
Preparation	• Install the Power PMAC IDE Software on the computer in use • Identify the list of related manuals described in this manual
Plug/Unplug	• Plug or unplug any of the following cables and accessories prior to power up of the Controller • Power cable, J-port cable, ethernet cable, EtherCAT cable, USB memory, diagnostic port cable and other stack accessories
Power PMAC Controller	• Establish communication between the Power PMAC and IDE Software • Load existing project or start from factory default settings • Use related manuals to the Controller form factor or IDE system setup
Encoder Settings	Implement encoder specific configuration and settings using encoder specification sheet and related manuals
Controller settings	Implement Controller specific settings described in this manual
Motor Settings	Implement motor specific configuration and settings using the motor specification sheet and related manuals

1-2-6 Test Run

Item	Reference
Verify encoder feedback	Using related hardware or user's manual
Amplifier/Motor Setup	Using related hardware, user's manual, or IDE system setup
Servo ON/OFF	Servo ON/OFF Check Motor operation



Additional Information

Alternately, this step can be performed using the automatic system setup in the IDE.

2

Specifications

This section describes the general specifications of the CK3C controller.

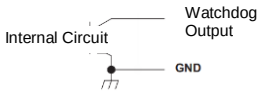
2-1 Controller Specifications 2-2
2-1-1 General/Mechanical 2-2
2-1-2 Environmental 2-4
2-1-3 Electrical 2-5
2-2 Part Number Designation 2-6
2-2-1 Controller Part Number 2-6
2-2-2 Stackable Accessory Part Numbers 2-7

2-1 Controller Specifications

This section describes the general, mechanical, environmental, functions, and electrical specifications of the controller.

2-1-1 General/Mechanical

Item		Specification
Memory		Main memory: 1 GB (non-ECC) Flash memory: 2 GB
External terminals		[Communications connector] For EtherCAT communications: • RJ45 × 1 (Shield supported) For Ethernet communications • RJ45 × 1 (Shield supported)
		[USB port] For external memory connection, USB 3.0 host × 1 Type A
Maximum number of stacked units		4 (including CK3C Controller) Max number of ACC-24S3's allowable*1: 1
Motion control	Maximum number of controlled axes	Analog: 8 axes max. (8 axis max. with ACC-24S3) EtherCAT: 8 axes max
	Control method	Direct Control : With Controller unit or ACC-24S3 Network : EtherCAT of Controller unit
EtherCAT communications specifications	Communications protocol	EtherCAT protocol
	Mode	DC sync, Free run
	Baud rate	100 Mbps
	Physical layer	100BASE-TX (IEEE 802.3)
	Topology	Line, daisy chain, and branching
	Transmission media	Twisted-pair cable of category 5 or higher (double-shielded cable with aluminum tape and braiding)
	Transmission distance	Distance between nodes: 100 m or less
	Maximum number of slaves	8
	Minimum EtherCAT Cycle Period	500usec min (8 axis slaves with total 32 PDO)
	Range of node addresses that can be Set	1 to 8
Ethernet communications specifications	Physical layer	1000BASE-T
	Frame length	1,514 bytes max.
	Media access method	CSMA/CD
	Modulation	Baseband
	Topology	Star
	Transmission media	Twisted-pair cable of category 5, 5e, or higher (shielded cable)
	Maximum transmission distance between Ethernet switch and node	100 m
	Maximum number of cascade connections	There are no restrictions if an Ethernet switch is used
USB port	Physical layer	USB 3.0 compliant, type A connector. Output voltage: 5 V, 0.9 A max.
	Transmission distance	3 m max.
Watch dog timer output	Output form	Open drain output (Under WD condition the output is open)
	Rated voltage	5 VDC

Item		Specification
	Operating load voltage range	20.4 to 26.4 VDC
	Maximum value of load current	0.1 A
	Leakage current	0.1 mA max
	Residual voltage	1.5 VDC max.
	ON/OFF response time	10 ms max./10 ms max.
	Isolation method	No isolation
	Circuit Configuration	
Dimensions (Height / Depth / Width) (Not including extended pins, CK3C-AX1□□□)		244.4 / 114.7 / 22.3 mm
Dimensions (Height / Depth / Width) (Including extended pins, CK3C-AX1□□□E)		244.4 / 114.7 / 29.2 mm
Weight		420 g

*1. Due to extended UBUS data connector, the ACC-24S3 has to be directly connected to the CK3C Controller.

Typical Stack Configurations		
Stack	**Amplifier Output**	**Encoder Feedback**
CK3C	Analog ±10V Filtered PWM	Refer to Section 1-1-1
ACC-51S ACC-8AS CK3C	Analog ±10V True DAC Analog ±10V Filtered PWM	Same as ACC-24E3 + Sinusoidal x 4096
ACC-84S Serial ACC-8AS CK3C	Analog ±10V True DAC Analog ±10V Filtered PWM	Same as ACC-24E3 + BiSS or EnDat 2.2
ACC-51S ACC-8FS CK3C	Digital Direct PWM Analog ±10V Filtered PWM	Same as ACC-24E3 + Sinusoidal x 4096
ACC-84S Serial ACC-8FS CK3C	Digital Direct PWM Analog ±10V Filtered PWM	Same as ACC-24E3 + BiSS or EnDat 2.2

2-1-2 Environmental

Item		Specification
Operating Environment	Operating ambient temperature	0 to 55 °C
	Operating ambient humidity	10 – 90 % RH (without condensation or icing)
	Storage ambient temperature	-25 to 70 °C
	Storage ambient humidity	10 – 90 % RH (without condensation or icing)
	Vibration resistance	Conforms to IEC 60068-2-6. 5 to 8.4 Hz with 3.5-mm amplitude, 8.4 to 150 Hz, acceleration of 9.8 m/s2 each in X, Y, and Z directions 100 min (10 sweeps of 10 min each = 100 min total)
	Shock resistance	Conforms to IEC 60068-2-27, 147 m/s2, 3 times each in X, Y, and Z directions
	Operating and storage atmosphere	Must be free of corrosive gases
	Maximum operating altitude	1,000 m
Enclosure		Mounted vertically in a panel
Grounding methods		Ground of 100 Ω or less
Insulation resistance		20 MΩ min. between isolated circuits (at 100 VDC)
Dielectric strength		510 VAC between isolated circuits for 1 minute at a leakage current of 5 mA max.
Pollution environment		Degree 2
Measurement category		Category III

2-1-3 Electrical

Digital Power Supply

The +5V and ground reference lines from the power supply should be connected to TB1 terminal block of the CK3C using 16 AWG stranded wire.

Item	Specification
Input voltage	5 VDC ± 0.2 VDC (3.8A max.)
Isolation	No isolation
Maximum input voltage ripple	0.1 Vp-p
Power down detection	4.6 VDC min. low voltage fault detection on 5 VDC input

Additional Information

The defined maximum current above indicates current consumption of the CK3C and any stack accessories, if any additional circuit is connected to the +5 VDC pins on any of the connectors additional current capacity should be considered by the user. Refer to terminal wiring diagrams for recommended wiring.

Analog I/O Power Supply

The ±12 VDC lines from the supply, including the ground reference, must be connected to TB1 terminal block.

Item	Specification
Input voltage	+12 VDC (+3.5 VDC/-0.2 VDC) (0.8 A max.)
	-12 VDC (0.2 VDC/-3.5 VDC) (0.8 A max.)
Isolation	No isolation
Maximum input voltage ripple	0.1 Vp-p

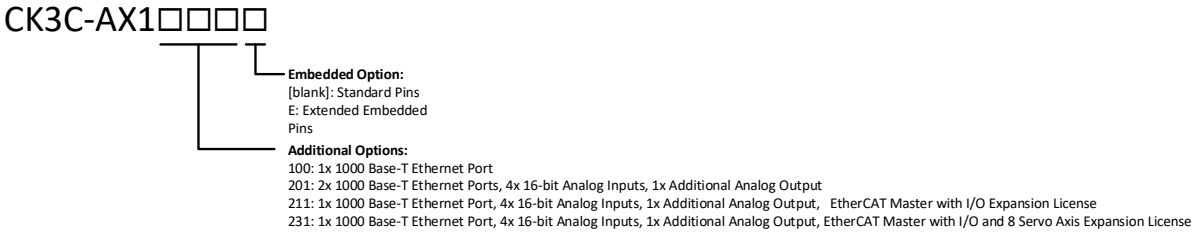
Precautions for Safe Use

Improper wiring of ±12 VDC to +5VDC input terminal pin can result in permanent damage to the controller circuits. Refer to terminal wiring diagrams for recommended wiring.

2-2 Part Number Designation

This section shows the part number designation.

2-2-1 Controller Part Number



Type of Unit	Model	CPU	Memory	Ethernet Port(s)	EtherCAT License	Analog I/O	Extended Pin
Controller Unit	CK3C-AX1100	Coretex-A7 Dual core 1GHz	RAM: 1 GB Flash: 2 GB	1	No	No	No
	CK3C-AX1201			2	No	4x 16-bit AI 1x Filtered PWM AO	No
	CK3C-AX1211			2	I/O Only	4x 16-bit AI 1x Filtered PWM AO	No
	CK3C-AX1231			2	8 Servo	4x 16-bit AI 1x Filtered PWM AO	No
	CK3C-AX1100E			1	No	No	Yes
	CK3C-AX1201E			2	No	4x 16-bit AI 1x Filtered PWM AO	Yes
	CK3C-AX1211E			2	I/O Only	4x 16-bit AI 1x Filtered PWM AO	Yes
	CK3C-AX1231E			2	8 Servo	4x 16-bit AI 1x Filtered PWM AO	Yes

2-2-2 Stackable Accessory Part Numbers

The CK3C comes standard with a powerful set of hardware and software capabilities, plus a full set of options and stackable accessories.

ACC-24S3

Part Number	Description
3-4050000-000-000000	4-Channel expansion
3-4050000-000-010000	4-Channel expansion with 4 analog inputs & 1 Filtered PWM output

ACC-51S

Part Number	Description
3-3674-0-0002-000000	2 channel 4096x interpolator board for sinusoidal encoders
3-3674-0-0012-000000	4 channel 4096x interpolator board for sinusoidal encoders

ACC-84S

Part Number	Description
3-3936-0-0002-000000	4-Channel SSI serial encoder protocol
3-3936-0-0003-000000	4-Channel ENDAT serial encoder protocol
3-3936-0-0006-000000	4-Channel Yaskawa Sigma II & III serial encoder protocol
3-3936-0-0008-000000	4-Channel Panasonic serial encoder protocol
3-3936-0-000B-000000	4-Channel BISS-C serial encoder protocol
3-3936-0-000D-000000	4-Channel Mitsubishi serial encoder protocol
3-3936-0-000E-000000	4-Channel 1S serial encoder protocol
3-3936-0-0XY2-000000	XY2-100 serial link for 2-axis and 3-axis laser scan heads and galvanometers
3-3936-0-0TBC-000000	4-Channel table based compare protocol

ACC-8AS

Part Number	Description
3A0-603673-10X	4-Channel dual 16-bit DAC analog board

ACC-8FS

Part Number	Description
3F0-603673-10X	4-Channel direct PWM breakout board

4-Channel Breakout Board

Part Number	Description
CBB4-00-000-000-1000	CK3C breakout board with separate connectors for each encoder, each flag set, and servo amplifier output

3

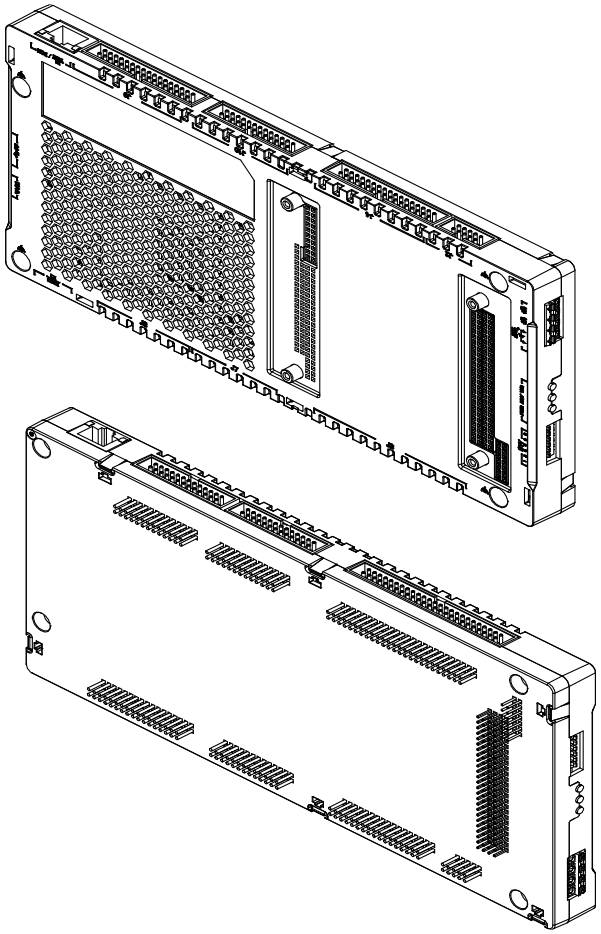
Mounting and Installation

This section describes the mounting and installation methods of the CK3C.

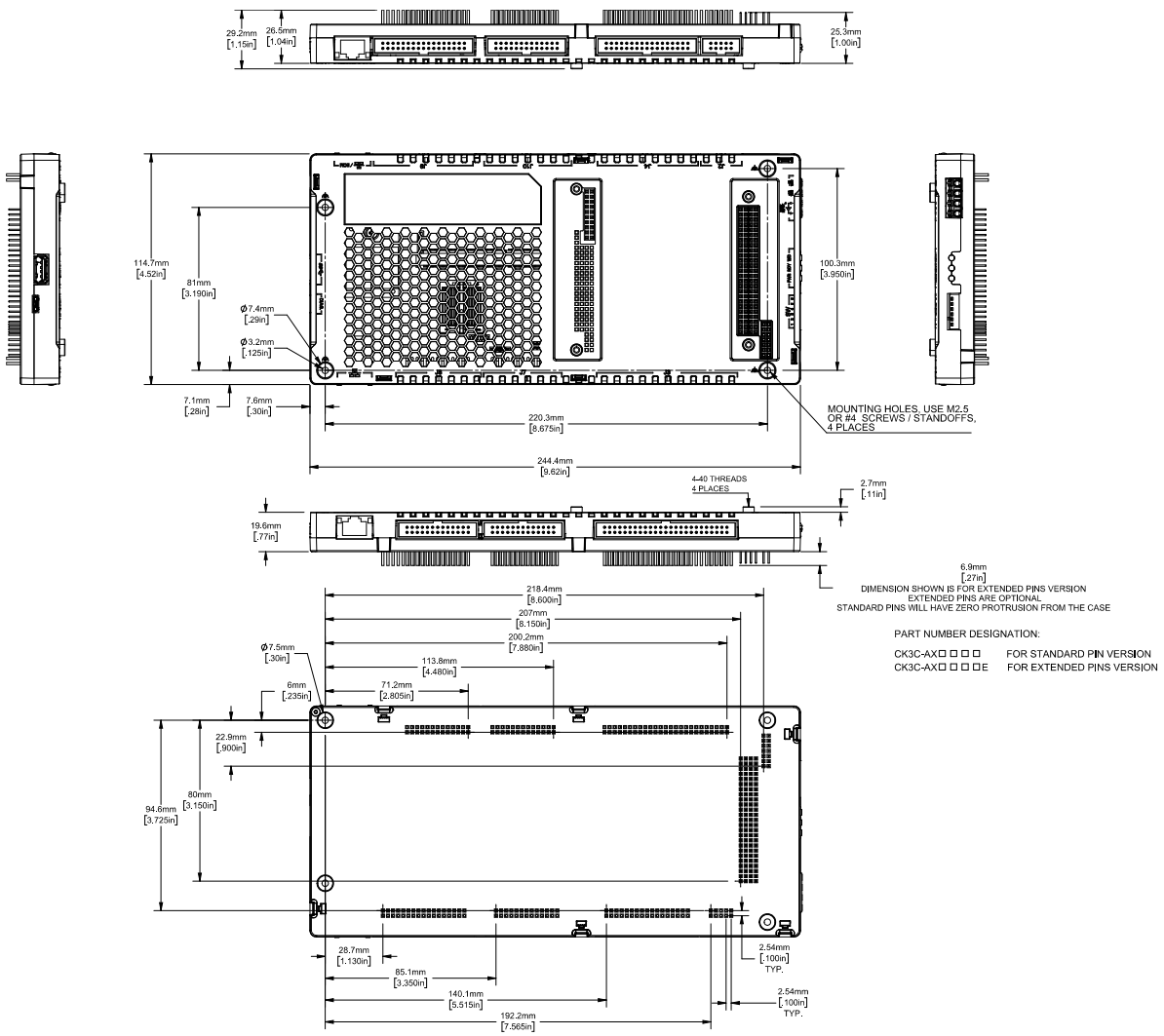
3-1 External and Mounting Dimensions 3-2
3-1-1 Isometric View 3-2
3-1-2 CK3C Dimensions and Mounting 3-3
3-2 Installation 3-4
3-2-1 Ventilation 3-5
3-2-2 Panel Clearance/Mounting Direction 3-5
3-2-3 Installation Conditions 3-5
3-2-4 Keeping Foreign Objects Out of Units 3-6

3-1 External and Mounting Dimensions

3-1-1 Isometric View



3-1-2 CK3C Dimensions and Mounting



3-2 Installation

Installation should be in an area that is protected from direct sunlight, corrosives, harmful gases or liquids, dust, metallic particles, and other contaminants. Exposure to these can reduce the operating life and degrade performance of the board.

Several other factors should be carefully evaluated when selecting a location for installation:

- For effective cooling and maintenance, the CK3C should be mounted on a smooth, nonflammable vertical surface.
- At least 61 mm (2.4 inches) top and bottom clearance must be provided for air flow and minimal bend of the RJ45 connectors.
- Temperature, humidity and Vibration specifications should also be taken in account.



Precautions for Safe Use

Unit must be installed in an enclosure that meets the environmental IP rating of the end product (ventilation or cooling may be necessary to prevent enclosure ambient from exceeding 55° C [113° F]).

The CK3C can be mounted as a stand-alone controller. At each of the four corners of the CK3C there are mounting holes that can be used for standoffs.

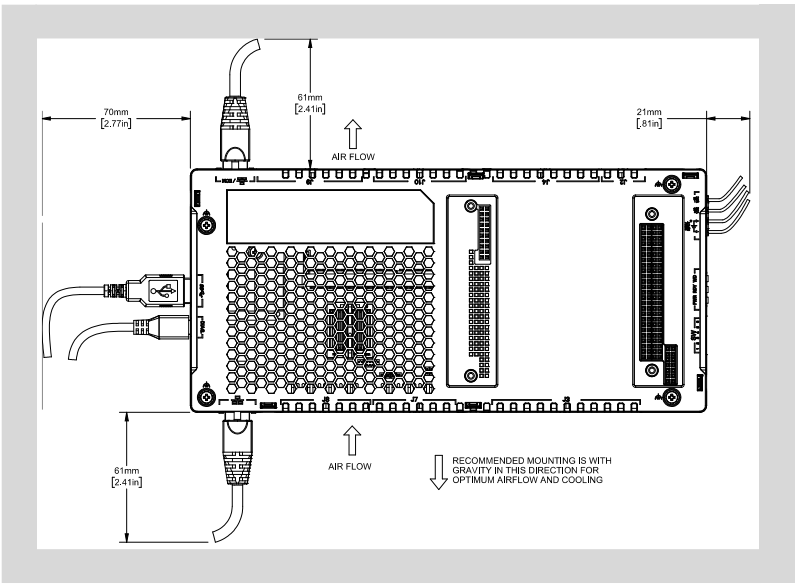
If the CK3C is mounted to a back panel, the back panel should be unpainted and electrically conductive to allow for reduced electrical noise interference. The back panel should be machined to accept the standoffs pattern of the board.

The controller can be mounted to the back panel using four conductive standoffs and internal-tooth lock washers. It is important that the teeth break through any oxidization on the controllers mounting gears to provide a good electrically conductive path in as many places as possible.

3-2-1 Ventilation

Install the Controller according to the dimension specifications shown below. Ensure proper dissipation of heat from the Controller and convection inside the panel.

3-2-2 Panel Clearance/Mounting Direction



Dimension	Distance
Top	61 mm minimum
Bottom	61 mm minimum
Left side	70 mm minimum
Right side	21 mm minimum
Front side	20 mm minimum (except for stack accessory installation)

3-2-3 Installation Conditions

- The panel environmental (such as operating temperature and humidity) conditions must abide by the environmental specifications stated in this manual. Operating the controller outside of these conditions may result in malfunction.
- Operating in an environment where temperature rise is minimal is recommended to maintain a high level of reliability.
- The use of the controller continuously in a hot environment shortens its lifetime. Use a fan or air conditioner to maintain the ambient temperature specified for this product.
- Remove any surface paint to provide electrical conductivity.

3-2-4 Keeping Foreign Objects Out of Units

- Take measures during installation and operation to keep foreign objects such as metal particles, debris, oil, machining oil, dust, or water out of the controller.
- Place a cover over the Controller or take other preventative measures to keep foreign objects, such as drill filings, out of the Controller during installation. Be sure to remove the cover after installation is complete. If the cover is left on during operation, heat dissipation from the CK3C is blocked, which may result in malfunction.

4

Wiring

This section describes the wiring methods of the CK3C.

4-1 DIP Switch configurations 4-3

4-1-1 Serial Encoder Interface Port Selection 4-3
4-1-2 General Purpose I/O Buffer Direction Selection 4-4

4-2 Power Supply Input 4-5

4-2-1 Connector Pinout 4-5
4-2-2 Wiring the Terminal 4-6

4-3 Machine Connector (J3) 4-9

4-3-1 Circuit Specifications 4-9
4-3-2 Connector Pinout 4-13
4-3-3 Wiring a Digital Quadrature Encoder 4-16
4-3-4 Wiring an Analog Input Servo Drive 4-17
4-3-5 Wiring the Analog Inputs 4-18

4-4 Machine Connector (J4) 4-19

4-4-1 Circuit Specifications 4-19
4-4-2 Connector Pinout 4-22
4-4-3 Wiring a Stepper Drive 4-24
4-4-4 Wiring the PMAC Reset Input 4-25
4-4-5 Wiring the Limits and Flags 4-25
4-4-6 Wiring the Compare Equal Outputs 4-26
4-4-7 Wiring an External Safety Circuit 4-26

4-5 Machine Connector (J7) 4-27

4-5-1 Circuit Specifications 4-27
4-5-2 Connector Pinout 4-30
4-5-3 Wiring a Hall Sensor 4-31
4-5-4 Wiring a Serial Encoder 4-32
4-5-5 Wiring the Brake Outputs 4-33
4-5-6 Wiring the Analog Inputs 4-33

4-6 GPIO Port (J8, J9) 4-34

4-6-1 Circuit Specifications 4-34
4-6-2 Connector Pinout (J8) 4-36
4-6-3 Connector Pinout (J9) 4-37
4-6-4 Wiring a General Purpose Input 4-39
4-6-5 Wiring a General Purpose Output 4-39

4-7 Handwheel/Pulse-Dir Port (J10) 4-40

4-7-1 Circuit Specifications 4-40
4-7-2 Connector Pinout 4-43
4-7-3 Wiring a Stepper Drive 4-44
4-7-4 Wiring a Quadrature Encoder 4-45
4-7-5 Wiring the Analog Output 4-45

4-8 Diagnostic Port 4-46

4-9 EtherCAT™/Ethernet Port 4-47

4-10 Ethernet Communications Port 4-48

4-11 USB Host Port 4-49

4-11-1 Circuit Specifications 4-49

4-12 Serial Port 4-50

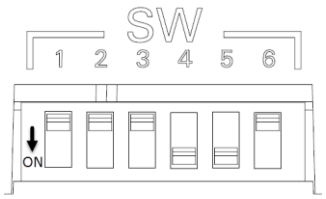
4-13 Wiring Conforming to EMC/EU Directives 4-51

4-14 Selecting Peripheral Components 4-52

4-14-1 Recommended Power Supply 4-52
4-14-2 Alternate Power Supply Solution 4-52

4-1 DIP Switch configurations

The DIP switch controls two set of functions. The first two switches control if the serial encoder interface for the first two channels are wired to connector J10 or J7. The last 4 switches control the IO buffer direction for the general purpose IO pins.



4-1-1 Serial Encoder Interface Port Selection

The Position 1 and Position 2 on the DIP Switch control which port is connected to the serial encoder interface on first two channels of the CK3C.

Position	Description	OFF	ON	Factory Default
1	Data+ input for channel 0	(J10) HWB0+	(J7) CHT0	ON
	Data- input for channel 0	(J10) HWB0-	(J7) CHU0	
	Clock+ input for channel 0	(J10) HWA0+	(J7) CHV0	
	Clock- input for channel 0	(J10) HWA0-	(J7) CHW0	
2	Data+ input for channel 1	(J10) HWB1+	(J7) CHT1	ON
	Data- input for channel 1	(J10) HWB1-	(J7) CHU1	
	Clock+ input for channel 1	(J10) HWA1+	(J7) CHV1	
	Clock- input for channel 1	(J10) HWA1-	(J7) CHW1	

If user intends to use serial encoders which are supported by CK3C, select the J7 as input port for serial encoder by setting the channel specific switch to ON position and wire the serial encoders based upon the pinout explained in section 4-5 of this manual.

If the user wants to wire additional quadrature encoders to the Handwheel port on J10, select J10 as input port for serial encoder by setting the channel specific switch to OFF position and wire the quadrature encoder based upon the pinout explained in section 4-7 of this manual.

Support for additional quadrature encoders is provided as a backward compatible wiring to Delta Tau's Turbo Clipper Controller product to allow users to keep the machine wiring the same.

Example Encoder Combinations

In the examples below, depending on DIP switches 1 & 2 setting, you can read the possible encoder protocols listed in the tables below.

Position	Channel	Encoder(s) per Channel	Total # of Encoder Types Used
1 (OFF)	1	Quadrature + HW Quadrature	• 4 Quadrature enc. (J3) • 2 HW Quadrature enc. (J10) • 2 Serial enc. (J7) Total # : 8
2 (OFF)	2	Quadrature + HW Quadrature	
-	3	Quadrature + Serial	
	4	Quadrature + Serial	

Position	Channel	Encoder(s) per Channel	Total # of Encoder Types Used
1 (OFF)	1	Quadrature + HW Quadrature	• 4 Quadrature enc. (J3) • 1 HW Quadrature enc. (J10) • 3 Serial enc. (J7) Total # : 8
2 (ON)	2	Quadrature + Serial	
-	3	Quadrature + Serial	
	4	Quadrature + Serial	

Position	Channel	Encoder(s) per Channel	Total # of Encoder Types Used
1 (ON)	1	Quadrature + Serial	• 4 Quadrature enc. (J3) • 4 Serial enc. (J7) Total # : 8
2 (ON)	2	Quadrature + Serial	
-	3	Quadrature + Serial	
	4	Quadrature + Serial	

4-1-2 General Purpose I/O Buffer Direction Selection

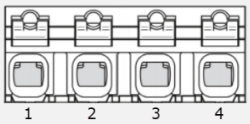
The DIP Switch positions 3 through 6 control the direction of the transceiver buffer for each group of 8 general purpose IO accessible through CK3C[0].GpioData[x] where x is between 0 - 31.Please note that the direction selection should match the CK3C[0].GpioDir[0] register setting.

Position	Description	OFF	ON	Factory Default
3	Gpio[07:00] (DATn) direction control (J8)	Input	Output	OFF
4	Gpio[15:08] (SELn) direction control (J8)	Input	Output	OFF
5	Gpio[23:16] direction control (J9)	Input	Output	OFF
6	Gpio[31:24] direction control (J9)	Input	Output	OFF

4-2 Power Supply Input

4-2-1 Connector Pinout

This 4-pin terminal block is used to bring the +5 VDC logic power and +/-12 VDC analog IO supply into the CK3C.

TB1: Power Supply 4-Pin Terminal Block				
Pin #	**Symbol**	**Function**	**Description**	**Notes**
1	GND	Common	Digital Common	
2	+5V	Input	Logic Supply	Supplies all PMAC digital circuits
3	+12V	Input	Analog Supply	Ref to Digital GND
4	-12V	Input	Analog Supply	Ref to Digital GND

Precautions for Safe Use

Improper wiring of ±12 VDC to +5 VDC input terminal pin can result in permanent damage to the controller circuits. Refer to terminal wiring diagrams for recommended wiring.

4-2-2 Wiring the Terminal

This section describes the wiring for the power supply connection terminal block.

Terminal Block Wiring

- **Applicable Wires**

  The wires that you can connect to the terminal block are twisted wires, solid wires. Ferrules are attached to the twisted wires. The following section describes the dimensions and processing methods for applicable wires.

Using Ferrules

If you use ferrules, attach the twisted wires to them. Observe the application instructions for your ferrules for the wire stripping length when attaching ferrules. Always use plated one-pin ferrules. Do not use un-plated ferrules or two-pin ferrules.

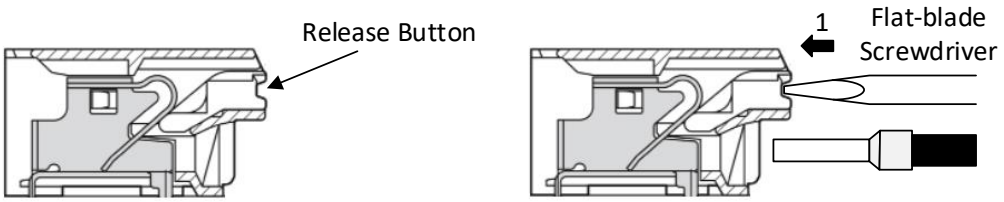
The applicable ferrules, wires, and crimping tools are listed in the following table.

Manufacturer	Ferrule Model	Applicable Wire mm^2 (AWG)	Crimping Tool (Applicable wire size in parentheses)
Phoenix Contact	Al1,0-10	1.0(#18)	Phoenix Contact CRIMPFOX 6 (0.25 to 6 mm2, AWG24 to 10)
	Al1,5-10	1.5(#16)	
Weidmüller	H1.0/16	1.0(#18)	Weidmüller PZ6 Roto (0.14 to 6 mm2, AWG26 to 10)
	H1.5/16	1.5(#16)	

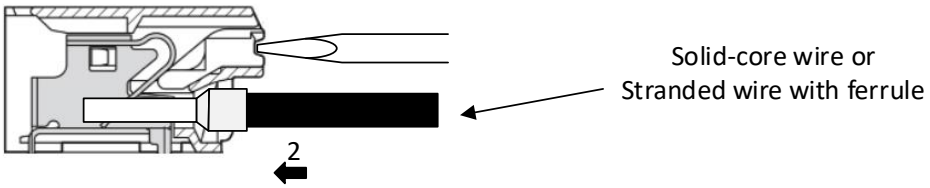
Connecting Wire

Use the following procedure to connect the twisted wires or solid wires to the terminal block.

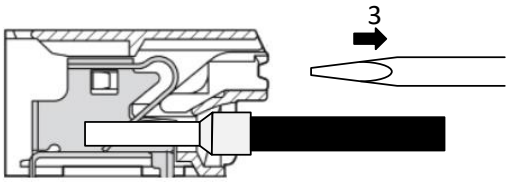
1 Press a flat-blade screwdriver straight against the release button from the terminal block front.



2 Leave the flat-blade screwdriver pressed against the release button and insert the stranded wire with ferrule or the solid-core wire into the terminal hole.
Insert the stranded wire with ferrule or the solid-core wire until the stripped portion is no longer visible, to prevent shorting.



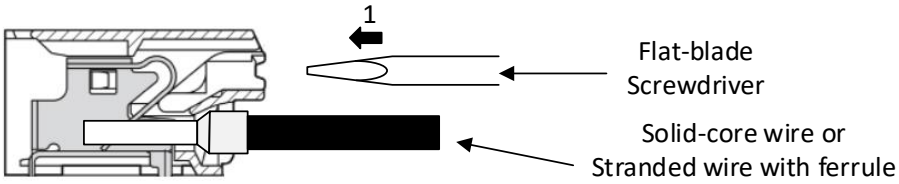
3 Pull the flat-blade screwdriver away from the release button.



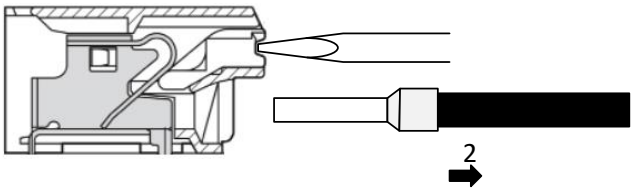
Removing Wire

Use the following procedure to remove the wires from the terminal block. The removal method is the same for ferrules, stranded wires with ferrule, and solid wires. If wires are secured firmly to the terminal block, release them first.

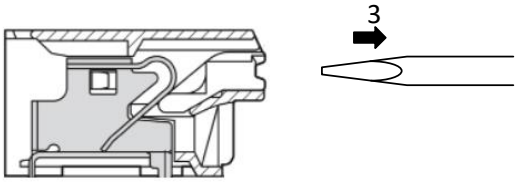
1 Press a flat-blade screwdriver straight against the release button from the terminal block front.



2 Leave the flat-blade screwdriver pressed against the release button and pull out the wire from the terminal hole.



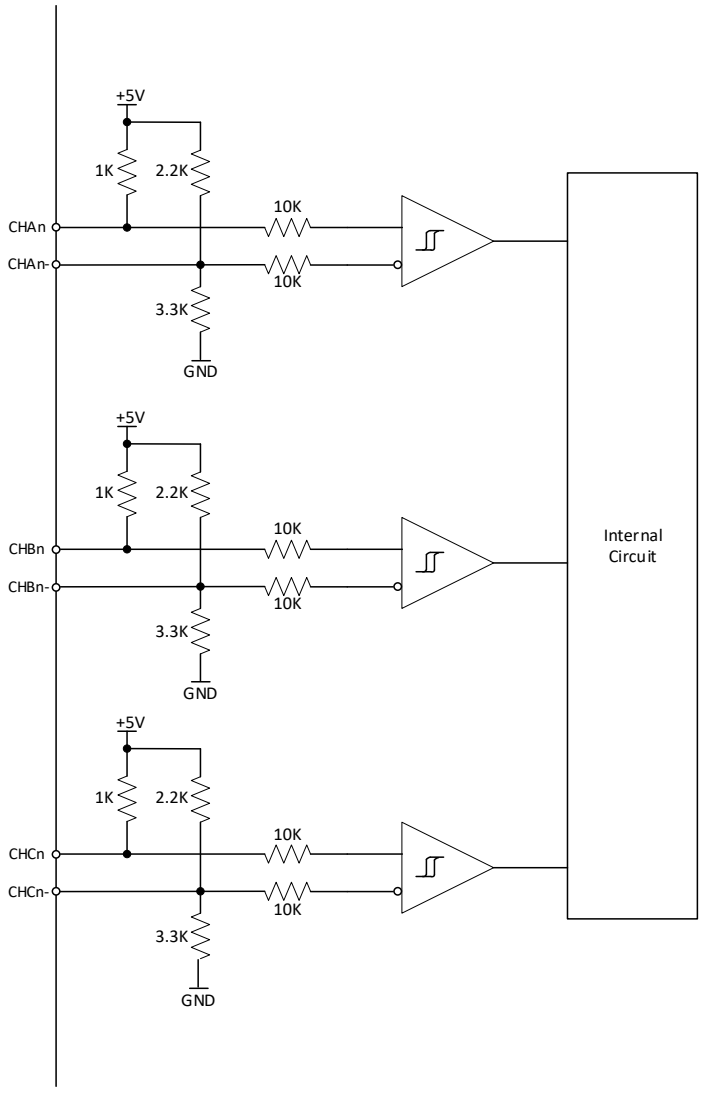
3 Pull the flat-blade screwdriver away from the release button.

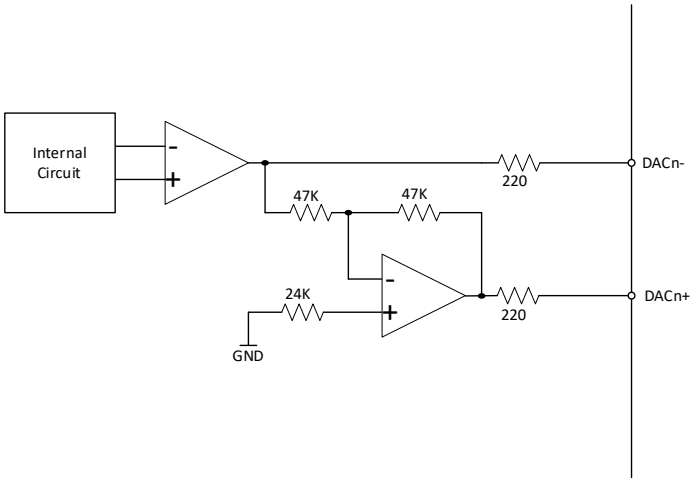


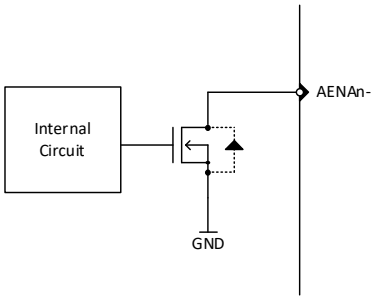
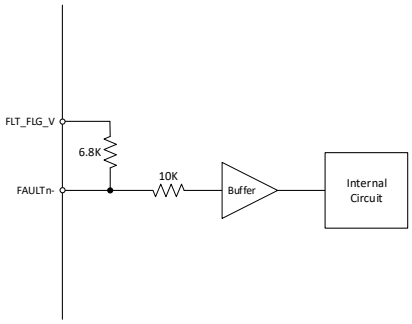
4-3 Machine Connector (J3)

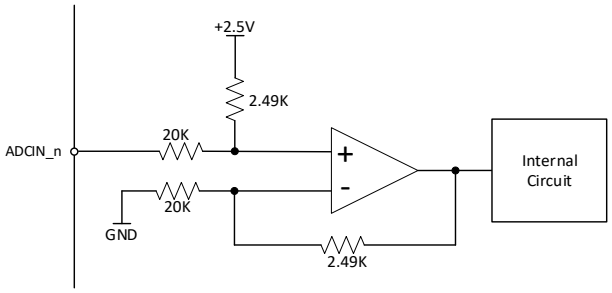
4-3-1 Circuit Specifications

The primary machine interface connector is J3 on the CK3C. It contains the pins for 4 channels of machine I/O: analog outputs, incremental encoder inputs, amplifier fault and enable signals, and power-supply connections.

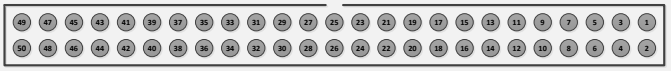
Item		Specification
Digital Encoder Input	I/O signals	CHAn+,CHAn-,CHBn+,CHBn-,CHCn+,CHCn- (n=0,1,2,3)
	Sensor type	Incremental quadrature encoder (differential and single ended)
	Rated input voltage	5 VDC
	Maximum input voltage	5.25 VDC
	Input current	4 mA typical
	ON voltage/ON current	2 VDC min. / 3 mA min.
	OFF voltage/OFF current	0.7 VDC max. / 4mA max.
	Isolation method	No Isolation
	Circuit Configuration	

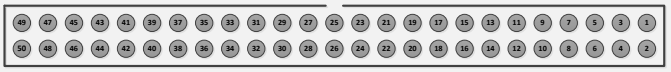
Item		Specification
DAC (Filtered PWM)	Number of points	1 output /channel
	Output method	Differential
	Output range	DACn+ : -10 to +10V (ref to GND) (n=0,1,2,3) DACn- : +10 to -10V (ref to GND) (n=0,1,2,3)
	Allowable load resistance	5 kΩ min.
	Output impedance	220 Ω typ.
	Resolution	1/10000 (13.28bit) with PWM Freq @ 30kHz
	Accuracy (to GND)	Same as existing Power Clipper 465 Controller ± 3% of full scale (25°C) ± 5% of full scale (0 to 55°C)
	Settling time (+/-1LSB)	Same as existing Power Clipper 465 Controller (At existing PMAC it is defined with 47K + 470pF CR filter)
	Isolation method	No isolation
	Circuit Configuration	

Item		Specification
Amp Enable	I/O signals	AENAn (n=0,1,2,3)
	Number of points	1 point / channel
	Output form	Open Drain Output
	Maximum switching capacity	24 VDC 0.1 A / channel
	Minimum switching capacity	5 VDC 1 mA / channel
	ON/OFF response time	10 ms max./10 ms max.
	Isolation method	No isolation
	Circuit Configuration	
Amp Fault	I/O signals	FAULTn/ (n=0,1,2,3)
	Number of points	1 point /channel
	Rated input voltage	5 VDC
	Maximum input voltage	5.25 VDC
	Input current	4 mA typical (at 5 VDC)
	ON voltage/ON current	2 VDC min. / 1 mA min.
	OFF voltage/OFF current	0.9 VDC max. 4 mA max.
	ON/OFF response time	1 µs max. / 1 µs max.
	Isolation method	No isolation
	Circuit Configuration	

Item		Specification
Analog Input	I/O signals	AIN0, AIN1, AIN2, AIN3
	Sensor type	Single ended analog input
	Rated input voltage	±10 VDC referenced to GND
	Input impedance	20 kΩ
	ADC Resolution	16-bit (0.305 mV/bit)
	Circuit Configuration	

4-3-2 Connector Pinout

J3: Machine Port Connector 50-Pin Header				
Pin #	**Symbol**	**Function**	**Description**	**Notes**
1	+5V	Output	+5V Power	For reference
2	+5V	Output	+5V Power	For reference
3	GND	Common	Digital Common	For reference
4	GND	Common	Digital Common	For reference
5	CHA0	Input	Encoder A Channel Positive	
6	CHA1	Input	Encoder A Channel Positive	
7	CHA0/	Input	Encoder A Channel Negative	1
8	CHA1/	Input	Encoder A Channel Negative	1
9	CHB0	Input	Encoder B Channel Positive	
10	CHB1	Input	Encoder B Channel Positive	
11	CHB0/	Input	Encoder B Channel Negative	1
12	CHB1/	Input	Encoder B Channel Negative	1
13	CHC0	Input	Encoder C Channel Positive	
14	CHC1	Input	Encoder C Channel Positive	
15	CHC0/	Input	Encoder C Channel Negative	1
16	CHC1/	Input	Encoder C Channel Negative	1
17	CHA2	Input	Encoder A Channel Positive	
18	CHA3	Input	Encoder A Channel Positive	
19	CHA2/	Input	Encoder A Channel Negative	1
20	CHA3/	Input	Encoder A Channel Negative	1
21	CHB2	Input	Encoder B Channel Positive	
22	CHB3	Input	Encoder B Channel Positive	
23	CHB2/	Input	Encoder B Channel Negative	1
24	CHB3/	Input	Encoder B Channel Negative	1
25	CHC2	Input	Encoder C Channel Positive	
26	CHC3	Input	Encoder C Channel Positive	
27	CHC2/	Input	Encoder C Channel Negative	1
28	CHC3/	Input	Encoder C Channel Negative	1

J3: Machine Port Connector 50-Pin Header				
29	DAC0	Output	Analog Output Positive 0	
30	DAC1	Output	Analog Output Positive 1	
31	DAC0/	Output	Analog Output Negative 0	2
32	DAC1/	Output	Analog Output Negative 1	2
33	AENA0/	Output	Amplifier-Enable 0	
34	AENA1/	Output	Amplifier -Enable 1	
35	FAULT0/	Input	Amplifier -Fault 0	
36	FAULT1/	Input	Amplifier -Fault 1	
37	DAC2	Output	Analog Output Positive 2	
38	DAC3	Output	Analog Output Positive 3	
39	DAC2/	Output	Analog Output Negative 2	2
40	DAC3/	Output	Analog Output Negative 3	2
41	AENA2/	Output	Amplifier -Enable 2	
42	AENA3/	Output	Amplifier -Enable 3	
43	FAULT2/	Input	Amplifier -Fault 2	
44	FAULT3/	Input	Amplifier -Fault 3	
45	ADC0	Input	Analog Input 0	
46	ADC1	Input	Analog Input 1	
47	FLT_FLG_V	Input	Amplifier Fault pull-up V+	
48	GND	Common	Digital Common	
49	+12V	Input	DAC Supply Voltage	
50	-12V	Input	DAC Supply Voltage	

Additional Information

Note 1: Leave this input floating if not used (i.e. digital single-ended encoders).
Note 2: Leave this output floating if not used. Do not tie to GND.

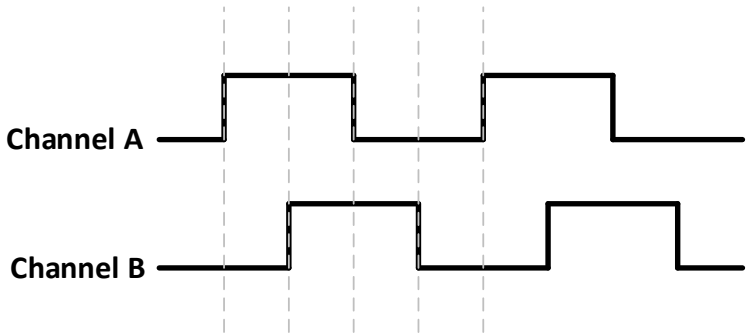
Additional Information

P/N 609-5041 by T&B Ansley or equivalent would be the mating connector for J3. If the user wishes to create a breakout, they can combine the following parts.

Part Number	Manufacturer	Description
P/N 609-5041	T&B Ansley	50-pin female flat cable connector
P/N 171-50	T&B Ansley	Standard flat cable stranded 50-wire
P/N 22 81 08 9	Phoenix Contact	Varioface module type FLKM 50 (male pins)

The standard encoder inputs on the CK3C are designed for differential quadrature type signals.

Quadrature encoders provide two digital signals to determine the position of the motor. Each nominally with 50% duty cycle, and nominally 1/4 cycle apart. This format provides four distinct states per cycle of the signal, or per line of the encoder. The phase difference of the two signals permits the decoding electronics to discern the direction of travel, which would not be possible with a single signal.



Typically, these signals are 5V TTL/CMOS level whether they are single-ended or differential.

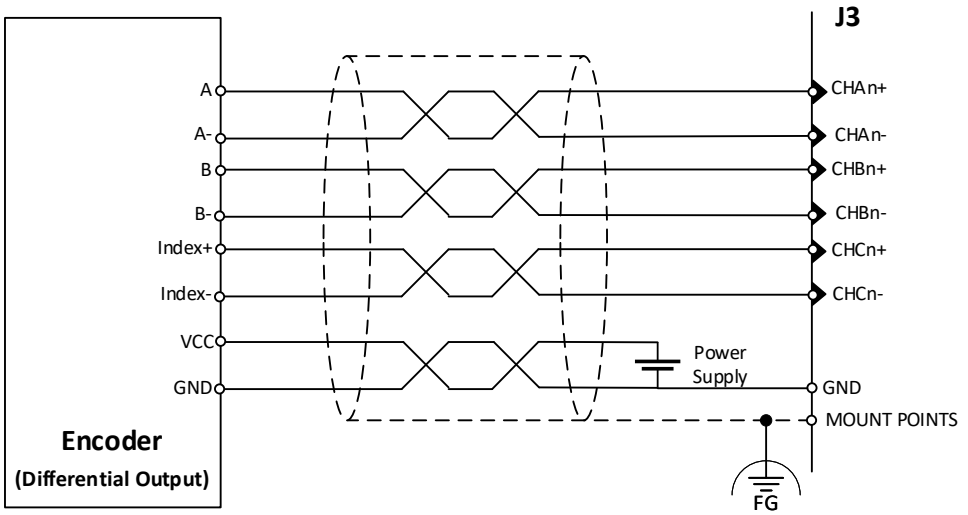
Additional Information

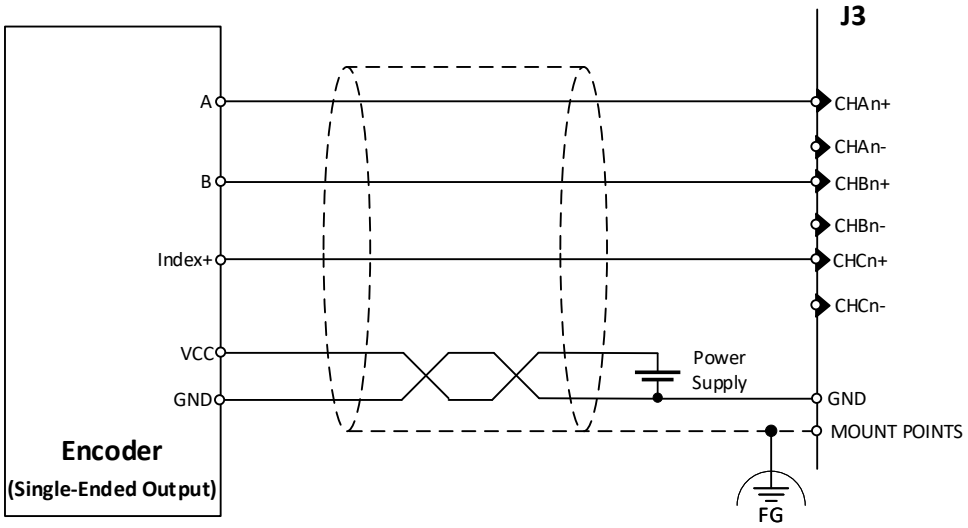
Differential signals can enhance noise immunity by providing common mode noise rejection. Modern design standards virtually mandate their use in industrial systems.

4-3-3 Wiring a Digital Quadrature Encoder

In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

Differential/Single-Ended Quadrature Encoder Wiring for Channel n





Additional Information

For single-ended encoders, leave the complementary signal pins floating – do not ground them. Some motor manufacturers bundle the hall sensors with the motor-lead cable. The hall sensors must be brought into the J7 connector.

4-3-4 Wiring an Analog Input Servo Drive

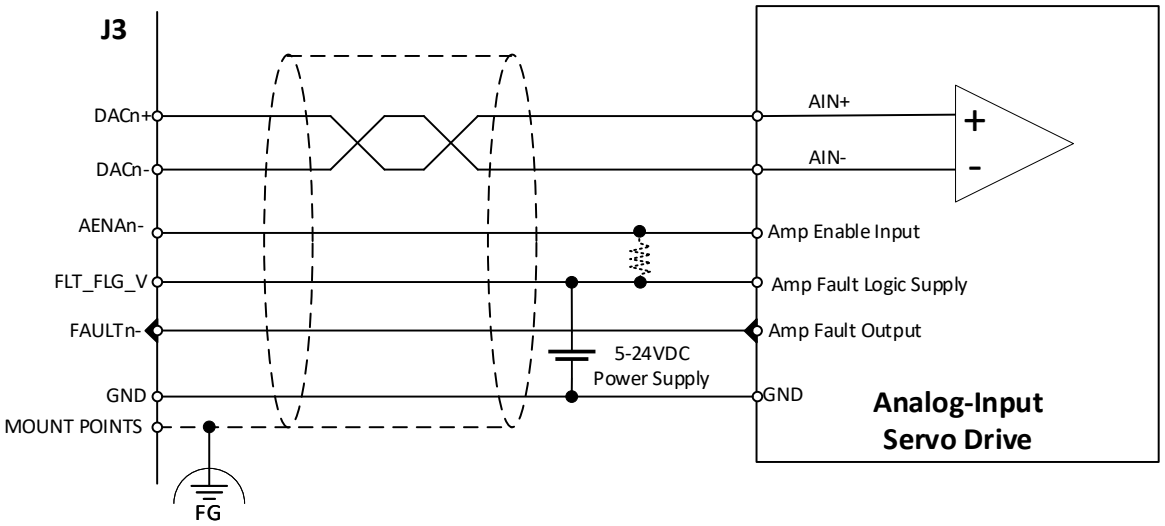
Most amplifiers have an enable/disable input that permits complete shutdown of the amplifier regardless of the voltage of the command signal. CK3C's AENA line is meant for this purpose. This signal is an open-collector output and an external 3.3 kΩ pull-up resistor can be used if necessary. In the drawing below, "n" represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel's pin number on the connector.

If Amp Enable signal on the servo drive requires any higher current than specified for AENAn- output on CK3C, use of an external isolation device such as a signal level relay is recommended.

The amplifier fault input on the J3 connector can take a signal from an amplifier (e.g. servo drive) so the CK3C knows when the amplifier is having problems and can shut down action.

The polarity is programmable with **Motor[x].AmpFaultLevel** (**Motor[1].AmpFaultLevel** for motor 1). With the default setup, this signal must actively be pulled low for a fault condition. In this setup, if nothing is wired into this input, the CK3C will consider the motor not to be in a fault condition. In the drawing below, "n" represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel's pin number on the connector.

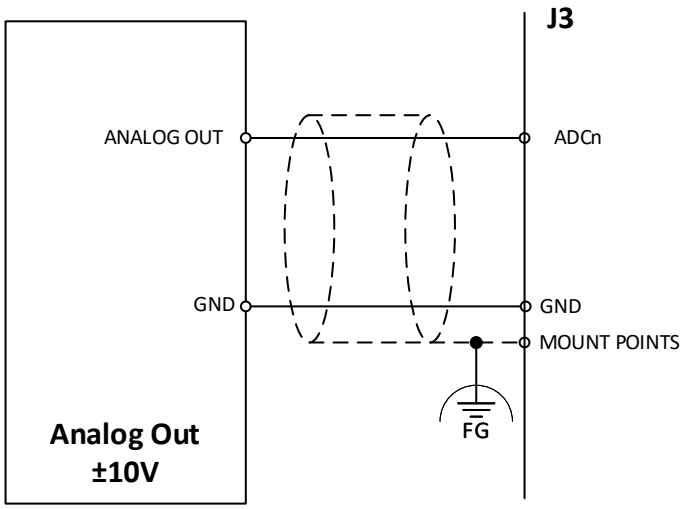
Example wiring for Channel n



4-3-5 Wiring the Analog Inputs

The CK3C provides four 16-bit analog inputs with a ±10 VDC range. The first two inputs are on J3, ADC0 and ADC1. The other ADC inputs are on the J7 connector, ADC2 and ADC3. All four inputs are referenced to digital ground on their respective connector. In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

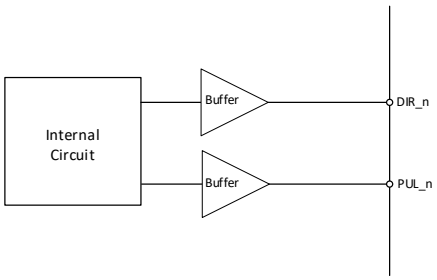
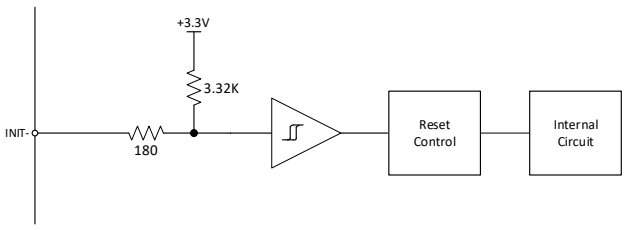
Example wiring for Channel n

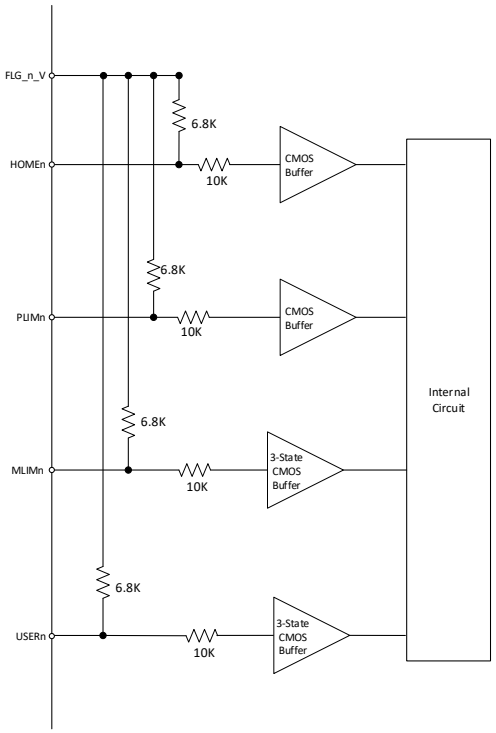


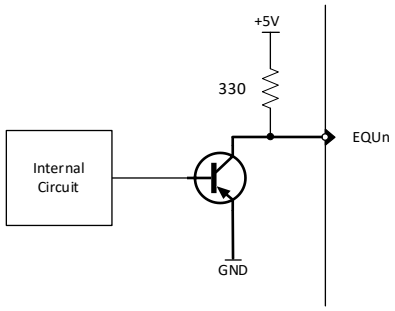
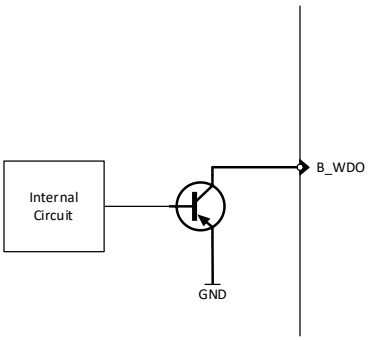
4-4 Machine Connector (J4)

4-4-1 Circuit Specifications

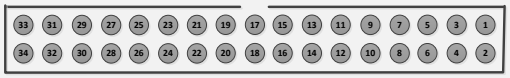
This machine interface connector is labeled J4 on the CK3C. It contains the pins for four channels of machine I/O.

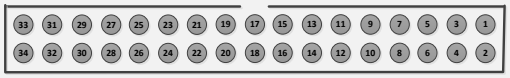
Item		Specification
Pulse & Direction (Single Ended)	I/O signals	PUL_n, DIR_n (n=0,1,2,3)
	Pulse output method	Pulse & direction outputs
	Output voltage	TTL
	Maximum output frequency	10 MHz ((PFM Clock:100MHz)/10=10MHz)
	Cable length	1 m max.
	Isolation method	No isolation
	Circuit Configuration	
Reset Input	I/O signals	INIT-
	Sensor type	Active low by pulling to GND
	Rated input voltage	Floating or 0 VDC
	Circuit Configuration	

Item		Specification
Flags	I/O signals	HOMEn, PLIMn, NLIMn, USERn / channel (n=0,1,2,3)
	Sensor type	Any sensor which can sink to GND
	Rated input voltage	5 to 24 VDC
	Maximum input voltage	26.4 VDC
	Input current	4 mA typical (at +24 VDC)
	ON voltage/ON current	2 VDC min. / 1 mA min.
	OFF voltage/OFF current	0.9 VDC max. / 4mA max.
	ON/OFF response time	HOME, PLIM, NLIM : 20 µs max./400 µs max. USER : 20 µs max./20 µs max.
	Isolation method	No Isolation
	Circuit Configuration	

Item		Specification
EQU	I/O signals	EQUn / channel (n=0,1,2,3)
	Rated voltage	5 VDC
	Maximum Load Current (L)	35 mA max.
	Pull up resistor	330 Ω ± 5%
	Leakage Current	0.1 mA max.
	Residual Voltage	0.4 VDC max.
	Isolation method	No isolation
	Circuit Configuration	
Watchdog Timer Output	I/O signals	1 Output (WDO)
	Rated voltage	5-24 VDC
	Operating load voltage range	5-24 VDC ± 200mV
	Maximum value of load current	0.1 A
	Leakage current	0.3 mA max
	Residual voltage	0.7 VDC max.
	ON/OFF response time	10 ms max./10 ms max.
	Isolation method	No isolation
	Circuit Configuration	

4-4-2 Connector Pinout

J4: Machine Port CPU Connector 34-Pin Header				
Pin #	**Symbol**	**Function**	**Description**	**Notes**
1	FLG_0_1_V	Input	Flags 0-1 Pull-Up	
2	FLG_2_3_V	Input	Flags 2-3 Pull-Up	
3	GND	Common	Digital Common	
4	GND	Common	Digital Common	
5	HOME0	Input	Home-Flag 0	1
6	HOME1	Input	Home-Flag 1	1
7	PLIM0	Input	Positive End Limit 0	
8	PLIM1	Input	Positive End Limit 1	
9	MLIM0	Input	Negative End Limit 0	
10	MLIM1	Input	Negative End Limit 1	
11	USER0	Input	User Flag 0	
12	USER1	Input	User Flag 1	
13	PUL_0	Output	Pulse Output 0	
14	PUL_1	Output	Pulse Output 1	
15	DIR_0	Output	Direction Output 0	
16	DIR_1	Output	Direction Output 1	
17	EQU0	Output	Encoder Comp-Equal 0	
18	EQU1	Output	Encoder Comp-Equal 1	
19	HOME2	Input	Home-Flag 2	1
20	HOME3	Input	Home-Flag 3	1
21	PLIM2	Input	Positive End Limit 2	
22	PLIM3	Input	Positive End Limit 3	
23	MLIM2	Input	Negative End Limit 2	
24	MLIM3	Input	Negative End Limit 3	
25	USER2	Input	User Flag 2	
26	USER3	Input	User Flag 2	
27	PUL_2	Output	Pulse Output 2	
28	PUL_3	Output	Pulse Output 3	
29	DIR_2	Output	Direction Output 2	
30	DIR_3	Output	Direction Output 3	

J4: Machine Port CPU Connector 34-Pin Header				
Pin #	**Symbol**	**Function**	**Description**	**Notes**
31	EQU2	Output	Encoder Comp-Equal 2	
32	EQU3	Output	Encoder Comp-Equal 3	
33	WDO	Output	Watchdog Out	Indicator/driver
34	INIT-	Input	PMAC Reset	Low is Reset. See note 2

Additional Information

Note 1: Functional polarity for homing or other trigger use of HOMEn controlled by Encoder/Flag Variable **CK3C[0].Chan[j].CaptCtrl**. HMFLn selected for trigger by Encoder/Flag structure **CK3C[0].Chan[j].CaptFlagSel**. Must be conducting to 0V (usually GND) to produce a 0 in PMAC software.
Note 2: Even if it is not used but connected, long cabling may pull this line low and cause PMAC to unintentionally reset.

Additional Information

P/N 609-3441 by T&B Ansley or equivalent would be the mating connector for J4. If the user wishes to create a breakout, they can combine the following parts.

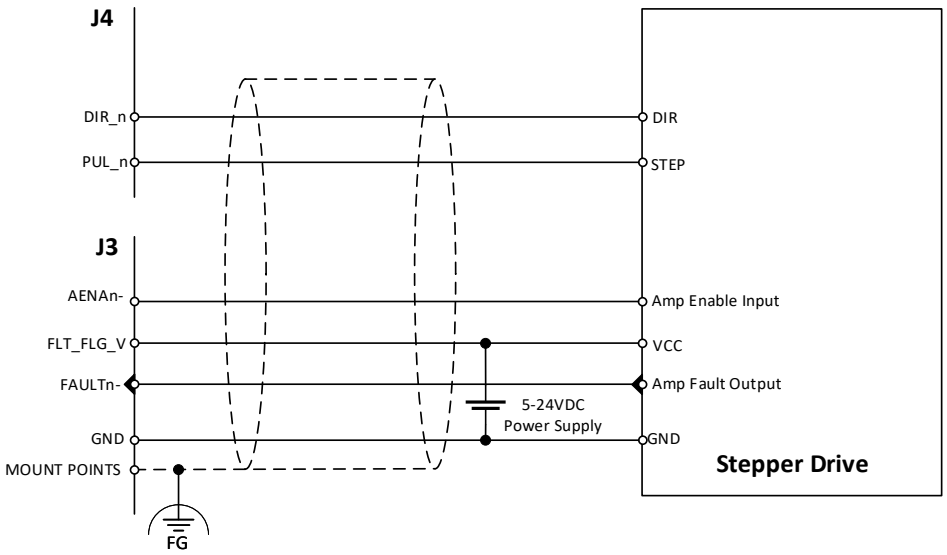
Part Number	Manufacturer	Description
P/N 609-3441	T&B Ansley	34-pin female flat cable connector
P/N 171-34	T&B Ansley	Standard flat cable stranded 34-wire
P/N 22 81 06 3	Phoenix Contact	Varioface module type FLKM 34 (male pins)

4-4-3 Wiring a Stepper Drive

The CK3C has the capability of generating pulse and direction (PFM) output signals to external stepper amplifiers. The pulse and direction outputs can be connected in single-ended configuration for 5V (input signal) amplifiers. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel's pin number on the connector.

In the drawing below, "n" represents all four channels. The primary purpose of the drawing is to illustrate the J4 connection.

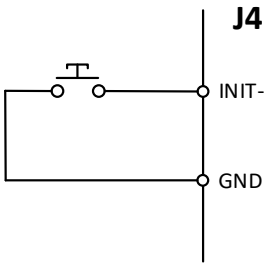
Example wiring for Channel n



4-4-4 Wiring the PMAC Reset Input

The Reset input will issue a PMAC Reset when the line is pulled low. The effect on the application is equivalent to cycling power off, then on. This can be convenient for a user that wants to force a hard reset on the controller without removing / disconnecting the power. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate pin number on the connector.

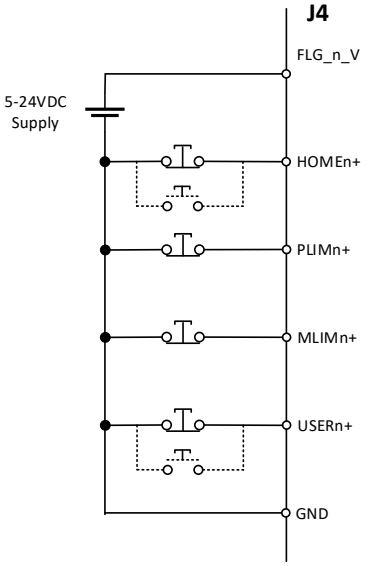
Example Wiring for PMAC Reset Input



4-4-5 Wiring the Limits and Flags

PLIMn and MLIMn are direction-sensitive over-travel limits that must conduct current to permit motion in that direction. If no over-travel switches will be connected to a particular motor, this feature must be disabled in the software by setting **Motor[x].pLimits= 0**. In the drawing below, "n" represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel's pin number on the connector.

Example wiring for Limits and Flags



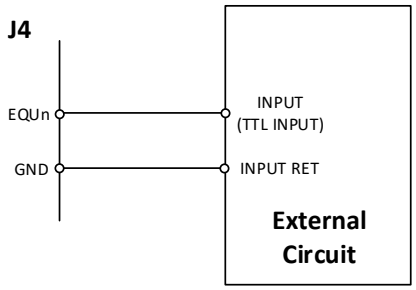
Additional Information

While normally closed-to-ground switches are required for the over-travel limits inputs, the home and user switches could be either normally close or normally open types. The polarity is determined by the homing setup, possibly through **CK3C[0].Chan[j].CaptCtrl**.

4-4-6 Wiring the Compare Equal Outputs

The compare-equal (EQU) outputs have a dedicated use of providing a signal edge when an encoder position reaches a pre-loaded value. This is very useful for scanning and measurement applications. Instructions for use of these outputs are covered in detail in the Power PMAC User Manual. In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

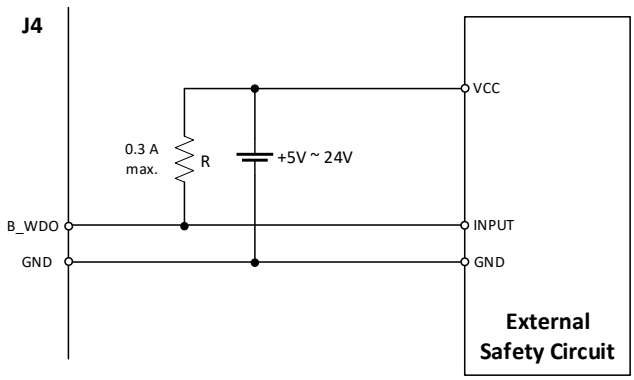
Example Wiring for Channel n



4-4-7 Wiring an External Safety Circuit

The drawing below describes how to wire the watchdog output to an external safety circuit. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate pin number on the connector.

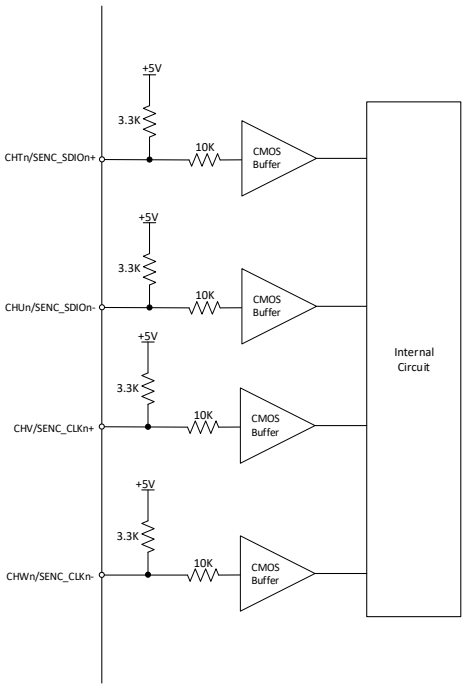
Wiring the Watchdog Output

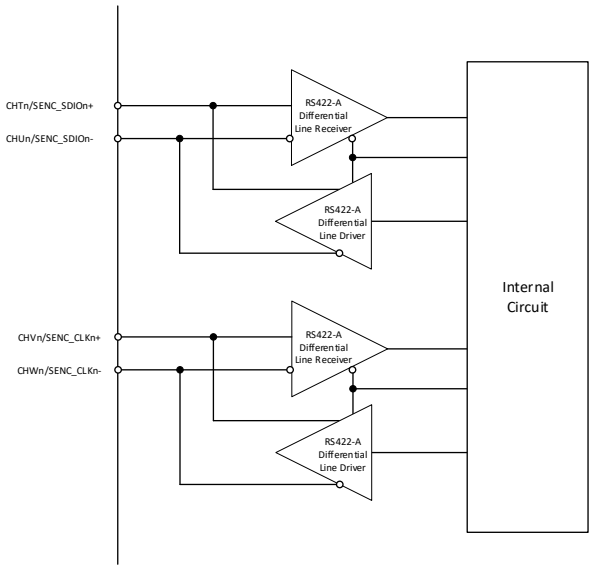


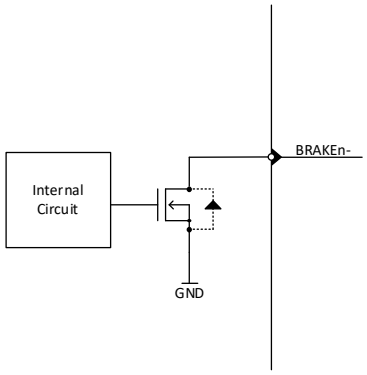
4-5 Machine Connector (J7)

4-5-1 Circuit Specifications

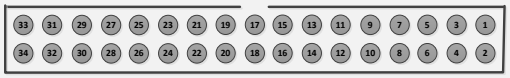
The J7 connector contains pins for four channels of Gate3 serial encoders and is shared with the T, U, V, and W flags normally used for hall device commutation. ADC inputs 3 and 4 are located on this connector. The J7 connector also has four channels of brake outputs.

Item		Specification
Hall Sensor Input	I/O signals	4 points/channel (CHUn, CHVn, CHWn, CHTn) Shared with Serial Encoder Interface (n=0,1,2,3)
	Sensor type	A sensor that detects the rotor position of the motor by detecting the magnetic field
	Rated input voltage	5 VDC
	Maximum input voltage	5.5 VDC
	Input current	0.5 mA typical (at 0 VDC)
	ON voltage/ON current	2 VDC min./0.5 mA min.
	OFF voltage/OFF current	0.8 VDC max./0.5 mA max.
	ON/OFF response time	10 µs max./10 µs max.
	Isolation method	Not isolated
	Circuit Configuration	

Item		Specification
Serial Encoder Input	I/O signals	4 points/channel (SENC_SDIOn+, SENC_SDIOn-, SENC_CLKn+, SENC_CLKn-) Shared with Hall sensor inputs (T, U, V, W) (n=0,1,2,3)
	Sensor type	Contact your OMRON representative for information on the supported protocols
	Interface	CLK/SDIO have RS-422/RS-485 interface. External termination resistor for SDIO± may be necessary to support long (+1.5m) cable
	Isolation method	Non-isolated
	Circuit Configuration	

Item		Specification
Brake Output	I/O signals	BRAKEn / channel (n=0,1,2,3)
	Output form	Open drain output (Under the brake condition, the output is closed)
	Rated voltage	5-24 VDC
	Operating load voltage range	5-24 VDC ± 0.2 VDC
	Maximum value of load current	0.1 A
	Leakage current	0.3 mA max
	Residual voltage	0.7 VDC max.
	ON/OFF response time	10 ms max./10 ms max.
	Isolation method	No Isolation
	Circuit Configuration	

4-5-2 Connector Pinout

J7: Machine Port 26-Pin Header			
Pin #	**Symbol**	**Function**	**Description**
1	ADC2	Input	General Purpose ADC 2
2	ADC3	Input	General Purpose ADC 3
3	+5V	Output	
4	+5V	Output	
5	BRAKE0	Output	Brake output for Channel 0
6	BRAKE1	Output	Brake output for Channel 1
7	BRAKE2	Output	Brake output for Channel 2
8	BRAKE3	Output	Brake output for Channel 3
9	CHT0	Input	T-flag/Serial Encoder Data+ Input for channel 0
10	CHT1	Input	T-flag/Serial Encoder Data+ Input for channel 1
11	CHT2	Input	T-flag/Serial Encoder Data+ Input for channel 2
12	CHT3	Input	T-flag/Serial Encoder Data+ Input for channel 3
13	GND	Common	
14	GND	Common	
15	CHU0	Input	U-flag/Serial Encoder Data- input for channel 0
16	CHU1	Input	U-flag/Serial Encoder Data- input for channel 1
17	CHV0	Input	V-flag/Serial Encoder Clock+ input for channel 0
18	CHV1	Input	V-flag/Serial Encoder Clock+ input for channel 1
19	CHW0	Input	W-flag/Serial Encoder Clock- input for channel 0
20	CHW1	Input	W-flag/Serial Encoder Clock- input for channel 1
21	CHU2	Input	U-flag/Serial Encoder Data- input for channel 2
22	CHU3	Input	U-flag/Serial Encoder Data- input for channel 3
23	CHV2	Input	V-flag/Serial Encoder Clock+ input for channel 2
24	CHV3	Input	V-flag/Serial Encoder Clock+ input for channel 3
25	CHW2	Input	W-flag/Serial Encoder Clock- input for channel 2
26	CHW3	Input	W-flag/Serial Encoder Clock- input for channel 3

Additional Information

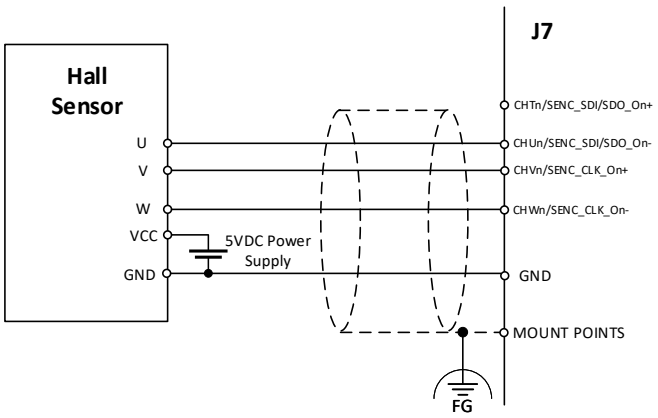
P/N 609-2641 by T&B Ansley or equivalent would be the mating connector for J7. If the user wishes to create a breakout, they can combine the following parts.

Part Number	Manufacturer	Description
P/N 609-2641	T&B Ansley	26-pin female flat cable connector
P/N 171.26	T&B Ansley	Standard flat cable stranded 26-wire
P/N 22 81 05 0	Phoenix Contact	Varioface module type FLKM 26 (male pins)

4-5-3 Wiring a Hall Sensor

In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

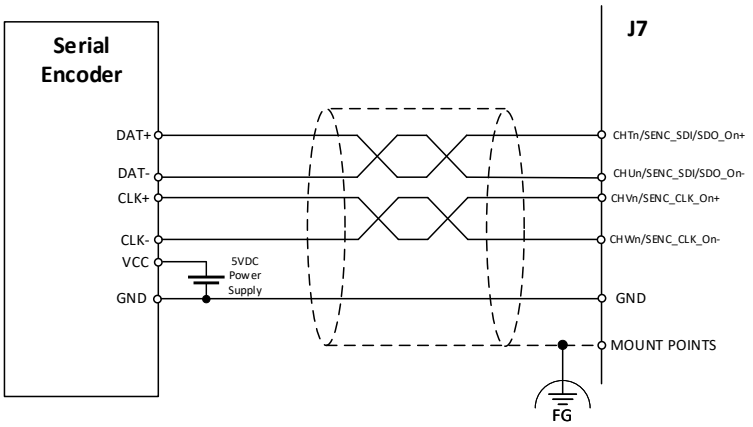
Example wiring for Channel n



4-5-4 Wiring a Serial Encoder

In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

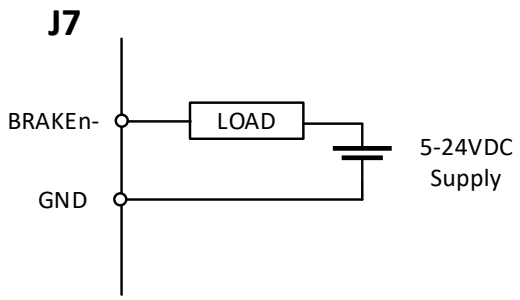
Example wiring for Channel n



4-5-5 Wiring the Brake Outputs

In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

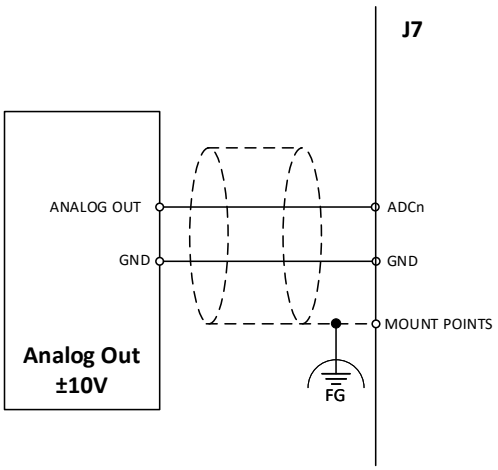
Example wiring for Channel n



4-5-6 Wiring the Analog Inputs

The CK3C provides four 16-bit analog inputs with a ±10 VDC range. The first two inputs are on J3, ADC0 and ADC1. The other ADC inputs are on the J7 connector, ADC2 and ADC3. All four inputs are referenced to digital ground on their respective connector. In the drawing below, “n” represents all four channels. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

Example wiring for Channel n



4-6 GPIO Port (J8, J9)

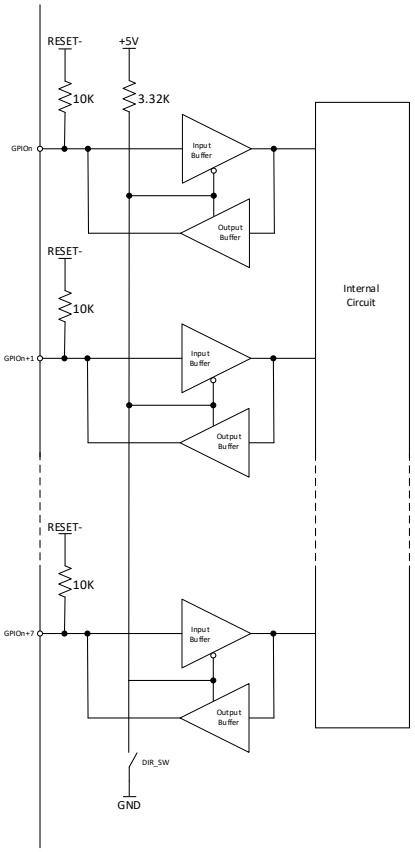
4-6-1 Circuit Specifications

The CK3C comes with 32 general-purpose digital I/O points which can be expanded through optional accessories or EtherCAT. The I/O can be configured as inputs or outputs in groups of eight. The default factory settings are 16 inputs and 16 outputs.

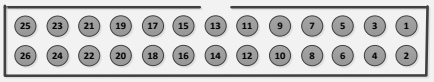
The J8 connector has 16 general-purpose digital I/O points. These may be used as general purpose I/O if the MuxIO feature is not used. With MuxIO feature of Power PMAC, this connector can be used to replicate the Thumbwheel Multiplexer port similar to legacy products such as Turbo Clipper. The direction of the input and output lines on this connector are set by Dip Switches described in section 4-1.

The J9 connector provides 16 general-purpose digital I/O points. Each input and each output has its own corresponding ground pin in the opposite row. The direction of the input and output lines on this connector are set Dip Switches described in section 4-1.

Item		Specification
Digital I/O	Digital Input (J8, J9)	
	Number of points	32 points (shared with output pins)
	Rated input voltage	5 VDC
	Maximum input voltage	5.5 VDC
	Input current	0.5 mA typical (at 5 VDC)
	ON voltage/ON current	2 VDC min./0.1 mA min.
	OFF voltage/OFF current	0.8 VDC max./0.5 mA max.
	ON/OFF response time	10 μs max./10 μs max.
	Isolation method	Not isolated
	Digital Output (J8, J9)	
	Number of points	32 points (shared with input pins)
	Rated voltage	5 VDC
	Maximum value of load current	20 mA/point
	Leakage current	5 uA max.
	Residual voltage	0.6 VDC max.
	ON/OFF response time	100 us max./100 us max.
	Number of Simultaneously ON Points	100% (at +5 VDC)
	Protective function	None
	Isolation method	Non-isolated

Item		Specification
	Circuit Configuration	

4-6-2 Connector Pinout (J8)

J8: Multiplexer Port Connector 26-Pin Header					
Pin #	**Symbol**	**Function**	**Description**	**Register**	**Notes**
1	GND	Common	PMAC Common		
2	GND	Common	PMAC Common		
3	GPIO 0	I/O	General Purpose IO	CK3C[0].GpioData[0].0	MuxIO DAT0
4	GPIO 8	I/O	General Purpose IO	CK3C[0].GpioData[0].8	MuxIO SEL0
5	GPIO 1	I/O	General Purpose IO	CK3C[0].GpioData[0].1	MuxIO DAT1
6	GPIO 9	I/O	General Purpose IO	CK3C[0].GpioData[0].9	MuxIO SEL1
7	GPIO 2	I/O	General Purpose IO	CK3C[0].GpioData[0].2	MuxIO DAT2
8	GPIO 10	I/O	General Purpose IO	CK3C[0].GpioData[0].10	MuxIO SEL2
9	GPIO 3	I/O	General Purpose IO	CK3C[0].GpioData[0].3	MuxIO DAT3
10	GPIO 11	I/O	General Purpose IO	CK3C[0].GpioData[0].11	MuxIO SEL3
11	GPIO 4	I/O	General Purpose IO	CK3C[0].GpioData[0].4	MuxIO DAT4
12	GPIO 12	I/O	General Purpose IO	CK3C[0].GpioData[0].12	MuxIO SEL4
13	GPIO 5	I/O	General Purpose IO	CK3C[0].GpioData[0].5	MuxIO DAT5
14	GPIO 13	I/O	General Purpose IO	CK3C[0].GpioData[0].13	MuxIO SEL5
15	GPIO 6	I/O	General Purpose IO	CK3C[0].GpioData[0].6	MuxIO DAT6
16	GPIO 14	I/O	General Purpose IO	CK3C[0].GpioData[0].14	MuxIO SEL6
17	GPIO 7	I/O	General Purpose IO	CK3C[0].GpioData[0].7	MuxIO DAT7
18	GPIO 15	I/O	General Purpose IO	CK3C[0].GpioData[0].15	MuxIO SEL7
19	N.C.	N.C.	No Connection		
20	GND	Common	PMAC Common		
21	N.C.	N.C.	No Connection		
22	GND	Common	PMAC Common		
23	N.C.	N.C.	No Connection		
24	GND	Common	PMAC Common		PMAC Common
25	+5V	Output	+5 VDC Supply		Power supply out
26	INIT-	Input	PMAC Reset		Low is Reset

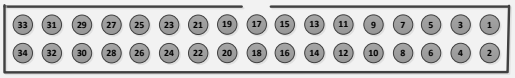
Additional Information

P/N 609-2641 by T&B Ansley or equivalent would be the mating connector for J8. If the user wishes to create a breakout, they can combine the following parts.

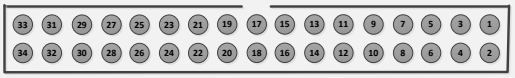
Part Number	Manufacturer	Description
P/N 609-2641	T&B Ansley	26-pin female flat cable connector
P/N 171.26	T&B Ansley	Standard flat cable stranded 26-wire
P/N 22 81 05 0	Phoenix Contact	Varioface module type FLKM 26 (male pins)

4-6-3 Connector Pinout (J9)

J9: I/O Port Connector
34-Pin Header



Pin #	Symbol	Function	Description	Register	Notes
1	GPIO 31	I/O	General Purpose IO	CK3C[0].GpioData[0].31	Legacy MI8
2	GND	Common	PMAC Common		
3	GPIO 30	I/O	General Purpose IO	CK3C[0].GpioData[0].30	Legacy MI7
4	GND	Common	PMAC Common		
5	GPIO 29	I/O	General Purpose IO	CK3C[0].GpioData[0].29	Legacy MI6
6	GND	Common	PMAC Common		
7	GPIO 28	I/O	General Purpose IO	CK3C[0].GpioData[0].28	Legacy MI5
8	GND	Common	PMAC Common		
9	GPIO 27	I/O	General Purpose IO	CK3C[0].GpioData[0].27	Legacy MI4
10	GND	Common	PMAC Common		
11	GPIO 26	I/O	General Purpose IO	CK3C[0].GpioData[0].26	Legacy MI3
12	GND	Common	PMAC Common		
13	GPIO 25	I/O	General Purpose IO	CK3C[0].GpioData[0].25	Legacy MI2
14	GND	Common	PMAC Common		
15	GPIO 24	I/O	General Purpose IO	CK3C[0].GpioData[0].24	Legacy MI1
16	GND	Common	PMAC Common		
17	GPIO 23	I/O	General Purpose IO	CK3C[0].GpioData[0].23	Legacy MO8
18	GND	Common	PMAC Common		
19	GPIO 22	I/O	General Purpose IO	CK3C[0].GpioData[0].22	Legacy MO7
20	GND	Common	PMAC Common		
21	GPIO 21	I/O	General Purpose IO	CK3C[0].GpioData[0].21	Legacy MO6
22	GND	Common	PMAC Common		

J9: I/O Port Connector 34-Pin Header					
Pin #	**Symbol**	**Function**	**Description**	**Register**	**Notes**
23	GPIO 20	I/O	General Purpose IO	CK3C[0].GpioData[0].20	Legacy MO5
24	GND	Common	PMAC Common		
25	GPIO 19	I/O	General Purpose IO	CK3C[0].GpioData[0].19	Legacy MO4
26	GND	Common	PMAC Common		
27	GPIO 18	I/O	General Purpose IO	CK3C[0].GpioData[0].18	Legacy MO3
28	GND	Common	PMAC Common		
29	GPIO 17	I/O	General Purpose IO	CK3C[0].GpioData[0].17	Legacy MO2
30	GND	Common	PMAC Common		
31	GPIO 16	I/O	General Purpose IO	CK3C[0].GpioData[0].16	Legacy MO1
32	GND	Common	PMAC Common		
33	+5V	Output	+5 VDC Power I/O		
34	GND	Common	PMAC Common		

Additional Information

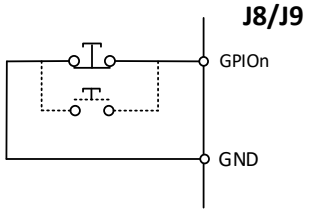
P/N 609-3441 by T&B Ansley or equivalent would be the mating connector for J9. If the user wishes to create a breakout, they can combine the following parts.

Part Number	Manufacturer	Description
P/N 609-3441	T&B Ansley	34-pin female flat cable connector
P/N 171-34	T&B Ansley	Standard flat cable stranded 34-wire
P/N 22 81 06 3	Phoenix Contact	Varioface module type FLKM 34 (male pins)

4-6-4 Wiring a General Purpose Input

In the drawing below, “n” represents the I/O point number. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate I/O point pin number on the connector.

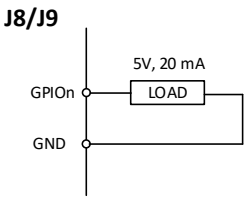
Example wiring of a General Purpose Input



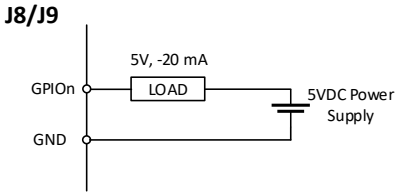
4-6-5 Wiring a General Purpose Output

In the drawing below, “n” represents the I/O point number. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate I/O point pin number on the connector.

Example Wiring of a General Purpose Output (PNP)



Example Wiring of a General Purpose Output (NPN)



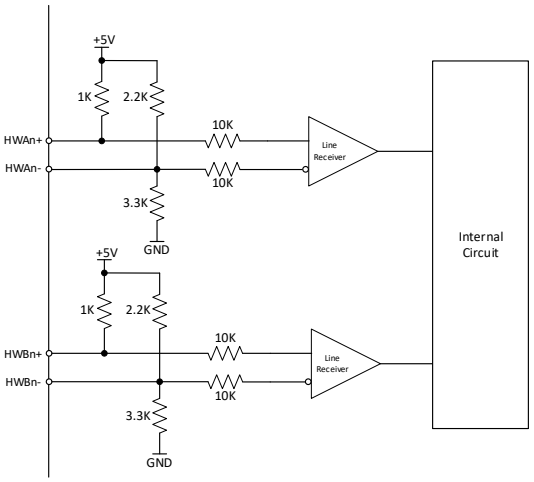
Additional Information

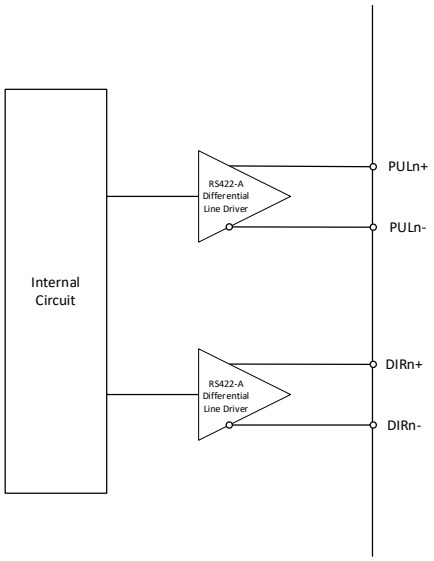
Use external +5V supply and connect to the same ground as the CK3C as illustrated in the image above.

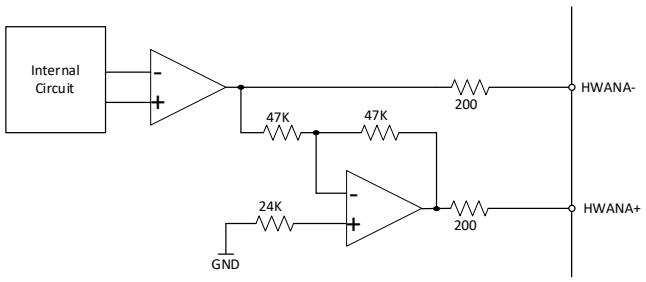
4-7 Handwheel/Pulse-Dir Port (J10)

4-7-1 Circuit Specifications

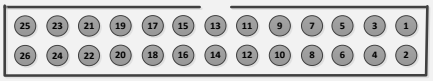
The J10 port provides two differential Quadrature encoder inputs (HW0 and HW1) and two differential PFM outputs or PWM output pairs. There is no index channel on HW0 and HW1. The serial encoders on the CK3C's channels 1 and 2 are shared with HW0 and HW1 respectively and DIP Switches 1 and 2 select which is active. Refer to section 4-1 for DIP switch settings.

Item		Specification
Handwheel Encoder Input	I/O signals	HWAn+,HWAn-,HWBn+,HWBn- for n=0,1
	Sensor type	Incremental quadrature encoder (differential and single ended)
	Rated input voltage	5 VDC
	Maximum input voltage	5.25 VDC
	Input current	4 mA typical
	ON voltage/ON current	2 VDC min. / 3 mA min.
	OFF voltage/OFF current	0.7 VDC max. / 4mA max.
	Isolation method	Not Isolated
	Circuit Configuration	

Item		Specification
Pulse and Direction Output (Differential)	I/O signals	Output: 1 pulse output pair, and 1 direction output pair / channel (for channels 0 and 1)
	Pulse output method	Pulse & direction outputs
	Pulse output form	Line driver output
	Minimum termination load	120 Ω
	Output voltage	RS422-A line driver level
	Maximum output frequency	10 MHz ((PFM Clock:100MHz)/10=10MHz)
	Cable length	Line driver outputs: 10 m max.
	Isolation method	Non-isolated
	Circuit Configuration	

Item		Specification
Analog Output	I/O signals	HWANA
	Number of points	1 points /channel
	Output method	Differential
	Output range	HWANA+ : -10 to +10V (ref to GND) HWANA - : +10 to -10V (ref to GND)
	Allowable load resistance	5 kΩ min.
	Output impedance	220 Ω
	Resolution	1/10000 (13.28bit) with PWM Freq @ 30kHz
	Accuracy (to GND)	Same as existing PMAC ± 3 % of full scale (25°C) ± 5 % of full scale (0 to 55°C)
	Settling time (+/-1LSB)	Same as existing PMAC (At Existing PMAC ,it is defined with 47K + 470pF CR filter)
	Isolation method	Non-isolated
	Circuit Configuration	

4-7-2 Connector Pinout

J10 (JHW) Handwheel Encoder Connector 26-Pin Header			
Pin #	**Symbol**	**Function**	**Description**
1	GND	Common	Reference voltage
2	+5V	Output	Supply voltage
3	HWA0+	Input	HW0 channel A+
4	HWA0-	Input	HW0 channel A-
5	HWB0+	Input	HW0 channel B+
6	HWB0-	Input	HW0 channel B-
7	HWA1+	Input	HW1 channel A+
8	HWA1-	Input	HW1 channel A-
9	HWB1+	Input	HW1 channel B+
10	HWB1-	Input	HW1 channel B-
11	PUL0+	Output	PULSE0+ output
12	PUL0-	Output	PULSE0- output
13	DIR0+	Output	DIRECTION0+ output
14	DIR0-	Output	DIRECTION0- output
15	PUL1+	Output	PULSE1+ output
16	PUL1-	Output	PULSE1- output
17	DIR1+	Output	DIRECTION1+ output
18	DIR1-	Output	DIRECTION1- output
19-22	NC		Not connected
23	HWANA+	Output	Filtered PWM DAC+
24	HWANA-	Output	Filtered PWM DAC-
25	GND	Common	Reference voltage
26	+5V	Output	Supply voltage

Additional Information

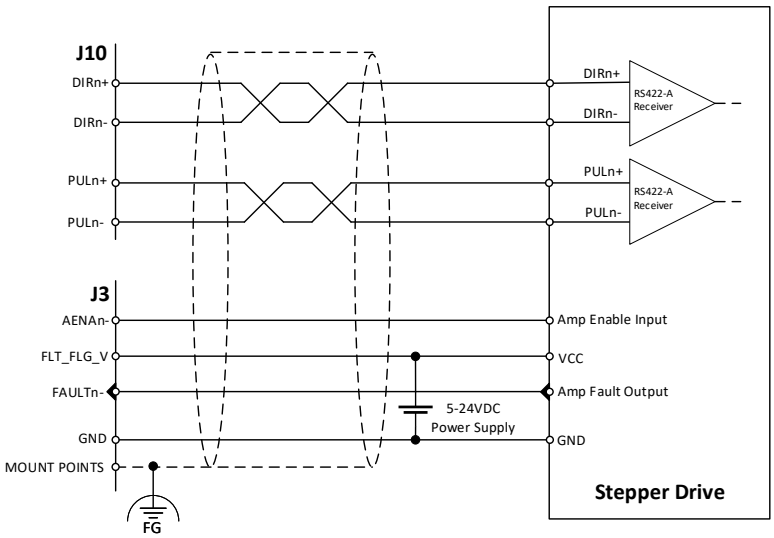
P/N 609-2641 by T&B Ansley or equivalent would be the mating connector for J10. If the user wishes to create a breakout, they can combine the following parts.

Part Number	Manufacturer	Description
P/N 609-2641	T&B Ansley	26-pin female flat cable connector
P/N 171.26	T&B Ansley	Standard flat cable stranded 26-wire
P/N 22 81 05 0	Phoenix Contact	Varioface module type FLKM 26 (male pins)

4-7-3 Wiring a Stepper Drive

In the drawing below, “n” represents the user preferred channel number. The primary purpose of the drawing is to illustrate the J10 connection. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

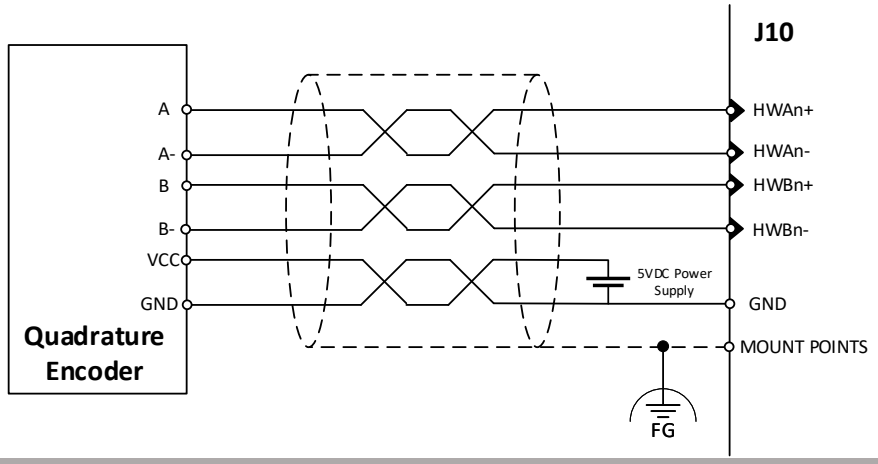
Example wiring for Channel n



4-7-4 Wiring a Quadrature Encoder

In the drawing below, “n” represents the user preferred channel number. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate channel’s pin number on the connector.

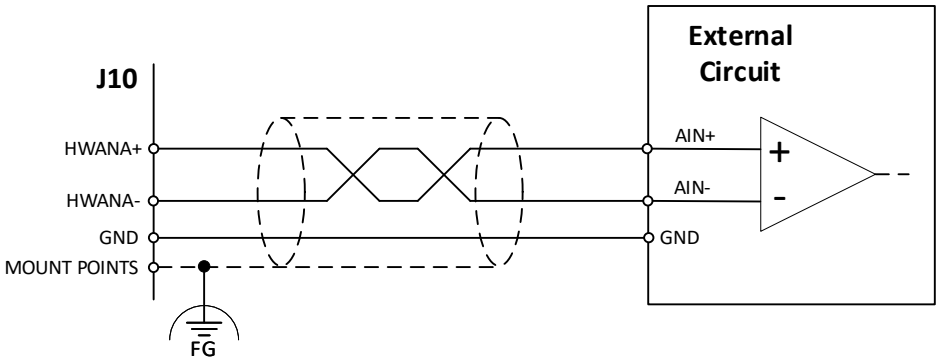
Example wiring for Channel n



4-7-5 Wiring the Analog Output

The CK3C provides an extra filtered PWM analog output with a ±10 VDC range on J10 connector. Refer to the connector pinout section to match the corresponding wire signals in the drawing to the appropriate pin number on the connector.

Example wiring for Channel n



4-8 Diagnostic Port

The USB “device” port is located on the right side of the CK3C. This port supports a Micro-USB connector and it is designed for accessing the on-board flash of the CK3C. When the CK3C is not powered, this port will allow a PC to view the flash memory as a USB drive.

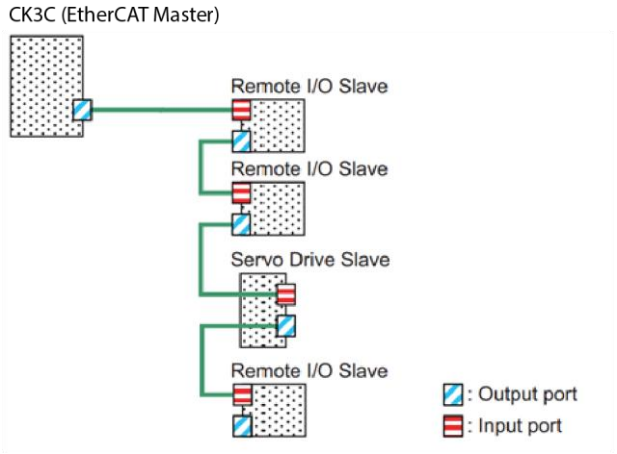
4-9 EtherCAT™/Ethernet Port

This connector is used to connect to an EtherCat™ network. It can also serve as another Ethernet communication port.
This section describes how to install EtherCAT networks.

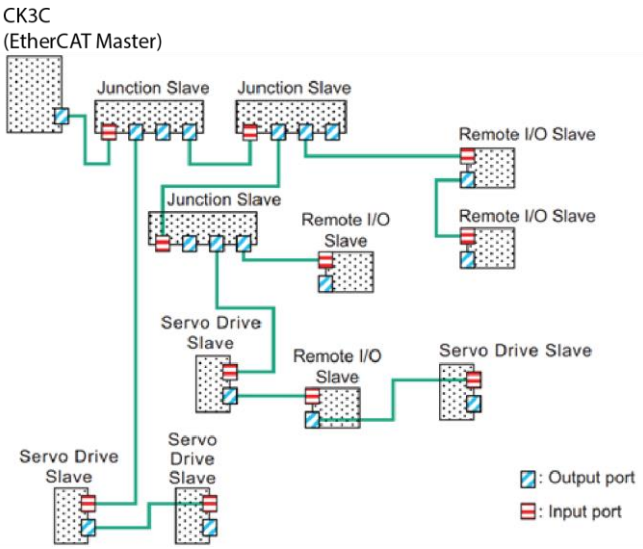
Supported Network Topologies

The EtherCAT port on the CK3C can be connected using daisy chain connections with no branching, or with branching connections using Junction Slaves. Examples of topology without branching and with branching (Junction Slaves) are shown below.

- No Branching.



- Branching.



4-10 Ethernet Communications Port

This connector is used to establish communication over Ethernet between the PC and the CK3C. It is recommended to use an RJ45 CAT5e or better shielded cable.

4-11 USB Host Port

The USB host port is located on the opposite side of the power input port. It is a USB Type-A connector. With this port, the CK3C acts as the host computer and various peripheral devices can be connected through this port.

The most common peripheral device used on this port is a USB flash drive. The CK3C board will automatically recognizes common standard formats (vfat, fat32, etc.) for flash drives connected to this port.

Model	Manufacturer	Appearance	Capacity
FZ-MEM2G	OMRON Corporation		2GB
FZ-MEM8G	OMRON Corporation		8GB

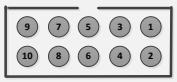
For using any other USB devices through this port, please consult with your Omron representative.

4-11-1 Circuit Specifications

Item		Specification
USB port	Physical layer	USB 3.0 compliant, type A connector. Output voltage: 5 V, 0.9 A max.
	Transmission distance	3 m max.
	Isolation	No isolation

4-12 Serial Port

This connector allows console communication with the CK3C through an RS-232 interface. This port is only intended for factory use and diagnosis.

J2: Serial Port Connector 10-Pin Header				
Pin #	**Symbol**	**Function**	**Description**	**Notes**
1	NC	N/A	No Connection	
2	DTR	Bidirect	Data Terminal Ready	Tied to "DSR"
3	TXD/	Output	Send Data	Host receive data
4	CTS	Input	Clear to Send	Host ready bit
5	RXD/	Input	Receive Data	Host transmit data
6	RTS	Output	Request to Send	PMAC ready bit
7	DSR	Bidirect	Data Set Ready	Tied to "DTR"
8	NC	N/A	No Connection	
9	GND	Common	Digital Common	
10	INIT-	Input	Hardware CPU Reset	Ground is Reset

Precautions for Safe Use

Confirm the wiring of standard IDC header ribbon cables to ensure PIN 10 is not connected to a +5V on the PC side. For correct use of this pin, please refer Reset Input specification in the J4 connector section

4-13 Wiring Conforming to EMC/EU Directives

The CK3C-series Units comply with EU and EMC Directives. When the product is installed, the customer must check to confirm that the overall machine conforms to EU/EMC Directives.
To ensure that the machine or device in which the CK3C is used complies with EU Directives, the following precautions must be observed.

The following are the conditions required for conformance to the EMC and EU Directives.

- The CK3C must be installed within a control panel.
- We recommend that you use the OMRON S8VK-S series and G series DC Power Supplies. EMC standard compliance was confirmed for the recommended Power Supplies.
- This is a Class A product (for industrial environments). In a residential environment, it may cause radio interference. If radio interference occurs, the user may be required to take appropriate measures.
- The CK3C complies with EU Directives. You must therefore confirm that the overall machine or equipment in which the CK3C is used complies with EU Directives.

4-14 Selecting Peripheral Components

4-14-1 Recommended Power Supply

The logic power supply must be chosen in compliance with the specifications listed in the Specifications section of this manual. The following OMRON power supplies are recommended for use with CK3C product. For achieving ±12V supply, two S8VK-G01512 units should be used in series with center rail to be connected to the +5V supply RET connection.

Power Supply	Manufacturer
S8VK-S series	OMRON

Power Supply	Output
S8VK-G03005	+5V / 5A
S8VK-G01512	+12V / 1.2A

4-14-2 Alternate Power Supply Solution

Third party power supplies can also be used with CK3C as long as it satisfies the voltage, current and voltage ripple requirements specified in the Specifications section of this manual.

Power Supply	Manufacturer
GRN-45-3001	INTEGRATED POWER DESIGNS

5

Software Configuration

This section describes the software configuration of the CK3C.

5-1 Outline of Software Configuration 5-2
5-1-1 Summary 5-2
5-1-2 Intended Users 5-3
5-1-3 Intended Hardware 5-3
5-1-4 Gate3 Alias Name 5-3
5-2 CK3C Basic Configuration 5-4
5-2-1 System and Gate3 Settings 5-4
5-2-2 Quadrature Encoder Configuration 5-5
5-2-3 Handwheel Encoder Configuration 5-6
5-2-4 Serial Encoder Configuration 5-7
5-2-5 Setting up the Analog (ADC) Inputs 5-34
5-2-6 Limit, Flag, and Brake [Channel 0-3] Structure Elements 5-36
5-2-7 General Purpose I/O (GPIO) Configuration 5-37

5-1 Outline of Software Configuration

This section provides an overview of the Software Configuration for better understanding.

5-1-1 Summary

The following topics are discussed in the Software Configuration section.

Topic	Content
Basic Configuration	System setup elements and CK3C (Gate3) settings
	Quadrature encoder configurations
	Handwheel encoder configurations
	Serial encoder configurations
	Setting up Analog (ADC) inputs
	Sample code for 4 channels of Limit, Flag, and Brake input status's

5-1-2 Intended Users

The Software Configuration section requires general understanding of Power PMAC Gate3 architecture.

5-1-3 Intended Hardware

The Software Configuration section pertains to the CK3C.

5-1-4 Gate3 Alias Name

In the IDE script environment, the structure element associated with **Gate3[i]** can be replaced with the hardware alias name **CK3C[i]**, specific to the CK3C. Using the hardware alias can be beneficial to projects that want to call out the type of hardware being used for better clarity to the programmer. For examples of how this alias is used, refer to all the coding examples below in the Software Configuration section.

5-2 CK3C Basic Configuration

This section describes the minimum necessary settings that allow the basic configuration of the CK3C.



Additional Information

This section only discusses using the CK3C without accessories for standard filtered PWM output. Refer to Section A-2 Sample Motor Setup Examples for setup using the CK3C with various accessories to output alternate signals, including Direct PWM and True DAC.

5-2-1 System and Gate3 Settings

For proper operation of the filtered PWM outputs on the CK3C, the PWM frequency must be set to 30 KHz. The PWM frequency is dictated by the Power PMAC phase frequency (setting of **CK3C[0].PhaseFreq**) and PWM frequency multiplier (setting of **CK3C[0].Chan[j].PwmFreqMult**) which have the following relationship:

$$\text{PWM Frequency} = \frac{\text{CK3C[0].Chan[j].PwmFreqMult} + 1}{2} \times \text{Phase Frequency}$$

```
Sys.WpKey = $AAAAAAAA                                                          DISABLE WRITE-PROTECTION
Sys.MaxCoords = 2                                                              MAX. COORD SYSTEMS+1 (EFFICIENCY)
Sys.MaxMotors = 5                                                              MAX. NO. OF MOTORS+1 (EFFICIENCY)

CK3C[0].PhaseFreq = 10000                                                      10 KHZ PHASE FREQUENCY
CK3C[0].ServoClockDiv = 1                                                      5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 0                                                            5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (CK3C[0].ServoClockDiv + 1) / CK3C[0].PhaseFreq       TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (CK3C[0].ServoClockDiv + 1)                     TYPICAL CALCULATION

CK3C[0].Chan[0].PwmFreqMult = 5                                                CH0-3 30 KHZ PWM FREQUENCY
CK3C[0].Chan[1].PwmFreqMult = 5
CK3C[0].Chan[2].PwmFreqMult = 5
CK3C[0].Chan[3].PwmFreqMult = 5

CK3C[0].Chan[0].OutputMode = 0                                                 CH0-3 FILTERED PWM OUTPUT ON PHASE A-D
CK3C[0].Chan[1].OutputMode = 0
CK3C[0].Chan[2].OutputMode = 0
CK3C[0].Chan[3].OutputMode = 0

CK3C[0].Chan[0].PwmDeadTime = 0
CK3C[0].Chan[1].PwmDeadTime = 0
CK3C[0].Chan[2].PwmDeadTime = 0
CK3C[0].Chan[3].PwmDeadTime = 0

CK3C[0].Chan[0].PackInData = 0
CK3C[0].Chan[1].PackInData = 0                                                 CH0-3 IN DATA
CK3C[0].Chan[2].PackInData = 0
CK3C[0].Chan[3].PackInData = 0

CK3C[0].Chan[0].PackOutData = 0
CK3C[0].Chan[1].PackOutData = 0                                                CH0-3 OUT DATA
CK3C[0].Chan[2].PackOutData = 0
CK3C[0].Chan[3].PackOutData = 0
```

5-2-2 Quadrature Encoder Configuration

The CK3C default settings are configured for quadrature encoders. Minimal setup is required to configure them; quadrature encoder signals are processed as a single 32-bit read in the encoder conversion table (ECT). 1/T extension is done in the Gate3 "hardware".

The default ECT settings for four quadrature encoders will be:

```
EncTable[1].Type = 1                          SINGLE 32-BIT READ
EncTable[1].pEnc = CK3C[0].Chan[0].ServoCapt.a PRIMARY SOURCE, CH 1 SERVO CAPTURE
EncTable[1].pEnc1 = Sys.Pushm                 SECONDARY SOURCE, NONE
EncTable[1].Index1 = 0                        LEFT SHIFT, NONE
EncTable[1].Index2 = 0                        RIGHT SHIFT, NONE
EncTable[1].Index3 = 0
EncTable[1].Index4 = 0
EncTable[1].Index5 = 0
EncTable[1].Index6 = 0
EncTable[1].ScaleFactor = 1/256               SCALE FACTOR, LSB LOCATION
```



Additional Information

The hardware 1/T extension produces 8 bits of fractional data, thus the (1 / 256) 0.00390625 scale factor.

Channel Number	Quadrature Encoder Source Address
0	CK3C[0].Chan[0].ServoCapt.a
1	CK3C[0].Chan[1].ServoCapt.a
2	CK3C[0].Chan[2].ServoCapt.a
3	CK3C[0].Chan[3].ServoCapt.a

Activating the corresponding channel is sufficient to display counts in the position window when the motor / encoder shaft is moved by hand.

```
Motor[1].ServoCtrl = 1                        CHANNEL ACTIVATION
```

The position and velocity source(s) must be pointing to the proper ECT result. With quadrature encoders, they are initialized by the firmware as:

```
Motor[1].pEnc = EncTable[1].a                 POSITION
Motor[1].pEnc2 = EncTable[1].a                VELOCITY
Motor[2].pEnc = EncTable[2].a                 POSITION
Motor[2].pEnc2 = EncTable[2].a                VELOCITY
Motor[3].pEnc = EncTable[3].a                 POSITION
Motor[3].pEnc2 = EncTable[3].a                VELOCITY
Motor[4].pEnc = EncTable[4].a                 POSITION
Motor[4].pEnc2 = EncTable[4].a                VELOCITY
```

Counts per User Units

With quadrature encoders, the number of counts per user units (usually revolution) is 4 times the specified number of lines of the encoder. For example, a 1,000–line rotary encoder should result in 4,000 motor units per revolution.

5-2-3 Handwheel Encoder Configuration

After wiring the quadrature/handwheel encoder on the J10 connector, you can begin setting up the appropriate software elements to activate the encoder channel and read in the feedback.

In order to read quadrature/handwheel encoders, the data is accessed using CK3C serial encoder interface internal to the board, the following settings should be used to ensure the proper reading of encoder data:

```
Sys.WpKey = $AAAAAAAA                                   DISABLE WRITE-PROTECTION
CK3C[0].SerialEncCtrl = $F4000000                       DEFAULT VALUE
CK3C[0].SerialEncEna = 0                                DEFAULT VALUE

EncTable[5].Type = 1
EncTable[5].pEnc = CK3C[0].Chan[0].SerialEncDataA.a     X4 QUADRATURE COUNTER WITH 1/T EXTENSION
EncTable[5].pEnc1 = Sys.pushm
EncTable[5].index1 = 0
EncTable[5].index2 = 0
EncTable[5].index3 = 0
EncTable[5].index4 = 0
EncTable[5].ScaleFactor = 1/256                         1/T EXTENSION X256 SUBCOUNT

EncTable[6].Type = 1
EncTable[6].pEnc = CK3C[0].Chan[1].SerialEncDataA.a     X4 QUADRATURE COUNTER WITH 1/T EXTENSION
EncTable[6].pEnc1 = sys.pushm
EncTable[6].index1 = 0
EncTable[6].index2 = 0
EncTable[6].index3 = 0
EncTable[6].index4 = 0
EncTable[6].ScaleFactor = 1/256                         1/T EXTENSION X256 SUBCOUNT

                                                        CHAN 0 CORRESPONDS TO ENC 5
//CK3C[0].Chan[0].SerialEncCmd = $400                   BIT 10 FOR RESETING THE ENCODER COUNTER TO 0
//CK3C[0].Chan[0].SerialEncCmd = $800                   BIT 11 FOR CHANGING THE COUNT DIRECTION
//CK3C[0].Chan[0].SerialEncDataB.31                     BIT 31 ENCODER COUNT ERROR FLAG

                                                        CHAN 1 CORRESPONDS TO ENC 6
//CK3C[0].Chan[1].SerialEncCmd = $400                   BIT 10 FOR RESETING THE ENCODER COUNTER TO 0
//CK3C[0].Chan[1].SerialEncCmd = $800                   BIT 11 FOR CHANGING THE COUNT DIRECTION
//CK3C[0].Chan[1].SerialEncDataB.31                     BIT 31 ENCODER COUNT ERROR FLAG
```



Additional Information

Dip Switch Positions 1 & 2 must be set to OFF for the Handwheel port to pass the quadrature encoder signals through the serial interface. This allows for use of two additional quadrature encoders to be used on Channels 0 and 1. Please refer to J10 Connector in Section 4-7 to ensure proper pinouts.

Activating the corresponding channel is sufficient to display counts in the position window when the motor / encoder shaft is moved by hand.

```
Motor[5].ServoCtrl = 1                                  CHANNEL ACTIVATION
Motor[6].ServoCtrl = 1
```

The position and velocity source(s) must be pointing to the proper ECT result. With quadrature encoders, they are initialized by the firmware as:

```
Motor[5].pEnc = EncTable[5].a                           POSITION
Motor[5].pEnc2 = EncTable[5].a                          VELOCITY
Motor[6].pEnc = EncTable[6].a                           POSITION
Motor[6].pEnc2 = EncTable[6].a                          VELOCITY
```

5-2-4 Serial Encoder Configuration

Configuring CK3C serial encoder protocols is achieved through the following structure elements:

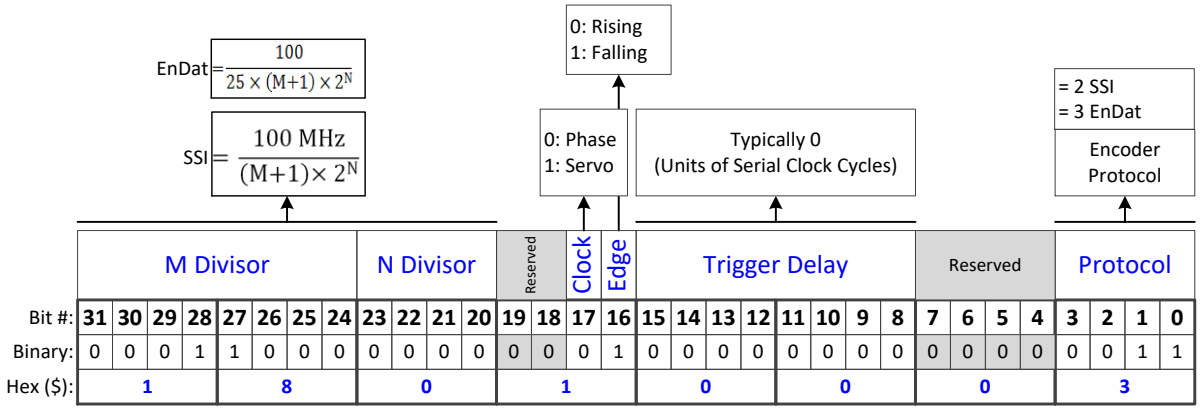
- Global Control Word, **CK3C[0].SerialEncCtrl**
- Channel Control Word, **CK3C[0].Chan[j].SerialEncCmd**
- Channel Enable Bit, **CK3C[0].Chan[j].SerialEncEna**

Global Control Register

The Global Control Word controls settings for all four channels (channels 0 – 3).

Channel Number	Global Control Register
0 - 3	CK3C[0].SerialEncCtrl

Following is a summary of the 32-bit serial control word. Detailed description can be found in the Power PMAC Software Reference and User's manual.



- Bits 20-31 specify the serial encoder interface transmission frequency, using the equations in the diagram. This frequency is usually specified by the encoder manufacturer, and typically set in the range of 1 – 16 MHz.
- Bit 17 specifies the trigger source; Phase clock is recommended.
- Bit 16 specifies the active edge; Falling edge is recommended.
- Bits 8-15 specify a trigger delay used to compensate for transmission over long encoder lines.
- Bits 0-3 specify the encoder protocol of the serial encoder.

In the example diagram above, an EnDat encoder is configured at a 4 MHz serial clock triggered at falling edge of phase clock. **CK3C[0].SerialEncCtrl** = $18010003.

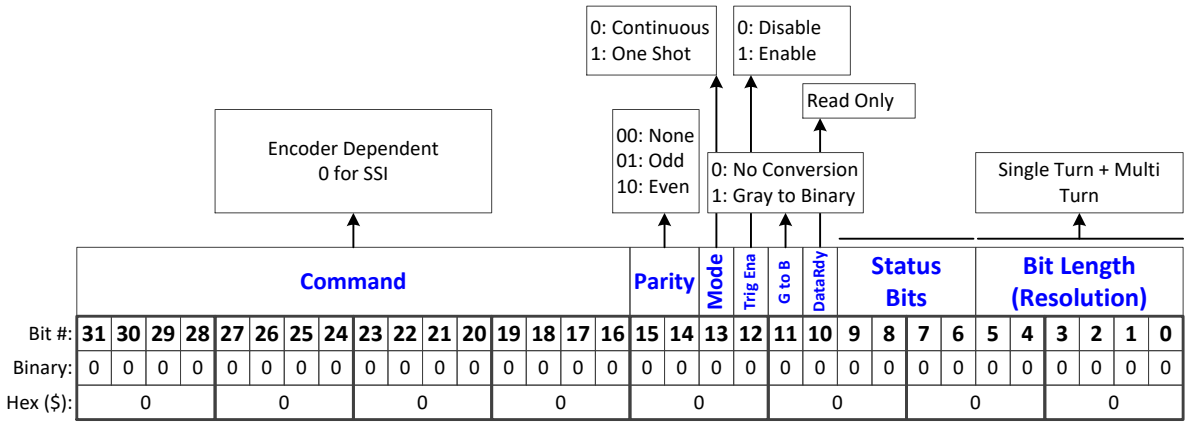
Channel Control Register

The channel control word is channel specific.

Channel Number	Channel Control Register
0	CK3C[0].Chan[0].SerialEncCmd
1	CK3C[0].Chan[1].SerialEncCmd
2	CK3C[0].Chan[2].SerialEncCmd
3	CK3C[0].Chan[3].SerialEncCmd

Following is a summary of the 32-bit channel control word. Detailed description can be found in the Power PMAC Software and User manuals.

- Bits 16-31 specify the command code. This field is protocol specific.
- Bits 14-15 specify the parity type, this is an encoder specific setting.
- Bit 13 specifies the trigger mode. Typically set to 0 for continuous (on-going position).
- Bit 12 is the trigger enable bit, must be set to 1 to trigger.
- Bit 11 specifies whether a Gray to Binary conversion is necessary.
- Bit 10 is a read-only bit.
- Bits 6-9 specify the encoder status field. This field is protocol specific.
- Bits 0-5 specify the serial encoder bit length (single-turn + multi-turn).

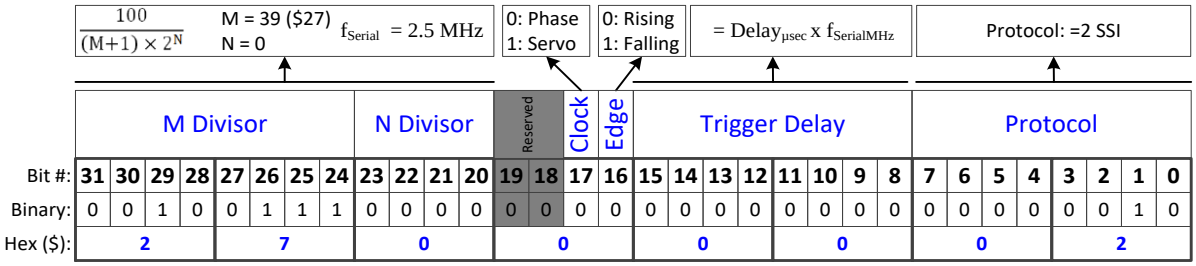


Serial Encoder Examples

● SSI Example

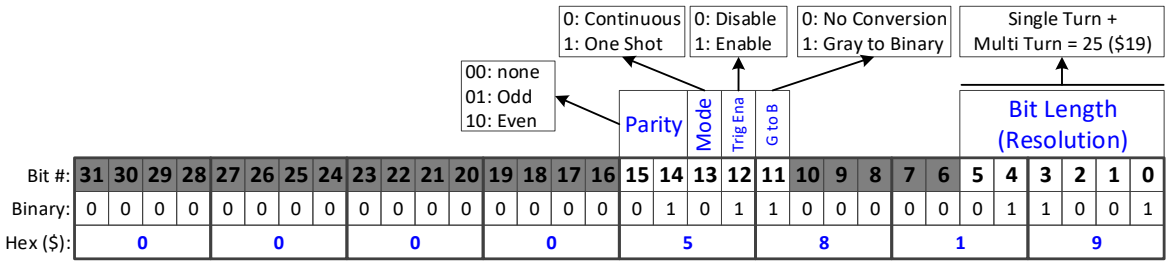
No trigger delay, rising edge of phase, and 2.5 MHz transmission

CK3C[0].SerialEncCtrl setting:



CK3C[0].Chan[j].SerialEncCmd setting:

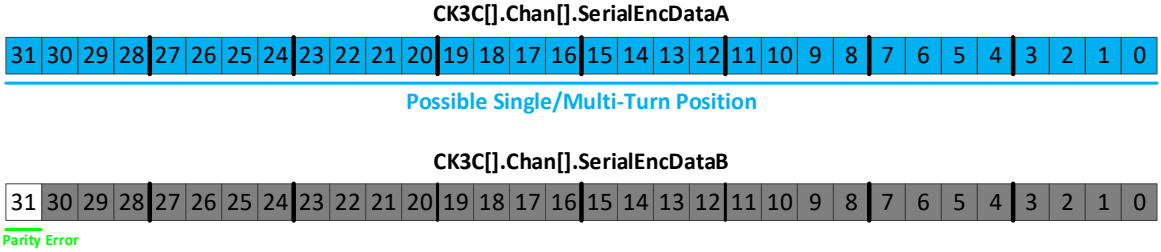
A 25-bit SSI encoder in Gray code, with odd parity




```
CK3C[0].SerialEncCtrl = $27000002
CK3C[0].Chan[0].SerialEncCmd = $5819
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

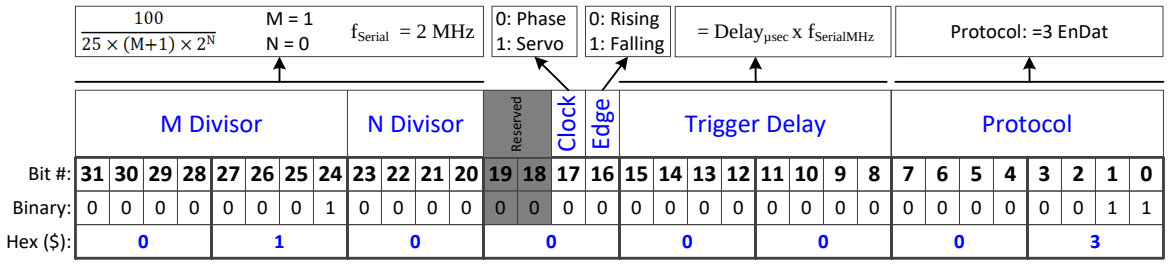
The resulting position data, status, and error bits for SSI are found in the following Serial Data Registers:



● EnDat 2.1/2.2 Example

No trigger delay, rising edge of phase, and 2.0 MHz transmission

CK3C[0].SerialEncCtrl setting:



CK3C[0].Chan[j].SerialEncCmd setting:

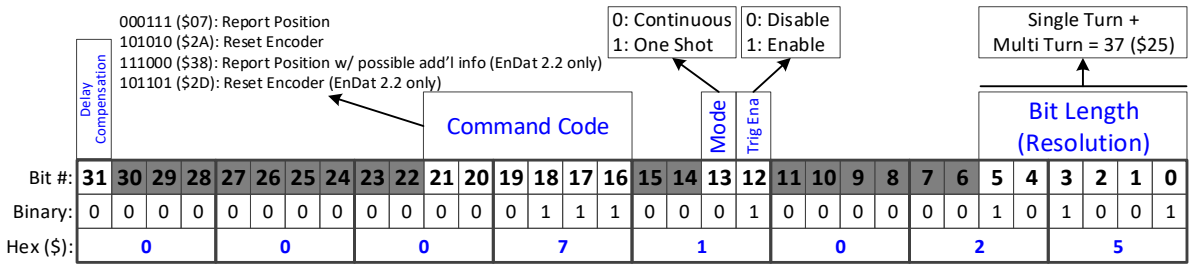
The DSPGate3 interface to EnDat supports four 6-bit command codes:

1. 000111 ($7) for reporting position (EnDat2.1/2.2).
2. 101010 ($2A) for resetting the encoder (EnDat2.1/2.2).
3. 111000 ($38) for reporting position with possible additional information (EnDat 2.2 only)
4. 101101 ($2D) for resetting the encoder (EnDat 2.2 only)

Additional Information

By the EnDat standard, EnDat 2.2 encoders should be able to accept and process EnDat 2.1 command codes. However, not all encoders sold as meeting the EnDat 2.2 standard can do this.

In this example below, a 37-bit EnDat 2.2 encoder for continuous position reporting:




```
CK3C[0].SerialEncCtrl = $1000003
CK3C[0].Chan[0].SerialEncCmd = $71025
CK3C[0].Chan[0].SerialEncEna = 1
```

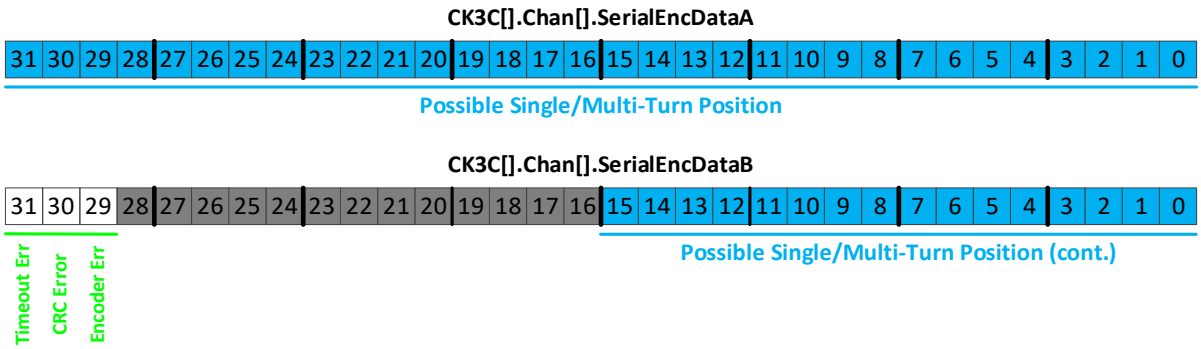
With EnDat 2.2, bit 31 is the StartDelayComp control bit. Setting this bit to 1 starts a delay identification and compensation cycle which measures the propagation delay between the encoder and the controller. The delay is measured three times and the average is used in the compensation. When these calculations are done, the StartDelayComp bit is automatically cleared. This delay identification operation must be performed after every power-up cycle. Delay compensation permits high bit transmission rates over very long cables.

To perform the delay identification and compensation cycle on this encoder, set **CK3C[0].Chan[j].SerialEncCmd** = $80071025, then wait for bit 31 to clear.

This same encoder can be reset with a command code of $2A sent in one-shot mode, so by setting **CK3C[0].Chan[j].SerialEncCmd** = $2A3025.

Serial Data Registers:

The resulting position data, status, and error bits for EnDat are found in the following Serial Data Registers:



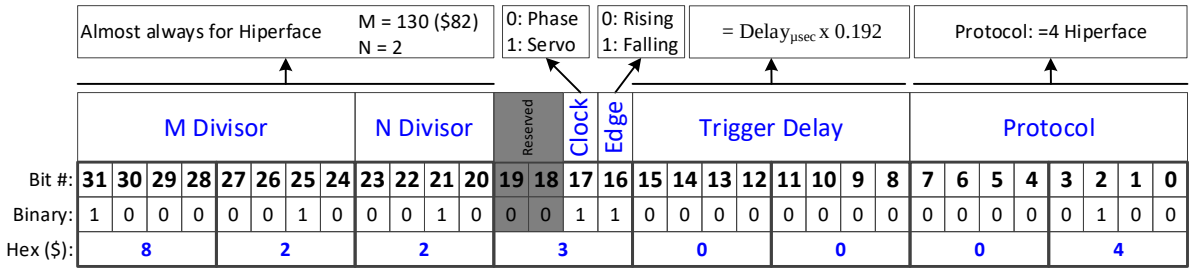
Hiperface Example

Because there is no explicit clock signal with Hiperface, the serial clock frequency is set 20 times higher than the bit transmission frequency to “oversample” the input data stream.

For the default 9600 baud transmission of the Hiperface encoder, this clock frequency should be 9.6 x 20 = 192 kHz.

Divide the 100 MHz clock by using M=130 ($82) and by N = 2 ($2) to get 192 kHz triggering on the falling edge of servo clock without delay. Since this is a “one-shot” read, the selection of the triggering clock edge does not matter much.

CK3C[0].SerialEncCtrl setting:

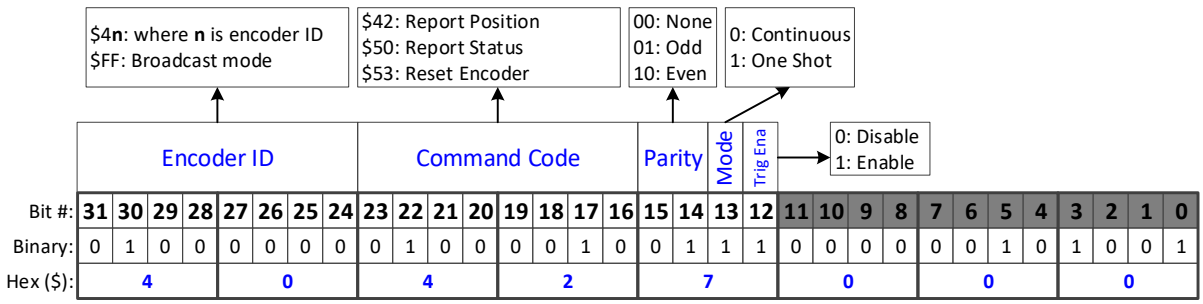


CK3C[0].Chan[j].SerialEncCmd setting:

The DSPGate3 interface to Hiperface supports three 8-bit command codes:000111 ($7) for reporting position (EnDat2.1/2.2).
$42 for reporting position.
$50 for reporting status
$53 for resetting the encoder

These command codes reside in the lower 8 bits of the Serial Encoder Command word. The upper 8 bits contain the address of the encoder in the interface. The Hiperface protocol permits up to 8 separate encoders to be “daisy-chained” on a single multi-drop interface. While this can be done, it is expected that each channel of the CK3C will be connected to a separate individual encoder, simplifying the wiring. In this configuration, this address field can either match the encoder’s address value (+ $40), or it can be set to $FF (broadcast mode).

A Hiperface encoder at user address 0 with odd parity would be set up for one-shot position reporting as follows:




```
CK3C[0].SerialEncCtrl = $82230004
CK3C[0].Chan[0].SerialEncCmd = $40427000  //or $FF427000
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

The resulting position data, status, and error bits for Hiperface are found in the following Serial Data Registers:

CK3C[].Chan[].SerialEncDataA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Possible Single/Multi-Turn Position																															

CK3C[].Chan[].SerialEncDataB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timeout Err.	Chksum Err.	Parity Err.	Encoder Err.	Encoder Error Code																											

● Yaskawa Sigma I Example

Because there is no explicit clock signal with Sigma I, the serial clock frequency is set 20 times higher than the bit transmission frequency to “oversample” the input data stream. For the default 9600 baud transmission of the Sigma I encoder, this clock frequency should be 9.6 x 20 = 192 kHz.

Divide the 100 MHz clock by M=130 ($83) and by 4 (N = 2) to get 192 kHz. Trigger on the falling edge of servo clock without delay. Since this is a “one-shot” read, the selection of the triggering clock edge does not matter much. Example settings:

CK3C[0].SerialEncCtrl setting:

	Always for Yaskawa								M = 0 N = 0				0: Phase 1: Servo	0: Rising 1: Falling	$= Delay_{\mu sec} \times f_{SerialMHz}$				Protocol = 6 Yaskawa			
	M Divisor								N Divisor				Reserved	Clock	Edge	Trigger Delay				Protocol		
Bit #:	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
Hex ($):	0				0				0				0				0				6			

CK3C[0].Chan[j].SerialEncCmd setting:

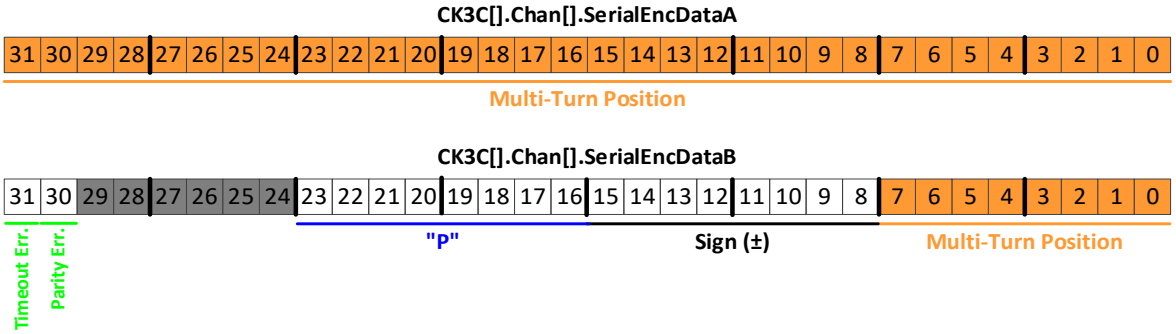
Yaskawa no longer produces Sigma I absolute encoders. However, newer generations of Yaskawa Sigma servo drives synthesize the Yaskawa Sigma I protocol for return to the controller even when using newer Sigma II, III, and V encoders. Bit 16 is set to strobe the encoder and Sigma I should use one-shot trigger, set as follows:

																0: Continuous 1: One Shot				0: Disable 1: Enable			
																Mode				Trig Ena			
Bit #:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary:	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
Hex ($):	0				0				0				1				3				0				0				0			

```
CK3C[0].SerialEncCtrl = $82230005
CK3C[0].Chan[0].SerialEncCmd = $13000
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

The resulting position data, status, and error bits for Sigma I are found in the following Serial Data Registers:



In **SerialEncDataA**, bits 0-7 represent the bits of the ASCII code for the “ones digit” of the turns count, bits 8-15 represent bits of the “tens digit”, bits 16-23 represent bits of the “hundreds digit”, bits 24-31 represent bits of the “thousands digit”.

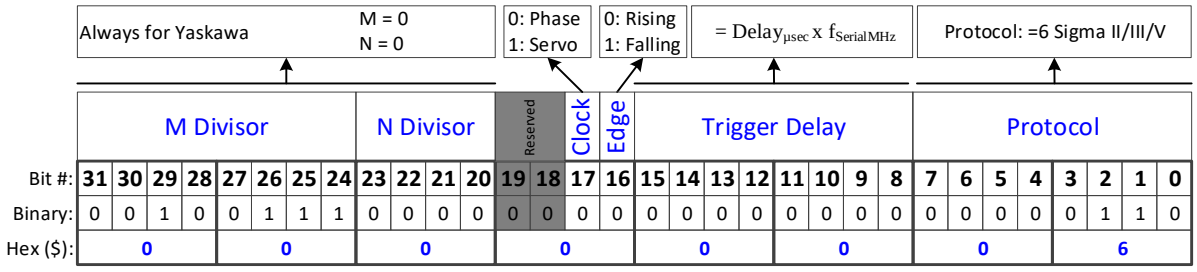
In **SerialEncDataB**, bits 0-7 represent bits of the “ten-thousands digit”, bits 8-15 represent bits of the ASCII code for the plus or minus sign, bits 16-23 represent bits of the ASCII code for the letter “P”, bits 30-31 represent bits of the error field (bit 30 is a parity error; bit 31 is a timeout error).

For each of the five numeric ASCII digits, the numeric value of the digit can be obtained by subtracting 48 ($30) from the value of the ASCII code.

- **Yaskawa Sigma II/III/V Example**

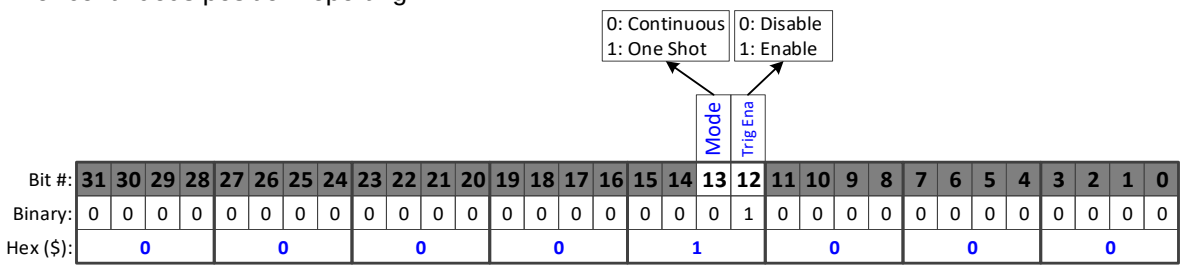
  No trigger delay, rising edge of phase, and 4.0 MHz transmission:

  CK3C[0].SerialEncCtrl setting:



CK3C[0].Chan[j].SerialEncCmd setting:

For continuous position reporting:

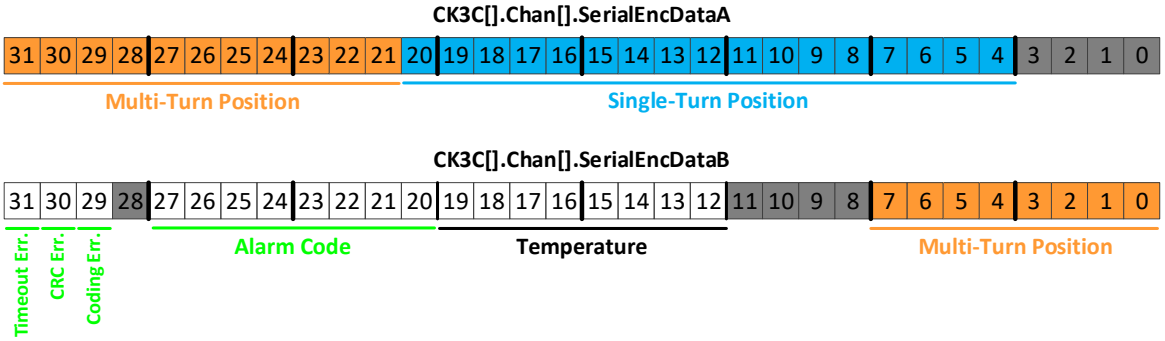



```
CK3C[0].SerialEncCtrl = $6
CK3C[0].Chan[0].SerialEncCmd = $1000
CK3C[0].Chan[0].SerialEncEna = 1
```

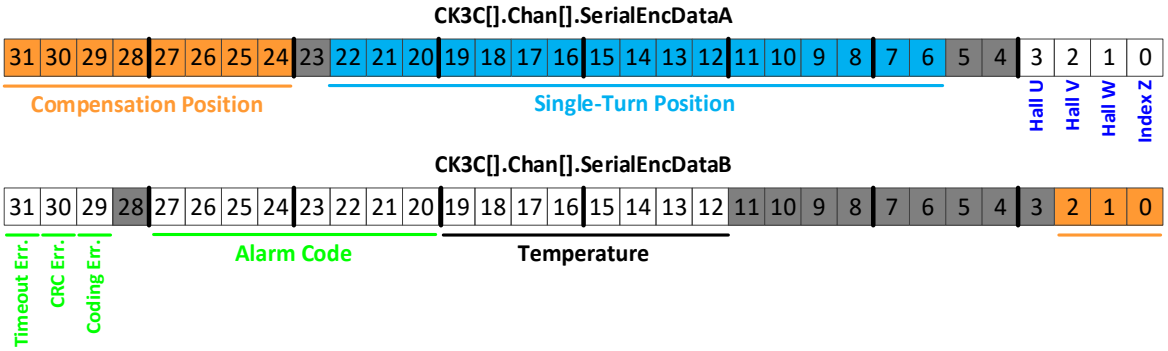
Serial Data Registers:

The resulting position data, status, and error bits for Sigma II/III/V are found in the following Serial Data Registers:

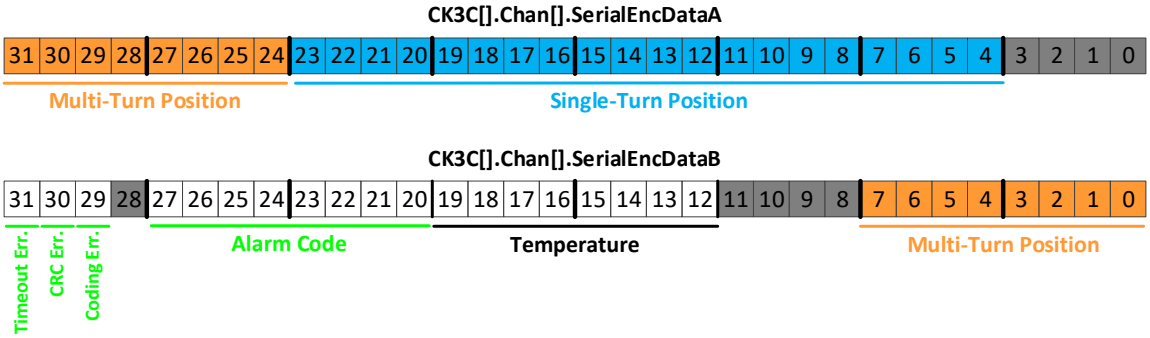
Yaskawa Sigma II (absolute 17-bit)



Yaskawa Sigma II (incremental 17-bit)



Yaskawa Sigma III/V (absolute 20-bit)



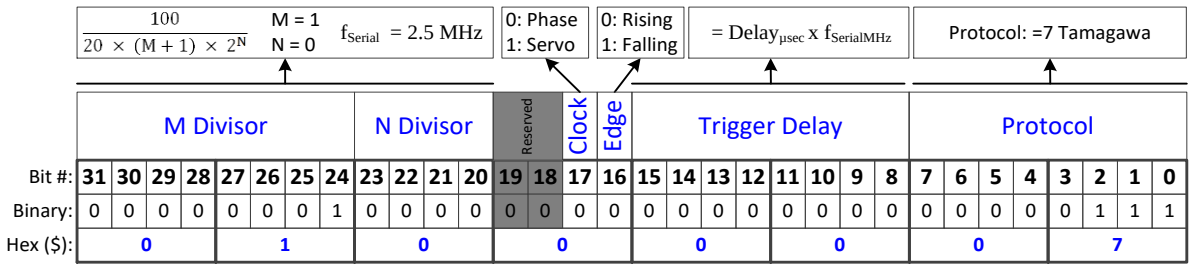
Yaskawa Sigma II/II/V Encoders Alarm Code (**SerialEncDataB**)

Bit #	Alarm Code
21	Power-on error self-detected
23	Revolution count (index to index) incorrect
26	Position reference (index) not found yet
21	Power-on error self-detected

● Tamagawa FA-coder Example

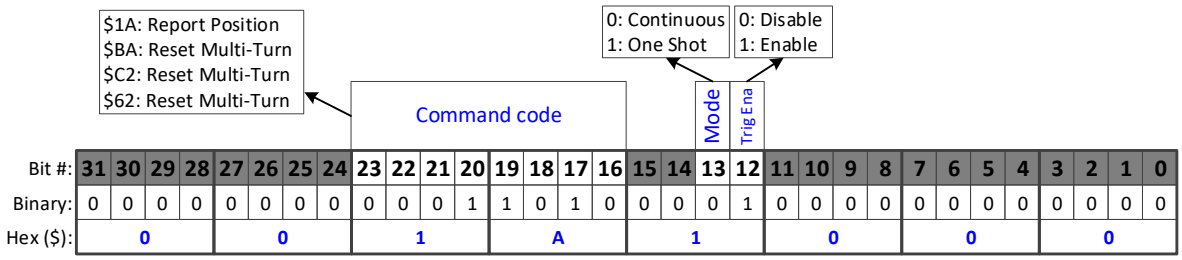
No trigger delay, rising edge of phase, and 2.5 MHz transmission:

CK3C[0].SerialEncCtrl setting:



CK3C[0].Chan[j].SerialEncCmd setting:

For continuous position reporting:



If the command code is set to $BA, $C2, or $62, the multi-turn position value in the encoder is reset to 0. This should be done in “one-shot” mode, making the element equal to $00BA3000, $00C23000, or $00623000, respectively. When the reset operation is done, the component should report as $00BA2000, $00C22000, or $00622000, respectively.

```
CK3C[0].SerialEncCtrl = $1000007
CK3C[0].Chan[0].SerialEncCmd = $1A1000
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

The resulting position data, status, and error bits for Tamagawa FA-Coder are found in the following Serial Data Registers:

CK3C[].Chan[].SerialEncDataA

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																Single-Turn Position															

CK3C[].Chan[].SerialEncDataB

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Timeout Err.	CRC Err.			Status Field				Alarm Code																Multi-Turn Position							

● Panasonic Example

No trigger delay, rising edge of phase, and 2.5 MHz transmission

CK3C[0].SerialEncCtrl setting:

$\frac{100}{20 \times (M+1) \times 2^N}$ M = 1 N = 0 f_{Serial} = 2.5 MHz | 0: Phase 1: Servo | 0: Rising 1: Falling | = $Delay_{\mu sec}$ x $f_{SerialMHz}$ | Protocol: =8 Panasonic

	M Divisor								N Divisor				Reserved		Clock	Edge	Trigger Delay								Protocol							
Bit #:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary:	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
Hex ($):	0				1				0				0				0				0				0				8			

CK3C[0].Chan[j].SerialEncCmd setting:

For continuous position reporting:

$2A: Report Absolute Position
$52: Single-Turn Position with Encoder ID code
$4A: Reset Multi-Turn
$7A: Reset Multi-Turn
$DA: Reset Multi-Turn
$F2: Reset Multi-Turn

0: Continuous 1: One Shot | 0: Disable 1: Enable

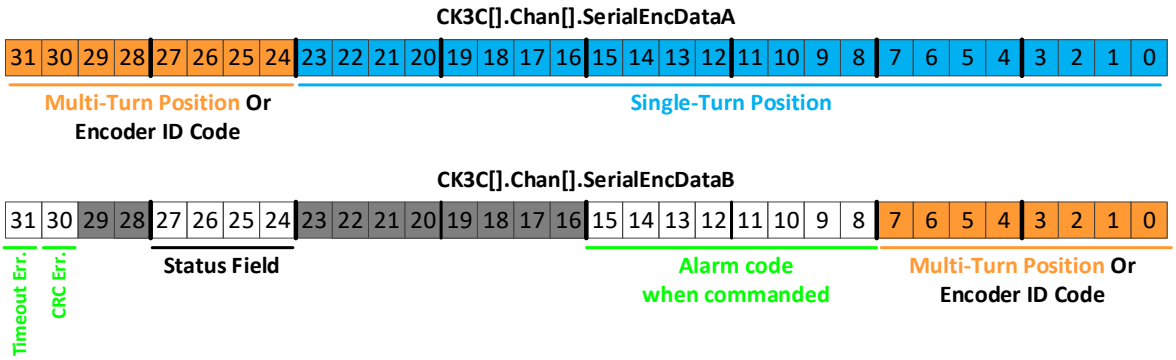
	Command code												Mode	Trig Ena																		
Bit #:	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Binary:	0	0	0	0	0	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	0	0	0	0	0	0	0	0	1	0	0	0
Hex ($):	0				0				2				A				1				0				0				0			

If the command code is set to $52 for single-turn position reporting with alarm code, the encoder ID value is reported where multi-turn position is normally reported.
If the command code is set to $4A, $7A, $DA, or $F2, the multi-turn position value in the encoder is reset to 0. This should be done in “one-shot” mode, making the element equal to $004A3000, $007A3000, $00DA3000, or $00F23000, respectively. When the reset operation is done, the component should report as $004A2000, $007A2000, $00DA2000, or $00F22000, respectively.

```
CK3C[0].SerialEncCtrl = $1000008
CK3C[0].Chan[0].SerialEncCmd = $2A1000
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

The resulting position data, status, and error bits for Panasonic are found in the following Serial Data Registers:



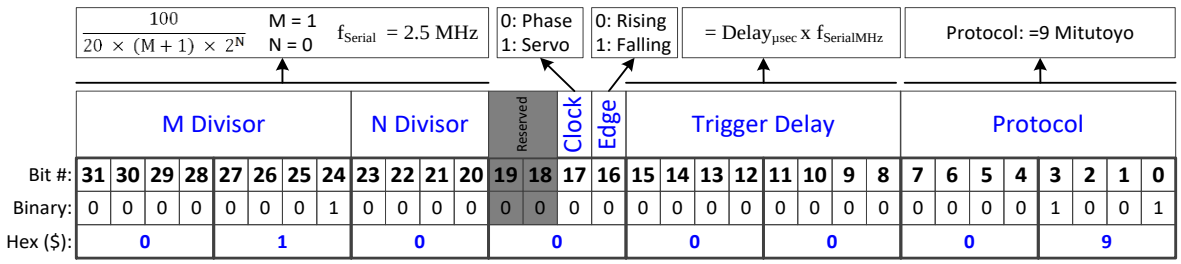
Bits 24 – 31 (**SerialEncDataA**) of the encoder ID code are fixed at a value of $11.

Bit #	Alarm Code
8	Overspeed error
9	Full resolution status; = 1 when over 100 rpm and reporting reduced resolution
10	Count Error
11	Counter Overflow
13	Multi-revolution error
14	System undervoltage error (< 2.5 V)
15	Battery low (< 3.1 V)

● Mitutoyo Example

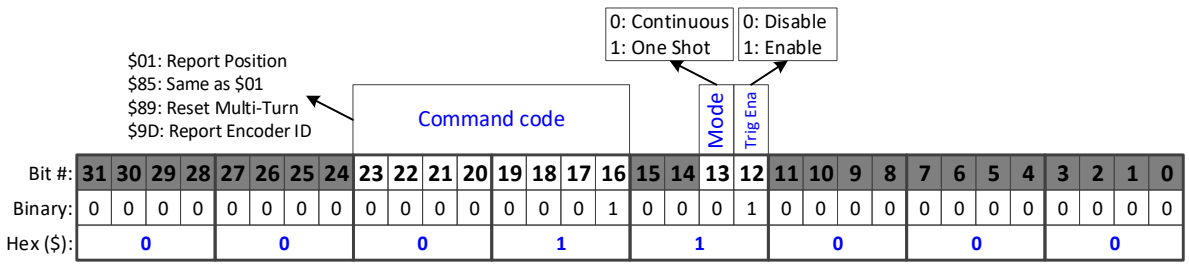
No trigger delay, rising edge of phase, and 2.5 MHz transmission

CK3C[0].SerialEncCtrl setting:



- **CK3C[0].Chan[j].SerialEncCmd** setting

  For continuous position reporting:

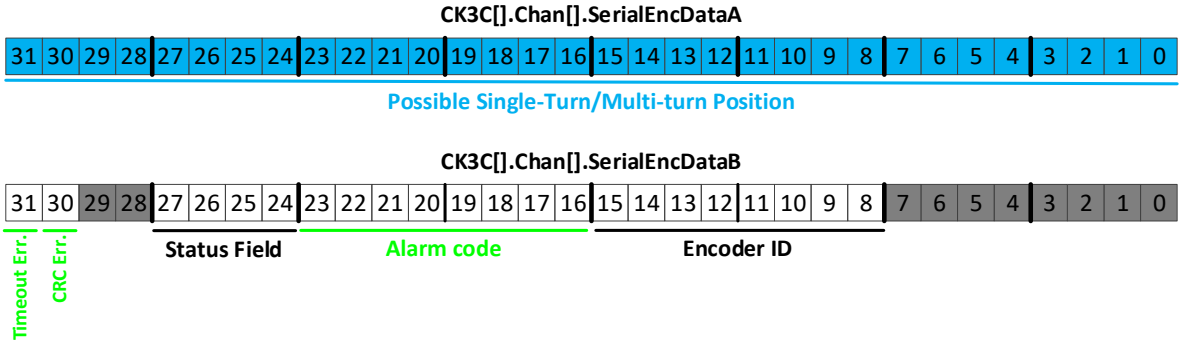


If the command code is set to $89, the multi-turn position value in the encoder is reset to 0 (after 8 cycles). If the command code is set to $9D, the encoder ID value is reported in bits 8 – 15 of SerialEncDataB. If the command code is set to $85, absolute position is reported, similarly to $01.

```
CK3C[0].SerialEncCtrl = $1000009
CK3C[0].Chan[0].SerialEncCmd = $11000
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

The resulting position data, status, and error bits for Mitutoyo are found in the following Serial Data Registers:



Bits 24 – 31 (**SerialEncDataA**) of the encoder ID code are fixed at a value of $11.

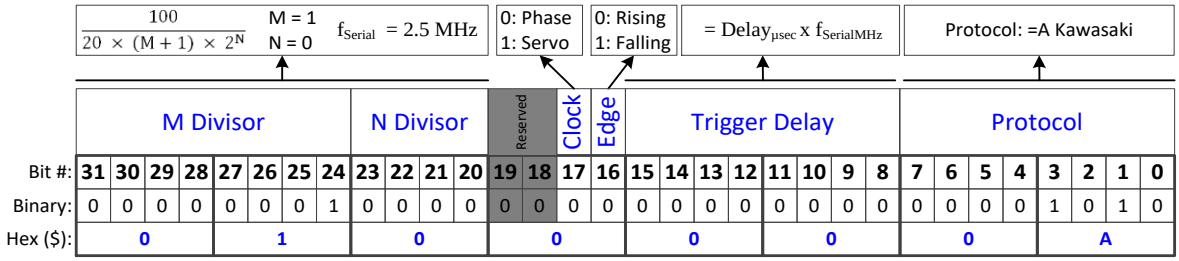
Bit #	Alarm Code
16	Initialization error
17	Mismatch of optical and capacitive sensors
18	Optical sensor error
19	Capacitive sensor error
20	CPU error (AT303); CPU/ROM/RAM error (AT503)
21	EEPROM error
22	ROM/RAM error (AT303); communication error (AT503)
23	Over-speed error

Bit #	Status Field
24	Fatal (unrecoverable) encoder error
26	Illegal command code from controller

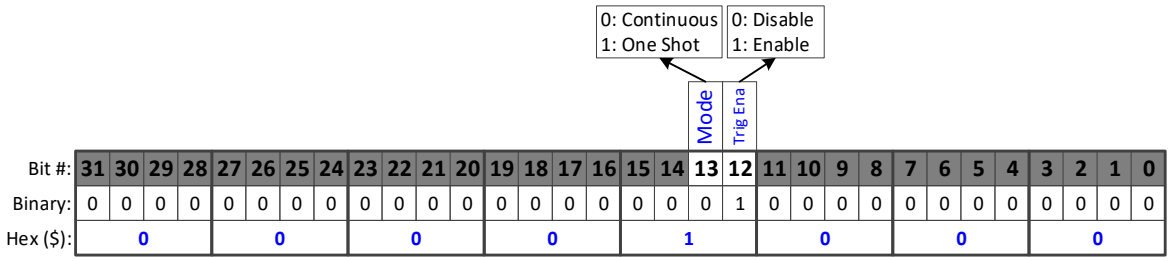
- **Kawasaki Example**

  No trigger delay, rising edge of phase, and 2.5 MHz transmission

CK3C[0].SerialEncCtrl setting:



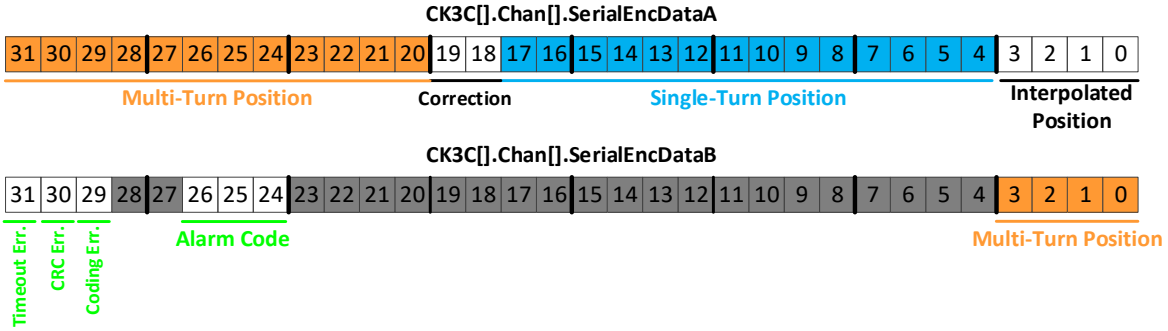
CK3C[0].Chan[j].SerialEncCmd setting:




```
CK3C[0].SerialEncCtrl = $100000A
CK3C[0].Chan[0].SerialEncCmd = $1000
CK3C[0].Chan[0].SerialEncEna = 1
```

Serial Data Registers:

The resulting position data, status, and error bits for Kawasaki are found in the following Serial Data Registers:



Bits 24 – 31 (**SerialEncDataA**) of the encoder ID code are fixed at a value of $11.

Bit #	Alarm Code
24	Interpolator error
25	Absolute track error
26	Busy flag

Serial Encoder Ongoing Position

For the on-going "incremental" position data, it is sufficient to process whatever position data (single-turn and/or multi-turn) is available in **CK3C[0].Chan[j].SerialEncDataA**. The PMAC firmware does not require processing the entire bit length, the difference change in between servo cycles is used to compute the on-going position. This will not limit the resolution or hinder the performance. Some people may choose to use strictly the single-turn data in the Encoder Conversion Table for simplicity.

A key step is to make sure that unwanted data has been cleared and the MSB of the data chosen is left-shifted to bit 31 in order to handle the rollover gracefully. **EncTable[i].index2** is set to the number of unwanted bits to the right of the desired data, so that a right shift can be performed to clear that unwanted data. **EncTable[i].index1** is then set to the number of bits the data must be shifted left (after the right shift) to make the MSB of your position data bit 31.

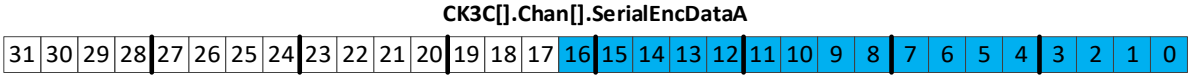
The following settings are required to read on-going position in counts. These settings depend primarily on the location of the position data in the **SerialEncDataA**.

Structure Element	Value
EncTable[i].type	1
EncTable[i].pEnc	CK3C[0].Chan[j].SerialEncDataA.a
EncTable[i].pEnc1	Sys.Pushm
EncTable[i].index1	Number of bits to left shift (second operation)
EncTable[i].index2	Number of bits to right shift (first operation)
EncTable[i].index3	0
EncTable[i].index4	0
EncTable[i].index5	0
EncTable[i].index6	0
EncTable[i].ScaleFactor	$1 / 2^{\text{EncTable[i].index1}}$

Structure Element	Value
Motor[x].ServoCtrl	1
Motor[x].pEnc	EncTable[i].a
Motor[x].pEnc2	EncTable[i].a

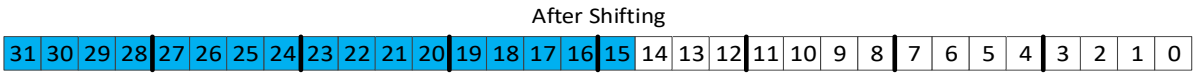
The following are examples for setting up the Encoder Conversion Table (ECT).

Example 1: A binary serial encoder with 17 bits of single-turn (or an equivalent 1 µm linear scale) position data starting at bit 0 of **SerialEncDataA**.



The position data should be shifted 15 bits left (using index1) so that the MSB is at bit 31 to handle the rollover gracefully. Also, the scale factor should reflect the new location of the LSB.

```
EncTable[1].type = 1
EncTable[1].pEnc = CK3C[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 15
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / EXP2(15)
```

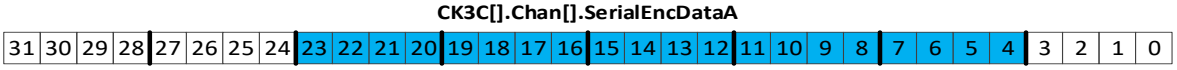


The settings below are sufficient to view motor position in the position window, in counts.

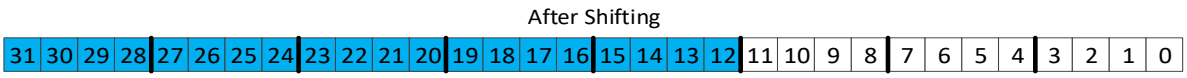
```
Motor[1].ServoCtrl = 1
Motor[1].pEnc = EncTable[1].a
Motor[1].pEnc2 = EncTable[1].a
```

In this case, the user will see $2^{SingleTurn} = 2^{17} = 131,072$ counts per revolution for a rotary encoder, and 1/0.001 = 1,000 counts per mm for a linear encoder.

Example 2: A binary serial encoder with 20 bits of single-turn (or an equivalent 50 nm linear scale) position data starting at bit 4 of **SerialEncDataA**. The low 4 bits may contain other information, irrelevant to position data.



The position data should be first shifted 4 bits to the right (using **index2**) to get rid of the unwanted data. Then shifted 12 bits to the left (using **index1**), so that the MSB is at bit 31 to handle the rollover gracefully. Also, the scale factor should reflect the new location of the LSB.



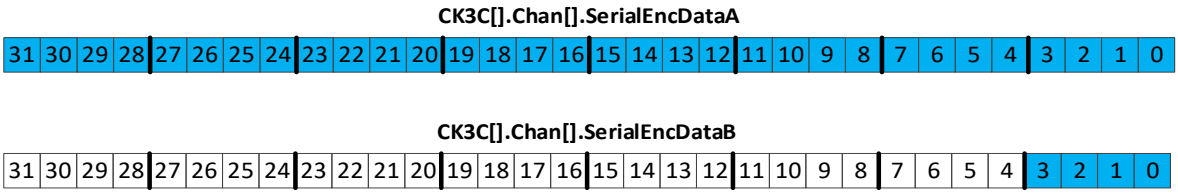

```
EncTable[1].type = 1
EncTable[1].pEnc = CK3C[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 12
EncTable[1].index2 = 4
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / EXP2(12)
```

The settings below are sufficient to view motor position in the position window, in counts.

```
Motor[1].ServoCtrl = 1
Motor[1].pEnc = EncTable[1].a
Motor[1].pEnc2 = EncTable[1].a
```

In this case, the user will see $2^{SingleTurn} = 2^{20} = 1{,}048{,}576$ counts per revolution for a rotary encoder. And 1 / 0.000050 = 20,000 counts per mm for a linear encoder.

Example 3: A binary serial encoder with 36 bits of single-turn (or an equivalent 1 nm linear scale) position data starting at bit 0 of **SerialEncDataA** and extending to bit 3 of **SerialEncDataB**.



Reading and processing the 32 bits of position data in **SerialEncDataA** is sufficient for producing the proper ongoing position.

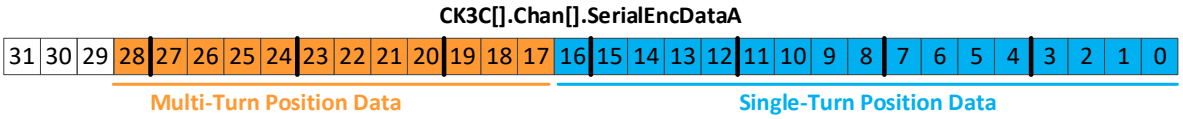
```
EncTable[1].type = 1
EncTable[1].pEnc = CK3C[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 0
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1
```

The settings below are sufficient to view motor position in the position window, in counts.

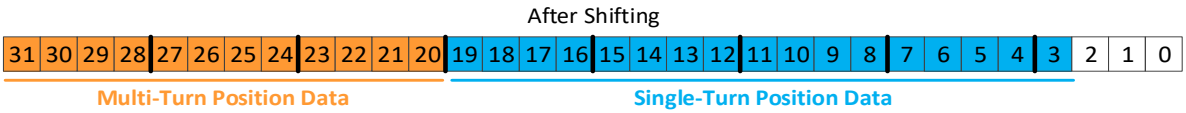
```
Motor[1].ServoCtrl = 1
Motor[1].pEnc = EncTable[1].a
Motor[1].pEnc2 = EncTable[1].a
```

In this case, the user will see $2^{SingleTurn} = 2^{36} = 68,719,476,736$ counts per revolution for a rotary motor. And 1 / 0.000001 = 1,000,000 counts per mm for a linear motor.

Example 4: A 29-bit binary serial encoder with 17 bits of single-turn and 12 bits of multi-turn position data starting at bit 0 of **SerialEncDataA**.



Both single-turn and multi-turn data can be used for ongoing position. The entire bit length is shifted left (using **index1**) 3 bits to place the MSB at bit 31.



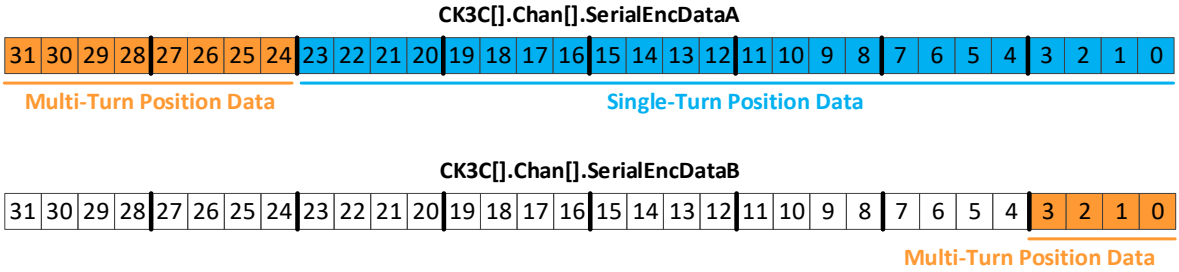

```
EncTable[1].type = 1
EncTable[1].pEnc = CK3C[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 3
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / EXP2(3)
```

The settings below are sufficient to view motor position in the position window, in counts.

```
Motor[1].ServoCtrl = 1
Motor[1].pEnc = EncTable[1].a
Motor[1].pEnc2 = EncTable[1].a
```

In this case, the user will see $2^{SingleTurn} = 2^{17} = 131{,}072$ counts per revolution.

Example 5: A 36-bit binary serial encoder with 24 bits of single-turn and 12 bits of multi-turn position data starting at bit 0 of **SerialEncDataA** and continuously extending to bit 3 of **SerialEncDataB**.



For on-going position, we are only interested in the position data residing in **SerialEncDataA**. Some people may elect to use only the single-turn data for on-going position processing. This would require shifting to the left 8 bits (**index1 = 8**), and setting up the **EncTable[i].ScaleFactor = 1 / 256**.

But, also it is possible to simply process the whole 32-bit word comprised of single-turn, and multi-turn position data with no shifting.

```
EncTable[1].type = 1
EncTable[1].pEnc = CK3C[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 0
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1
```

The settings below are sufficient to view motor position in the position window, in counts.

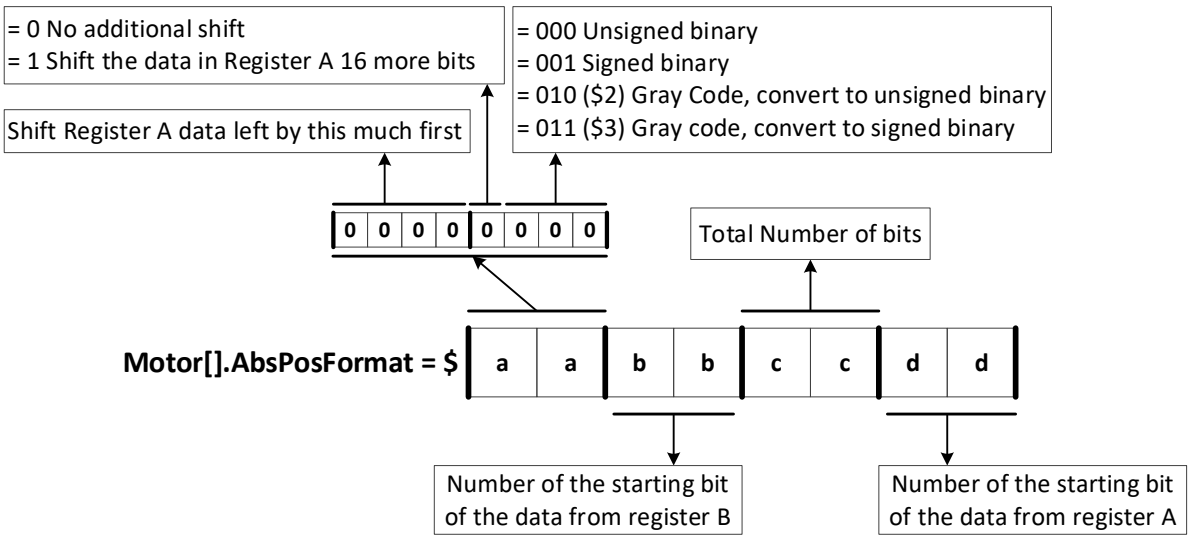
```
Motor[1].ServoCtrl = 1
Motor[1].pEnc = EncTable[1].a
Motor[1].pEnc2 = EncTable[1].a
```

In this case, the user will see $2^{SingleTurn} = 2^{24} = 16{,}777{,}216$ counts per revolution.

Serial Encoder Power-on Absolute Position

The absolute position is computed directly from the serial data registers, and set up using the following key structure elements:

- **Motor[x].pAbsPos**, typically = **CK3C[0].Chan[j].SerialEncDataA.a**
- **Motor[x].AbsPosSf** = **Motor[x].PosSf**
  - These settings should appear after Scaling to Engineering Units (in your motor setup file) so that **Motor[x].PosSf** is already set.
- **Motor[x].AbsPosFormat**:
  - Encoders with no multi-turn position data are unsigned. Rotary encoders with multi-turn position data are signed.



- **Motor[x].HomeOffset = 0**

Additional Information

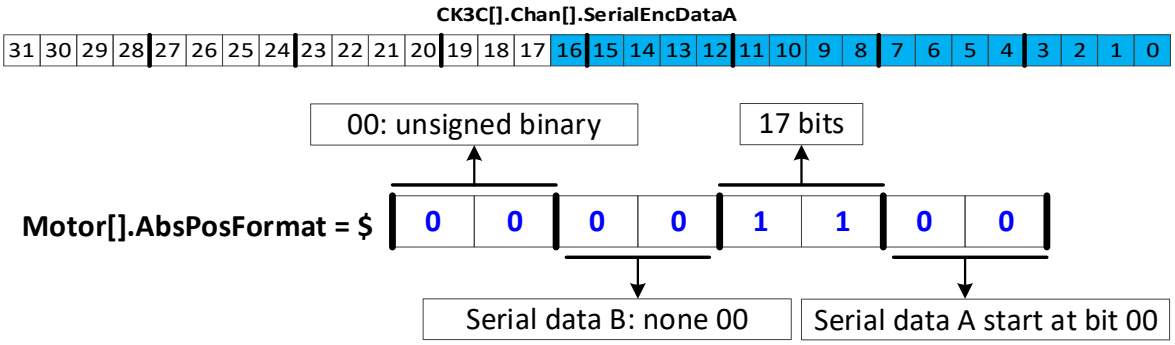
Gray code conversion should be omitted here if it had been already implemented in **CK3C[0].Chan[j].SerialEncCmd** word.

Additional Information

Motor[x].PowerOnMode bit 2 (value of 4) specifies an absolute position read on power up. Alternately, **#1HMZ** from the online terminal or a **HOMEZ 1** from a PLC can be issued to retrieve the absolute position.

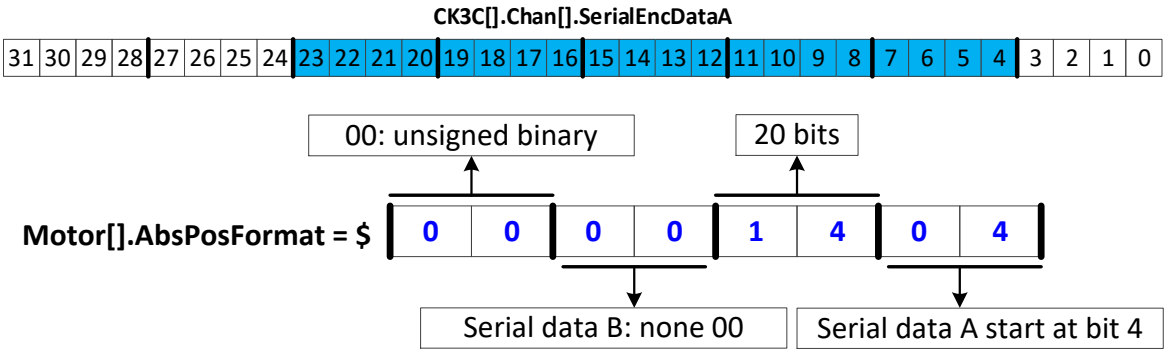
Following are examples for setting up the absolute position read with various serial encoders. These settings depend primarily on the location of the position data in **SerialEncDataA** and **SerialEncDataB**.

Example 1: A binary serial encoder with 17 bits of single-turn (or an equivalent 1 µm linear scale) position data starting at bit 0 of **SerialEncDataA**.



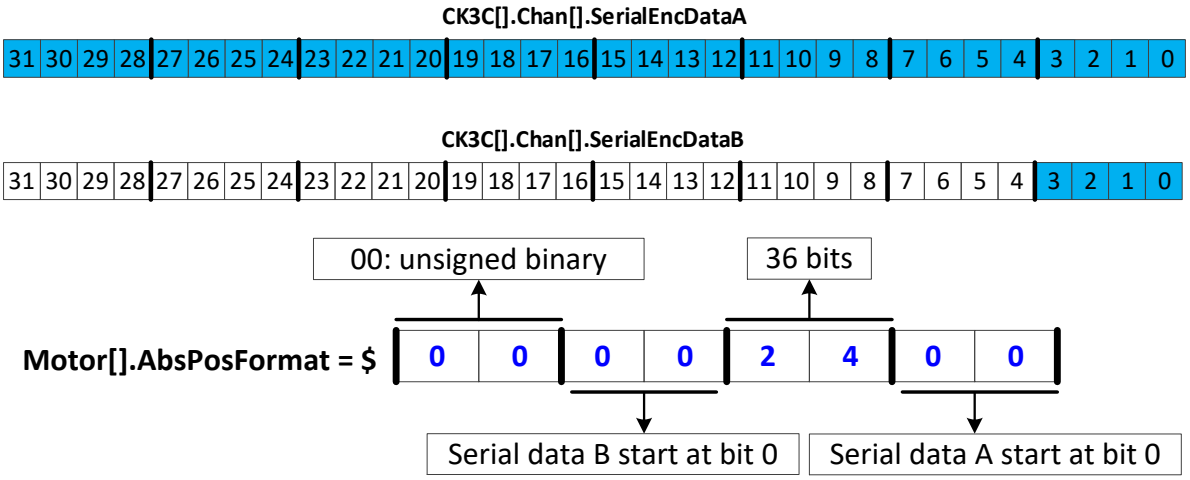

```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $00001100
Motor[1].HomeOffset = 0
```

Example 2: A binary serial encoder with 20 bits of single-turn (or an equivalent 50 nm linear scale) position data starting at bit 4 of **SerialEncDataA**. The low 4 bits may contain other information, irrelevant to position data.



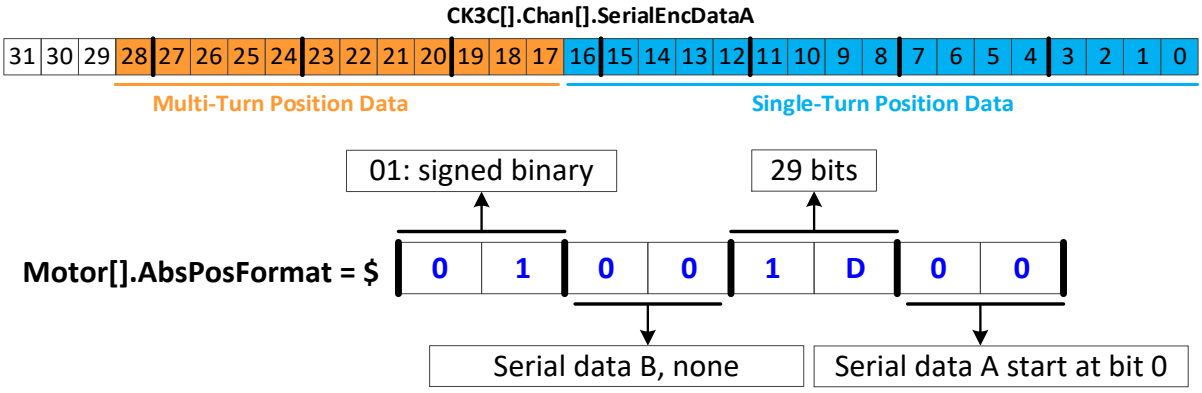

```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $00001404
Motor[1].HomeOffset = 0
```

Example 3: A binary serial encoder with 36 bits of single-turn (or an equivalent 1 nm linear scale) position data starting at bit 0 of **SerialEncDataA** and extending to bit 3 of **SerialEncDataB**.



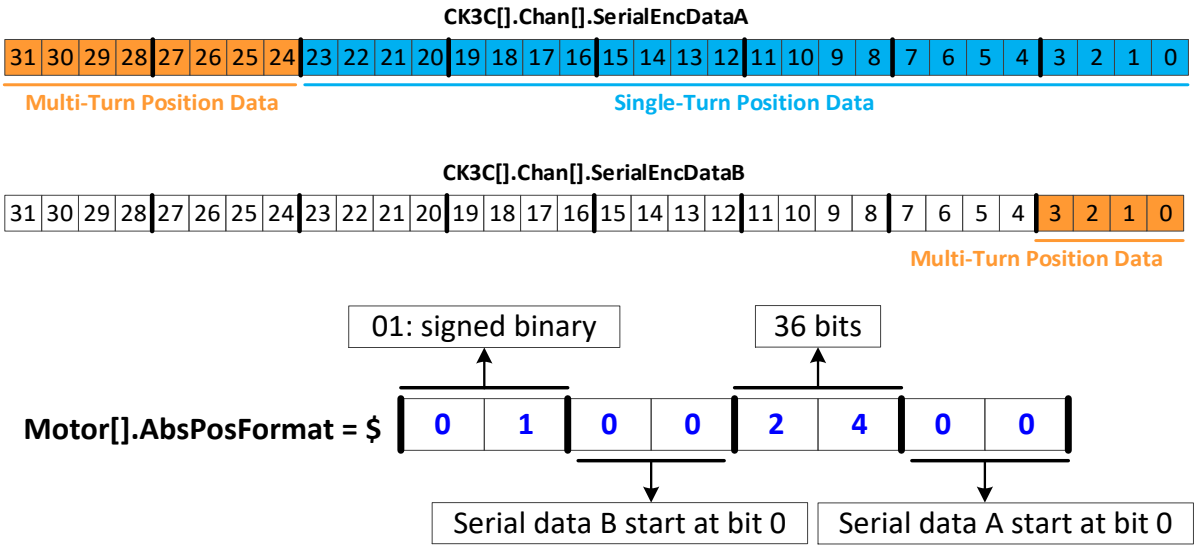

```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $00002400
Motor[1].HomeOffset = 0
```

Example 4: A 29-bit binary serial encoder with 17 bits of single-turn and 12 bits of multi-turn position data starting at bit 0 of **SerialEncDataA**.



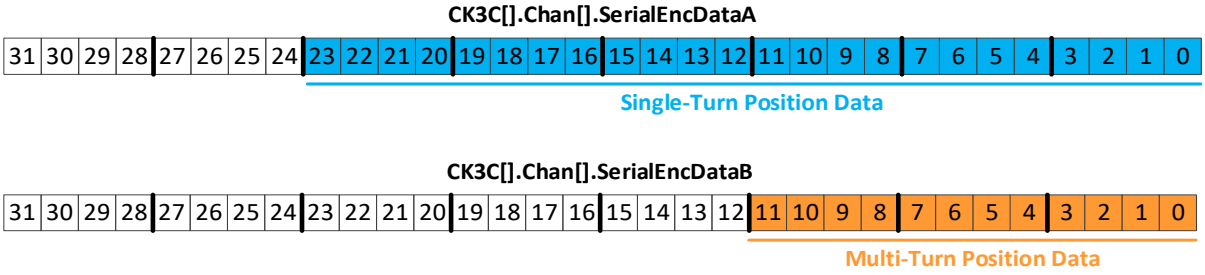

```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $01001D00
Motor[1].HomeOffset = 0
```

Example 5: A 36-bit binary serial encoder with 24 bits of single-turn and 12 bits of multi-turn position data starting at bit 0 of **SerialEncDataA** and continuously extending to bit 3 of **SerialEncDataB**.

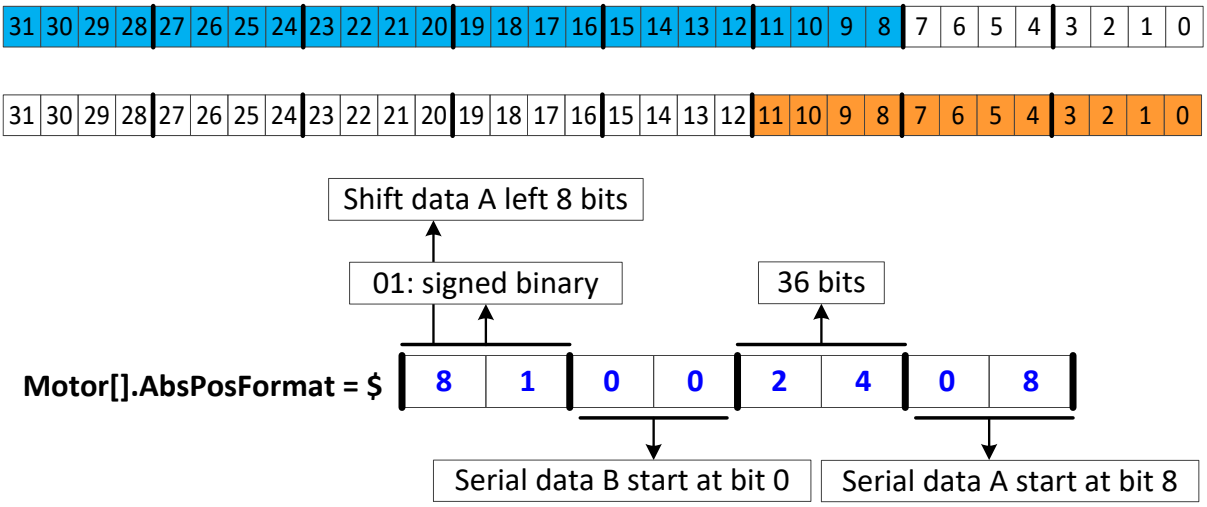



```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $01002400
Motor[1].HomeOffset = 0
```

Example 6: A 36-bit binary serial encoder with 24 bits of single-turn data starting at bit 0 of **SerialEncDataA**, and 12 bits of multi-turn position data starting at bit 0 of **SerialEncDataB**.

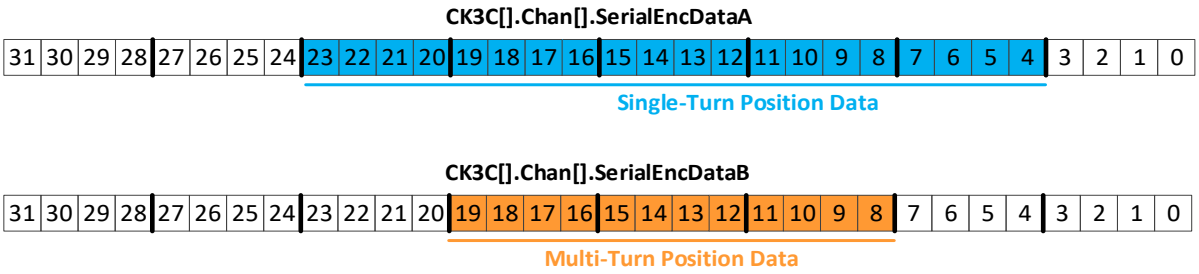


The single turn data must be shifted 8 bits left first, to make it contiguous with the multi-turn data. This shift is done using the upper 5 bits of the **$aa** byte of **Motor[x].AbsPosFormat**. The data would then look like:

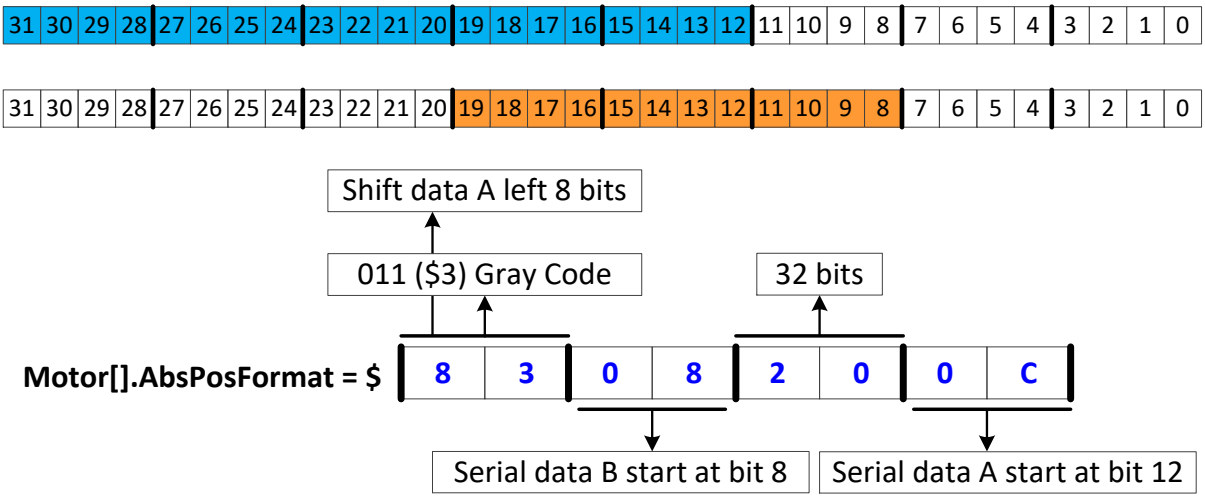



```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $81002408
Motor[1].HomeOffset = 0
```

Example 7: A 32-bit Gray code serial encoder with 20 bits of single-turn data starting at bit 4 of **SerialEncDataA**, and 12 bits of multi-turn position data starting at bit 8 of **SerialEncDataB**.



The single turn data must be shifted 8 bits left first. This shift is done using the upper 5 bits of the **$aa** byte of **Motor[x].AbsPosFormat**. The data would then look like:




```
Motor[1].pAbsPos = CK3C[0].Chan[0].SerialEncDataA.a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $8308200C
Motor[1].HomeOffset = 0
```

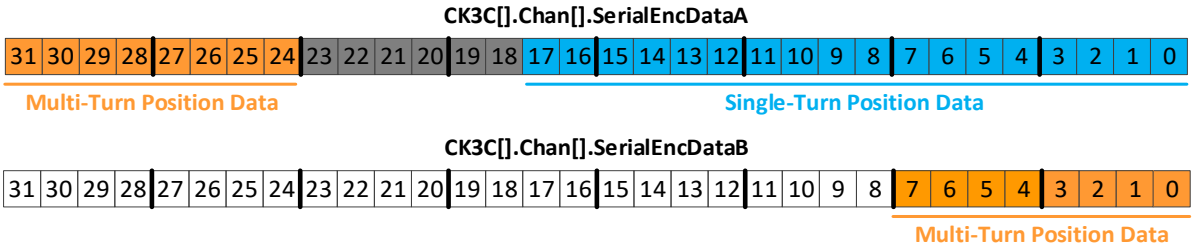
Additional Information

Encoders with multi-turn position data are typically set up as signed.

Additional Information

Gray code conversion should be omitted here if it had been already implemented in **CK3C[0].Chan[j].SerialEncCmd**.

Example 8: A 34-bit binary serial encoder (for example, Panasonic) with 18 bits of single-turn and 16 bits of multi-turn position data in the following fields:



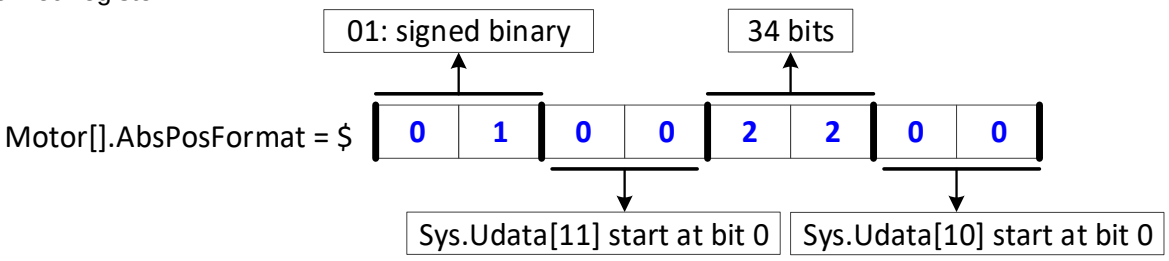
The automatic settings are not suitable for the discontinuity between the single-turn and multi-turn data.

We will assemble the absolute position word manually (in a background or initialization PLC), and hold the data in two consecutive open memory registers to feed the automatic settings. Below is the example PLC for performing this operation:

```
GLOBAL Enc1StData
GLOBAL Enc1MtData
GLOBAL Enc1Data
#define Enc1AbsPos1    Sys.Udata[10]
#define Enc1AbsPos2    Sys.Udata[11]

OPEN PLC PanasonicAbsPosPLC
LOCAL Enc1MtDataA, Enc1MtDataB
Enc1StData = CK3C[0].Chan[0].SerialEncDataA & $3FFFF
Enc1MtDataA = (CK3C[0].Chan[0].SerialEncDataA & $FF000000) >> 24
Enc1MtDataB = CK3C[0].Chan[0].SerialEncDataB & $000000FF
Enc1MtData = Enc1MtDataA + Enc1MtDataB * EXP2(8)
IF (Enc1MtData > EXP2(15)) // NEGATIVE?
{
   Enc1Data = (Enc1StData - EXP2(18)) + (Enc1MtData - EXP2(16)) << 18
}
ELSE                       // POSITIVE?
{
   Enc1Data = Enc1StData + Enc1MtData * << 18
}
Enc1AbsPos1 = Enc1Data & $FFFFFFFF
Enc1AbsPos2 = (Enc1Data >> 32) & $3
DISABLE PLC PanasonicAbsPosPLC
CLOSE
```

The automatic settings can now be set up to read the absolute position at the first corresponding user defined register:




```
Motor[1].pAbsPos = Sys.Udata[10].a
Motor[1].AbsPosSf = Motor[1].PosSf
Motor[1].AbsPosFormat = $01002200
Motor[1].HomeOffset = 0
```

Once the example code in the sample PLC is executed, an HMZ command can be issued for an absolute position read.

5-2-5 Setting up the Analog (ADC) Inputs

The analog inputs accept ±10V single-ended signals only.

The ADC data resides in the upper 16 bits of the 32-bit structure elements in the following table. The structure elements do not allow bit definitions of the upper 16 bits, hence scaling (shifting) would be required to obtain the raw ADC data. Using the explicit address registers with PTR definitions is one alternative:

ADCIN_n/ Connector	Structure	Address
ADC0, J3	CK3C[0].Chan[0].AdcEnc[0]	$900030
ADC1, J3	CK3C[0].Chan[0].AdcEnc[1]	$900034
ADC2, J7	CK3C[0].Chan[0].AdcEnc[2]	$900038
ADC3, J7	CK3C[0].Chan[0].AdcEnc[3]	$90003C



Additional Information

The explicit address register(s) can be found by subtracting **Sys.piom** from **CK3C[0].Chan[0].AdcEnc[n].a** (n=0-3).



Additional Information

The ADC input data must be in the “unpacked” format to be read properly; **CK3C[0].Chan[0].PackInData = 0**.

Raw ADC Data (in bits)

```
Sys.WpKey = $AAAAAAAA                    DISABLE WRITE-PROTECTION
CK3C[0].Chan[0].PackInData = 0           UNPACK INPUT DATA ALL ADCS J3, J7
PTR ADCIN_1->S.IO:$900030.16.16           ADC0 J3
PTR ADCIN_2->S.IO:$900034.16.16           ADC1 J3
PTR ADCIN_3->S.IO:$900038.16.16           ADC2 J7
PTR ADCIN_4->S.IO:$90003C.16.16           ADC3 J7
```

ADC for Servo Feedback Example

```
EncTable[1].Type = 1                      SINGLE 32 BIT READ
EncTable[1].pEnc = CK3C[0].Chan[0].AdcEnc[0].a
EncTable[1].pEnc1 = Sys.pushm
EncTable[1].index1 = 16
EncTable[1].index2 = 16
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / EXP2(16)

Motor[1].pEnc = EncTable[1].a
Motor[1].pEnc2 = EncTable[1].a
```

Since the analog inputs have 16 bits of resolution (65,536 software counts) spanning over the full range of the input voltage, wiring a ±10V produces the following counts in software:

Single-Ended (VDC)	Software Counts
-10	-32768
0	0
10	32767

5-2-6 Limit, Flag, and Brake [Channel 0-3] Structure Elements

Either the user flags or other unassigned axis flags on the base board can be used as general-purpose I/O providing up to 20 inputs and 4 outputs at 5-24 VDC levels. The indicated Structure Elements allow accessing each particular line as shown below:

```
PTR Ch0UserFlag -> CK3C[0].Chan[0].UserFlag                 USER 0 FLAG INPUT STATUS
PTR Ch0HomeFlag -> CK3C[0].Chan[0].HomeFlag                 HOME FLAG 0 INPUT STATUS
PTR Ch0PlusLimit -> CK3C[0].Chan[0].PlusLimit               POSITIVE LIMIT 0 FLAG INPUT STATUS
PTR Ch0MinusLimit -> CK3C[0].Chan[0].MinusLimit             NEGATIVE LIMIT 0 FLAG INPUT STATUS

PTR Ch1UserFlag -> CK3C[0].Chan[1].UserFlag                 USER 1 FLAG INPUT STATUS
PTR Ch1HomeFlag -> CK3C[0].Chan[1].HomeFlag                 HOME FLAG 1 INPUT STATUS
PTR Ch1PlusLimit -> CK3C[0].Chan[1].PlusLimit               POSITIVE LIMIT 1 FLAG INPUT STATUS
PTR Ch1MinusLimit -> CK3C[0].Chan[1].MinusLimit             NEGATIVE LIMIT 1 FLAG INPUT STATUS

PTR Ch2UserFlag -> CK3C[0].Chan[2].UserFlag                 USER 2 FLAG INPUT STATUS
PTR Ch2HomeFlag -> CK3C[0].Chan[2].HomeFlag                 HOME FLAG 2 INPUT STATUS
PTR Ch2PlusLimit -> CK3C[0].Chan[2].PlusLimit               POSITIVE LIMIT 2 FLAG INPUT STATUS
PTR Ch2MinusLimit -> CK3C[0].Chan[2].MinusLimit             NEGATIVE LIMIT 2 FLAG INPUT STATUS

PTR Ch3UserFlag -> CK3C[0].Chan[3].UserFlag                 USER 3 FLAG INPUT STATUS
PTR Ch3HomeFlag -> CK3C[0].Chan[3].HomeFlag                 HOME FLAG 3 INPUT STATUS
PTR Ch3PlusLimit -> CK3C[0].Chan[3].PlusLimit               POSITIVE LIMIT 3 FLAG INPUT STATUS
PTR Ch3MinusLimit -> CK3C[0].Chan[3].MinusLimit             NEGATIVE LIMIT 3 FLAG INPUT STATUS
```

The following settings are required to synchronize the enabling/disabling of the motor with the brake output signal.

```
Motor[1].pBrakeOut = CK3C[0].Chan[0].OutFlagB.a             SINGLE 32 BIT READ
Motor[1].BrakeOutBit = 9
Motor[1].BrakeOffDelay = 1                                  MSEC, BRAKE OFF DELAY --USER INPUT
Motor[1].BrakeOnDelay = 1                                   MSEC, BRAKE ON DELAY --USER INPUT

Motor[2].pBrakeOut = CK3C[0].Chan[1].OutFlagB.a
Motor[2].BrakeOutBit = 9                                    MSEC, BRAKE OFF DELAY --USER INPUT
Motor[2].BrakeOffDelay = 1                                  MSEC, BRAKE ON DELAY --USER INPUT
Motor[2].BrakeOnDelay = 1

Motor[3].pBrakeOut = CK3C[0].Chan[2].OutFlagB.a             MSEC, BRAKE OFF DELAY --USER INPUT
Motor[3].BrakeOutBit = 9                                    MSEC, BRAKE ON DELAY --USER INPUT
Motor[3].BrakeOffDelay = 1
Motor[3].BrakeOnDelay = 1
                                                            MSEC, BRAKE OFF DELAY --USER INPUT
Motor[4].pBrakeOut = CK3C[0].Chan[3].OutFlagB.a             MSEC, BRAKE ON DELAY --USER INPUT
Motor[4].BrakeOutBit = 9
Motor[4].BrakeOffDelay = 1
Motor[4].BrakeOnDelay = 1
```

If you do not wish to use the brake functionality, leave **Motor[x].pBrakeOut = 0**.

Additional Information

For diagnostics, set **Motor[x].pBrakeOut = 0** and set the output using the **CK3C[0].Chan[j].OutFlagB** bit element.

5-2-7 General Purpose I/O (GPIO) Configuration

To configure the I/O the following must be set:

```
CK3C[0].GpioDir[0] = $00FFFF00      USER DEPENDENT, VALUE OF 0 IN BIT SPECIFIES AS INPUT, VALUE OF 1 IN BIT
                                    SPECIFIES OUTPUT

CK3C[0].GpioPol[0] = $00000000      USER DEPENDENT, VALUE OF 0 IN BIT SPECIFIES HIGH-TRUE OUTPUT, VALUE OF 1 IN
                                    BIT SPECIFIES LOW-TRUE OUTPUT
```



Additional Information

Make sure to set **Sys.WpKey = $AAAAAAAA** in order to make changes to write protected structures.

The lines on the general-purpose I/O connector (J9) are accessed with the following structures:

```
                                    INPUTS
PTR AllInputs->Gate3[0].GpioData[0].24.8    INPUTS AS AN 8-BIT BYTE
PTR Input1->Gate3[0].GpioData[0].24         GPIO 24
PTR Input2->Gate3[0].GpioData[0].25         GPIO 25
PTR Input3->Gate3[0].GpioData[0].26         GPIO 26
PTR Input4->Gate3[0].GpioData[0].27         GPIO 27
PTR Input5->Gate3[0].GpioData[0].28         GPIO 28
PTR Input6->Gate3[0].GpioData[0].29         GPIO 29
PTR Input7->Gate3[0].GpioData[0].30         GPIO 30
PTR Input8->Gate3[0].GpioData[0].31         GPIO 31

                                    OUTPUTS
PTR AllOutputs->Gate3[0].GpioData[0].16.8   OUTPUTS AS AN 8-BIT BYTE
PTR Output1->Gate3[0].GpioData[0].16        GPIO 16
PTR Output2->Gate3[0].GpioData[0].17        GPIO 17
PTR Output3->Gate3[0].GpioData[0].18        GPIO 18
PTR Output4->Gate3[0].GpioData[0].19        GPIO 19
PTR Output5->Gate3[0].GpioData[0].20        GPIO 20
PTR Output6->Gate3[0].GpioData[0].21        GPIO 21
PTR Output7->Gate3[0].GpioData[0].22        GPIO 22
PTR Output8->Gate3[0].GpioData[0].23        GPIO 23
```

An M-variable can be assigned to an individual bit of an element, or to a consecutive set of bits. When the assignment is made through the IDE, an application-specific name can be given to the variable. For example:

```
ptr LaserOn->CK3C[0].GpioData[0].21          1-BIT VALUE
ptr OverrideKnob->CK3C[0].GpioData[0].24.4   4-BIT VALUE
```

Once the assignment is made, the application can use the declared variable name in the application.

6

Troubleshooting

This section describes the LED indications, error display codes, and troubleshooting methods of the CK3C.

6-1 LED Indicators 6-2
6-1-1 Controller Operation Status Indicators 6-2
6-1-2 Ethernet / EtherCAT Indicators 6-2
6-2 Error Processing 6-3
6-2-1 Classification of Errors 6-3
6-2-2 Fatal Errors 6-3

6-1 LED Indicators

6-1-1 Controller Operation Status Indicators

The following indicators are located next to the power input terminal connection. Below is a description of their color designation and description:

Indicator	Color	Status	Description
PWR	Green	Lit	Power is supplied to the Unit.
		Not lit	Power is not supplied to the Unit.
RDY	Green	Lit	Unit is in operation-ready status.
		Not lit	Initialization process is in progress.
ERR	Red	Lit	Watchdog or another hardware error
		Not lit	No watchdog error

6-1-2 Ethernet / EtherCAT Indicators

The following indicators are located on each corresponding Ethernet jack. Below is a description of their color designation and description:

Indicator	Color	Status	Description
ECAT LINK	Green	Lit	The EtherCAT link is established (1 Gbps)
		Not lit	The EtherCAT link is not established
	Orange	Lit	The EtherCAT link is established (100 Mbps)
		Not lit	The EtherCAT link is not established
ECAT ACT	Yellow	Lit	The EtherCAT link is established
		Flashing	Data communications are in progress after the EtherCAT link is established Flashes every time data is sent or received
		Not lit	The EtherCAT link is not established
Ethernet LINK	Green	Lit	The EtherCAT link is established (1Gbps)
		Not lit	The EtherCAT link is not established
	Orange	Lit	The Ethernet link is established (100 Mbps)
		Not lit	The Ethernet link is not established
Ethernet ACT	Yellow	Lit	The Ethernet link is established
		Flashing	Data communications are in progress after the Ethernet link is established Flashes every time data is sent or received
		Not lit	The Ethernet link is not established

6-2 Error Processing

6-2-1 Classification of Errors

Fatal errors in the CK3C are errors that occurred as the result of the controller being disabled.

Non-fatal errors in the CK3C are errors that can be detected and managed by the controller itself that is still operating.

In section 6-2-2 we describe the fatal errors that cause the CPU unit to stop.

6-2-2 Fatal Errors

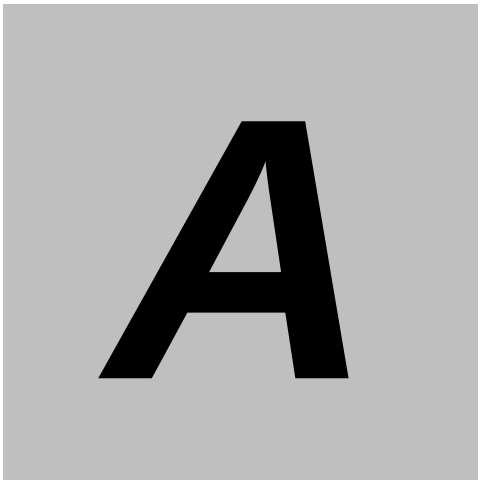
During normal operation, the CPU unit should have the following indicators:

Indicators			CPU Unit operating status
PWR (green)	RDY (green)	ERR (red)	
Lit	Lit	Not lit	Normal operation

The following are the errors that cause the operation of the CPU Unit to stop.

- Power supply error
- Initial process error
- Watchdog timer error

Indicators			CPU Unit operating status
PWR (green)	RDY (green)	ERR (red)	
Not lit	Not lit	Not lit	Power Supply Error
Lit	Not lit	Not lit	Initial process error
Lit	Not lit	Lit	Watchdog timer error



Appendices

This section contains appendices pertaining to the CK3C.

A-1 CK3C Project Management Example..... A-2

A-1-1 Creating an IDE Project..... A-2
A-1-2 Basic Optimization, System and Gate Settings..... A-6
A-1-3 Encoder On-going Position, Verification..... A-7
A-1-4 Motor Setup and Commissioning..... A-11
A-1-5 Reversing Motor Operation Direction..... A-36

A-2 Sample Motor Setup Examples..... A-37

A-2-1 Filtered PWM Output with Quadrature Encoder..... A-38
A-2-2 Direct PWM Output with Quadrature Encoder (ACC-8FS)..... A-40
A-2-3 Direct PWM Output with Sinusoidal Encoder (ACC-8FS, ACC-51S)..... A-42
A-2-4 Direct PWM Output with BiSS Encoder (ACC-8FS, ACC-84S)..... A-44

A-1 CK3C Project Management Example

This section describes a general example procedure for manually setting up a CK3C using the Power PMAC IDE and some illustration of how to organize the contents of those files. The procedure steps are shown below.

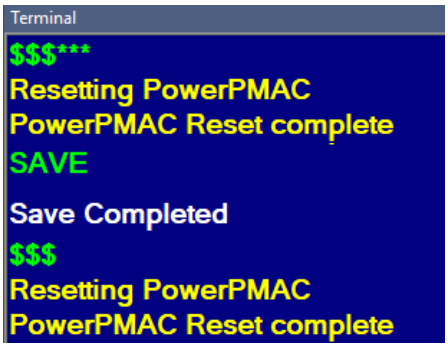
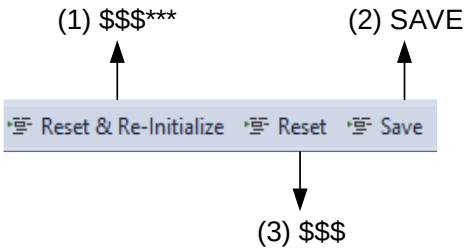
1. Creating an IDE project
2. Basic optimization, system and gates settings
3. Encoder on-going position, verification
4. Motor setup and commissioning

A-1-1 Creating an IDE Project

Factory Default Reset

For new projects, starting from factory default settings is highly recommended to ensure a clean starting point. Factory default reset can be performed by issuing a global reset **$$$*****, followed by a **SAVE**, and a normal reset **$$$**.

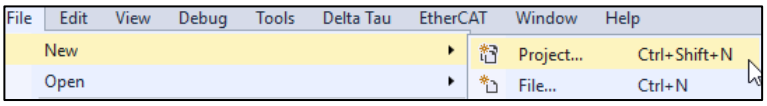
This can be done either from the terminal window, or using the IDE toolbar shortcuts.

Terminal Window	IDE Toolbar
	

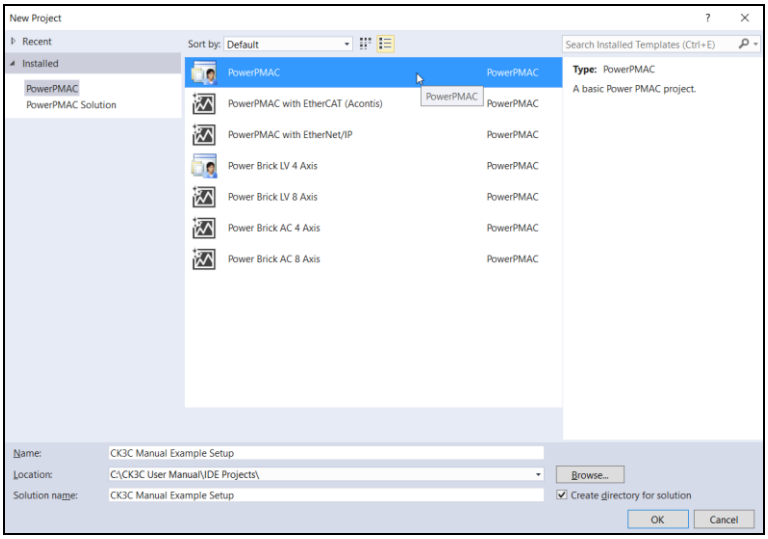
New IDE Project

Creating a new project is done from the file menu.

- File > New > Project



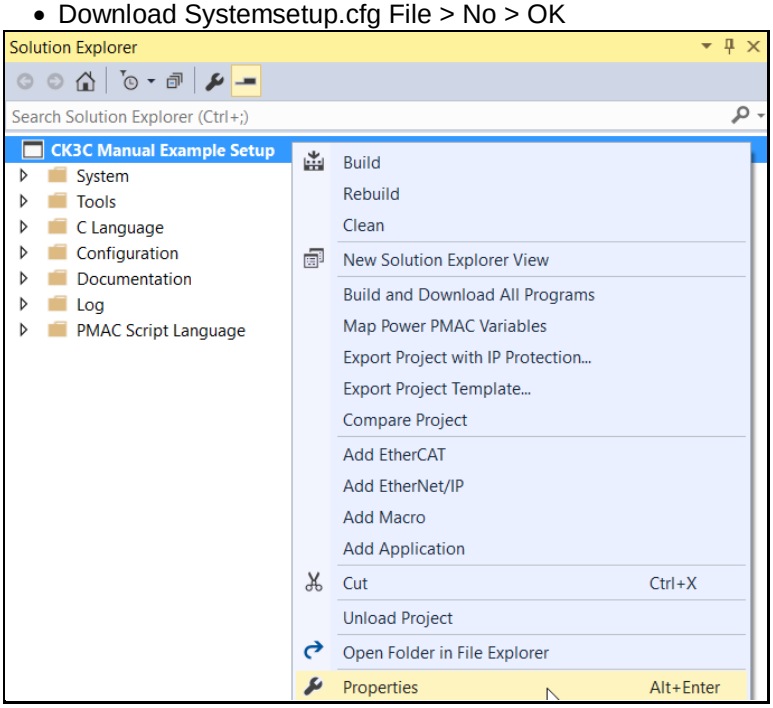
- Choose template
- Choose project location
- Enter project name

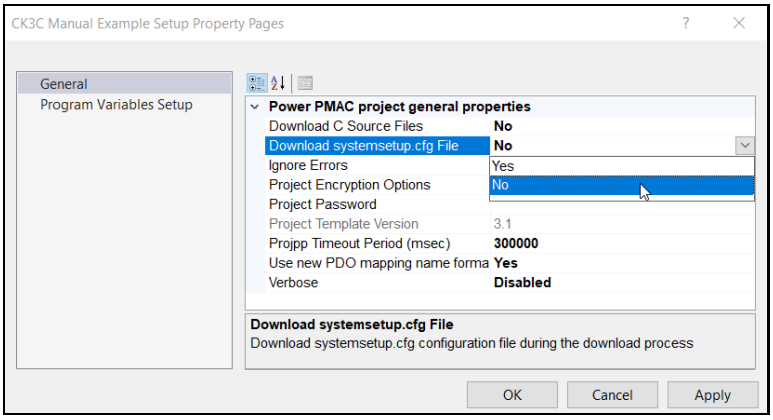


Systemsetup.cfg

If the IDE System setup tool is not used, systemsetup.cfg file should be disabled.

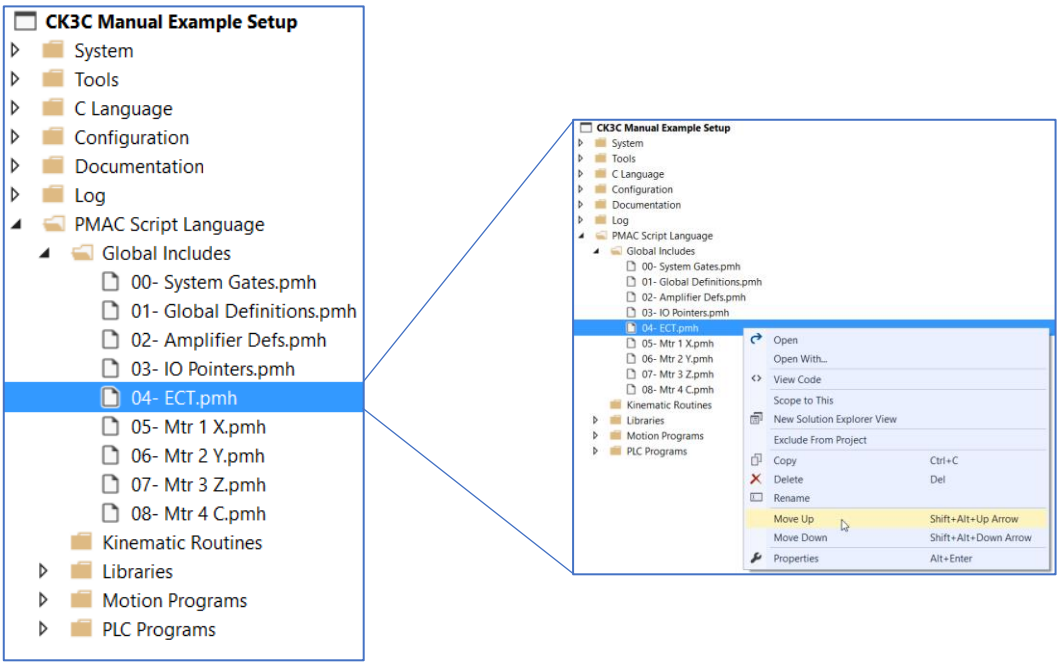
- Right-click on project name
- Properties
- Download Systemsetup.cfg File > No > OK





Project Organization

The majority of the parameters described in the following sections are typically placed under Global Includes. Files in this folder can be managed per the user preference. They can be added, deleted, inserted from an existing project, re-named, moved up or down, amongst other actions. Refer to the IDE Manual to learn about these manipulations. One suggested assortment is shown below.

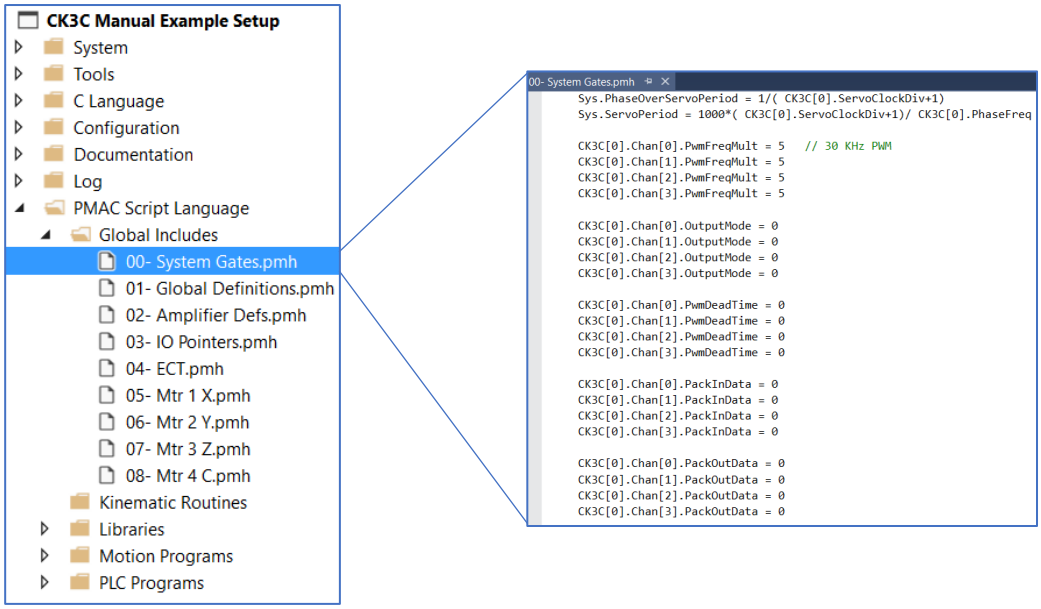


The table below provides a brief description of the suggested files:

File	Typical Content	Content Example
System Gates	System structure elements	Sys.MaxMotors
	Gate3 structure elements	CK3C[0].PhaseFreq
	Channel structure elements	CK3C[0].Chan[0].PwmFreqMult
Global Defs	User-defined variables	GLOBAL MyVariable
Amplifier Defs	Amp specific Variables	GLOBAL AMP1DcBus
IO Pointers	Digital I/O pointers	PTR Input1->CK3C[0].GpioData[0].0.1 PTR Output1-> CK3C[0].GpioData[0].16.1
	Analog Input pointers	PTR ADC_IN1-> S.IO:$900030.20.12
ECT	Encoder conversion table	EncTable[1].type
Mtr1 X	Motor structure elements	Motor[1].ServoCtrl

A-1-2 Basic Optimization, System and Gate Settings

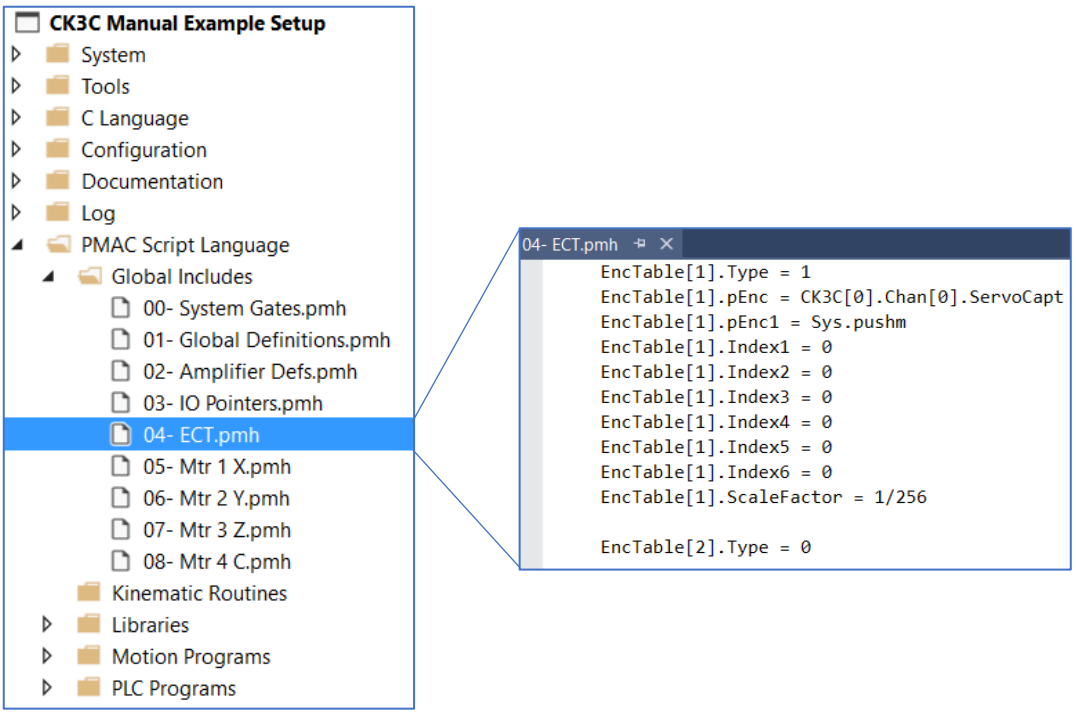
This section describes basic optimization, and clock frequencies.



Structure Element	Description and Examples
Sys.WpKey	**Write protect key for Gate3** Disable write protection, typical, = $AAAAAAAA
Sys.MaxCoords	**Highest number of coordinate systems to be used + 1** Example, = 3 (using coordinate systems 0..1, 0 is usually unused)
Sys.MaxMotors	**Highest number of motors to be used + 1** Example, = 5 (using Motors #0..4, 0 is usually unused)
CK3C[0].PhaseFreq	**Phase clock frequency in [Hz]** Example, = 10000 (10KHz)
CK3C[0].ServoClockDiv	**Servo clock divider (defines servo frequency)** Servo frequency = Phase frequency / (ServoClockDiv + 1) Example, = 1 (5KHz)
Sys.ServoPeriod	**Servo update period for interpolation** Necessary for correct frequencies setup. It can be set as fixed expression: Example, = 1000 * (CK3C[0].ServoClockDiv + 1) / CK3C[0].PhaseFreq
Sys.PhaseOverServoPeriod	**Ratio of phase to servo** Necessary for correct frequencies setup. It can be set as fixed expression: Example, = 1 / (CK3C[0].ServoClockDiv + 1)
CK3C[0].Chan[j].PwmFreqMult	**PWM frequency multiplier (defines PWM frequency)** Example, PWM frequency = (PwmFreqMult + 1) * PhaseFreq / 2
CK3C[0].Chan[j].OutputMode	**(PWM, DAC, PFM)** Determines which output format is used on the command signal line for each phase (A to D) of the IC's channel Example, = 0
CK3C[0].Chan[j].PwmDeadTime	**PWM deadtime** Example, = 2 / 0.0533 (2 µsec for CK3A-G305L) Example, = 3 / 0.0533 (3 µsec for CK3A-G310L)
CK3C[0].Chan[j].PackInData	**ADC input pack data enable** Example, = 0
CK3C[0].Chan[j].PackOutData	**ADC output pack data enable** Example, = 0

A-1-3 Encoder On-going Position, Verification

The Encoder Conversion Table ECT is the primary encoder input processing structure that produces on-going actual position. The result of an ECT is typically used in Motor servo loops.



Additional Information

- Refer to the Power PMAC Software Reference Manual for detailed descriptions.
- **EncTable[n].Type = 0** indicates the end of the ECT (strongly recommended for efficiency).

Below, are general guidelines for setting up an ECT entry for most common types of encoder feedback devices.

Additional Information

EncTable[n].Index1 through 6 are typically =0 if omitted.

ECT Entry for Digital Quadrature

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= CK3C[0].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].ScaleFactor	= 1 / 256

ECT Entry for Standard Analog Sinusoidal x4096 (ACC51S)

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= CK3C[0].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].ScaleFactor	= 1

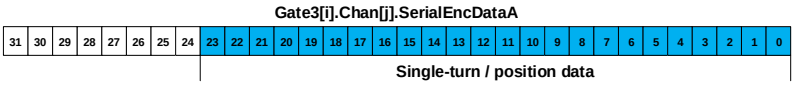
● Other CK3C (ACC-51S) settings necessary for this type of processing

Structure Element	Setting
Motor[x].EncType	= 3
CK3C[0].Chan[j].AtanEna	= 1

ECT Entry for Serial with CK3C (Gate3)

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= CK3C[0].Chan[j].SerialEncDataA.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].Index1	= Number of data bits to shift left to bit 31 + index2. This is the 2nd shift operation. In example diagram below, = 8
EncTable[n].Index2	= Number of data bits to shift right to bit 0 (rare case). This is the 1st shift operation. In example diagram below, = 0
EncTable[n].ScaleFactor	= $1 / 2^{EncTable[n].Index1}$ In example diagram below, = $1 / 2^{8}$

Example:



● Other necessary settings for this type of processing

Structure Element	Setting
CK3C[0].Chan[j].SerialEncEna	= 1
CK3C[0].SerialEncCtrl	Refer to the Power PMAC Software Reference Manual
CK3C[0].Chan[j].SerialEncCmd	Refer to the Power PMAC Software Reference Manual

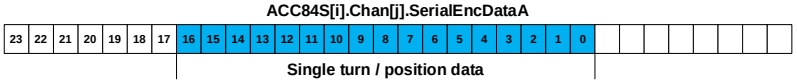
Additional Information

After setting up **CK3C[0].SerialEncCtrl** and **CK3C[0].Chan[j].SerialEncCmd** successfully, and setting **CK3C[0].Chan[j].SerialEncEna=1,** raw serial encoder data can be seen in **CK3C[0].Chan[j].SerialEncDataA** and **CK3C[0].Chan[j].SerialEncDataB**. Depending on the protocol, information may reside in **CK3C[0].Chan[j].SerialEncDataC**, or **CK3C[0].Chan[j].SerialEncDataD**.

ECT Entry for Serial with ACC-84S

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= ACC84S[i].Chan[j].SerialEncDataA.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].Index1	= Number of data bits to shift left to bit 31 + index2. This is the 2nd shift operation. In example diagram below, = 15
EncTable[n].Index2	= 8 + Number of data bits to shift right to bit 0 (rare case). This is the 1st shift operation. In example diagram below, = 8
EncTable[n].ScaleFactor	= 1 / $2^{EncTable[n].Index1}$ In example diagram below, = 1 / 2^{15}

Example:



- **Other necessary settings for this type of processing**

Structure Element	Setting
ACC84S[i].SerialEncCtrl	Refer to the Power PMAC Software Reference Manual
ACC84S[i].Chan[j].SerialEncCmd	Refer to the Power PMAC Software Reference Manual

Additional Information

After setting up **ACC84S[i].SerialEncCtrl** and **ACC84S[i].Chan[j].SerialEncCmd**, raw serial encoder data can be seen in **ACC84S[i].Chan[j].SerialEncDataA** and **ACC84S[i].Chan[j].SerialEncDataB**. Depending on the protocol, some information may reside in **ACC84S[i].Chan[j].SerialEncDataC**, or **ACC84S[i].Chan[j].SerialEncDataD**.

Finishing and Verifying Encoder Feedback

Precautions for Safe Use

- The absence of encoder data could be a very dangerous condition in closed-loop control because the servo loop no longer knows what the true physical position of the Motor is – usually it thinks it is "stuck" – and it can react wildly, often causing a runaway condition.
- Make sure that encoder data is updating and counting correctly before issuing any Motor closed loop commands.
- **Motor[x].PosSf** and **Motor[x].Pos2Sf** are part of the Motor structure elements, including servo loop. If they are changed after initial setup, many other elements need to be adjusted such as acceleration, speed settings, and servo loop gains.

The resulting output of an ECT entry is usually tied to a Motor to be used in the position and velocity servo loops using the structure elements below.

Additional Information

- The (processed) output of an ECT appears in **EncTable[n].PrevEnc**.
- **EncTable[n].PrevEnc** data is not multiplied by **EncTable[n].ScaleFactor**.

Structure Element	Description and Examples
Motor[x].ServoCtrl	**Activate Motor channel** Typical, = 1
Motor[x].pEnc	**Position feedback pointer** Typical, = EncTable[n].a (corresponding ECT)
Motor[x].pEnc2	**Velocity feedback pointer, always =Motor[x].pEnc for single feedback** Typical, = EncTable[n].a (corresponding ECT)
Motor[x].PosSf	**Position scale factor (allows conversion to user units)** 20-bit rotary Motor in degrees, = 360 / 1048576 Linear Motor with 1 nm scale, = 1 / 1000000
Motor[x].Pos2Sf	**Velocity scale factor, always =Motor[x].PosSf for single feedback** 20-bit rotary Motor in degrees, = 360 / 1048576 Linear Motor with 1 nm scale, = 1 / 1000000
Motor[x].Posunit	**Setting for IDE position window units** For degrees, = 11 For mm, = 3

- **Verifying Encoder Counting in Both Directions**

  This can be done by moving the Motor by hand e.g. clockwise, counter-clockwise, positive or negative while monitoring the position window in the IDE.

  The user must also check if the position is stable (within an inherent dithering amount) at standstill.

  For troubleshooting purposes, if no position change is seen in the IDE position window, looking at the “raw encoder counts” may be helpful. The table below shows these registers per encoder type.

Encoder “processing” Type	Raw Counts Structure Element
Digital Quadrature	CK3C[0].Chan[j].ServoCapt
Analog sinusoidal	CK3C[0].Chan[j].ServoCapt
Analog sinusoidal ACI	CK3C[0].Chan[j].ServoCapt
Serial w/ Gate3	CK3C[0].Chan[j].SerialEncDataA CK3C[0].Chan[j].SerialEncDataB (does not always have data)
Serial w/ ACC-84S	ACC84S[i].Chan[j].SerialEncDataA ACC84S[i].Chan[j].SerialEncDataB (does not always have data)

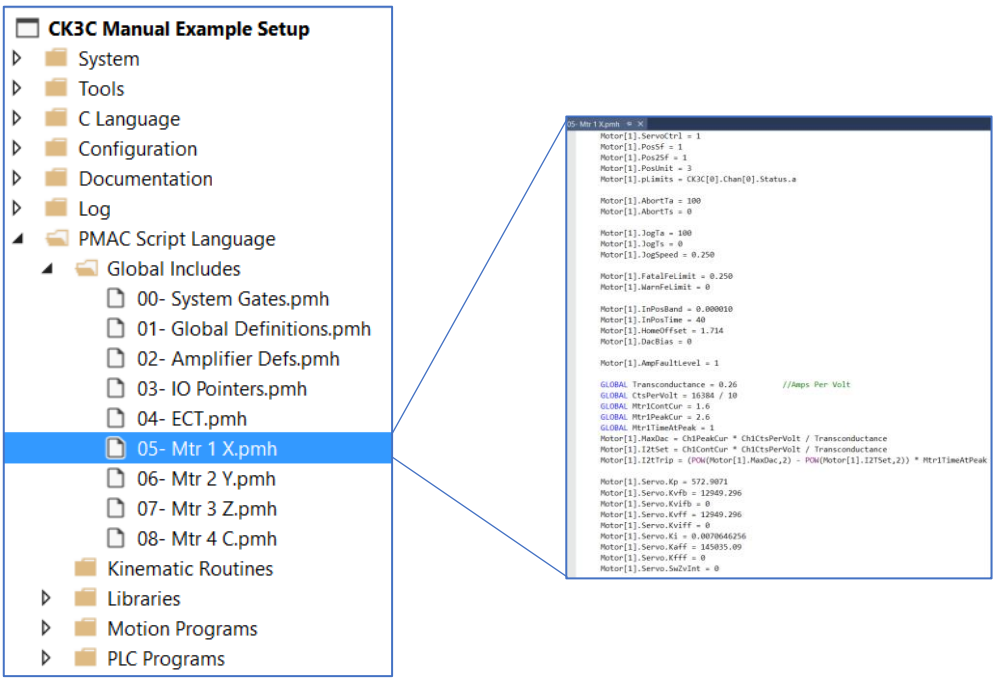
- **Reporting the Correct Distance**

  The user can verify if the feedback device is counting correctly by moving the Motor a known amount and recording the elapsed distance shown in the IDE position window. In some cases, the **#nHMZ** (where n is the Motor number) command can be used to zero the position display.

  If the counting is incorrect, make sure that **EncTable[n].ScaleFactor**, **Motor[x].PosSf**, **Motor[x].Pos2Sf**, and **Motor[x].EncType** are set up correctly.

A-1-4 Motor Setup and Commissioning

This section discusses the general guidelines for the remaining Motor structure elements, and certain commissioning procedures for CK3C with various accessories.



Generic Motion Elements

Structure Element	Description
Motor[x].AbortTa	**Abort deceleration** Time [msec] if >= 0 Inverse rate [msec^2 / user unit] if < 0
Motor[x].AbortTs	**Abort s-curve deceleration** Time [msec] if >= 0 Inverse jerk rate [msec^3 / user unit] if < 0 (must be set if AbortTa < 0)
Motor[x].JogTa	**Jog acceleration/deceleration** Time [msec] if >= 0 Inverse rate [msec^2 / user unit] if < 0
Motor[x].JogTs	**Jog s-curve acceleration/deceleration** Time [msec] if >= 0 Inverse jerk rate [msec^3 / user unit] if < 0 (must set if AbortTa < 0)
Motor[x].JogSpeed	**Jog speed** [user units / msec]
Motor[x].FatalFeLimit	**Max. position error before following error fault** [user units]
Motor[x].WarnFeLimit	**Max. position error before following error warning** [user units]
Motor[x].InPosBand	**In-position threshold** [user units]
Motor[x].InPosTime	**Time (number of consecutive servo cycles) that the position error is within Motor[x].InPosBand before Motor[x].InPos is set to true** = {time in msec} / Sys.ServoPeriod



Additional Information

Motor[x].pLimits is automatically set up by the firmware by hardware recognition. **Motor[x].pLimits** specifies the (flag pointer) source of the hardware over-travel limits. If they are not wired, or desired to be disabled then **Motor[x].pLimits** can be set = 0.

Common Analog Amplifier Elements

Structure Element	Description
Motor[x].AmpFaultLevel	**Amplifier fault logical state** Default, = 0
Motor[x].DacBias	**Servo Output Offset** Specifies the offset to be added to the calculated output command from the position/velocity servo loop. Default, = 0

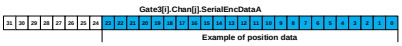
Common Amplifier and Direct PWM Control Elements

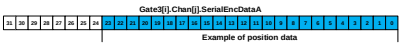
Structure Element	Description
Motor[x].AmpFaultLevel	**Amplifier fault logical state** Default, = 0, = 1 for CK3A
Motor[x].AdcMask	**Specifies current feedback resolution (in bits)** Default, = $FFF00000 , = $FFFF0000 for CK3A
Motor[x].PhaseOffset	**Angular distance between Motor phases** Typical, = 683
Motor[x].PwmSf	**PWM output scale factor (voltage divider)** Motor rated voltage greater or equal main supply voltage, = 16384 * 0.95 Motor rated voltage less than main supply voltage, = 16384 * Motor Voltage / Main Voltage

On-going Phase (Commutation) Position

The following definitions are used typically when configuring a brushless motor with Direct PWM Amplifier. This system configuration would require a CK3C + ACC-8FS to output a Direct PWM signal.

Name	Description
ST	Single-turn serial rotary encoder position data
NoOfPolePairs	Number of pair poles of a rotary brushless Motor
CtsPerRev	Number of counts per revolution of a rotary Motor (usually lines x 4)
LinesPerRev	Number of sine cycles of a sinusoidal rotary encoder
ECLmm	Linear Motor electrical cycle length or magnetic pitch in mm (e.g. 30.48mm)
RESmm	Linear encoder resolution in mm (e.g. 1µm = 0.001mm)

Structure Element	Description and Examples
Motor[x].PhaseCtrl	**Motor commutation control** Typical, = 4
Motor[x].pPhaseEnc	**Commutation on-going position (pointer) source** Digital quadrature = CK3C[0].Chan[j].PhaseCapt.a Analog sinusoidal = CK3C[0].Chan[j].PhaseCapt.a Serial w/ Gate3 = CK3C[0].Chan[j].SerialEncDataA.a Serial w/ ACC-84S = ACC84S[i].Chan[j].SerialEncDataA.a
Motor[x].PhaseEncRightShift	**Number of bits to shift phase-position source data right (to bit 0)** Digital quadrature = 0 Analog sinusoidal = 0 Analog sinusoidal ACI = 0 Serial w/ Gate3 = 0 in this example (most serial protocols w/ Gate3)  Serial w/ ACC-84S = 8 in this example (typical w/ ACC-84S) 

Structure Element	Description and Examples
Motor[x].PhaseEncLeftShift	**Number of bits to shift phase-position source data left (to bit 31)** Digital quadrature = 0 Analog sinusoidal = 0 Analog sinusoidal ACI = 0 Serial w/ Gate3 = 8 in this example  Serial w/ ACC-84S = 15 in this example (7 data bits + PhaseEncRightShift) 
Motor[x].PhasePosSf	**Commutation angle scale factor**
Digital quadrature	Rotary Motor = 2048 * NoOfPolePairs / CtsPerRev Linear Motor = 2048 * RES_{mm} / (256 * ECL_{mm})
Analog sinusoidal	Rotary Motor = 2048 * NoOfPolePairs / (LinesPerRev * 16384) Linear Motor = 2048 * RES_{mm} / (16384 * ECL_{mm})
Analog sinusoidal ACI	Rotary Motor = 2048 * 4 * NoOfPolePairs / (LinesPerRev * 65536) Linear Motor = 2048 * 4 * RES_{mm} / (65536 * ECL_{mm})
Serial w/ Gate3	Rotary Motor = 2048 * NoOfPolePairs / ($2^{Motor[x].PhaseEncLeftShift + ST}$) Linear Motor = 2048 * RES_{mm} / ($2^{Motor[x].PhaseEncLeftShift}$ * ECL_{mm})

Serial w/ ACC-84S	Rotary Motor Linear Motor	= 2048 * NoOfPolePairs / ($2^{Motor[x].PhaseEncLeftShift + ST}$) = 2048 * RES_{mm} / ($2^{Motor[x].PhaseEncLeftShift}$ * ECL_{mm})

PMAC I2T Protection

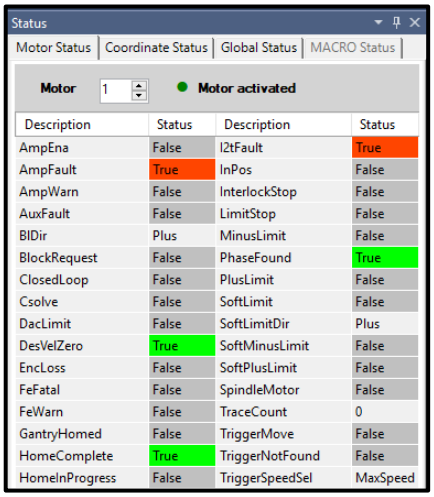
Precautions for Safe Use

- PMAC I2T must be set correctly, especially if the Motor current is lower than the Amplifier. Not doing so could risk damaging the Motor.
- The lower current specifications between the Amplifier and the Motor must always be used in the I2T calculation for proper protection.

This section describes the settings of the PMAC I2T function that protects against Motor overheating.

PMAC I2T fault occurs when **Motor[x].I2tSum** (integrator value) reaches **Motor[x].I2tTrip**. When this happens, the Amplifier enable signal is set to OFF (killing the Motor) immediately, and the amplifier fault and I2T fault status bits are set (as seen in the IDE Motor status window).

These status bits can be accessed using the Motor structure elements **Motor[x].AmpFault** and **Motor[x].I2TFault**.



PMAC I2T setup requires the knowledge of the maximum current, rated current, time allowed at maximum current, and current scaling in the Amplifier.

The lower current specification between the Amplifier and the Motor must always be used in the I2T calculation for proper protection. The following table summarizes this selection process.

Rated RMS Current Condition	**Specification to Use in I2T Calculation**
Motor ≤ Amplifier	Motor rated RMS current
Motor > Amplifier	Amplifier rated RMS current
Maximum RMS Current Condition	**Specification to Use in I2T Calculation**
Motor ≤ Amplifier	• Motor maximum RMS current • Motor time allowed at maximum current
Motor > Amplifier	• Amplifier maximum RMS current • Amplifier time allowed at maximum current

Assuming the following definitions, below are the expressions for setting up PMAC I2T with an Analog Drive/Amplifier.

Name	Description
Transconductance	Amps per Volt
CtsPerVolt	16384 / 10 is typical for filtered PWM
ContCur	Lowest rated current between Motor and Amplifier
PeakCur	Lowest maximum current between Motor and Amplifier
TimeAtPeak	Time at maximum current in seconds

Structure Element	Description and Example
Motor[x].MaxDac	**Maximum servo output** = PeakCur * CtsPerVolt / Transconductance
Motor[x].I2tSet	**Continuous current limit** = Ch1ContCur * Ch1CtsPerVolt / Transconductance
Motor[x].I2tTrip	**Maximum (shutdown) current limit** = (POW(Motor[1].MaxDac,2) - POW(Motor[1].I2TSet,2)) * Mtr1TimeAtPeak

Assuming the following definitions, below are the expressions for setting up PMAC I2T with a Direct PWM Drive/Amplifier.

Name	Description
MaxRmsCur	Lowest maximum RMS current between Motor and Amplifier
RatedRmsCur	Lowest rated RMS current between Motor and Amplifier
TimeAtMaxCur	Time at maximum current in seconds
AmplifierMaxAdc	Amplifier current scaling (defined in amperes peak)

Structure Element	Description and Example
Motor[x].MaxDac	**Maximum servo output** = MaxRmsCur * 32768 * SQRT(2) * COSD(30) / AmplifierMaxAdc
Motor[x].I2tSet	**Continuous current limit** = RatedRmsCur * 32768 * SQRT(2) * COSD(30) / AmplifierMaxAdc
Motor[x].I2tTrip	**Maximum (shutdown) current limit** = ((Motor[x].MaxDac * Motor[x].MaxDac) - (Motor[x].I2tSet * Motor[x].I2tSet)) * TimeAtMaxCur

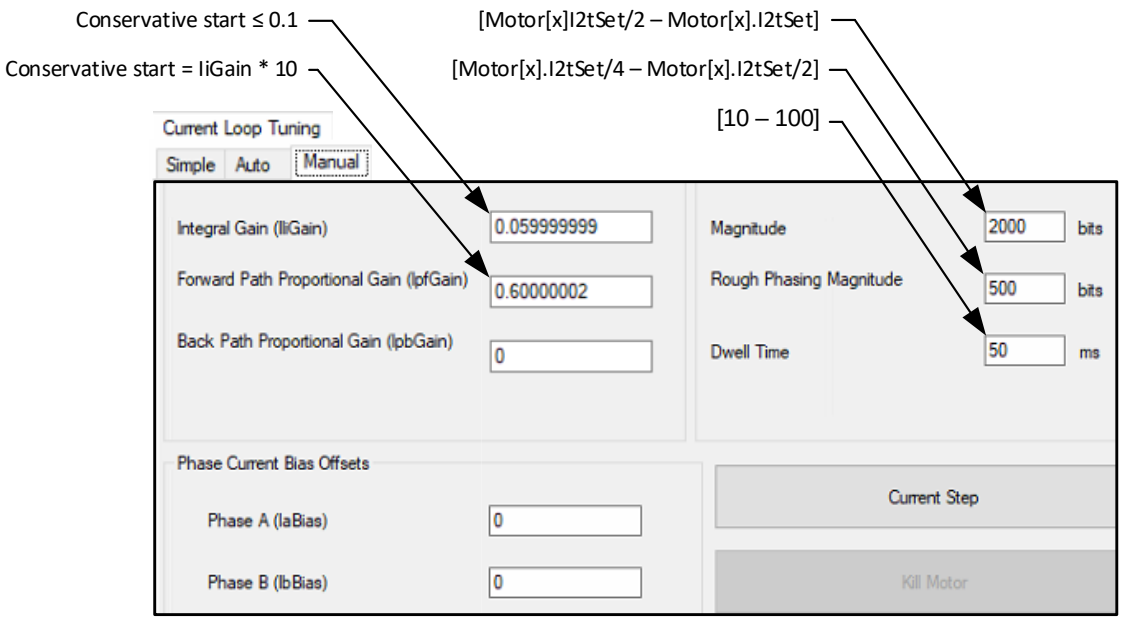


Additional Information

SQRT() and **COSD()** are square root and cosine (degrees) in PMAC script.

Current Loop Tuning

This section shows an overview of the manual current loop tuning tool in the IDE when typically working with a CK3C + ACC-8FS, and Direct PWM amplifier.



Field	Description and Range
Magnitude	**Current applied during current step** Typical range, [Motor[x].I2tSet/2 - Motor[x].I2tSet]
Rough Phasing Magnitude	**Current applied to force Motor to zero electrical cycle before current step** Typical range, [Motor[x].I2tSet/4 - Motor[x].I2tSet/2]
Dwell Time	**Time of current step** Typical range, [10 - 100] msec

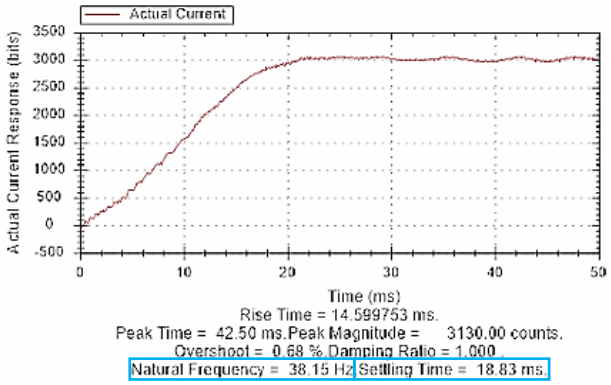
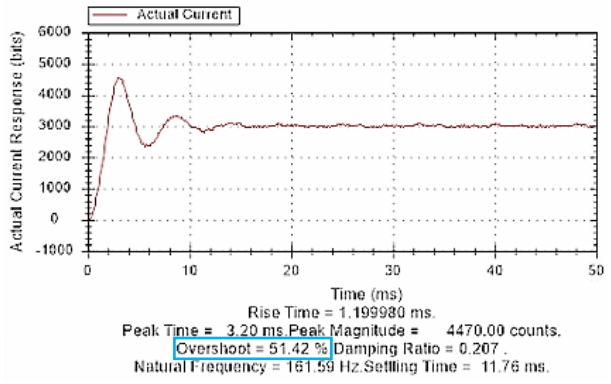
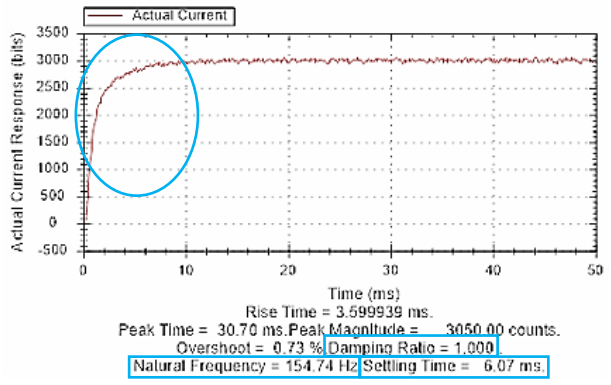
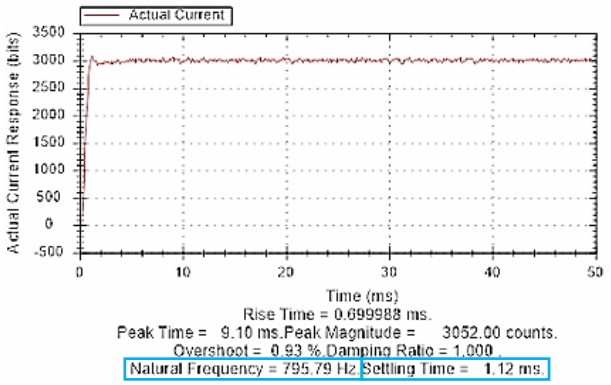
Structure Element	Description and Example
Motor[x].IiGain	**Current loop integral gain** Conservative start, = 0.1 or less
Motor[x].IpfGain	**Current loop forward-path proportional gain** Conservative start, = Motor[x].IiGain * 10
Motor[x].IpbGain	**Current loop back-path proportional gain** Not used in this example guide

Most modern motors' current loop can be tuned using **Motor[x].IiGain** and **Motor[x].IpfGain** only. The goal is to achieve a step response natural frequency in the range of 200-1200Hz (usually 10x faster than position loop), settling time of 1-3 msec or less, and a 3-5% overshoot or less. These characteristics are adequate for most servo Motor applications.

Additional Information

With basic knowledge of Motor and amplifier parameters, it is possible to calculate (starting) current-loop gains analytically. This is described in the Power PMAC User manual.

Below are general guidelines for tuning the current loop.

Step Response	Observation	Counter-measure
	"Sluggish" response. Very low natural frequency and long settling time.	Increase Motor[x].IiGain and Motor[x].IpfGain (proportionally).
	"Under-damped" response. Too much overshoot.	Increase Motor[x].IpfGain.
	"Over-damped" response (curve shape). Causing lower natural frequency and longer settling time.	Increase Motor[x].IiGain or decrease Motor[x].IpfGain.
	Good "Critically damped" response. Natural frequency ~800Hz, Settling time ~1 msec.	None (finished).

Additional Information

Over-tightening the current loop could result in overshoot and oscillation of the response, especially with slower response motors. This is not desirable because it could have adverse effects on the position loop performance.

Establishing Phase Reference (Motor Phasing)

This section shows an overview of the types of phasing in the IDE when typically working with a CK3C + ACC-8FS, Direct PWM amplifier, and brushless motor.



Precautions for Safe Use

Motor phasing in this section will cause Motor movement. Care must be taken.

This section discusses the two most common “initial” phasing methods. There are no fundamental differences between them. Either method can be used with any type of brushless Motor and encoder. The table below provides a suggestion and further information.

“Initial” Phasing	Suggested Use	Final Application Phasing Method
Manual phasing	If Motor assembly has absolute sensor.	Set up absolute phasing using absolute sensor (discussed in a later section)
Automatic stepper	If Motor assembly does not have absolute sensor.	Save stepper phasing method parameters to PMAC, and issue phase command after power-up.



Additional Information

- For details about establishing phase reference, refer to the Power PMAC User’s Manual.
- If the Motor assembly has an absolute sensor such as halls or absolute encoder, an absolute phasing method can be implemented and saved into the Controller. This is discussed in a later section.
- Initial motor phasing is best performed with uncoupled Motor.

Manual Phasing Method

The manual phasing method consists of locking the Motor tightly onto the zero position of the commutation electrical cycle (conventionally phase B in PMAC) by forcing current into the **Motor[x].IbBias** offset. The tighter the Motor is locked, the better is the phase reference. This procedure is usually performed from the terminal window in the IDE using the steps described below.

In the table below, “x” represents the motor number.

Step	Command	Explanation
1	#xK	Make sure Motor is killed and stationary
2	Motor[x].IbBias = Motor[x].I2tSet/4	Amount of current (torque) [Motor[x].I2tSet/4 - Motor[x].I2tSet]
3	#xOUT0	Enable Amplifier (this will move the Motor) Wait for Motor to settle (stop moving)
4	Motor[x].PhasePos = 0	Tell PMAC that this is the zero phase position location
5	Motor[x].PhaseFound = 1	Set status bit
6	Motor[x].IbBias = 0 #xK	Reset Motor[x].IbBias, and kill Motor



Additional Information

- The Motor may oscillate during this procedure, **Motor[x].IbBias** can be reduced. Alternately, the Motor could be stabilized by hand if safe and possible.
- If the Motor assembly does not have an absolute sensor, the manual phasing method can be implemented (and saved) in a PLC program for on-demand phasing.
- An open loop test (discussed further below) verifies the success of this method.

Automatic Stepper-Motor Phasing Method

The automatic stepper phasing method is one of two phase referencing routines built into the Power PMAC firmware. It requires the setting of the following structure elements.

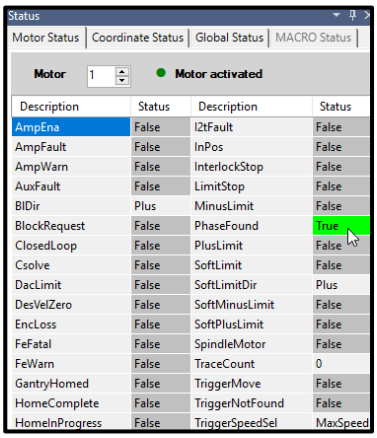
Structure Element	Description and Example
Motor[x].PhaseFindingDac	**Phasing search output (current/torque) magnitude** Typical range, [Motor[x].I2tSet/4 – Motor[x].I2tSet/2]
Motor[x].PhaseFindingTime	**Phasing search time** = {Time in msec} / (2 * Sys.ServoPeriod * (Sys.RtIntPeriod + 1))
Motor[x].AbsPhasePosOffset	**Minimum motion that qualifies a valid phasing search move** Typical, = 2048 / 5

Once these settings are downloaded into the PMAC, initiating a phasing search move could be issued from either the terminal, or PLC program:

- From the terminal window: **#x$** (where x is the Motor number)
- From PLC or subroutine: **Motor[x].PhaseFindingStep = 1**

If the phasing search move is successful, the **Motor[x].PhaseFound** status bit will be set to true.

If the phasing search move is not successful, try one of the following tips:



- Increase **Motor[x].PhaseFindingDac**
  The result must be greater than 256 or else PMAC will use the 4-guess method.
- Increase **Motor[x].PhaseFindingTime**

- Reverse encoder decode (for incremental encoders)
  CK3C[0].Chan[j].EncCtrl =7 <-> **3**

- For serial encoders, it is not simple to reverse counting direction. Try setting **Motor[x].PwmSf**, and **Motor[x].PhaseOffset** simultaneously to the opposite sign to reverse the commutation direction.

- Swap two of the Motor leads

- De-couple the Motor from the load and try again

Open Loop Test

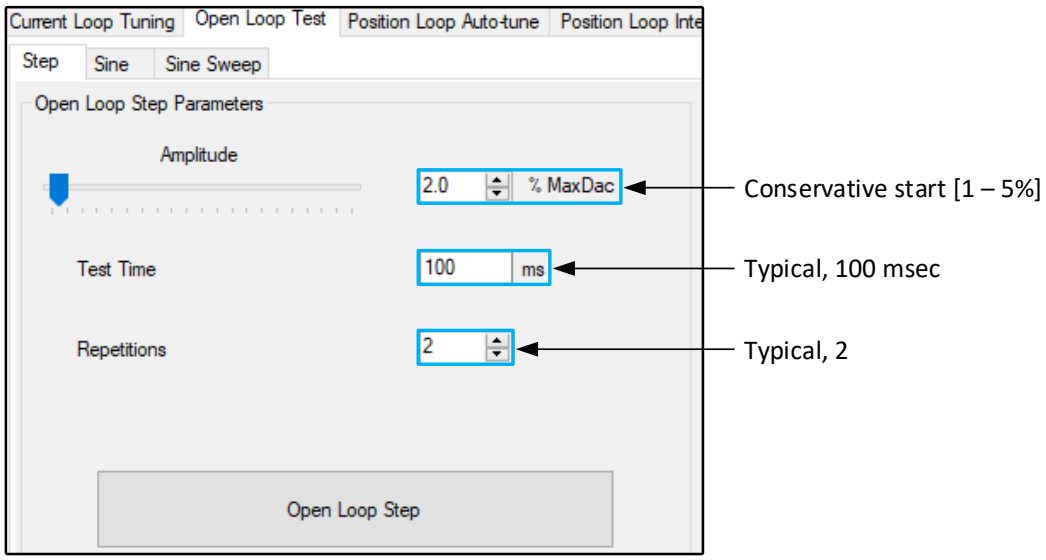
Precautions for Safe Use

- Do not attempt to close the position loop on a Motor which open loop test has not passed. This may lead to dangerous runaway conditions.
- The open loop test will move the Motor. Care must be taken.

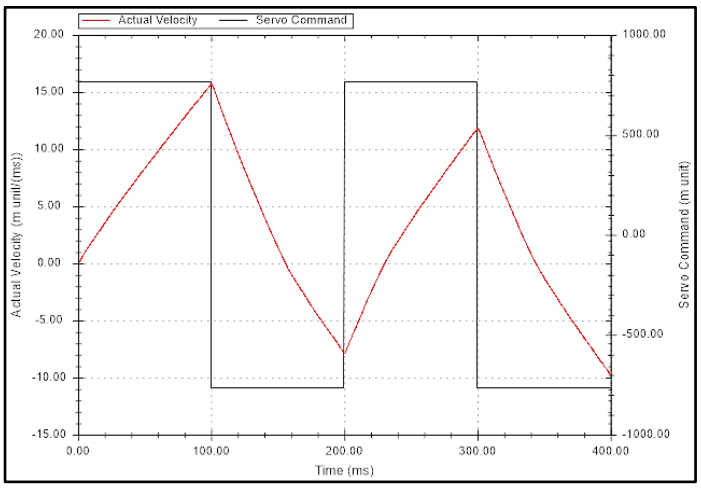
The open loop test is a critical step in the setup of a commutated “brushless” Motor. Primarily, it provides verification for the following functions.

- Initial phasing success
- Amplifier output is in the same direction as the encoder counting sense

The open loop test is typically performed using the IDE tuning tool shown below.



The goal of the open loop test is to obtain positive (velocity) movement when a positive (open loop) command is issued, and vice-versa. The expected “passing” result looks like the following “saw tooth”.



Limited Amount of Travel

Some Motors may have a limited amount of travel which makes it difficult to perform an open loop test. In this case, a manual open loop test could be performed from the terminal window in the IDE.

- **#xK** kills (servo OFF) Motor x.
- **#xOUT1**, is the Motor moving in the positive direction?
- **#xOUT-1**, is the Motor moving in the negative direction?

This could be verified by looking at the Motor velocity window in the IDE. The percentage output can be increased as needed.

Troubleshooting the Open Loop Test

Generally, the open loop test could fail in one of two following ways.

- Motor does not move (feels locked)
- Motor moves in opposite direction of open loop command (response shows inverted “saw tooth”)

This indicates that one or a combination of the following. If any one item is changed, re-phasing the Motor is necessary before attempting another open loop test.

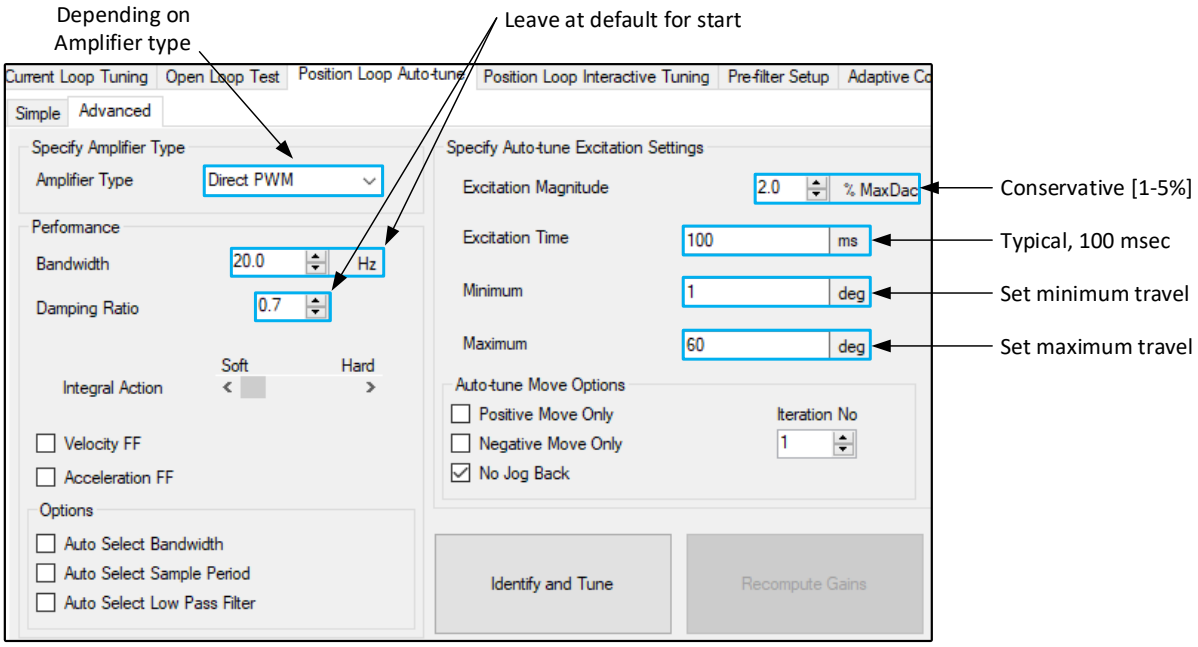
- Incorrect commutation cycle size, **Motor[x].PhasePosSf** setting
- Initial phasing was not performed successfully
- Reversed incremental encoder direction, **CK3C[0].Chan[j].EncCtrl = 7** <-> **3**
  (does not apply to serial encoders)
- Serial encoder direction does not match Amplifier output. In this case, reversing the commutation counting direction is suggested by swapping two of the Motor leads. Or setting **Motor[x].PwmSf**, and **Motor[x].PhaseOffset** simultaneously to the opposite sign.

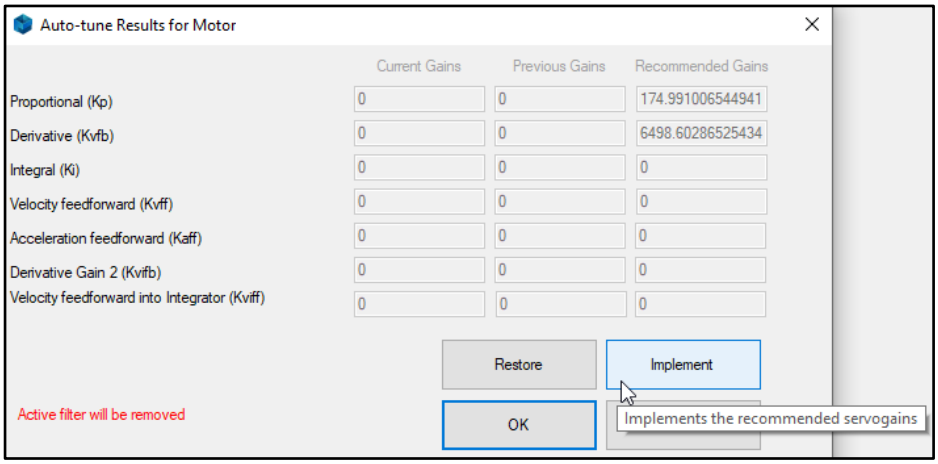
Position Loop (Basic) Tuning

Precautions for Safe Use

Do not attempt to perform position loop tuning prior to passing the open loop test. This may lead to dangerous runaway conditions.

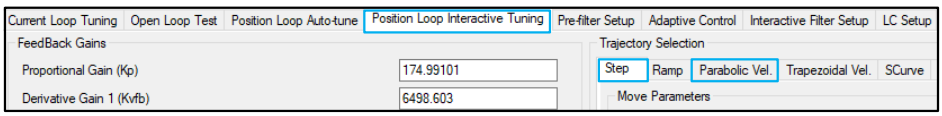
Using the same parameters as the open loop test (output percentage, and excitation time), the auto-tuning tool in the IDE can be used to acquire starting gains.





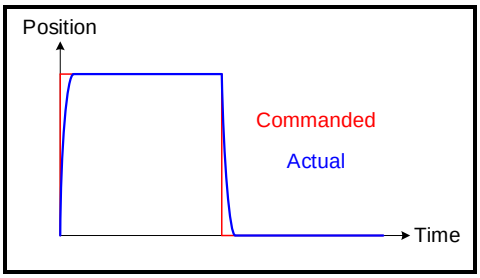
Implementing the gains proposed by the auto-tune automatically changes the window to “interactive tuning”.

In the interactive tuning, we will discuss the most commonly used step and parabolic tuning moves.



● Step Move (Basic) Tuning

The goal of the step move tuning is to “superimpose” the actual position on top of the commanded “step” position.

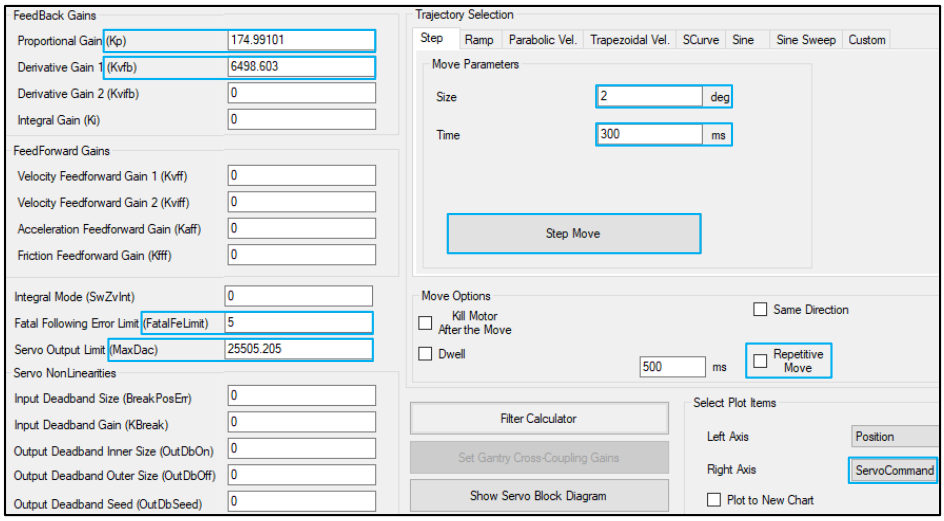


That is, by optimizing the following structure element gains.

- **Motor[x].Servo.Kp**
- **Motor[x].Servo.Kvfb**

Preparing for the step move tuning:

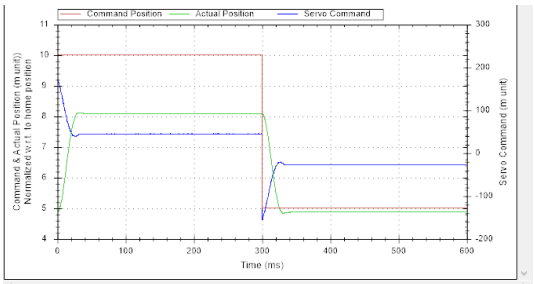
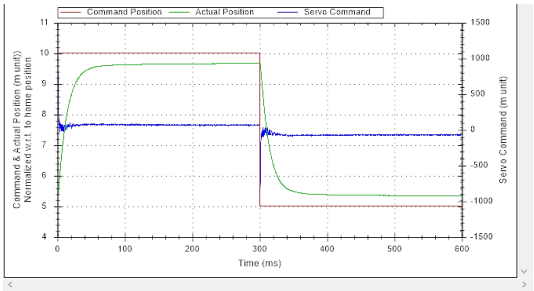
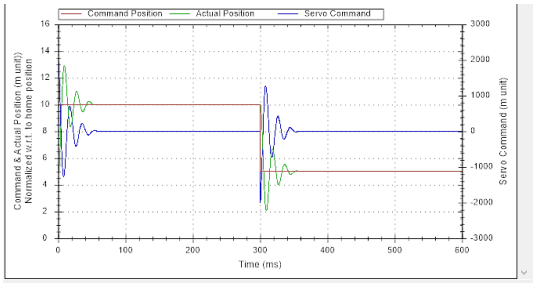
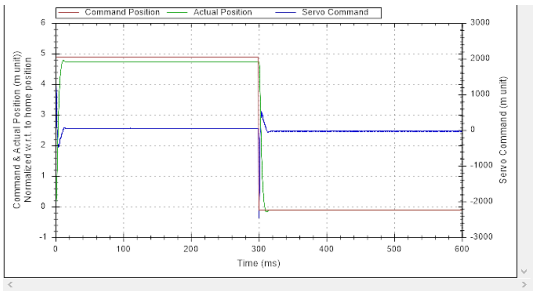
- Make sure all other gains are set to zero
- Choose move size that moves the load
  Rotary Motor typical range [1 – 5 deg]
  Linear Motor typical range [0.1 – 0.5 mm]
- Choose move time, typical range [300 – 2000 msec]
- The fatal following error must be greater than the step move. It can be slightly increased temporarily for the step move tuning exercise
- Keep in mind servo output limit (Motor[x].MaxDac), compare to servo “effort” command in transient part of the step move response



Additional Information

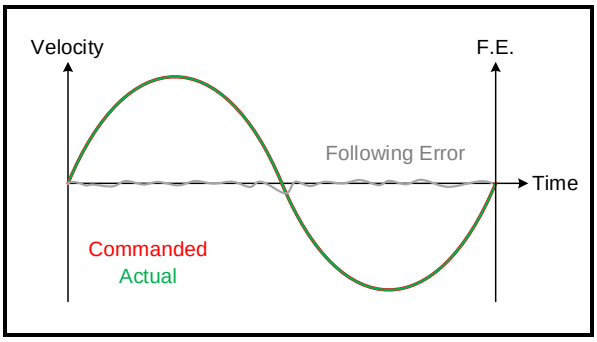
Try (single) step move at first. If safe and stable, check the repetitive move box, then change gains on the fly while monitoring the change on the plot.

Below are general guidelines for step move tuning.

Step Response	Observation	Counter-measure
 Stats Decimal Places 2 Natural Frequency 23.70 Hz | Overshoot 1.20% Damping Ratio 0.82 | Rise Time 17.20 ms Settling Time 24.83 ms | Peak Magnitude 3.24 mu Peak Time 30.60 ms | RMS Following Error 1.57	• “Sluggish” response. Motor may feel loose. • Too little servo “effort” command. About 1% of Motor[x].MaxDac in this example. • Relatively long settling time	Increase Motor[x].Servo.Kp and Motor[x].Servo.Kvfb (proportionally)
 Stats Decimal Places 2 Natural Frequency 18.69 Hz | Overshoot 0.05% Damping Ratio 1.00 | Rise Time 29.80 ms Settling Time 43.14 ms | Peak Magnitude 4.33 mu Peak Time 278.60 ms | RMS Following Error 0.88	• “Over-damped” response	Increase Motor[x].Servo.Kp and or decrease Motor[x].Servo.Kvfb
 Stats Decimal Places 2 Natural Frequency 59.56 Hz | Overshoot 58.71% Damping Ratio 0.17 | Rise Time 3.20 ms Settling Time 39.92 ms | Peak Magnitude 7.93 mu Peak Time 9.20 ms | RMS Following Error 0.65	• “Under-damped” response • Too much overshoot	Increase Motor[x].Servo.Kvfb
 Stats Decimal Places 2 Natural Frequency 99.47 Hz | Overshoot 0.89% Damping Ratio 1.00 | Rise Time 5.60 ms Settling Time 8.06 ms | Peak Magnitude 4.89 mu Peak Time 11.60 ms | RMS Following Error 0.50	• “Critically-damped” response • For coupled Motors, should try to achieve ~80% Motor[x].MaxDac • Fast settle time (<10 msec) • Little to no overshoot (<1%) • High natural frequency (~100 Hz)	Good step move response

● Parabolic Move (Basic) Tuning

The goal of the parabolic move tuning is to "superimpose" the actual velocity on top of the commanded "parabolic" velocity with minimum and centered (around zero) following error.

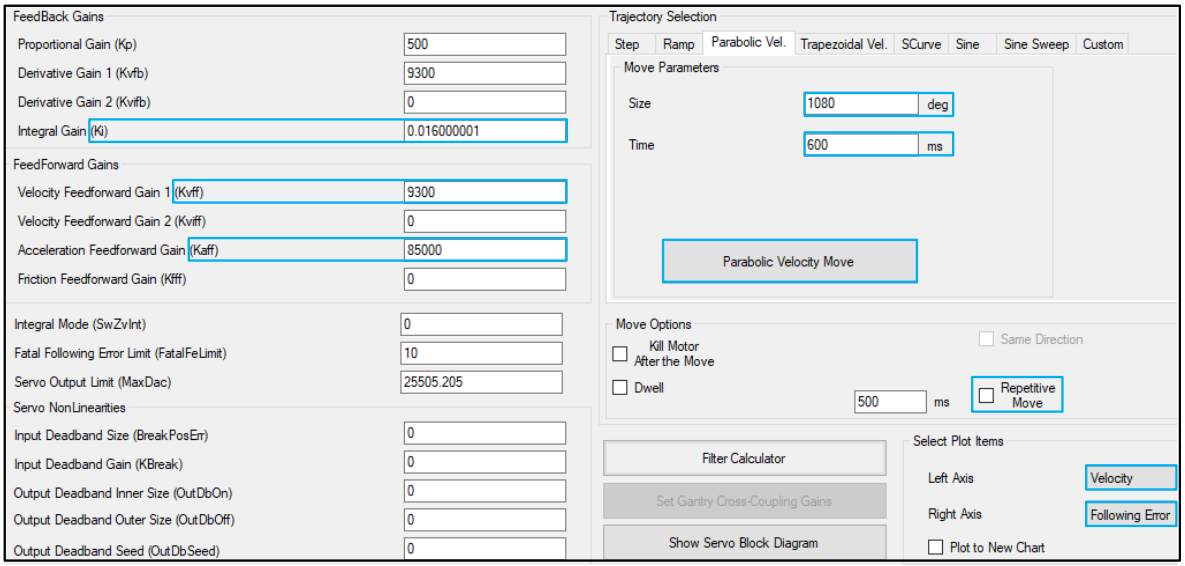


That is, by optimizing the following structure element gains.

- **Motor[x].Servo.Kvff**
- **Motor[x].Servo.Kaff**
- **Motor[x].Servo.Ki**

Preparing for the parabolic move tuning:

- Copy **Motor[x].Servo.Kvfb** into **Motor[x].Servo.Kvff** (these two gains are almost always equal)
- Choose move size that moves the load
  Rotary Motor typical range [1 – 4 revolutions]
  Linear Motor typical range [10 – 80 mm]
- Choose move time, typical range [600 – 2000 msec]
- Typically, it is desirable to achieve ~20-40% of anticipated Motor maximum speed

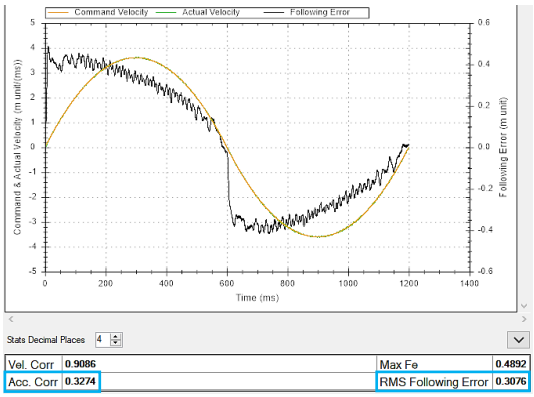
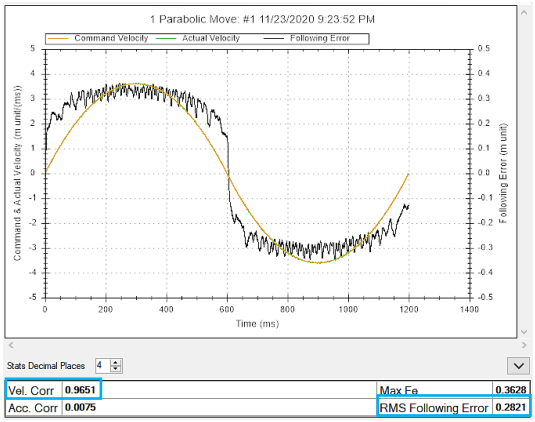
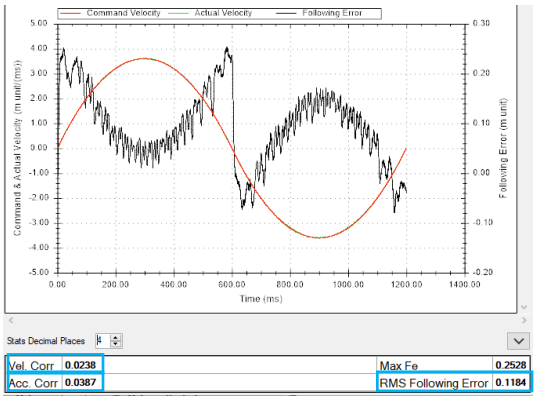
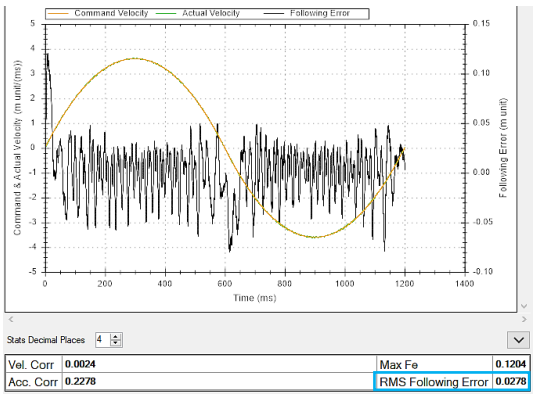


Additional Information

Try (single) parabolic move at first. If safe and stable, check the repetitive move box, then change gains on the fly while monitoring the change on the plot.

It is recommended to optimize **Motor[x].Servo.Kaff** first before **Motor[x].Servo.Ki**.

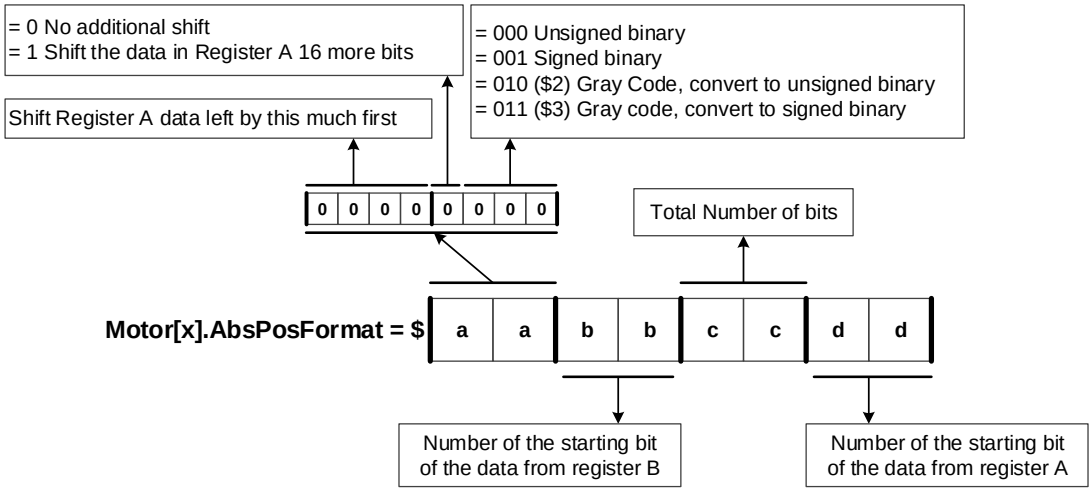
Below, are general guidelines for parabolic move tuning.

Step Response	Observation	Counter-measure
	High acceleration correlation (following error looks like V shape)	Increase Motor[x].Servo.Kaff
	High velocity correlation (following error looks like step) Note, that for move and settle (point to point) applications, it is not necessary to adjust Motor[s].Servo.Kvff, always keep equal to Motor[x].Servo.Kvfb	Increase Motor[x].Servo.Kvff
	Little to no acceleration, and velocity correlations Notice the following error scale Following error shape centered around zero	Increase Motor[x].Servo.Ki
	Following error optimized around zero For the spike in following error (change of direction) in the middle of the plot, it is possible (for contouring applications) to add a small amount of Motor[x].Servo.Kfff.	Good parabolic move response

Absolute Position Setup (Absolute Homing)

This section describes the basic operation for setting up “absolute” position read for absolute serial encoders. Typically performed with supporting hardware that may include a CK3C+ACC-8FS+ACC-84S, Direct PWM Amplifier, and brushless motor combination. The necessary elements to set up for this feature are shown below.

Structure Element	Description and Example
Motor[x].pAbsPos	**Power-on absolute position pointer** Typically, for Gate3, = CK3C[0].Chan[j].SerialEncDataA.a Typically, for ACC-84S, = ACC84S[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPosSf	**Absolute position read scale factor** = Motor[x].PosSf
Motor[x].AbsPosFormat	**Power-on absolute data format** See description and examples (further) below.
Motor[x].HomeOffset	**Positioning reference offset** Set by the user (in Motor units) to indicate desired zero location

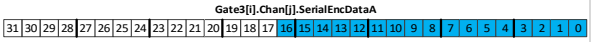
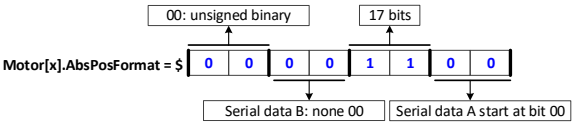
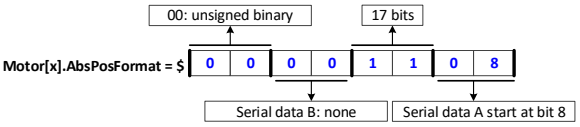
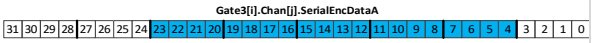
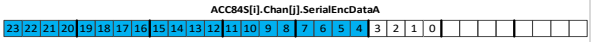
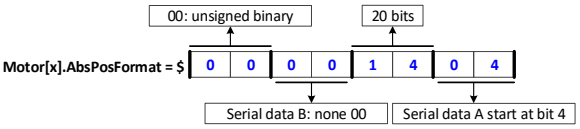
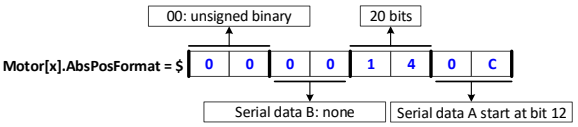
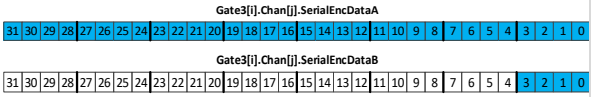
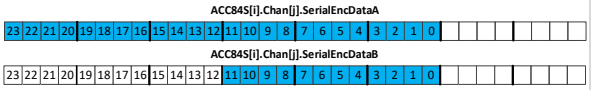
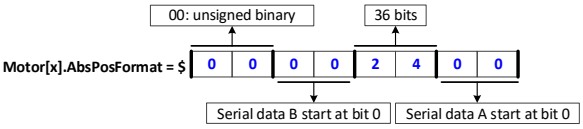
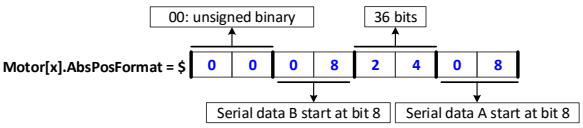
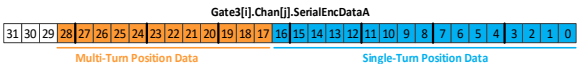
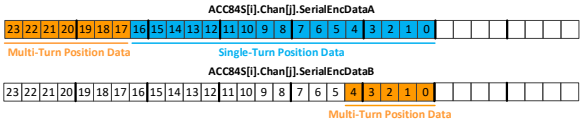
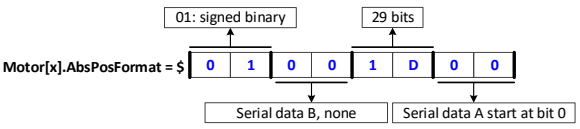
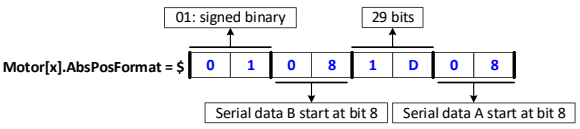


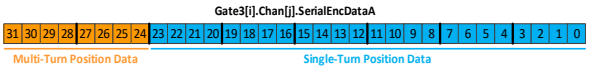
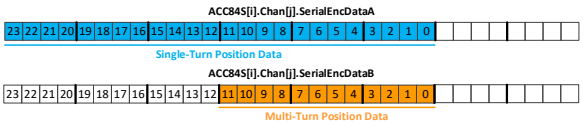
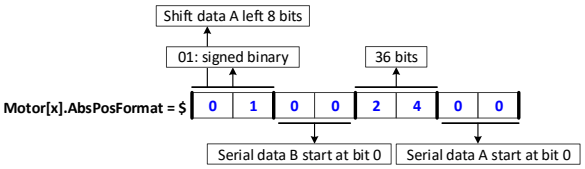
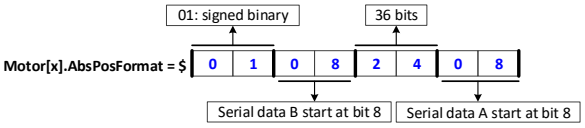
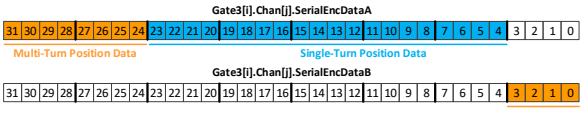
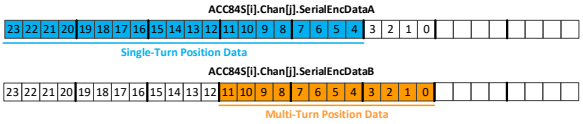
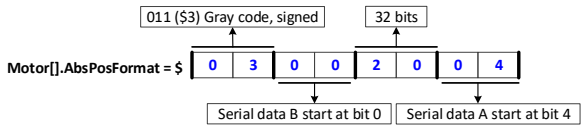
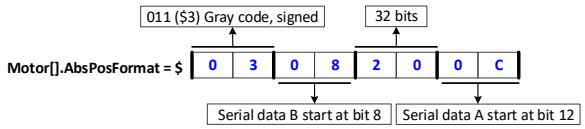
Once these settings have been downloaded to PMAC. This function can be used as follows.

- From terminal window: **#xHMZ** will read absolute position of Motor x
- From program (PLC or Motion): **HOMEZ** x will read absolute position of Motor x
- Alternately, bit 1 of **Motor[x].PowerOnMode** (= $2) specifies absolute position read on power-up. Make sure encoder power is available (if power source is external to axis card) at power-up.

Additional Information

Gray code conversion should be omitted here if it had been already implemented in the Serial Encoder Command word **CK3C[0].Chan[j].SerialEncCmd**, or **ACC84S[i].Chan[j].SerialEncCmd**.

Example	Motor[x].AbsPosFormat Setting
An absolute serial encoder with 17 bits of binary single-turn (or linear scale), and no multi-turn, position data located in the lower fields of serial data A register. • Gate3  • ACC-84S 	• Gate3  • ACC-84S 
An absolute serial encoder with 20 bits of binary single-turn (or linear scale), and no multi-turn, position data starting at bit 4 of serial data A register. • Gate3  • ACC-84S 	• Gate3  • ACC-84S 
An absolute serial encoder with 36 bits of single-turn (or linear scale) position data located in serial data A and B registers consecutively. • Gate3  • ACC-84S 	• Gate3  • ACC-84S 
A 29-bit binary absolute serial encoder with 17 bits of single-turn and 12 bits of multi-turn position data starting at bit 0 of serial data A register and continuously extending to bit 28. • Gate3  • ACC-84S 	• Gate3  • ACC-84S 

Example	Motor[x].AbsPosFormat Setting
A 36-bit binary absolute serial encoder with 24 bits of single-turn position data, and 12 bits of multi-turn position data. • Gate3  • ACC-84S  	• Gate3  • ACC-84S 
A 32-bit Gray code absolute serial encoder with 20 bits of single-turn position data (starting at bit 4), and 12 bits of multi-turn position data. • Gate3  • ACC-84S  **Note** If gray code is already set up in global command word, then it should not be set up here.	• Gate3  • ACC-84S 



Additional Information

If, in the same serial encoder data register, there is a gap between the single-turn and multi-turn data (non-contiguous case), then the position data must be assembled manually (e.g. in a PLC), and placed in user memory registers (e.g. **Sys.Udata[i]**) to eliminate the gap. Then the absolute homing structure elements can be applied to this register.

Absolute Phase Reference Setup

Precautions for Correct Use

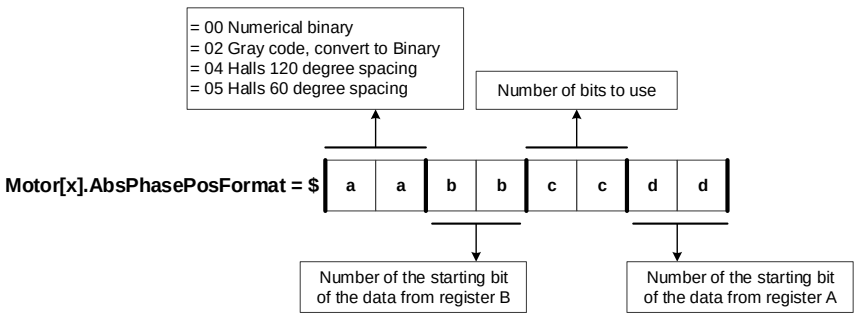
- Motor must have been successfully phased prior to setting up absolute phase reference.
- **Motor[x].PhaseFindingDac**, and **Motor[x].PhaseFindingTime** must be equal to zero.

This section describes the setup of absolute phase (commutation) reference for Motors with Hall effect sensors and absolute serial encoder. Typically performed with supporting hardware that may include a CK3C+ACC-8FS + ACC-84S, Direct PWM Amplifier, and brushless motor combination. The following definitions are used in this section.

Name	Description
ST	Single-turn serial rotary encoder position data
NoOfPolePairs	Number of pair poles of a rotary brushless motor
ECLmm	Linear Motor electrical cycle length or magnetic pitch in mm (e.g. 30.48mm)
RESmm	Linear encoder resolution in mm (e.g. 1µm = 0.001mm)

The primary structure elements for setting up the absolute phase reference are shown below.

Structure Element	Description and Example
Motor[x].pAbsPhasePos	**Power-on absolute phase position pointer** For halls, = CK3C[0].Chan[j].Status.a For absolute serial encoders, = CK3C[0].Chan[j].SerialEncDataA.a
Motor[x].AbsPosSf	**Absolute position read scale factor** For halls, = ±2048 / 12 (depends on UVW transition) For serial absolute rotary encoders, = 2048 * NoOfPolePairs / 2^{ST} For serial absolute linear encoders, = 2048 * RESmm / ECLmm
Motor[x].AbsPhasePosFormat	**Power-on absolute phase data format** For halls 120° spacing, = $400030C For absolute serial encoders, see description and examples (further) below.
Motor[x].AbsPhasePosOffset	**Absolute phase position reference** For halls, = value of phase position that intersect the 1 <-> 3 transition of UVW For absolute serial encoders, = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf



Once these settings have been downloaded to PMAC. This function can be used as follows.

- From terminal window: **#x$** will read absolute position of Motor x
- From PLC program: **Motor[x].PhaseFindingStep = 1** will read absolute position of Motor x
- Alternately, bit 2 of **Motor[x].PowerOnMode** (= $4) specifies absolute phase reference on power-up. Make sure encoder power is available (if power source is external to axis card) at power-up.

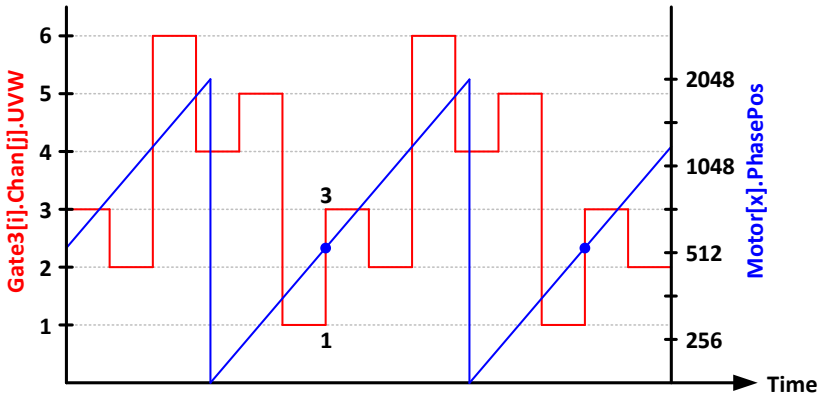
● Absolute Phase Reference Using Hall Effect Sensors

Absolute phase reference setup using the conventional 120° hall effect sensor spacing is described in this section.

Precautions for Correct Use

- Motor must have been successfully phased prior to setting up absolute phase reference.
- **Motor[x].PhaseFindingDac**, and **Motor[x].PhaseFindingTime** must be equal to zero.

Gather, using the IDE plot, structure elements **CK3C[0].Chan[j].UVW** and **Motor[x].PhasePos** while moving the Motor **in the positive direction** (typically slow).



This plot reveals two important halls settings:

- **Motor[x].PhasePosSf** **= 2048 / 12** if the transition of UVW is from 1 to 3
  = **–2048 / 12** if the transition of UVW is from 3 to 1

- **Motor[x].AbsPhasePosOffset** = {value of **Motor[x].PhasePos**} at 1<-> 3 transition

● Absolute Phase Reference Using an Absolute encoder



Precautions for Correct Use

- Motor must have been successfully phased prior to setting up absolute phase reference.
- **Motor[x].PhaseFindingDac**, and **Motor[x].PhaseFindingTime** must be equal to zero.

The maximum number of bits that can be used for absolute phase reference is 32. It is not necessary to use the full position data stream. For simplicity, it is recommended to only use the single-turn data. However, one important point is to always include to the MSB of the position data to be used.

Motor[x].AbsPhasePosForce is only a holding register. Its value is found by performing the following absolute phase reference test procedure in the table below where **x** designates the Motor number.

Phase Force test procedure:

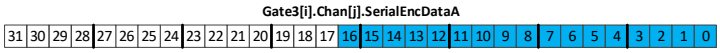
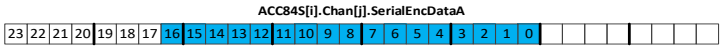
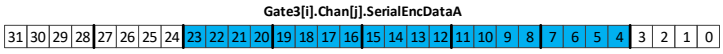
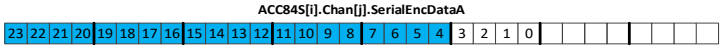
Step	Command	Explanation
1	#xK	Make sure Motor is killed and stationary
2	Motor[x].IbBias = Motor[x].I2tSet/4	Amount of current (torque) [Motor[x].I2tSet/4 - Motor[x].I2tSet]
3	#xOUT0	Enable Amplifier (this will move the Motor) Wait for Motor to settle (stop moving)
4	Record position from CK3C[0].Chan[j].SerialEncDataA register	Record value of SerialEncDataA
5	#xK	Kill the Motor
6	Motor[x].IbBias = 0	Reset Motor[x].IbBias
7	Motor[x].AbsPhasePosForce = (see examples on next page)	Refer to appropriate PhaseForce equations in the examples on the next page. Use the value retrieved in Step 4 for SerialEncDataA

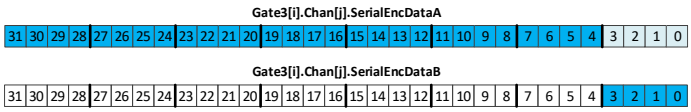
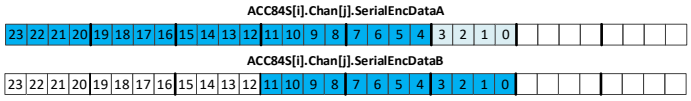
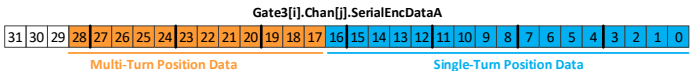
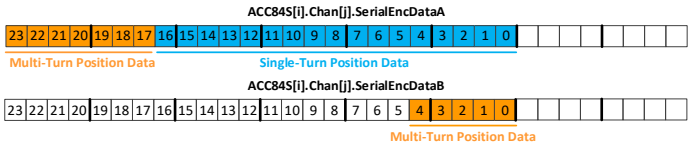


Additional Information

The Motor may oscillate during this procedure, **Motor[x].IbBias** can be reduced. Alternately, the Motor could be stabilized by hand if safe and possible.

See examples on the next page for how the absolute phase reference parameters are set and grouped in motor setup global includes file.

Ex.	Absolute Phase Reference Settings
1	An absolute serial encoder with 17 bits of binary single-turn (or linear scale), and no multi-turn, position data located in the lower fields of serial data A register. • Gate3  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm Motor[x].pAbsPhasePos = CK3C[0].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001100 Motor[x].AbsPhasePosForce = value of {**CK3C[0].Chan[j].SerialEncDataA & $1FFFF**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf • ACC-84S  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm Motor[x].pAbsPhasePos = ACC84S[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001108 Motor[x].AbsPhasePosForce = value of {**ACC84S[i].Chan[j].SerialEncDataA & $1FFFF**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf
2	An absolute serial encoder with 20 bits of binary single-turn (or linear scale), and no multi-turn, position data starting at bit 4 of serial data A register. • Gate3  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{20} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm Motor[x].pAbsPhasePos = CK3C[0].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001404 Motor[x].AbsPhasePosForce = value of {**(CK3C[0].Chan[j].SerialEncDataA & $FFFFF0) >> 4**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf • ACC-84S  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{20} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm Motor[x].pAbsPhasePos = ACC84S[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $0000140C Motor[x].AbsPhasePosForce =value of {**(ACC84S[i].Chan[j].SerialEncDataA & $FFFFF0) >> 4**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

Example	Absolute Phase Reference Settings
3	An absolute serial encoder with 36 bits of single-turn (or linear scale) position data located in serial data A and B registers consecutively. • Gate3  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{32} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm * 2^4 / ECLmm Motor[x].pAbsPhasePos = CK3C[0].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00002004 (ignore lower 4 bits, use upper 32 bits to include MSB) Motor[x].AbsPhasePosForce = value of {**CK3C[0].Chan[j].SerialEncDataA>>4+(CK3C[0].Chan[j].SerialEncDataB&$F)<<28**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf • ACC-84S  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{32} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm * 2^4 / ECLmm Motor[x].pAbsPhasePos = ACC84S[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $0008200C (ignore lower 4 bits, use upper 32 bits to include MSB) Motor[x].AbsPhasePosForce = value of {**ACC84S[i].Chan[j].SerialEncDataA>>4+(ACC84S[i].Chan[j].SerialEncDataB&$FFF)<<20**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf
4	A 29-bit binary absolute serial encoder with 17 bits of single-turn and 12 bits of multi-turn position data starting at bit 0 of serial data A register and continuously extending to bit 28. • Gate3  Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17} Motor[x].pAbsPhasePos = CK3C[0].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001100 (use only 17-bit single-turn data in register A) Motor[x].AbsPhasePosForce = value of {**CK3C[0].Chan[j].SerialEncDataA & $1FFFF**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf • ACC-84S  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17} Motor[x].pAbsPhasePos = ACC84S[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001108 (use only 17-bit single-turn data in register A) Motor[x].AbsPhasePosForce = value of {**ACC84S[i].Chan[j].SerialEncDataA & $1FFFF**} Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

A-1-5 Reversing Motor Operation Direction



Precautions for Safe Use

- The following setting modifications should not be applied while the Motor is energized.
- Motor must be re-phased after implementing operation direction change.
- Over-travel and soft limits must be modified accordingly when a Motor operation direction is changed.

Often times, in certain applications, it is desired to reverse the Motor operation direction after set up has been complete. For brushless Motors, the “new” settings of the structure elements necessary to apply this change are shown below. This depends on the encoder type.

Motors with Digital Quadrature, and Analog Sinusoidal Encoders

Structure Element	New Value
Motor[x].PhaseOffset	= – present value
Motor[x].PwmSf	= – present value
CK3C[0].Chan[j].EncCtrl	= 3 or 7 (opposite decode value)
Motor[x].AbsPhasePosSf (if using halls)	= – present value
Motor[x].AbsPhasePosOffset (if using halls)	= 2048 – present value

Motors with Serial Encoders

Structure Element	New Value
Motor[x].PhaseOffset	= – present value
Motor[x].PwmSf	= – present value
Motor[x].PhasePosSf	= – present value
EncTable[n].ScaleFactor	= – present value
Motor[x].AbsPosSf (if absolute encoder)	= – present value
Motor[x].AbsPhasePosSf (if absolute encoder)	= – present value
Motor[x].AbsPhasePosOffset (if absolute encoder)	= 2048 – present value

A-2 Sample Motor Setup Examples

Brushless Motor setup samples with CK3C, ACC-8AS, ACC-8FS, ACC-84S, and ACC-51S.



Precautions for Safe Use

The samples provided in this section are for reference only, they may not operate your Motor immediately. Care must be taken for the following items:

- Encoder and Motor specification parameters
- Current loop tuning must be performed
- Initial phasing must be performed
- Open loop test must be performed
- Position loop tuning must be performed
- Absolute position setup (if applicable) should be reviewed
- Absolute phase reference setup (if applicable) should be reviewed
- Generic motion parameters (e.g. Jog speed) must be reviewed



Additional Information

For simplicity, the samples in this section refer to Motor 1 and CK3C[0].

A-2-1 Filtered PWM Output with Quadrature Encoder

Hardware Configuration

Item	Notes
Controller	CK3C
Amplifier/Drive	R88D-KT01L
Motor	OMRON R88M-K10030L-S2, 168 VDC, 1.6/3.2 A
Encoder	Digital quadrature, 2,500-line (10,000 counts per revolution with x4 decode)

System Gates

```
Sys.WpKey = $AAAAAAAA                                          DISABLE WRITE-PROTECTION
Sys.MaxCoords = 3                                              MAX. COORD SYSTEMS+1 (EFFICIENCY)
Sys.MaxMotors = 5                                              MAX. NO. OF MOTORS+1 (EFFICIENCY)

CK3C[0].PhaseFreq = 10000        // 10 KHz Phase               10 KHZ PHASE FREQUENCY
CK3C[0].PhaseClockDiv = 0
CK3C[0].ServoClockDiv = 3        // 2.5 KHz Servo              2.5 KHZ SERVO FREQUENCY

Sys.RtIntPeriod = 0                                            2.5 KHZ RTI FREQUENCY
Sys.PhaseOverServoPeriod = 1/( CK3C[0].ServoClockDiv+1)        TYPICAL CALCULATION
Sys.ServoPeriod = 1000*( CK3C[0].ServoClockDiv+1)/ CK3C[0].PhaseFreq   TYPICAL CALCULATION

CK3C[0].Chan[0].PwmFreqMult = 5                                CH0-3 30 KHZ PWM FREQUENCY
CK3C[0].Chan[1].PwmFreqMult = 5
CK3C[0].Chan[2].PwmFreqMult = 5
CK3C[0].Chan[3].PwmFreqMult = 5

CK3C[0].Chan[0].OutputMode = 0                                 CH0-3 FILTERED PWM OUTPUT ON PHASE A-D
CK3C[0].Chan[1].OutputMode = 0
CK3C[0].Chan[2].OutputMode = 0
CK3C[0].Chan[3].OutputMode = 0

CK3C[0].Chan[0].PwmDeadTime = 0
CK3C[0].Chan[1].PwmDeadTime = 0
CK3C[0].Chan[2].PwmDeadTime = 0
CK3C[0].Chan[3].PwmDeadTime = 0

CK3C[0].Chan[0].PackInData = 0                                 CH0-3 IN DATA
CK3C[0].Chan[1].PackInData = 0
CK3C[0].Chan[2].PackInData = 0
CK3C[0].Chan[3].PackInData = 0

CK3C[0].Chan[0].PackOutData = 0                                CH0-3 OUT DATA
CK3C[0].Chan[1].PackOutData = 0
CK3C[0].Chan[2].PackOutData = 0
CK3C[0].Chan[3].PackOutData = 0
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                           TYPICAL QUADRATURE ENCODER ENTRY
EncTable[1].pEnc = CK3C[0].Chan[0].ServoCapt.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 0
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / 256
EncTable[1].MaxDelta = 0

EncTable[2].Type = 0                                           END OF ECT (FOR EFFICIENCY)
```

```
GLOBAL Transconductance = 0.26                                          TRANSDUCTANCE (AMPS PER VOLT)
GLOBAL CtsPerVolt = 16384 / 10                                          COUNTS PER VOLT
GLOBAL Mtr1ContCur = 1.6                                                MAX RMS CONT CUR.(MTR<AMP)
GLOBAL Mtr1PeakCur = 2.6                                                RATED/PEAK RMS CUR.(MTR<AMP)
GLOBAL Mtr1TimeAtPeak = 1                                               TIME ALLOWED AT MAX CUR.

Motor[1].ServoCtrl = 1                                                  ACTIVATE CHANNEL
Motor[1].PosUnit = 3                                                    IDE WINDOW UNITS, MM
Motor[1].PosSf = 5/10000                                                SCALED POSITION TO MM PER REV
Motor[1].Pos2Sf = 5/10000                                               SCALED VELOCITY TO MM PER REV
Motor[1].pLimits = CK3C[0].Chan[0].Status.a                             ENABLE LIMITS, =0 TO DISABLE

Motor[1].AbortTa = 100                                                  ABORT DECELERATION, 100 MSEC
Motor[1].AbortTs = 0                                                    ABORT S-CURVE, NONE

Motor[1].JogTa = 100                                                    JOG ACCELERATION, 100 MSEC
Motor[1].JogTs = 0                                                      JOG S-CURVE, NONE
Motor[1].JogSpeed = 0.0025                                              JOG SPEED, 2.5 MM/MSEC

Motor[1].FatalFeLimit = 0.1                                             FE LIMIT, 0.1 MM
Motor[1].WarnFeLimit = 0                                                WARNING FE LIMIT, NONE

Motor[1].InPosBand = 0.005                                              INPOS BAND, 5 MICROMETERS
Motor[1].InPosTime = 20 / Sys.ServoPeriod                               INPOS TIME
Motor[1].HomeOffset = 0                                                 MOTOR HOME OFFSET
Motor[1].DacBias = 0                                                    DAC OUTPUT BIAS OFFSET

Motor[1].AmpFaultLevel = 1                                              AMPLIFIER DEPENDENT

Motor[1].MaxDac = Ch1PeakCur * Ch1CtsPerVolt / Transconductance         TYPICAL I2T FOR BRUSHED MOTOR
Motor[1].I2tSet = Ch1ContCur * Ch1CtsPerVolt / Transconductance
Motor[1].I2tTrip = (POW(Motor[1].MaxDac,2)-POW(Motor[1].I2TSet,2))*Mtr1TimeAtPeak


Motor[1].Servo.Kp = 0                                                   POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                                 (MUST TUNE / USER INPUT)
Motor[1].Servo.Kvifb = 0
Motor[1].Servo.Kvff = 0
Motor[1].Servo.Kviff = 0
Motor[1].Servo.Ki = 0
Motor[1].Servo.Kaff = 0
Motor[1].Servo.Kfff = 0
Motor[1].Servo.SwZvInt = 0
```



Additional Information

Please use the **Tune Tool** in the IDE to determine the values of the tuning parameters in the code above (i.e. **Motor[x].Servo.Kp**) for the Position Servo Loop gains.

A-2-2 Direct PWM Output with Quadrature Encoder (ACC-8FS)

Hardware Configuration

Item	Notes
Controller	CK3C + ACC-8FS
Amplifier	CK3A-G310L (three-phase 208 VAC main supply voltage)
Motor	Rotary servo Motor, 200 VAC, 11/23 A, 4 pole pairs
Encoder	Digital quadrature, 2,000-line (8,000 counts per revolution with x4 decode)

System Gates

```
Sys.WpKey = $AAAAAAAA                                                    DISABLE WRITE-PROTECTION
Sys.MaxCoords = 2                                                        MAX. COORD SYSTEMS+1 (EFFICIENCY)
Sys.MaxMotors = 5                                                        MAX. NO. OF MOTORS+1 (EFFICIENCY)

CK3C[0].PhaseFreq = 10000                                                10 KHZ PHASE FREQUENCY
CK3C[0].ServoClockDiv = 1                                                5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                      2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (CK3C[0].ServoClockDiv + 1) / CK3C[0].PhaseFreq TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (CK3C[0].ServoClockDiv + 1)               TYPICAL CALCULATION

CK3C[0].Chan[0].PwmFreqMult = 1                                          CH0-3 10 KHZ PWM FREQUENCY
CK3C[0].Chan[1].PwmFreqMult = 1
CK3C[0].Chan[2].PwmFreqMult = 1
CK3C[0].Chan[3].PwmFreqMult = 1

CK3C[0].Chan[0].OutputMode = 0                                           CH0-3 FILTERED PWM OUTPUT ON PHASE A-D
CK3C[0].Chan[1].OutputMode = 0
CK3C[0].Chan[2].OutputMode = 0
CK3C[0].Chan[3].OutputMode = 0

CK3C[0].Chan[0].PwmDeadTime = 3 / 0.0533                                 CH0-3 MIN. 3µSEC FOR G310
CK3C[0].Chan[1].PwmDeadTime = 3 / 0.0533
CK3C[0].Chan[2].PwmDeadTime = 3 / 0.0533
CK3C[0].Chan[3].PwmDeadTime = 3 / 0.0533

CK3C[0].Chan[0].PackInData = 0                                           CH0-3 IN DATA
CK3C[0].Chan[1].PackInData = 0                                           MUST BE 0 FOR CK3A
CK3C[0].Chan[2].PackInData = 0
CK3C[0].Chan[3].PackInData = 0

CK3C[0].Chan[0].PackOutData = 0                                          CH0-3 OUT DATA
CK3C[0].Chan[1].PackOutData = 0                                          MUST BE 0 FOR CK3A
CK3C[0].Chan[2].PackOutData = 0
CK3C[0].Chan[3].PackOutData = 0

CK3C[0].AdcAmpStrobe = $901001                                           CH0-3 STROBE WORD (FOR CK3A)
CK3C[0].AdcAmpClockDiv = 5                                               CH0-3 3.125 MHZ ADC CLOCK
CK3C[0].AdcAmpHeaderBits = 4                                             CH0-3 MUST BE 4 (FOR CK3A)
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                                     TYPICAL QUADRATURE ENCODER ENTRY
EncTable[1].pEnc = CK3C[0].Chan[0].ServoCapt.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 0
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / 256
EncTable[1].MaxDelta = 0

EncTable[2].Type = 0                                                     END OF ECT (FOR EFFICIENCY)
```

```
GLOBAL Ck3aMaxAdc = 31.35                                                  CK3A MAX ADC
GLOBAL Mtr1CtsPerRev = 8000                                                COUNTS PER REVOLUTION
GLOBAL Mtr1PolePairs = 4                                                   NO. OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 20                                                  MAX RMS CUR.(MTR>CK3A)
GLOBAL Mtr1RatedRmsCur = 10                                                RATED RMS CUR.(MTR>CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2                                                TIME ALLOWED AT MAX CUR.

CK3C[0].Chan[0].EncCtrl = 7                                                ENC. COUNTING DIRECTION
Motor[1].ServoCtrl = 1                                                     ACTIVATE CHANNEL
Motor[1].PosUnit = 11                                                      IDE WINDOW UNITS, DEGREES
Motor[1].PosSf = 360 / Mtr1CtsPerRev                                       SCALE POSITION TO DEGREES
Motor[1].Pos2Sf = 360 / Mtr1CtsPerRev                                      SCALE VELOCITY TO DEGREES
Motor[1].pLimits = CK3C[0].Chan[0].Status.a                                ENABLE LIMITS, =0 TO DISABLE

Motor[1].AbortTa = 150                                                     ABORT DECELERATION, 150 MSEC
Motor[1].AbortTs = 0                                                       ABORT S-CURVE, NONE

Motor[1].JogTa = 300                                                       JOG ACCELERATION, 300 MSEC
Motor[1].JogTs = 0                                                         JOG S-CURVE, NONE
Motor[1].JogSpeed = 4.5                                                    JOG SPEED, 4.5 DEG/MSEC

Motor[1].FatalFeLimit = 5                                                  FE LIMIT, 5 DEGREES
Motor[1].WarnFeLimit = 0                                                   WARNING FE LIMIT, NONE

Motor[1].InPosBand = 0.05                                                  INPOS BAND, 0.05 DEGREES
Motor[1].InPosTime = 20 / Sys.ServoPeriod                                  INPOS TIME, 20 MSEC

Motor[1].pPhaseEnc = CK3C[0].Chan[0].PhaseCapt.a                           TYPICAL FOR QUADRATURE
Motor[1].PhaseCtrl = 4                                                     TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 0                                             NONE FOR QUADRATURE
Motor[1].PhaseEncRightShift = 0                                            NONE FOR QUADRATURE
Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / (256 * Mtr1CtsPerRev)         TYPICAL FOR QUADRATURE

Motor[1].AmpFaultLevel = 1                                                 =1 FOR CK3A
Motor[1].AdcMask = $FFFF0000                                               =$FFFF0000 FOR CK3A
Motor[1].PhaseOffset = 683                                                 TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PwmSf = 0.95 * 16384                                              TYPICAL IF MTR VOLTAGE>INPUT

Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc TYPICAL I2T FOR BRUSHLESS
Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc ...
Motor[1].I2tTrip =(POW(Motor[1].MaxDac,2)-POW(Motor[1].I2TSet,2))*Mtr1TimeAtMaxCur ...

Motor[1].CurrentNullPeriod = 0                                             =0 FOR CK3A
Motor[1].IaBias = 0                                                        =0 DURING NORMAL OPERATION
Motor[1].IbBias = 0                                                        =0 DURING NORMAL OPERATION

Motor[1].IiGain = 0                                                        CURRENT LOOP GAINS
Motor[1].IpfGain = 0                                                       (MUST TUNE / USER INPUT)
Motor[1].IpbGain = 0                                                       ...

Motor[1].Servo.Kp = 0                                                      POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                                    (MUST TUNE / USER INPUT)
Motor[1].Servo.Kvifb = 0                                                   ...
Motor[1].Servo.Kvff = 0                                                    ...
Motor[1].Servo.Kviff = 0                                                   ...
Motor[1].Servo.Ki = 0                                                      ...
Motor[1].Servo.Kaff = 0                                                    ...
Motor[1].Servo.Kfff = 0                                                    ...
Motor[1].Servo.SwZvInt = 0                                                 ...

Motor[1].PhaseFindingTime = 8000 / (2 * Sys.ServoPeriod * (Sys.RtIntPeriod + 1)) STEPPER PHASING (8 SECONDS)
Motor[1].PhaseFindingDac = 0.8 * Motor[1].I2tSet                           80% OF I2TSET
Motor[1].AbsPhasePosOffset = 2048 / 5                                      TYPICAL SETTING
Motor[1].PowerOnMode = 0                                                   NO ABS. PHASE ON POWER-UP
```



Additional Information

Please use the **Tune Tool** in the IDE to determine the values of the tuning parameters in the code above (i.e. **Motor[x].IiGain**, **Motor[x].Servo.Kp**) for the Current Loop and Position Servo Loop gains.

A-2-3 Direct PWM Output with Sinusoidal Encoder (ACC-8FS, ACC-51S)

Hardware Configuration

Item	Notes
Controller	CK3C + ACC-8FS + ACC-51S
Amplifier	CK3A-G310L (three-phase 208 VAC main supply voltage)
Motor	Linear servo Motor, 200 VAC, 2.6/8.3 A, 60.96mm electrical cycle length
Encoder	Sinusoidal, 20µm (resolution) sine cycle length

System Gates

```
Sys.WpKey = $AAAAAAAA                                                   DISABLE WRITE-PROTECTION
Sys.MaxCoords = 2                                                       MAX. COORD SYSTEMS+1 (EFFICIENCY)
Sys.MaxMotors = 5                                                       MAX. NO. OF MOTORS+1 (EFFICIENCY)

CK3C[0].PhaseFreq = 10000                                               10 KHZ PHASE FREQUENCY
CK3C[0].ServoClockDiv = 1                                               5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 0                                                     5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (CK3C[0].ServoClockDiv + 1) / CK3C[0].PhaseFreq  TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (CK3C[0].ServoClockDiv + 1)              TYPICAL CALCULATION

CK3C[0].Chan[0].PwmFreqMult = 1                                         CH0-3 10 KHZ PWM FREQUENCY
CK3C[0].Chan[1].PwmFreqMult = 1
CK3C[0].Chan[2].PwmFreqMult = 1
CK3C[0].Chan[3].PwmFreqMult = 1

CK3C[0].Chan[0].OutputMode = 0                                          CH0-3 FILTERED PWM OUTPUT ON PHASE A-D
CK3C[0].Chan[1].OutputMode = 0
CK3C[0].Chan[2].OutputMode = 0
CK3C[0].Chan[3].OutputMode = 0

CK3C[0].Chan[0].PwmDeadTime = 3 / 0.0533                                CH0-3 MIN. 3µSEC FOR G310
CK3C[0].Chan[1].PwmDeadTime = 3 / 0.0533
CK3C[0].Chan[2].PwmDeadTime = 3 / 0.0533
CK3C[0].Chan[3].PwmDeadTime = 3 / 0.0533

CK3C[0].Chan[0].PackInData = 0                                          CH0-3 IN DATA
CK3C[0].Chan[1].PackInData = 0
CK3C[0].Chan[2].PackInData = 0
CK3C[0].Chan[3].PackInData = 0

CK3C[0].Chan[0].PackOutData = 0                                         CH0-3 OUT DATA
CK3C[0].Chan[1].PackOutData = 0
CK3C[0].Chan[2].PackOutData = 0
CK3C[0].Chan[3].PackOutData = 0

CK3C[0].AdcAmpStrobe = $901001                                          CH0-3 STROBE WORD (FOR CK3A)
CK3C[0].AdcAmpClockDiv = 5                                              CH0-3 3.125 MHZ ADC CLOCK
CK3C[0].AdcAmpHeaderBits = 4                                            CH0-3 MUST BE 4 (FOR CK3A)

CK3C[0].EncClockDiv = 3                                                 ENC. SAMPLING 12.5MHZ
CK3C[0].AdcEncClockDiv = 3                                              ENC. ADC SAMPLING 12.5MHZ
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 4                                    SOFTWARE ARCTANGENT EXTENSION CONVERSION
EncTable[1].pEnc = CK3C[0].Chan[0].Status.a             REGISTER WITH QUADRATURE STATE
EncTable[1].pEnc1 = Sys.Piom + $A00000                  REGISTER WITH SINE INPUT (SYS.PIOM + ADDRESS OF ADC'S ON ACC-51S
EncTable[1].index1 = 0
EncTable[1].index2 = 0
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 1                                  USE PMAC3-STYLE IC FORMAT
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1/65536                       FOR RESULT IN ENCODER LINE
EncTable[1].MaxDelta = 0

EncTable[2].Type = 0                                    END OF ECT (FOR EFFICIENCY)
```

```
GLOBAL Ck3a1MaxAdc = 31.35                                          CK3A MAX ADC
GLOBAL Mtr1Res = 0.020                                              RESOLUTION, ENC. PITCH [mm]
GLOBAL Mtr1CtsPerMm = 1 / Mtr1Res                                   COUNTS PER MM
GLOBAL Mtr1Ecl = 60.96                                              ELECTRICAL CYCLE LENGTH [mm]
GLOBAL Mtr1MaxRmsCur = 4                                            MAX RMS CUR.(MTR<CK3A)
GLOBAL Mtr1RatedRmsCur = 2                                          RATED RMS CUR.(MTR<CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2                                         TIME ALLOWED AT MAX CUR.

Motor[1].EncType = 6                                                TYPICAL FOR SINUSOIDAL USING ACC-51S
CK3C[0].Chan[0].EncCtrl = 7                                         ENC. COUNTING DIRECTION

Motor[1].ServoCtrl = 1                                              ACTIVATE CHANNEL
Motor[1].PosUnit = 3                                                IDE WINDOW UNITS [mm]
Motor[1].PosSf = 1 / Mtr1CtsPerMm                                   SCALE POSITION TO mm
Motor[1].Pos2Sf = 1 / Mtr1CtsPerMm                                  SCALE VELOCITY TO mm
Motor[1].pLimits = CK3C[0].Chan[0].Status.a                         ENABLE LIMITS, =0 TO DISABLE

Motor[1].AbortTa = 100                                              ABORT DECELERATION, 100 MSEC
Motor[1].AbortTs = 0                                                ABORT S-CURVE, NONE

Motor[1].JogTa = 100                                                JOG ACCELERATION, 100 MSEC
Motor[1].JogTs = 0                                                  JOG S-CURVE, NONE
Motor[1].JogSpeed = 0.050                                           JOG SPEED, 50 mm/sec

Motor[1].FatalFeLimit = 0.500                                       FE LIMIT, 500 µm
Motor[1].WarnFeLimit = 0                                            WARNING FE LIMIT, NONE

Motor[1].InPosBand = 0.000100                                       INPOS BAND, 100 nm
Motor[1].InPosTime = 10 / Sys.ServoPeriod                           INPOS TIME, 10 MSEC

Motor[1].pPhaseEnc = CK3C[0].Chan[0].PhaseCapt.a                    TYPICAL FOR SINUSOIDAL
Motor[1].PhaseCtrl = 4                                              TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 0                                      NONE FOR SINUSOIDAL
Motor[1].PhaseEncRightShift = 0                                     NONE FOR SINUSOIDAL
Motor[1].PhasePosSf = 2048 * Mtr1Res / (Mtr1Ecl * 1024)             TYPICAL FOR SINUSOIDAL

Motor[1].AmpFaultLevel = 1                                          =1 FOR CK3A
Motor[1].AdcMask = $FFFF0000                                        =$FFFF0000 FOR CK3A
Motor[1].PhaseOffset = 683                                          TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PwmSf = 0.95 * 16384                                       TYPICAL IF MTR VOLTAGE>INPUT

Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc   TYPICAL I2T FOR BRUSHLESS
Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc
Motor[1].I2tTrip =(POW(Motor[1].MaxDac,2)-
POW(Motor[1].I2TSet,2))*Mtr1TimeAtMaxCur

Motor[1].CurrentNullPeriod = 0                                      =0 FOR CK3A
Motor[1].IaBias = 0                                                 =0 DURING NORMAL OPERATION
Motor[1].IbBias = 0                                                 =0 DURING NORMAL OPERATION

Motor[1].IiGain = 0                                                 CURRENT LOOP GAINS
Motor[1].IpfGain = 0                                                (MUST TUNE / USER INPUT)
Motor[1].IpbGain = 0

Motor[1].Servo.Kp = 0                                               POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                             (MUST TUNE / USER INPUT)
Motor[1].Servo.Kvifb = 0
Motor[1].Servo.Kvff = 0
Motor[1].Servo.Kviff = 0
Motor[1].Servo.Ki = 0
Motor[1].Servo.Kaff = 0
Motor[1].Servo.Kfff = 0
Motor[1].Servo.SwZvInt = 0

Motor[1].pAbsPhasePos = CK3C[0].Chan[0].Status.a
Motor[1].AbsPhasePosFormat = $400030C                               HALLS ABS. PHASING
Motor[1].AbsPhasePosSf = 2048 / 12                                  MUST SET ± PER HALLS PROCEDURE
Motor[1].AbsPhasePosOffset = 0                                      MUST FIND PER HALLS PROCEDURE
Motor[1].PowerOnMode = 2                                            ABS HALLS PHASE ON POWER-UP
```



Additional Information

Please use the **Tune Tool** in the IDE to determine the values of the tuning parameters in the code above (i.e. **Motor[x].IiGain**, **Motor[x].Servo.Kp**) for the Current Loop and Position Servo Loop gains.

A-2-4 Direct PWM Output with BiSS Encoder (ACC-8FS, ACC-84S)

Hardware Configuration

Item	Notes
Controller	CK3C + ACC-8FS + ACC-84S(BiSS)
Amplifier	CK3A-G310L (three-phase 208 VAC main supply voltage)
Motor	Linear servo Motor, 200 VAC, 2.6/8.3 A, 60.96mm electrical cycle length
Encoder	BiSS, 32-bit, 1nm resolution

System Gates

```
Sys.WpKey = $AAAAAAAA                                                    DISABLE WRITE-PROTECTION
Sys.MaxCoords = 2                                                        MAX. COORD SYSTEMS+1 (EFFICIENCY)
Sys.MaxMotors = 5                                                        MAX. NO. OF MOTORS+1 (EFFICIENCY)

CK3C[0].PhaseFreq = 10000                                                10 KHZ PHASE FREQUENCY
CK3C[0].ServoClockDiv = 1                                                5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                      2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (CK3C[0].ServoClockDiv + 1) / CK3C[0].PhaseFreq TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (CK3C[0].ServoClockDiv + 1)               TYPICAL CALCULATION

CK3C[0].Chan[0].PwmFreqMult = 1                                          CH0-3 10 KHZ PWM FREQUENCY
CK3C[0].Chan[1].PwmFreqMult = 1
CK3C[0].Chan[2].PwmFreqMult = 1
CK3C[0].Chan[3].PwmFreqMult = 1

CK3C[0].Chan[0].OutputMode = 0                                           CH0-3 FILTERED PWM OUTPUT ON PHASE A-D
CK3C[0].Chan[1].OutputMode = 0
CK3C[0].Chan[2].OutputMode = 0
CK3C[0].Chan[3].OutputMode = 0

CK3C[0].Chan[0].PwmDeadTime = 3 / 0.0533                                 CH0-3 MIN. 3µSEC FOR G310
CK3C[0].Chan[1].PwmDeadTime = 3 / 0.0533
CK3C[0].Chan[2].PwmDeadTime = 3 / 0.0533
CK3C[0].Chan[3].PwmDeadTime = 3 / 0.0533

CK3C[0].Chan[0].PackInData = 0
CK3C[0].Chan[1].PackInData = 0                                           CH0-3 IN DATA
CK3C[0].Chan[2].PackInData = 0                                           MUST BE 0 FOR CK3A
CK3C[0].Chan[3].PackInData = 0

CK3C[0].Chan[0].PackOutData = 0
CK3C[0].Chan[1].PackOutData = 0                                          CH0-3 OUT DATA
CK3C[0].Chan[2].PackOutData = 0                                          MUST BE 0 FOR CK3A
CK3C[0].Chan[3].PackOutData = 0

CK3C[0].AdcAmpStrobe = $901001                                           CH0-3 STROBE WORD (FOR CK3A)
CK3C[0].AdcAmpClockDiv = 5                                               CH0-3 3.125 MHZ ADC CLOCK
CK3C[0].AdcAmpHeaderBits = 4                                             CH0-3 MUST BE 4 (FOR CK3A)

ACC84S[0].SerialEncCtrl = $31000B                                        SERIAL ENC. CTRL (BISS)
ACC84S[0].Chan[0].SerialEncCmd = $2114A0                                 SERIAL ENC. COMMAND (32 BITS)
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                                     TYPICAL SERIAL ACC-84S ENTRY
EncTable[1].pEnc = ACC84S[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 8
EncTable[1].index2 = 8
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = EXP2(EncTable[1].index1)
EncTable[1].MaxDelta = 0

EncTable[2].Type = 0                                                     END OF ECT (FOR EFFICIENCY)
```

```
GLOBAL Ck3a1MaxAdc = 31.35                                                  CK3A MAX ADC
GLOBAL Mtr1Res = 0.000001                                                   1nm RESOLUTION [mm]
GLOBAL Mtr1CtsPerMm = 1 / Mtr1Res                                           COUNTS PER MM
GLOBAL Mtr1Ecl = 60.96                                                      ELECTRICAL CYCLE LENGTH [mm]
GLOBAL Mtr1MaxRmsCur = 8.3                                                  MAX RMS CUR.(MTR<CK3A)
GLOBAL Mtr1RatedRmsCur = 2.6                                                RATED RMS CUR.(MTR<CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2                                                 TIME ALLOWED AT MAX CUR.

Motor[1].ServoCtrl = 1                                                      ACTIVATE CHANNEL
Motor[1].PosUnit = 3                                                        IDE WINDOW UNITS [mm]
Motor[1].PosSf = 1 / Mtr1CtsPerMm                                           SCALE POSITION TO mm
Motor[1].Pos2Sf = 1 / Mtr1CtsPerMm                                          SCALE VELOCITY TO mm
Motor[1].pLimits = CK3C[0].Chan[0].Status.a                                 ENABLE LIMITS, =0 TO DISABLE

Motor[1].AbortTa = 100                                                      ABORT DECELERATION, 100 MSEC
Motor[1].AbortTs = 0                                                        ABORT S-CURVE, NONE

Motor[1].JogTa = 100                                                        JOG ACCELERATION, 100 MSEC
Motor[1].JogTs = 0                                                          JOG S-CURVE, NONE
Motor[1].JogSpeed = 0.050                                                   JOG SPEED, 50 mm/sec

Motor[1].FatalFeLimit = 0.500                                               FE LIMIT, 500 µm
Motor[1].WarnFeLimit = 0                                                    WARNING FE LIMIT, NONE

Motor[1].InPosBand = 0.000100                                               INPOS BAND, 100 nm
Motor[1].InPosTime = 10 / Sys.ServoPeriod                                   INPOS TIME, 10 MSEC

Motor[1].pPhaseEnc = ACC84S[0].Chan[0].SerialEncDataA.a                     TYPICAL FOR SERIAL
Motor[1].PhaseCtrl = 4                                                      TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 8                                              ACC-84
Motor[1].PhaseEncRightShift = 8                                             ACC-84
Motor[1].PhasePosSf = 2048 * Mtr1Res / (EXP2(Motor[1].PhaseEncLeftShift) * Mtr1Ecl)  TYPICAL FOR SERIAL

Motor[1].AmpFaultLevel = 1                                                  =1 FOR CK3A
Motor[1].AdcMask = $FFFF0000                                                =$FFFF0000 FOR CK3A
Motor[1].PhaseOffset = 683                                                  TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PwmSf = 0.95 * 16384                                               TYPICAL IF MTR VOLTAGE>INPUT

Motor[1].MaxDac = Mtr1MaxRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc  TYPICAL I2T FOR BRUSHLESS
Motor[1].I2TSet = Mtr1RatedRmsCur * 32768 * COSD(30) * SQRT(2) / Ck3a1MaxAdc  ...
Motor[1].I2tTrip =(POW(Motor[1].MaxDac,2)-POW(Motor[1].I2TSet,2))*Mtr1TimeAtMaxCur  ...

Motor[1].CurrentNullPeriod = 0                                              =0 FOR CK3A
Motor[1].IaBias = 0                                                         =0 DURING NORMAL OPERATION
Motor[1].IbBias = 0                                                         =0 DURING NORMAL OPERATION

Motor[1].IiGain = 0                                                         CURRENT LOOP GAINS
Motor[1].IpfGain = 0                                                        (MUST TUNE / USER INPUT)
Motor[1].IpbGain = 0                                                        ...

Motor[1].Servo.Kp = 0                                                       POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                                     (MUST TUNE / USER INPUT)
Motor[1].Servo.Kvifb = 0                                                    ...
Motor[1].Servo.Kvff = 0                                                     ...
Motor[1].Servo.Kviff = 0                                                    ...
Motor[1].Servo.Ki = 0                                                       ...
Motor[1].Servo.Kaff = 0                                                     ...
Motor[1].Servo.Kfff = 0                                                     ...
Motor[1].Servo.SwZvInt = 0                                                  ...

Motor[1].pAbsPos = ACC84S[0].Chan[0].SerialEncDataA.a                       TYPICAL FOR SERIAL ABS. ENC.
Motor[1].AbsPosFormat = $00082008                                           UNSIGNED 32 BITS
Motor[1].AbsPosSF = Motor[1].PosSf                                          SAME AS POSITION SCALE
Motor[1].HomeOffset = 0                                                     USER CONFIGURABLE POS. OFFSET

Motor[1].pAbsPhasePos = ACC84S[0].Chan[0].SerialEncDataA.a                  TYPICAL FOR ABS. SERIAL ENC.
Motor[1].AbsPhasePosFormat = $00082008                                      USE 32 BITS FOR ABS. PHASE
Motor[1].AbsPhasePosSf = 2048 * Mtr1Res / Mtr1Ecl                           TYPICAL FOR ABS. SERIAL ENC.
//Motor[1].AbsPhasePosForce =                                               READ 32 BITS, TEST VALUE
//ACC84S[0].Chan[0].SerialEncDataA+(ACC84S[0].Chan[0].SerialEncDataB&$FF)*16777216
Motor[1].AbsPhasePosForce = 253109861                                       STORE TEST VALUE
Motor[1].AbsPhasePosOffset =-Motor[1].AbsPhasePosForce * Motor[1].AbsPhasePosSf  TYPICAL SETTING
Motor[1].PowerOnMode = $6                                                   ABS. PHASE & POS ON POWER-UP
```

Additional Information

Please use the **Tune Tool** in the IDE to determine the values of the tuning parameters in the code above (i.e. **Motor[x].IiGain**, **Motor[x].Servo.Kp**) for the Current Loop and Position Servo Loop gains.

OMRON Corporation **Industrial Automation Company**

Kyoto, JAPAN **Contact : www.ia.omron.com**

Regional Headquarters

OMRON EUROPE B.V.
Wegalaan 67-69, 2132 JD Hoofddorp
The Netherlands
Tel: (31) 2356-81-300 Fax: (31) 2356-81-388

OMRON ELECTRONICS LLC
2895 Greenspoint Parkway, Suite 200
Hoffman Estates, IL 60169 U.S.A.
Tel: (1) 847-843-7900 Fax: (1) 847-843-7787

OMRON ASIA PACIFIC PTE. LTD.
438B Alexandra Road, #08-01/02 Alexandra
Technopark, Singapore 119968
Tel: (65) 6835-3011 Fax: (65) 6835-3011

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-6023-0333 Fax: (86) 21-5037-2388

Authorized Distributor:

©OMRON Corporation 2025 All Rights Reserved.
In the interest of product improvement,
specifications are subject to change without notice.

Cat. No. O054-E1-03 0825 (0325)