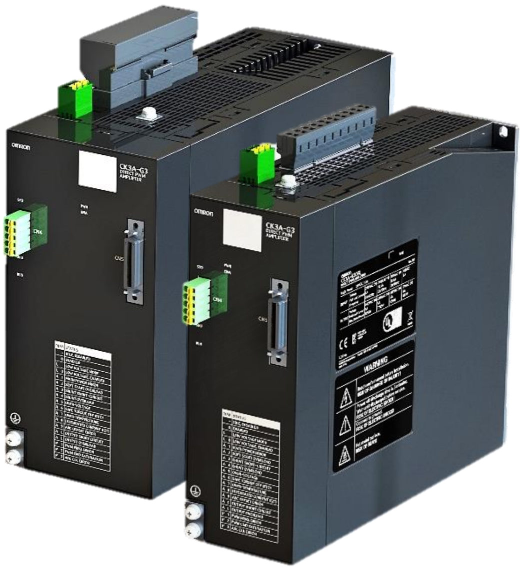
OMRON

CK3A-series

Direct PWM Amplifier

User's Manual

CK3A-G305L
CK3A-G310L



O050-E1-01

Introduction

Thank you for purchasing the CK3A Direct PWM Amplifier.

This manual contains the necessary information for proper mounting, installation, wiring, configuration, and troubleshooting of the CK3A Direct PWM Amplifier.

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 CK3A Amplifier.

Read this Manual in its entirety to gain full understanding of the proper method to use the CK3A Amplifier. Keep this manual in a convenient place for quick reference, and make sure that it is provided to the end-user.

Applicable Products

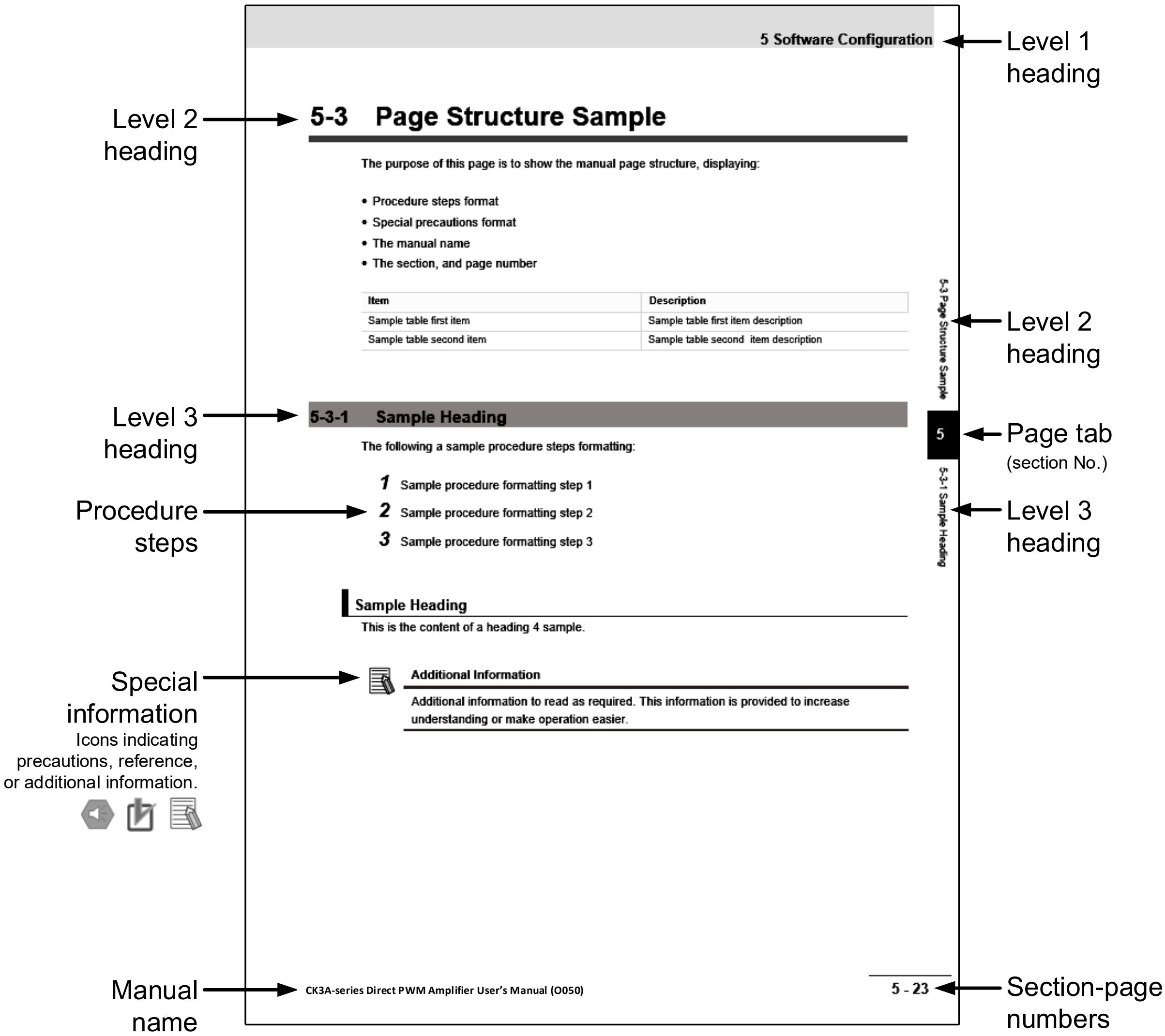
The applicable part numbers for this manual are:

- CK3A-G305L
- CK3A-G310L

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

CONTENTS

Section 1 Introduction to the Amplifier

Section 2 Specifications

Section 3 Mounting and Installation

Section 4 Wiring

Section 5 Software Configuration

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.

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 CK3A Direct PWM Amplifier, 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

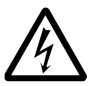
The safety precautions described in this manual are extremely important for safety. Always read and heed the information provided in all safety precautions.

The following notation is used to provide precautions required to ensure safe usage of the CK3A Direct PWM Amplifier:

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.

Transporting and Unpacking

⚠ WARNING

When transporting the Amplifier, do not carry it by the cables.
Doing so may result in injury, unit damage, or malfunction.

Do not step on the Amplifiers or place heavy articles on it.
Doing so may result in injury, unit damage, or malfunction.

Installation, Wiring and Maintenance

⚠ WARNING

Connect this Amplifier to power supplies, Motor, and other equipment correctly according to the instructions in this manual. Not doing so may result in serious injury.

Use an appropriate External Shunt Resistor. Install an external protective device such as temperature sensor to ensure safety when using the External Shunt Resistor.
Not doing so run the risk of burnout.

Install the product and peripheral equipment on non-flammable materials such as metals.
Not doing so may cause a fire.

Follow the instructions in this manual to correctly perform terminal block and connector wiring and insertion. Double-check all wiring and connector insertion before turning ON the power supply. Not doing so may result in electric shock, fire, equipment damage, malfunction, or injury.

Do not damage, pull, bend strongly, or put excessive stress or heavy objects on the cables. Doing so may cause malfunction or burning.

Do not place flammable material near the Amplifier or peripheral equipment. Doing so may cause a fire.	
Do not apply AC power to the Amplifier if the low voltage short-circuit wire (CN1) is installed. Doing so may cause damage, malfunction, or injury.	
Make sure that the Amplifier (and Motor) are tied to ground using 100Ω or less connection. Not doing so may cause electric shock.	
Do not remove any of the Amplifier covers. Doing so may cause electric shock.	
Use appropriate tools to wire terminals and connectors. Make sure that there is no short-circuit before use. Not doing so may cause electric shock.	
Connect the frame ground wire of the Motor cable securely to the ⏚ or FG of the Amplifier. Not doing so may cause electric shock.	
Provide safety measures, such as a fuse, molded case circuit breaker, or Earth Leakage Circuit Breaker to protect against short circuiting of external wiring and failure of the product. Not doing so may cause a fire.	

Software Configuration

⚠ WARNING

Make sure that the Motor-specific protection parameters in the Controller, especially I2T thermal protection including Motor[x].I2tSet, Motor[x].I2tTrip, and Motor[x].MaxDac are set correctly. Not doing so may result in Motor burning, fire, or serious injury.	

While Main Power Is applied

⚠ WARNING

Before carrying out wiring or inspection, turn OFF the main circuit power and wait for at least 5 minutes. Not doing so may cause electric shock or burning.	

Fail-safe Measures

⚠ WARNING

Provide safety measures, such as cutting OFF main circuit power, in external circuits to ensure safety in the system if an Amplifier error or abnormality occurs due to malfunction of the products or due to other external factors affecting operation. Not doing so may result in serious accidents.

Emergency stop, interlock, limit circuits and similar safety measures must be provided in external control circuits. Not doing so may result in serious accidents.

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. Not doing so may result in serious accidents.

For vertical axes, use brake or counter-balance mechanism to prevent them from falling down when the servo control is stopped. Not doing so may result in serious injury.

Test Run

⚠ WARNING

Program the Amplifier properly according to the instructions in this and related manual. Not doing so may result in equipment damage.

Always confirm safety when replacing the Power PMAC Controller, or changing the configuration (from the IDE software). Devices or machines may perform unexpected operation regardless of the operating mode of the product.
Not doing so may result in equipment damage, malfunction, or injury.

During Operation

WARNING

Do not enter the machine operating area during operation. Doing so may cause injury.	⚠
Do not touch the Amplifier heatsink, Shunt Resistor, or Motor while the power is supplied or for at least 5 minutes after the power had been turned OFF because they get hot. Doing so may cause fire or a burn injury.	⚠
Take appropriate measures to ensure that the specified power with the rated voltage is supplied. Be particularly careful in locations where the power supply is unstable. Not doing so may cause failure.	⚠
When the power is restored after a momentary power interruption, the machine may restart suddenly. Do not come close to the machine when restoring power. Doing so may cause injury. Implement measures to ensure safety of people nearby even when the machine is restarted.	⚠
Do not plug/unplug any connection to the Amplifier while main circuit power is ON. Doing so may cause failure, equipment damage, or serious injury.	⚠
The Amplifier does not error if the Controller is sending Motor commands while the Motor connector is unplugged. Do NOT plug the Motor connector until you make sure that the Controller is not sending Motor commands or main circuit power is OFF. Doing so may cause failure, equipment damage, or serious injury.	⚠
Do not enable the Motor during Soft Start. Wait a minimum of 5 seconds before enabling the Motor after turning main AC power ON. Doing so may cause failure, equipment damage, or serious injury.	⚠

Precautions for Safe Use

Transporting

- Do not drop any unit or subject it to abnormal vibration or shock.

Doing so may result in unit malfunction, burning, or electric shock.

Mounting

- Be sure to observe the mounting direction and clearance with the surroundings.
- Check that terminal blocks, connectors, and other locking devices are properly latched before use.

Installation

- Do not block the intake or exhaust openings. Malfunction, operation stop, or equipment damage may result.
- Do not allow foreign objects to enter the Amplifier.
- Do not operate or store in the following locations:
  a) Locations subject to direct sunlight
  b) Locations subject to temperatures outside of the range specified in the specifications
  c) Locations subject to humidity outside of the range specified in the specifications
  d) Locations subject to condensation (e.g. severe changes in temperature)
  e) Locations subject to corrosive or flammable gases
  f) Locations subject to dust (especially iron dust) or salts
  g) Locations subject to exposure to water, oil, or chemicals
  h) Locations subject to shock or vibration

  Doing so may result in electric shock, fire, equipment damage, or malfunction.

Wiring

- Follow the instructions in this manual to correctly perform terminal block and connector wiring and insertion. Double-check all wiring and connector insertion before turning ON the power supply.
- Do not pull on the cables or bend them beyond their natural limit.
- Do not place heavy objects on top of the cables or other lines. Doing so may break the cables.
- Do not allow wire clippings, shavings, or other foreign material to enter the Amplifier. Otherwise, Amplifier burning, failure, or malfunctions may occur. Cover the Amplifier or take other suitable countermeasures, in particular when carrying out wiring work.
- Use a cable track or appropriate mechanism to separate moving and fixed parts of the system.
- Connect the Amplifier to the Motor without a contactor or other electrical relay device.

Power Supply Design

- Power supply must be within the rated capacity range specified in this manual.
- Install external breakers and take other safety measures against short-circuit and over-current.
- Do not apply voltages in excess of the range specified in this manual.

Motor Selection

- Check the specifications carefully before selecting a Motor to be connected to the Amplifier.

Software Configuration

- Make sure that the Amplifier-specific parameters in the Controller, described in the Software Configuration section of this manual, are set correctly.
- Make sure that the Motor-specific protection parameters in the controller are set correctly. In particular, I2T thermal protection settings.

Turning Power ON

- Be sure that the terminal blocks, connectors, and other items with locking or screwing devices are correctly locked into place before turning the power ON.
- Following, is the correct procedure for turning the Amplifier ON:
  1. Turn ON 24VDC logic power first (always before main circuit power)
  2. Wait a minimum of 10 seconds for the Amplifier to boot up
  3. Turn ON main circuit power (e.g. 200 VAC)
  4. Wait a minimum of 5 seconds
- When the Amplifier is booting up, do NOT turn ON main power or issue any command to the Amplifier from the controller.
- Before turning main circuit power ON, make sure that the controller is not sending any Motor command to the Amplifier.
- After turning main circuit power ON, allow a minimum of 5 seconds before sending any Motor commands from the controller to the Amplifier.

Not doing all of the above can cause serious injury, equipment damage, or malfunction.

While Power is ON

- Do not touch the Amplifier heatsink, Shunt Resistor, or Motor while the power is supplied or for at least 5 minutes after the power had been turned OFF because they get hot. Doing so may cause fire or a burn injury.
- If an error occurs, remove the cause of the error and ensure safety before performing reset and restarting system. Not doing so may cause damage, malfunction, or injury.

Turning Power OFF

- Following, is the correct procedure for turning the Amplifier OFF:

  1 Turn OFF main circuit power (e.g. 200VAC) – always before 24VDC logic power

  2 If the Amplifier is programmed (set up by the strobe word in the Controller) to discharge capacitor energy, wait a minimum of 3 seconds.

  3 Turn OFF 24VDC logic power.

- Always turn OFF power before attempting to perform any of the following:

  a) Mounting or removing the units.
  b) Connecting cables or wiring the system.
  c) Connecting or disconnecting the terminal blocks or connectors.

Not doing all of the above can cause serious injury, equipment damage, or malfunction.

Maintenance

- Do not use corrosive chemicals to clean the Amplifier. Doing so may result in failure or malfunction of the Amplifier.
- Dispose of the product according to local ordinances as they apply.

Precautions for Correct Use

General Precaution

- Take appropriate and sufficient countermeasures when installing the Amplifier in the following locations.
  a) Locations subject to strong, high-frequency noise
  b) Locations subject to static electricity or other forms of noise
  c) Locations subject to strong electromagnetic fields
  d) Locations subject to possible exposure to radioactivity
  e) Locations close to power lines

Installation and Mounting

- Follow the instructions in this manual for proper installation of the Amplifier.
- Use the specified tightening torque for mounting the Amplifier.
- Tighten the mounting screws for the product and peripheral equipment to the specified torque.
- Take appropriate and sufficient countermeasures when installing the Amplifier in the following locations.
  a) Locations subject to strong, high-frequency noise
  b) Locations subject to static electricity or other forms of noise
  c) Locations subject to strong electromagnetic fields
  d) Locations subject to possible exposure to radioactivity
  e) Locations close to power lines
- Install the Amplifier away from sources of heat and ensure proper ventilation.
- After installing or replacing the product, check the operation sufficiently before moving to the actual operation.

Not doing all of the above may result in injury, equipment damage, malfunction or fire.

Wiring

- Wire the cables correctly and securely.
- Carefully perform the wiring and assembling.
- Use crimp terminals to wire screw type terminal blocks. Do not connect bare stranded wires directly to terminals blocks.
- Always use the power supply voltage specified in the manual.
- When constructing a system that includes safety functions, be sure you understand the relevant safety standards and all related information in user documentation, and design the system to comply with the standards.
- Disconnect all connections to the Amplifier before attempting a megger test (insulation resistance measurement) on the Amplifier. Do not perform a dielectric strength test on the Amplifier.
- Keep conductive or flammable foreign objects such as screws, metal pieces, and oil out of the Amplifier and connectors. Pay particular attention to the connectors on the top part of the Amplifier.
- Wear protective equipment when installing or removing connectors, especially pay attention to the main circuit connector. Do not apply a force after the protrusion of the connector opener reaches the bottom dead center. (As a guide, do not apply a force of 100 N or more).
- Do not apply excessive force to wire terminals and connectors.

Not doing all of the above may result in injury, equipment damage, malfunction or fire.

Power ON/OFF

- Do not cycle ON/OFF main circuit power if logic power is OFF.
- After turning logic power ON, wait 10 seconds before turning ON main circuit power.

Adjustment

- If a problem occurs in communications to the controller or the computer during a test operation, you have no means to stop the Motor. Connect an externally installed emergency stop switch, etc. to the Error Stop Input of the general-purpose input so that the Motor can be stopped without fail.
- Do not try to adjust parameters to extreme values in the controller. This may make the operation unstable.

Not doing all of the above may result in injury, equipment damage, malfunction or fire.

Operation

- Make sure the connectors are firmly plugged in, and the terminal block are tightly screwed in.
- Install a proper stopping device on the machine to ensure safety. The dynamic brake mechanism is not a stopping device.
- Install an immediate stop device externally to the machine so that the operation can be stopped and the power supply is cut off immediately.
- After an earthquake, lightning, flooding, or power surge be sure to conduct safety checks.
- Do not place flammable material near the Amplifier.
- Do not use the cable when it is laying in oil or water.
- If the Amplifier fails, cut off the power supply to the Amplifier at the power source.
- Turn OFF the power supply when not using the Amplifier for a prolonged period of time.
- Immediately stop the operation and turn OFF the power supply when unusual smell, noise, smoking, abnormal heat generation, or vibration occurs.
- Use the appropriate External Shunt Resistor. Install an external protective device such as temperature sensor to ensure safety when using the External Shunt Resistor.
- Not doing all of the above may result in injury, equipment damage, malfunction or fire.

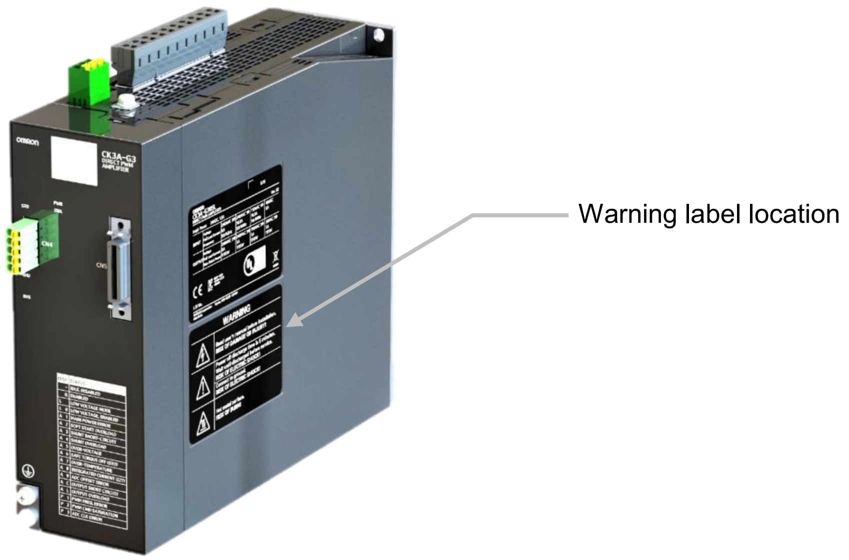
Maintenance and Troubleshooting

- Always turn OFF power before attempting to perform any of the following:
  a) Mounting or removing the units.
  b) Connecting cables or wiring the system.
  c) Connecting or disconnecting the terminal blocks or connectors.
- Not doing all of the above can cause serious injury, equipment damage, or malfunction.

- Do not attempt to disassemble, repair, or modify the Amplifier. Doing so may result in damage, malfunction, or injury.

Location of Warning Label

The Amplifier bears a warning label at the following location. Be sure to observe the instructions provided on this label.



- **Instructions on Warning Display**

Note the above is an example of warning display.

Location of Display Code

The Amplifier bears a display code label at the following location. This is intended for quick reference. Refer to the troubleshooting section of this manual for display code details.

Display code label location

● Information on Display code

DISP.	STATUS
–	IDLE, DISABLED
0.	ENABLED
L	LOW VOLTAGE MODE
L 0	LOW VOLTAGE, ENABLED
A 1	MAIN POWER ERROR
A 2	SOFT START OVERLOAD
A 3	SHUNT SHORT–CIRCUIT
A 4	SHUNT OVERLOAD
A 5	OVER–VOLTAGE
A 6	SAFE TORQUE OFF (STO)
A 7	OVER–TEMPERATURE
A 8	INTEGRATED CURRENT (I2T)
A 9	ADC OFFSET ERROR
A C	OUTPUT SHORT CIRCUIT
A L	OUTPUT OVERLOAD
P 1	PWM FREQ. ERROR
P 2	PWM CMD SATURATION
P 3	ADC CLK ERROR

Disposal

Dispose of in accordance with WEEE Directive.

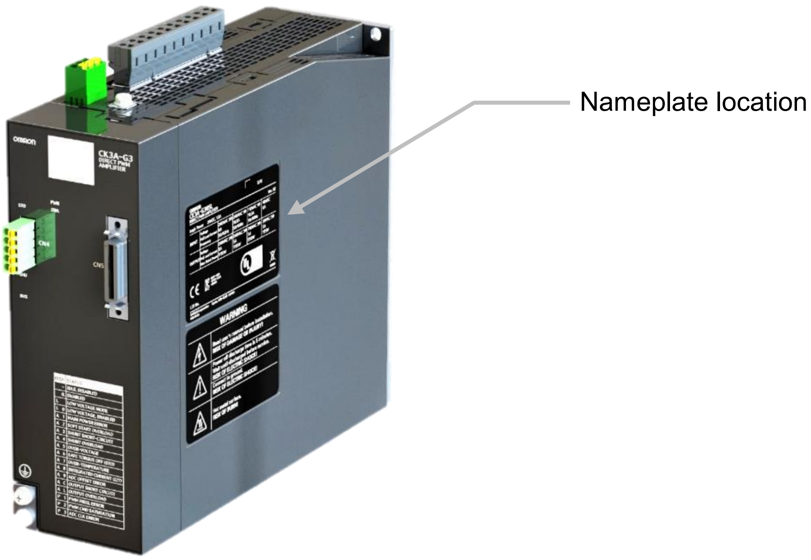
Items to Check after Unpacking

After you unpack the product, check the following items:

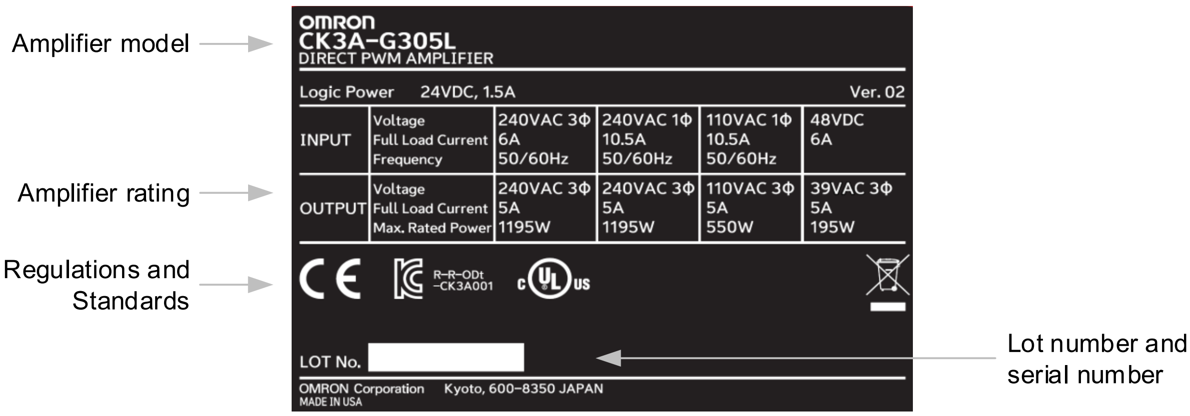
- Is this the model you ordered?
- Was there any damage sustained during shipment?

Location of Nameplate

The Amplifier bears a nameplate label at the following location.



Information on Nameplate



Note nameplate example for CK3A-G305L model.

Accessories

The Amplifier 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 Amplifier.

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	Main Power Connector (CN1)	Control Power Connector (CN2)	Motor Connector (CN3)	STO Connector (CN4)	PWM Connector/Cable (CN5)
CK3A-G305L	Included	Included	Included	Included	NOT Included
CK3A-G310L					

If any item is missing or a problem is found such as Amplifier 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 CK3A Direct PWM Amplifier.

Conformance to EU Directives

Item	Standard
EMC Directive	EN61800-3 second environment
Low Voltage Directive	EN61800-5-1 C2 category
Functional Safety	EN61800-5-2 SIL3 (STO)

Conformance to UL Directives

Item	Standard
UL Standards	UL 61800-5-1
CSA Standards	CSA C22.2 No. 274

Conformance to KC Standards

When you use the Amplifier 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.

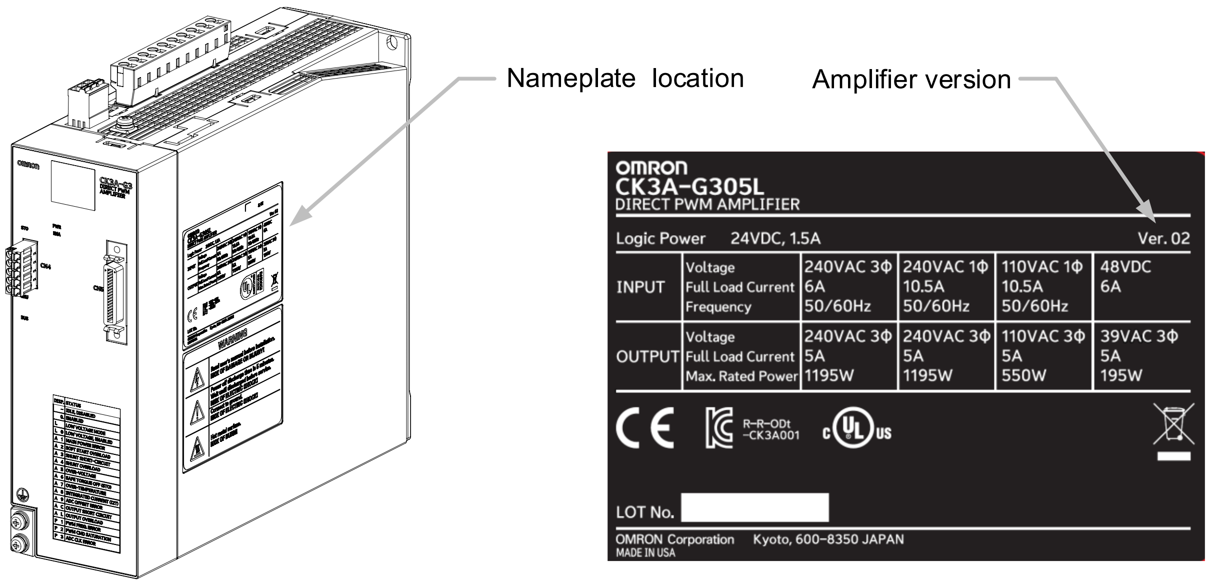
This product meets the electromagnetic compatibility requirements for business use. There is a risk of radio interference when this product is used in home.

Versions

The CK3A Direct PWM Amplifier uses unit version. Unit versions are useful to manage differences in product modifications, upgrades etc...

Hardware Version

The hardware unit version of the CK3A Direct PWM Amplifier is displayed on the nameplate.



Firmware Version

The amplifier firmware version can be retrieved per the instructions in the Software Configuration section of this manual.

Related Manuals

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

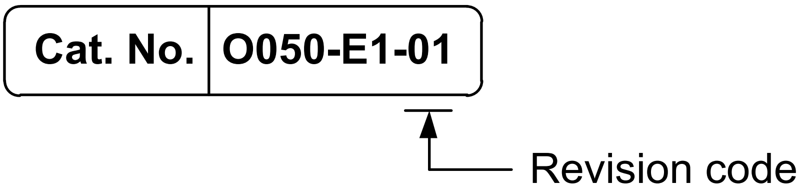
Manual Name	Cat. No.	Application	Description
CK3A-series Direct PWM Amplifier User's Manual (this manual)	O050	Learning about the specifications, including installation, wiring, basic software configuration, maintenance, and troubleshooting.	• Introduction to the Amplifier • Configuration, features, and specifications • Mounting, installation and wiring • Basic software configuration • Maintenance, and troubleshooting
CK3M-series Programmable Multi-Axis Controller Hardware User's Manual	O036	Learning the basic specifications of the CK3M, including introductory information, design, installation, and maintenance. Mainly hardware information.	• Features and system configuration • Introduction • Part names and functions • General specifications • Installation and wiring • Maintenance and inspection
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 CK3M and UMAC
Power PMAC User's Manual	O014	Learning the features and usage examples of the Power PMAC Controller.	• Parameter settings relevant to the Amplifier • 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 Amplifier using system setup
ACC-24E3 Hardware Reference Manual	N/A	Learning the basic specifications of the UMAC accessory ACC-24E3, 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 generation of the CPU at the time of creation of this manual
CK3M	CK3M-series Programmable Multi-Axis Controller (Power PMAC CPU inside)
UMAC	• Acronym for Universal Machine and Automation Controller • Typically, rack containing the Power PMAC CPU and other accessories • Originally designed by Delta Tau Data Systems, Inc.
Gate3	Delta Tau ASIC in CK3M/UMAC axis servo card which can have Direct PWM option
Controller	In this manual, used as a short for the Power PMAC Controller w/ Gate3 ASIC
Amplifier	In this manual, used as a short for the CK3A Direct PWM Amplifier
IDE	• Acronym for Integrated Development Environment • The primary software that is used to program the Power PMAC Controller
F.L.A.	Acronym for Full Load Amperage
IPM	• Acronym for Intelligent Power Module • In this manual, refers to the power electronic circuit of the Amplifier
FPGA	Field Programmable Gate Arrays
ADC	• Analog to Digital Converter • In this manual, refers to the ADC current sensors in the Amplifier
ADC Strobe Word	Refers to Power PMAC structure element AdcAmpStrobe
PWM	Acronym for Pulse Width Modulation
Direct PWM	In this manual, refers to the Controller-CK3A interface
Main circuit power	Refers to the AC main power supply to the Amplifier
Logic power	Refers to the 24VDC control power supply to the Amplifier
Bus voltage	Refers to the DC bus link inside the Amplifier
PLC	• Acronym for Programmable Logic Controller • In this manual, referring to a Power PMAC PLC program
I2T	Thermal protection model, can refer to Amplifier built-in I2T or PMAC configurable I2T
ECT	Acronym for Encoder Conversion Table

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	April 2022	Original production

1

Introduction to the Amplifier

This section describes the features, overall configuration, name of each part, and operating procedure of the CK3A Direct PWM Amplifier.

1-1 Outline

This section describes the features, and basic configuration of the CK3A Direct PWM Amplifier.

1-1-1 Amplifier Features

About

The CK3A Direct PWM (hereinafter referred to occasionally as Amplifier) is a line of Amplifier products designed to interface with Digital Direct PWM servo cards, typically integrated with OMRON and Delta Tau Data systems, Inc. (hereinafter referred to as Delta Tau) series of Turbo (legacy) and Power PMAC controllers.

Ultra-Low Latency Servo Control

This Amplifier, using Digital Direct PWM, takes the actual power-transistor ON/OFF signals directly from the Controller via ASIC (DSPGATE) machine interface while providing digital phase-current feedback and status for closed-loop servo control.

This architecture offers one of the fastest Controller-Amplifier interface for industrial applications, allowing ultra-fast servo update rates.

Flexibility

The CK3A Amplifier, combined with the flexibility of Power PMAC Controller and the ability to interface with various encoder feedback devices, allows the support of the following types of Motors:

- AC/DC synchronous brushless (rotary or linear) – e.g. servo Motor.
- DC Brushed – e.g. voicecoil actuator.
- AC asynchronous – e.g. Induction Motor (contact support for setting up this type of Motor).

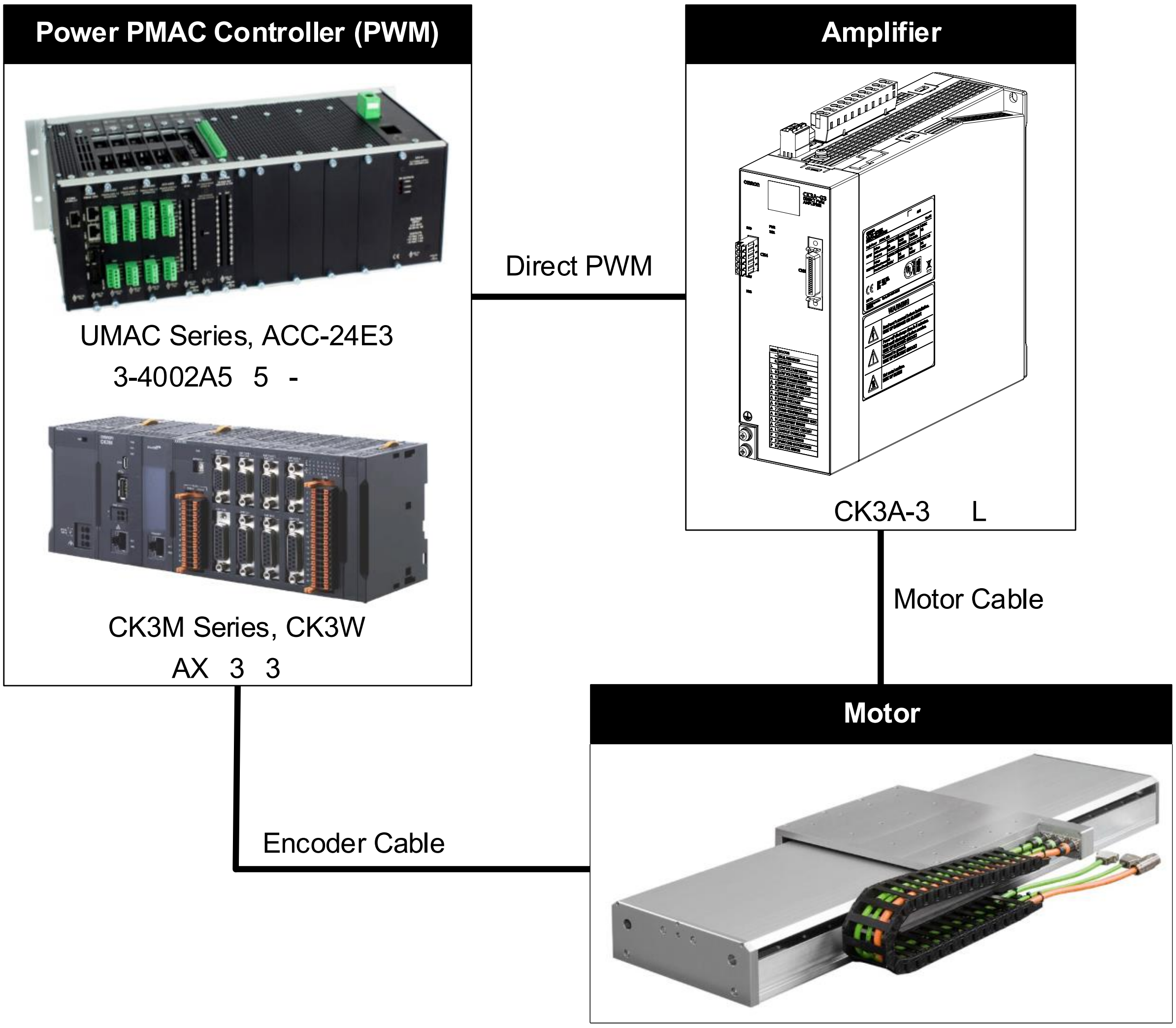
Amplifier Features

- Nano-scale linear servo positioning accuracy
- High PWM frequency – up to 20 KHz
- High resolution current sensing – 16-bit ADCs
- High speed current ADC sampling – up to 6.125MHz
- Dual STO inputs
- STO status output
- Basic functions – Energy discharge, dynamic braking, fan control
- Basic Data reporting – DC bus voltage, power module temperature, firmware version
- Dual 7-segment LED status display
- Built-in or external shunt resistor
- Support of low voltage main power operation

1-1-2 Typical Configuration

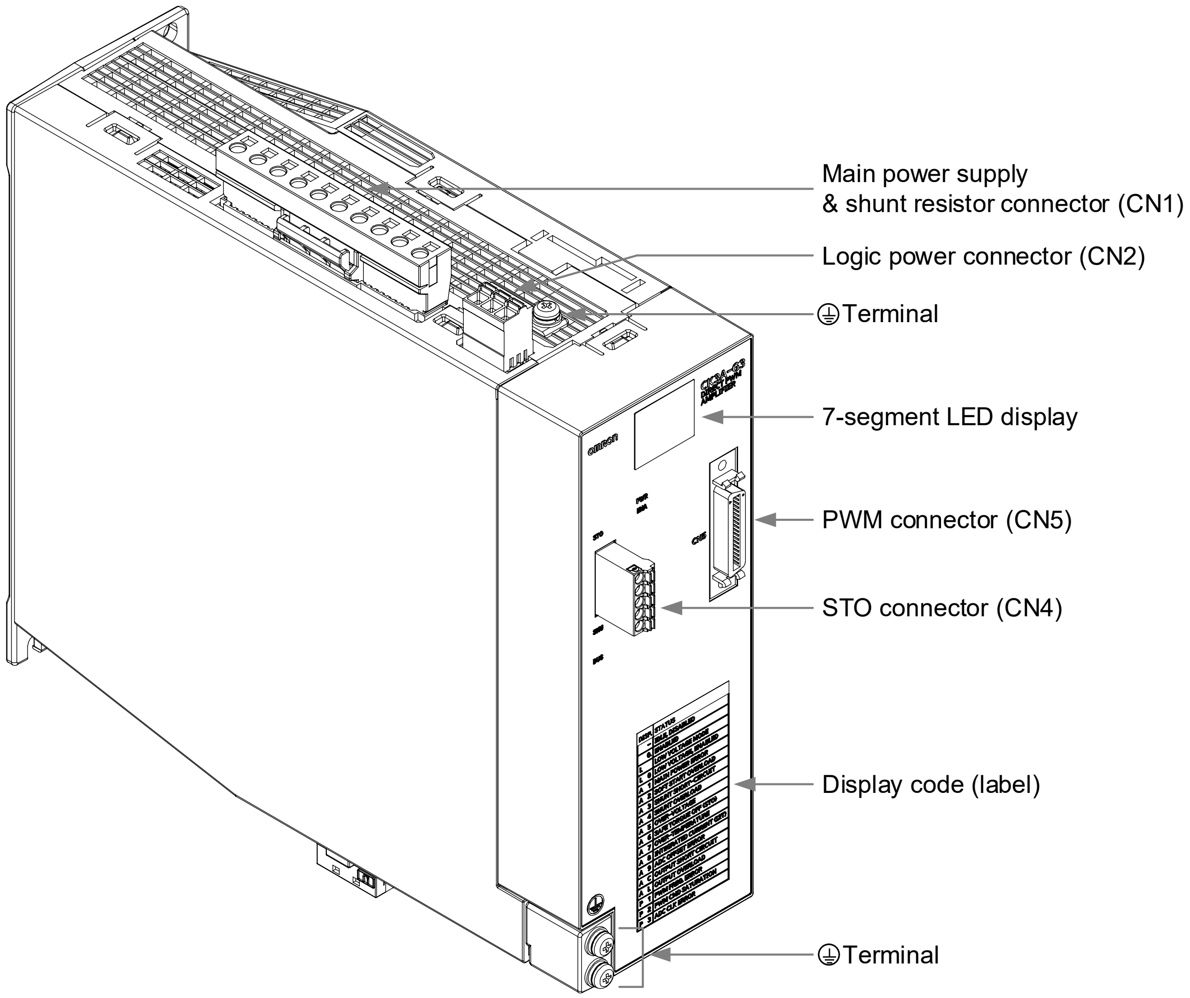
The CK3A Direct PWM Amplifier connects to the CK3M or UMAC Controller via PWM cable.

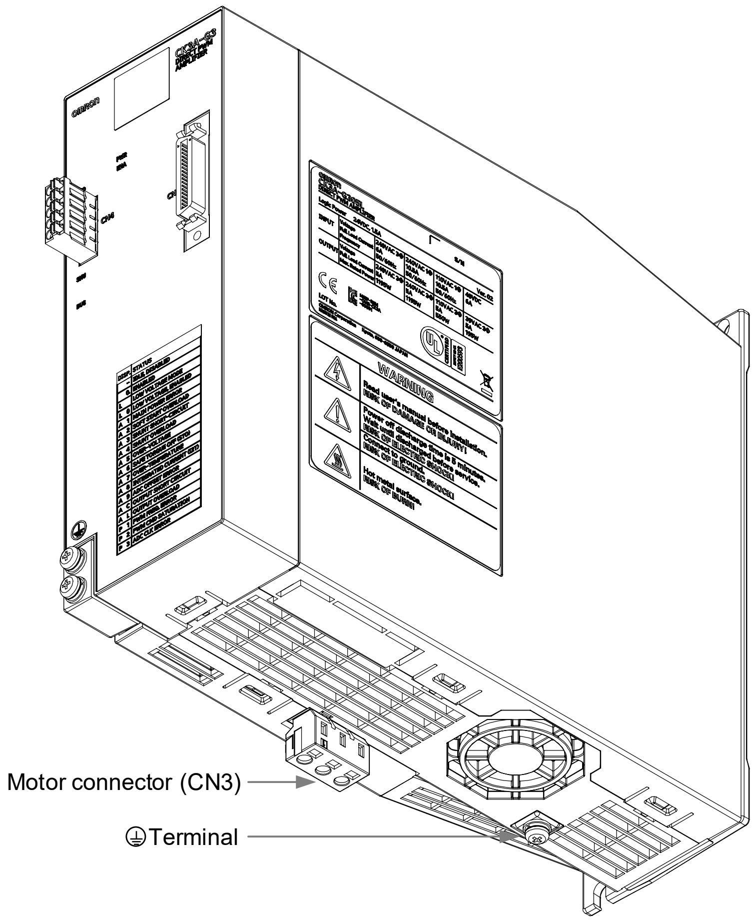
- The Motor connects directly to the CK3A Direct PWM Amplifier.
- The encoder connects directly to the Controller.

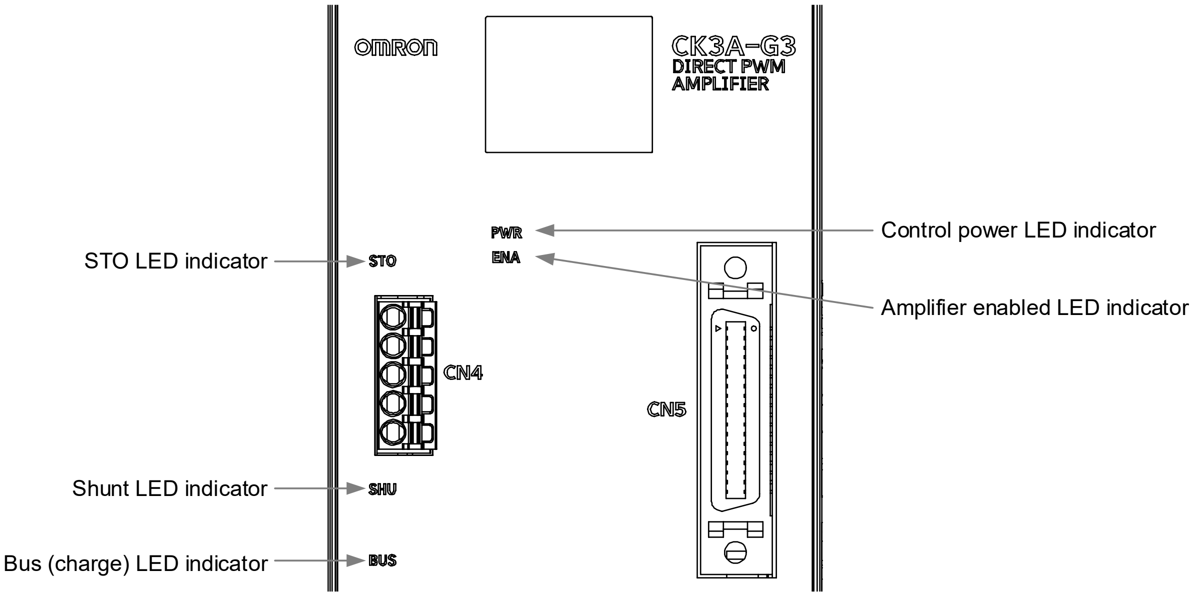


1-1-3 Part Names and Locations

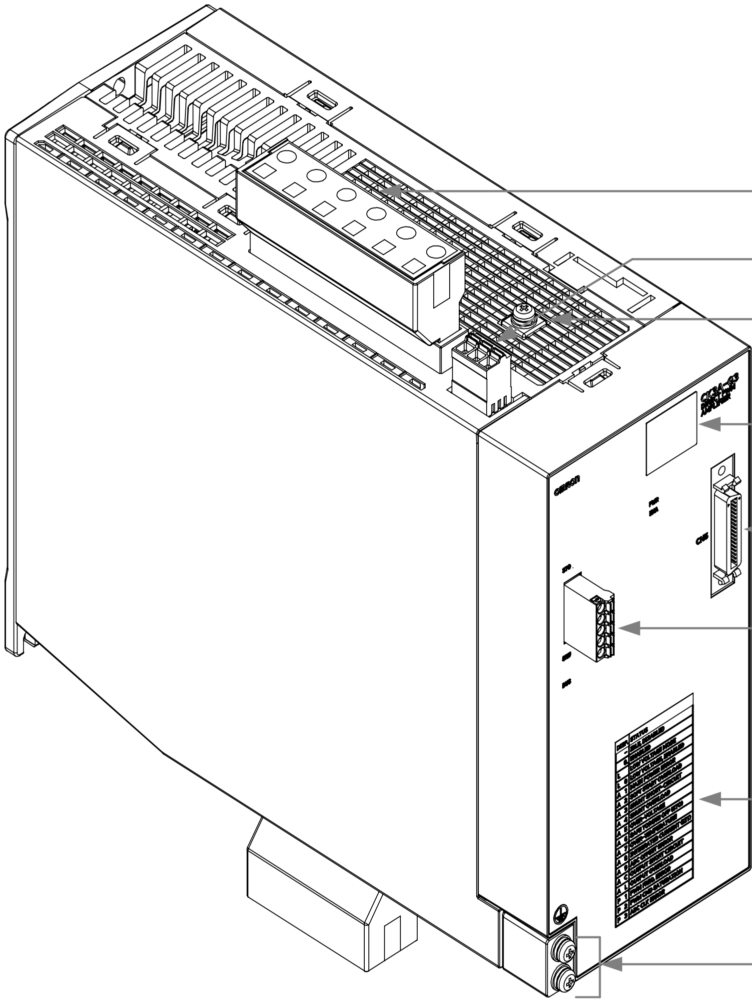
CK3A-G305L



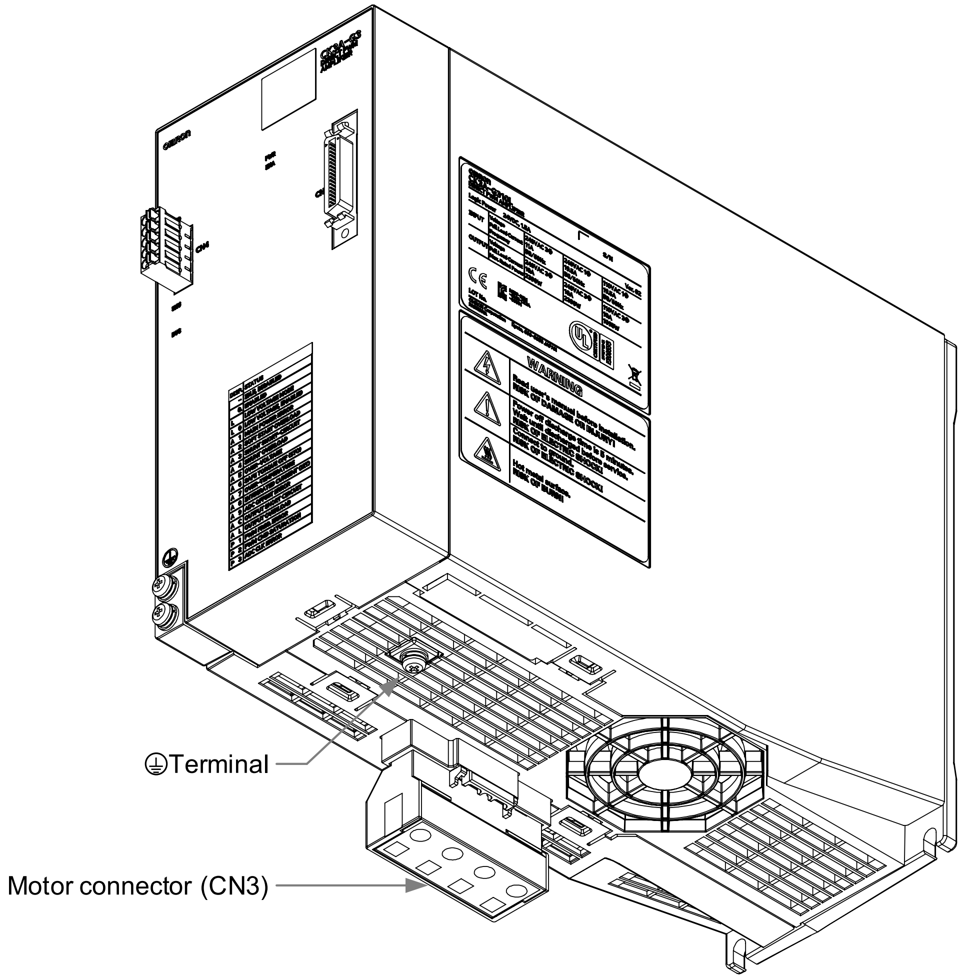

Motor connector (CN3)
⏚Terminal

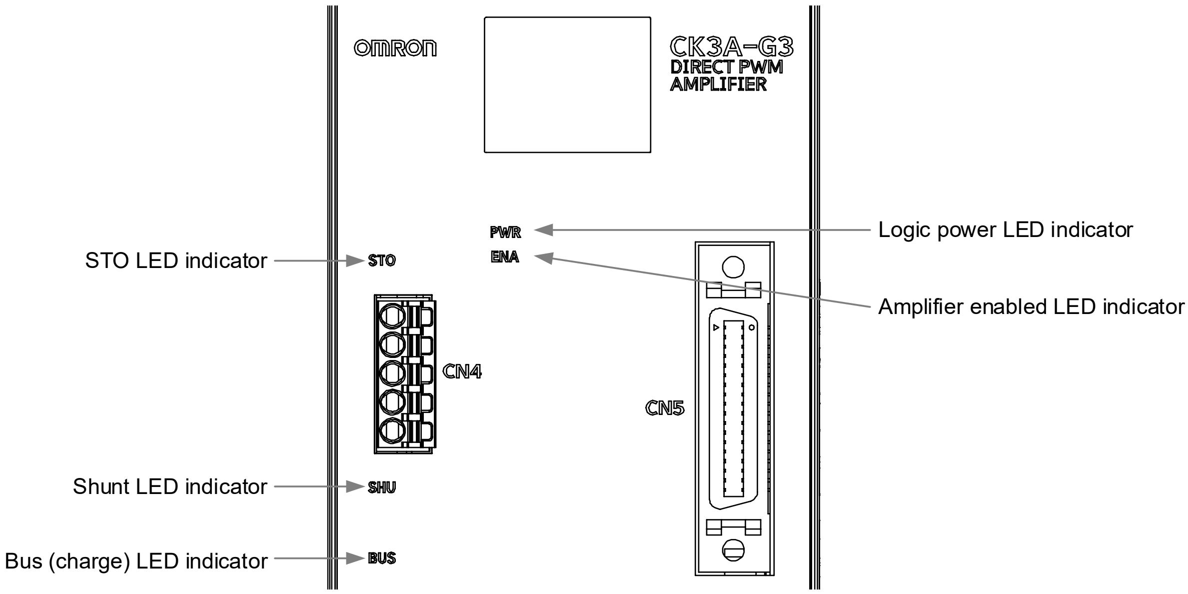

OMRON
CK3A-G3
DIRECT PWM
AMPLIFIER
PWR
ENA
Control power LED indicator
Amplifier enabled LED indicator
STO LED indicator
STO
CN4
CN5
Shunt LED indicator
SHU
Bus (charge) LED indicator
BUS

CK3A-G310L



Main power supply
& shunt resistor connector (CN1)
Logic power connector (CN2)
⏚Terminal
7-segment LED display
PWM connector (CN5)
STO connector (CN4)
Display code (label)
⏚Terminal


Terminal
Motor connector (CN3)


omron
CK3A-G3
DIRECT PWM
AMPLIFIER
PWR
ENA
Logic power LED indicator
Amplifier enabled LED indicator
STO LED indicator
STO
CN4
CN5
Shunt LED indicator
SHU
Bus (charge) LED indicator
BUS

1-1-4 Part Functions

Status LED Indicators

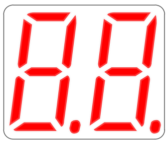
The following LED indicators are located on the front of the Amplifier:

Name	Color	Description
PWR	Green	Indicates the logic power supply status
ENA	Green	Indicates the amplifier enabled status
SHU	Red	Indicates if shunt operation is active
STO	Red/Green	Indicates STO input status
BUS	Red	Indicates bus charge status

For details, refer to the troubleshooting section.

7-segment LED Displays

Dual 7-segment displays are located on the front of the Amplifier. These displays report Amplifier operation and error status. On power-up, the 7-segment displays scroll through all indicators six times. More details about error codes can be found in the troubleshooting section.



Additional Information

When the Amplifier is OFF, it may be difficult to see the indicators. The displays may appear as solid white surface.

Main Circuit and Shunt Connector (CN1)

The CN1 connector is used for the following functions:

● CK3A-G305L (10-pin)

- Main circuit power supply connection
- Internal shunt resistor connection (by short-circuit wire), or external shunt resistor connection
- Low voltage mode (by short-circuit wire)

● CK3A-G310L (6-pin)

- Main circuit power supply connection
- Internal shunt resistor connection (by short-circuit wire), or external shunt resistor connection

Logic Power Supply Connector (CN2)

The CN2 connector (3-pin) is used to supply 24VDC logic power to the Amplifier.

Motor Connector (CN3)

The CN3 connector is used to connect the Motor to the Amplifier. It is a 3-pin connector for the CK3A-G305L model and 4-pin connector for the CK3A-G310L model.

Safe Torque OFF Connector (CN4)

The CN4 connector (5-pin) is used to disable or connect the STO input(s), and STO status output. The short-circuit wire to disable the STO is installed on the connector from the factory.

Direct PWM Connector (CN5)

The CN5 connector (36-pin) is used to connect the Amplifier to the Controller. This is a pre-configured cable connector.

Ground Terminals

Following, are the location and number of ground ⏚ terminals of the Amplifier:

Location	No. of Terminals	Suggested Use
Top	1	Protective earth PE of main circuit power input
Front	2	Frame Ground FG inside the control panel
Bottom	1	Frame Ground FG of Motor cable and shield

1-2 Operating Procedure

This section describes the necessary steps to integrate the CK3A Direct PWM Amplifier into a system.

1-2-1 Preparation

Item	Reference
Power PMAC Controller	Must have PWM Option shown in system configuration
Logic power supply	Must be within the specifications stated in this manual
Main circuit power supply	Must be within the specifications stated in this manual
Motor specifications	Must be within needed operating conditions and Amplifier specifications
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 • Multi-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
Safety Test	Test ON/OFF safety measures implemented (e.g. STO, E-Stop)
Precautions for Safe Use	Review Precautions for Safe Use section of this manual
Turn power ON	Referring to Turning Power ON procedure in Precautions for Safe Use

1-2-5 Controller Settings for Amplifier

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
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
Amplifier settings	Implement Amplifier specific settings described in this manual
Motor Settings	Implement Motor specific configuration and settings using Motor specification sheet and related manuals

1-2-6 Test Run

Item	Reference
Verify encoder feedback	Using related hardware or user's manual
Enable ON/OFF	• Enable ON and OFF (open loop) with current loop gains in controller set to zero • Make sure there are no Amplifier faults per status display described in this manual
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 CK3A Direct PWM Amplifier.

2-1 Amplifier Specifications

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

2-1-1 General/Mechanical

Item		Specification
Number of axes		1
Enclosure		Panel mount
Protective case		IP20 (built into IP54 panel)
Grounding		200V class D grounding, 100 Ω or less
Vibration resistance		10 to 60 Hz at an acceleration of 5.88 m/s^2 or less (Not to be run continuously at the resonant frequency)
Air flow clearance		Refer to installation section
Mounting screws tightening torque		1.2 Nm
Cooling		Natural convection and built-in fan
Weight	CK3A-G305L	1.81 kg
	CK3A-G310L	2.67 kg
Dimensions	CK3A-G305L	212.5 x 65.0 x 180.0 mm
	CK3A-G310L	238.0 x 90.0 x 180.0 mm

2-1-2 Environmental

Item	Specification
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)
Operating and storage atmosphere	Must be free of corrosive gases
Maximum operating altitude	1,000 m

2-1-3 Electrical

Values in parentheses indicate the range of acceptable variation. Rated values are outside of parentheses.

	Item		CK3A-G305L	CK3A-G310L
Logic power supply	Voltage		24 VDC (22.0 – 26.4)	
	Current consumption		1.5 A	
	Inrush current		2.5 A	
	Inrush time		5 msec	
Main circuit power supply	3-Phase AC	Voltage	240 VAC (170 – 252 VAC)	
		F.L.A.	6 A_{rms}	11 A_{rms}
		Frequency	50 / 60 Hz	
	1-Phase AC	Voltage	110 – 240 VAC (85 – 252 VAC)	
		F.L.A.	10.5 A_{rms}	19.5 A_{rms}
		Frequency	50 / 60 Hz	
	1-Phase DC[*1]	(Low) Voltage	48 VDC[*1,3] (44 – 60 VDC)	N/A[*2,3]
		F.L.A	6 A_{rms}	N/A
Output	Rated Current		5 A_{RMS}	10 A_{RMS}
	Maximum (peak) Current		10 A_{RMS}	20 A_{RMS}
	Maximum Rated Power (3-Phase AC)		1195 W	2390 W
	Maximum Rated Power (1-Phase 240 VAC)		1195 W	2390 W
	Maximum Rated Power (1-Phase 110 VAC)		550 W	1095 W
	Maximum Rated Power (1-Phase DC)		195 W	N/A[*2]
	Time at Peak Current		2 sec	
PWM Interface	Current feedback resolution		16 bits	
	Maximum current ADC reading		15.735 A	31.470 A
	Minimum PWM deadtime		2 μsec	3 μsec
	PWM Frequency		8 – 20 KHz	
Shunt Resistor	Internal shunt resistor		30 W	
	External shunt resistor		20 Ω, 60 W	17 Ω, 60 W

*1. Must install low voltage short-circuit wire and set up ADC Strobe Word correctly.

*2. The CK3A-G310L requires a special part number, and factory modification to operate with low voltage main power input. Contact your local Omron representative for this option.

*3. The CK3A (both models) can functionally operate in the 100-350 VDC main supply input range without short-circuit wire or special consideration. Operation and setup are same as described in this manual however, the specifications and performance are not defined in this case.

2-1-4 Performance

Precautions for Safe Use

Install a proper stopping device on the machine to ensure safety. The dynamic brake mechanism is not a stopping device.

Precautions for Correct Use

- If the dynamic brake function is ON, the user must enable the Amplifier and wait a minimum of 50 milliseconds before commanding motion.
- If dynamic braking has been performed (after Motor KILL while in motion), the user must wait a minimum of 500 milliseconds before re-enabling the Amplifier.

Specification	Value	Notes
STO input to power drivers OFF	< 30 msec	
Overcurrent I2T to IPM OFF	< 10 msec	A8 fault
Phase short to IPM OFF	< 3 μsec	AC fault
Current loop response time	< 1 msec	1 mH 3-Ph brushless Motor Y-winding
Dynamic brake relay response time	< 20 msec	Mechanical relay time constant
I2T time to Amplifier OFF	< 2.5 sec	At 200% output
Soft start time	< 650 msec	Do not enable Amplifier during soft start
Hold at momentary power interruption	10 msec	3-ph 208VAC @ rated load
DC bus discharge to less than < 36 VDC	< 2.5 sec	Forced discharge to shunt resistor
DC bus discharge to less than < 36 VDC	< 5 min	Natural discharge
Current ADC clock frequency range	2.450 – 6.250 MHz	Set in Controller
Time between main circuit power cycles	1 min minimum	If discharge ON
Time between main circuit power cycles	10 sec minimum	If discharge OFF

2-1-5 Regenerative Energy Absorption

The Amplifier has built-in capacitors, which absorb the regenerative energy produced during motor deceleration and dynamic braking due to counter EMF. When the built-in capacitors cannot absorb all regenerative energy, the Internal (or external) shunt resistor absorbs the rest of the energy.

If the amount of regenerative energy from the Motor is too large, the regeneration operation is stopped, and the Motor is disabled with an overvoltage error (A5) in order to prevent the internal shunt resistor from burning.

If the continuous duration of regeneration exceeds 2 seconds, the regeneration operation is stopped, and the Motor is disabled with a shunt overload error (A3) to prevent overheating and damage. If this happens, it is recommended to allow 5 minutes for cool down.

If these cases occur, the regenerative energy produced by the Motor must be reduced. The calculation of the regenerative energy and recommendations are described in this section.

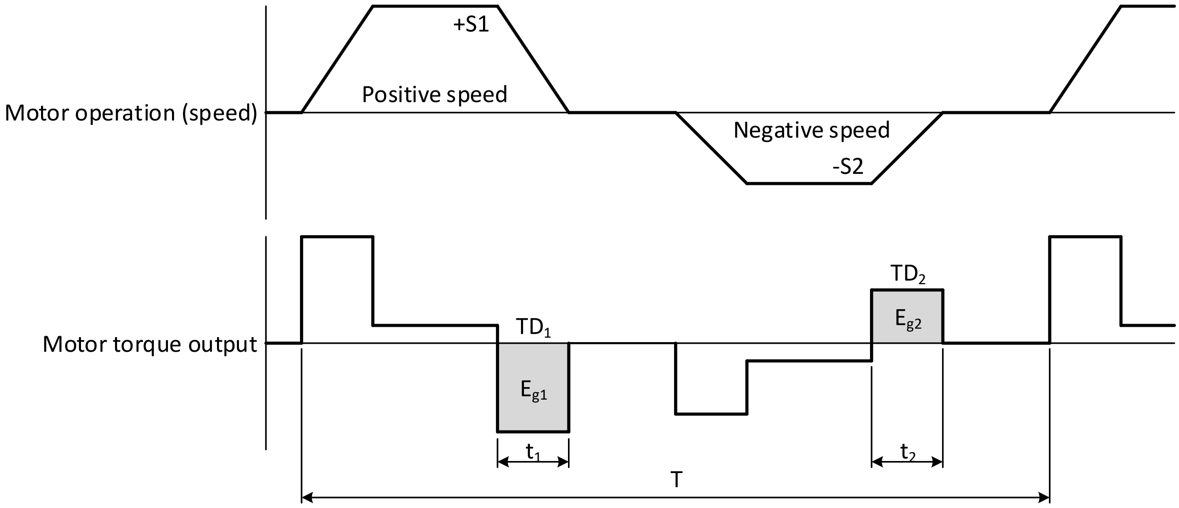
Amplifier Internal Regeneration Absorption Capacity

The following table shows the Amplifier power, internal shunt resistor specifications, regenerative power absorption capacity and maximum duration. These values are based on a 200VAC main power supply.

Model	CK3A-G305L	CK3A-G310L
Rated RMS power [W]	1195W	2390W
Internal shunt resistor specification	25Ω 30W	17Ω 80W
Built-in capacitors absorption energy [J]	46J	62J
Internal shunt resistor average regeneration energy [W]	18W	32W
Maximum duration of continuous regeneration [sec]	2 sec	2 sec
Minimum allowable shunt resistance [Ω]	20Ω	15Ω

Regenerative Energy Calculation

- **Horizontal Axis**



Symbol	Description
S1, S2	Motor speed at the start of deceleration [r/min] for rotary Motor, [m/s] for linear Motor
TD_1, TD_2	Deceleration torque [N.m]
t_1, t_2	Deceleration time [s]

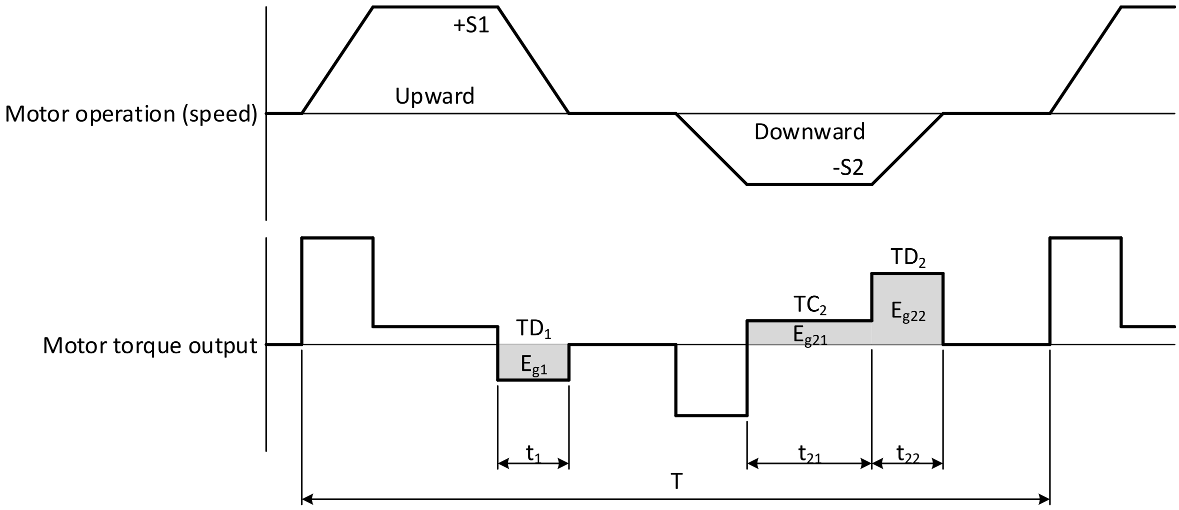
The regenerative energy absorption Eg_1, and Eg_2 [J] can be calculated as follows.

Rotary Motor Regenerative Energy [J]	Linear Motor Regenerative Energy [J]
$E_{g1} = \frac{1}{2} \times \frac{2\pi}{60} \times S1 \times TD_1 \times t_1$	$E_{g1} = \frac{1}{2} \times S1 \times T_1 \times t_1$
$E_{g2} = \frac{1}{2} \times \frac{2\pi}{60} \times S2 \times TD_2 \times t_2$	$E_{g2} = \frac{1}{2} \times S2 \times T_2 \times t_2$

Additional Information

Because of Motor winding resistance, the actual regenerative energy will be approximately 90% of the calculated values.

● Vertical Axis



Symbol	Description
S1, S2	Motor speed at the start of deceleration [r/min] for rotary Motor, [m/s] for linear Motor
TD_1, TD_2	Deceleration torque [N.m]
TC_2	Downward constant-speed torque [N.m]
t_1, t_{22}	Deceleration time [s]
t_{21}	Downward constant-speed time [s]

The regenerative energy absorption for each area Eg_1, Eg_{21}, and Eg_{22} [J] can be calculated as follows.

Rotary Motor Regenerative Energy [J]	Linear Motor Regenerative Energy [J]
$E_{g1} = \frac{1}{2} \times \frac{2\pi}{60} \times S1 \times TD_1 \times t_1$	$E_{g1} = \frac{1}{2} \times S1 \times TD_1 \times t_1$
$E_{g21} = \frac{2\pi}{60} \times S2 \times TC_2 \times t_{21}$	$E_{g21} = S2 \times TC_2 \times t_{21}$
$E_{g22} = \frac{1}{2} \times \frac{2\pi}{60} \times S2 \times TD_2 \times t_{22}$	$E_{g22} = \frac{1}{2} \times S2 \times TD_2 \times t_{22}$
$E_{g2} = E_{g21} + E_{g22}$	

Note that the total regenerative energy in the downward movement is $Eg_2 = Eg_{21} + Eg_{22}$.

Additional Information

Because of Motor winding resistance, the actual regenerative energy will be approximately 90% of the calculated values.

Necessary Regeneration Energy vs. Amplifier Absorption Capacity

If both of the previously calculated regenerative energy E_{g1}, and E_{g2} [J] values are smaller than or equal to the Amplifier's capacitors regenerative absorption capacity E_C [J], then there is nothing to do.

However, if either E_{g1} or E_{g2} [J] is larger than the Amplifier's capacitors regenerative absorption capacity E_C [J], then use the following equation to calculate the average regenerative power P_r [W].

$E_g = (E_{g1} – E_C) + (E_{g2} – E_C)$ [J]
$P_r = E_g / T$ [W]

Symbol	Description
P_r	Average regenerative power absorption in one cycle of operation [W]
E_g	Regenerative energy absorption in one cycle of operation [J]
E_C	Built-in capacitors absorption energy [J]
T	Operation cycle time [s]

Note If $(E_{g1} – E_C)$, or $(E_{g2} – E_C) \leq 0$, then use 0.

If the average regenerative power P_r [W] is less than or equal to the average regenerative power of the Amplifier's built-in resistor, then there is nothing to do.

However, if the average regenerative power P_r [W] is greater than the average regenerative power of the Amplifier's built-in resistor, then one or more of the following measures can be taken.

- Connect an external shunt resistor.
- Decrease the motion profile speed. The regenerative energy is proportional to the square of the Motor speed.
- Decrease the motion profile deceleration time, thus reducing the regenerative energy per unit time.
- Increase the operation cycle time, allowing more time between consecutive moves.

2-2 Functions and Data Reporting

The CK3A Direct PWM Amplifier interfaces with the Controller via PWM cable.
The Amplifier has functions that can be configured by the Controller. Also, data and status bits that can be read in the Controller.

Precautions for Correct Use

If the PWM cable is unplugged or logic power is OFF, the functions, data and status information is not accessible.

Additional Information

- The Amplifier data reporting is not intended for real-time use. Depending on the method of acquisition from the Controller, and number of channels, the update rate can be in the hundreds of milliseconds in some cases.
- The Amplifier data reporting is intended for general purpose use during programming, troubleshooting, or machine operator status display.
- The Amplifier functions and data reporting are controlled by the setting of the ADC Strobe Word in the Controller.

2-2-1 Amplifier Functions

The following, are the Amplifier functions.

Item	Function Options
Cooling fan control	• Automatic (by Amplifier firmware) OR • Always ON
Dynamic brake control	• Let Motor coast (free run) after servo OFF (or error) event OR • Make Motor come to a stop after servo OFF (or error) event
Clear fault function	Process of clearing Amplifier faults (using ADC Strobe Word)
Low voltage mode control	Normal AC input (per specifications) OR • Low voltage 44 - 60VDC main circuit power input (only for CK3A-G305L)
Bus discharge control	• Do not dissipate residual energy when main power circuit is turned OFF OR • Dissipate residual energy when main power circuit is turned OFF

Precautions for Correct Use

Dynamic braking is only intended to stop the Motor in the case of a fault condition. Do not KILL (servo OFF) frequently a Motor in motion if the dynamic brake function is ON.

2-2-2 Data Reporting

The Amplifier can report (to the Controller) the following data.

Item	Specification
DC bus voltage	Reports the DC link bus voltage in VDC
Power module temperature	Reports the power module temperature in ℃
Firmware version	Reports the Amplifier firmware version
Current rating	Reports the current rating of the Amplifier in Amperes

2-2-3 Status Bits

The Amplifier can report (to the Controller) the following status bits.

Status Bits Set 1	Status bits Set 2	Status Bits Set 3
Main power circuit fault	Main power circuit status	I2T integrator status
Soft start fault	Over temperature fault	ADC offset fault
STO status	Shunt overload fault	Amplifier ready status
Shunt short fault	PWM frequency fault	-
Over voltage fault	PWM command fault	-
I2T Fault	Amplifier enabled status	-
Short circuit fault	-	-
Overload fault	-	-

2-3 Part Number Designation

This section shows the part number designation.

2-3-1 Amplifier Part Number

C	K	3	A	-	G	3			L

05:	5/10A model
10:	10/20A model

3

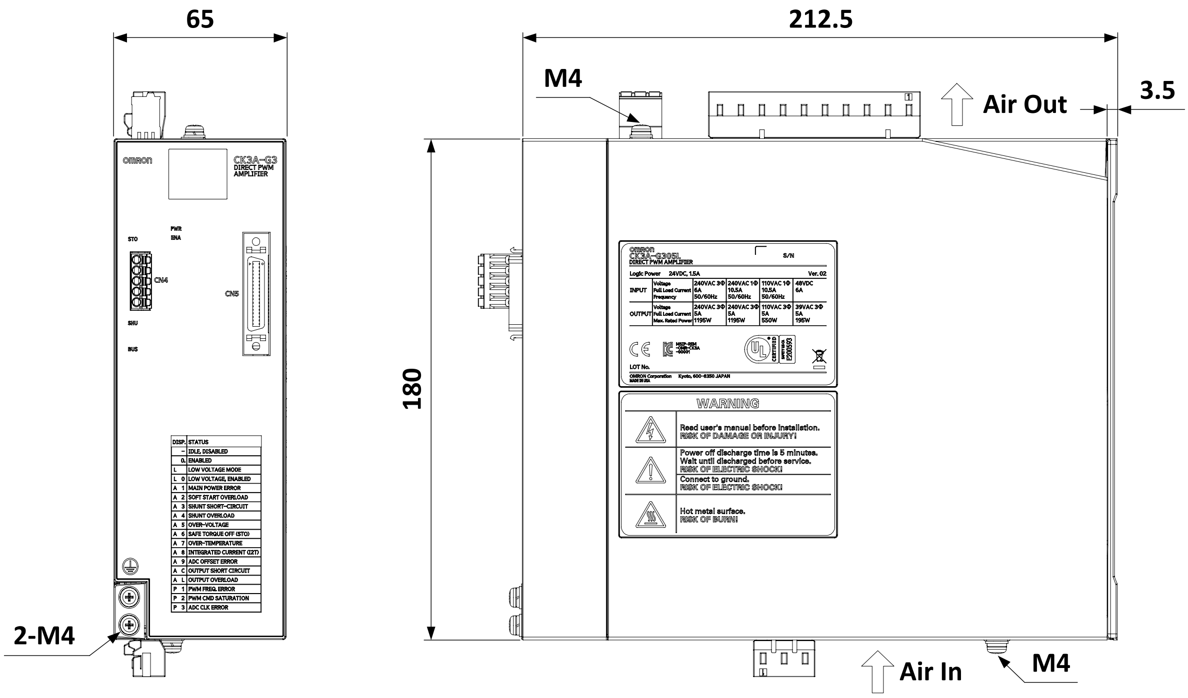
Mounting and Installation

This section describes the mounting and installation methods of the CK3A Direct PWM Amplifier. All dimensions given in millimeters.

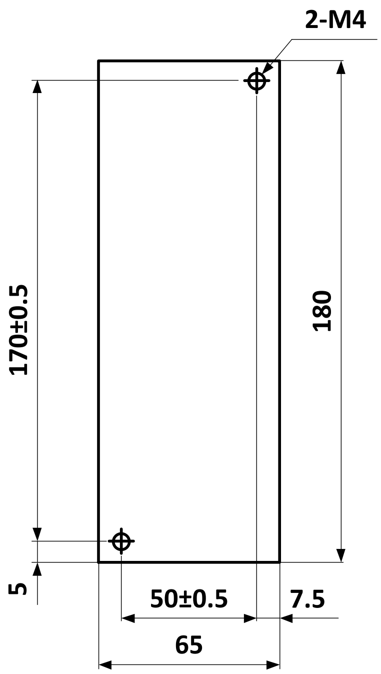
3-1 External and Mounting Dimensions

3-1-1 CK3A-G305L Dimensions and Mounting

External Dimensions

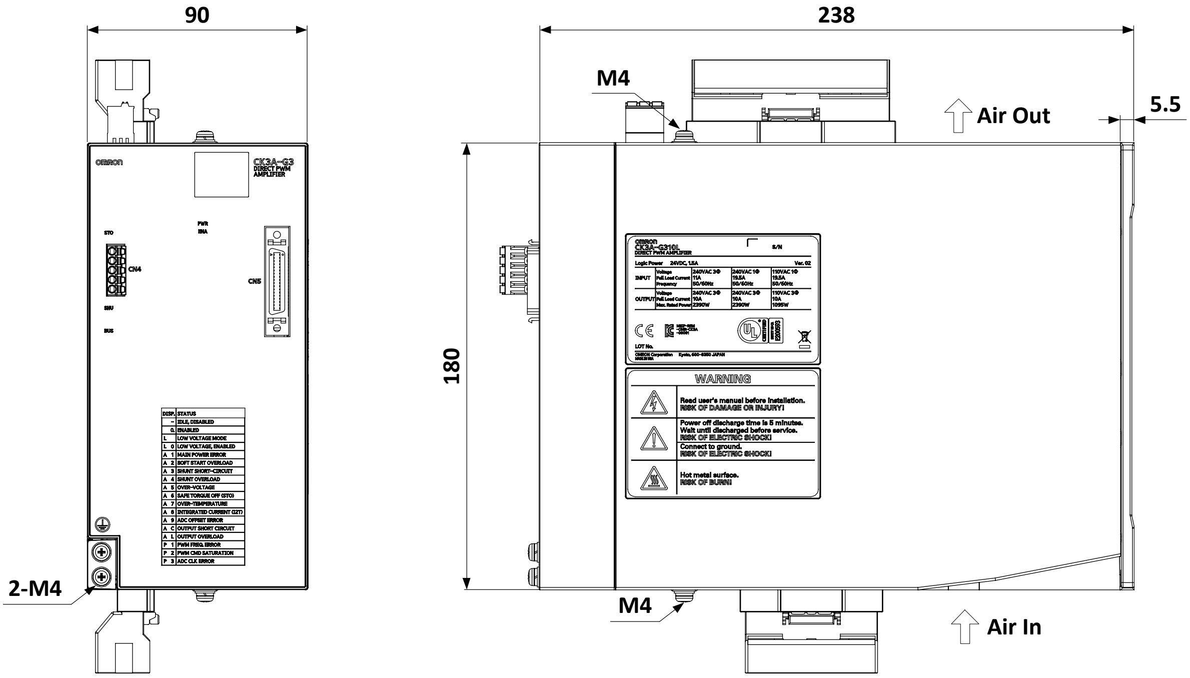


Mounting Dimensions

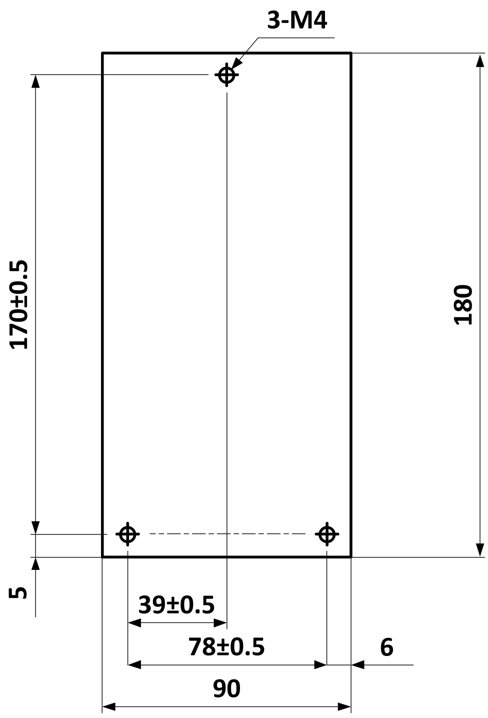


3-1-2 CK3A-G310L Dimensions and Mounting

External Dimensions



Mounting dimensions



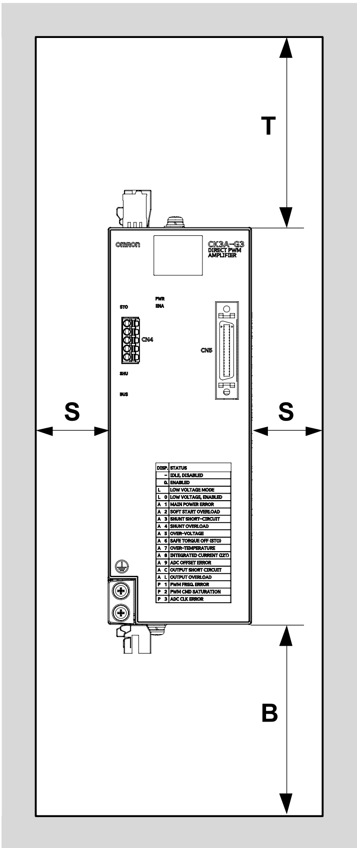
3-2 Installation

3-2-1 Ventilation

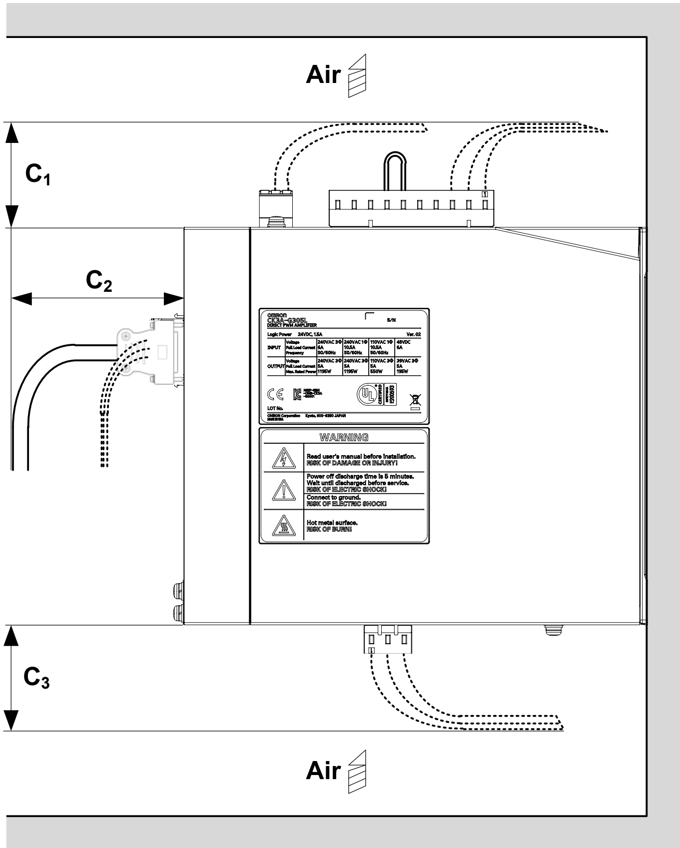
Install the Amplifier according to the dimension specifications shown below. Ensure proper dissipation of heat from the Amplifier and convection inside the panel. If the Amplifiers are installed side by side, install a ventilation system or fan for air circulation to prevent uneven temperatures inside the panel.

3-2-2 Panel Clearance

Single-Unit Installation

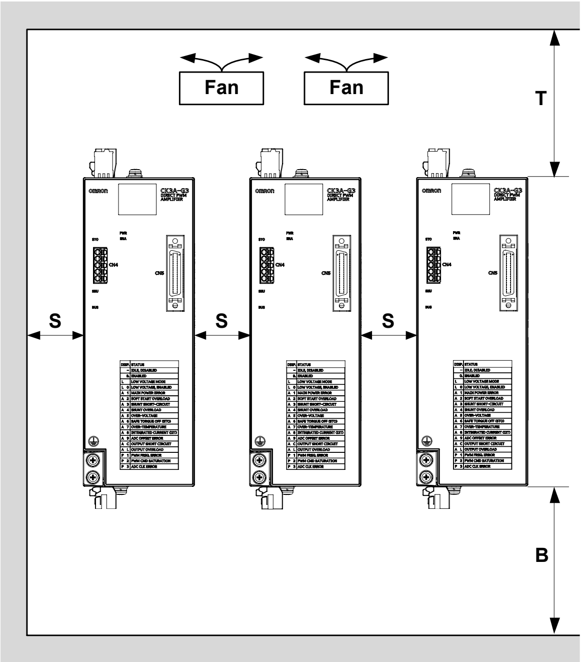


Front View

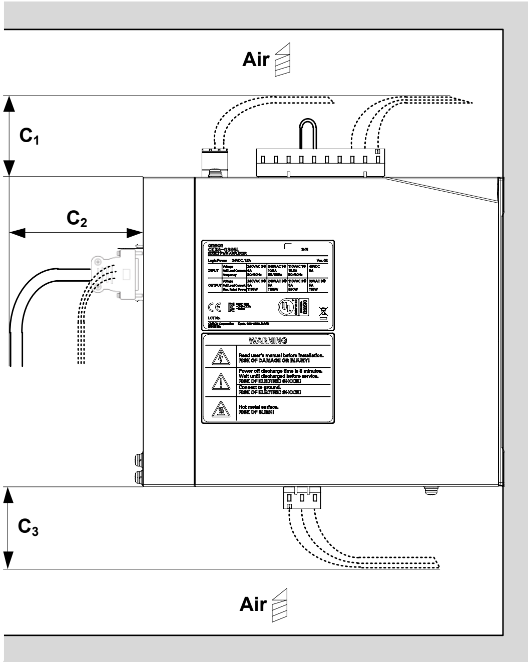


Side View

Multi-Unit Side-by-side



Front View



Side View

- **CK3A-G305L**

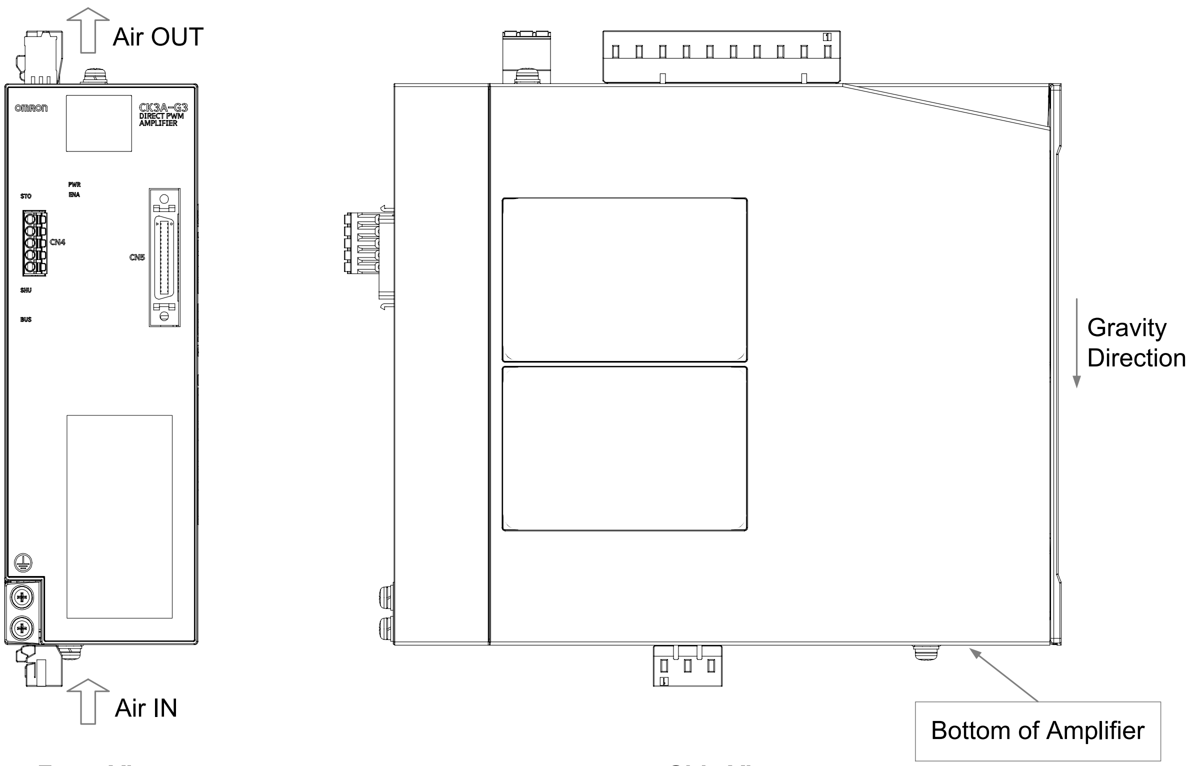
Dimension	Minimum Distance [mm]
T (top)	110
B (bottom)	110
S (side)	10
C_1	70
C_2	100
C_3	70

- **CK3A-G310L**

Dimension	Minimum Distance [mm]
T (top)	140
B (bottom)	140
S (side)	10
C_1	100
C_2	100
C_3	100

3-2-3 Mounting Direction

The Amplifier should be mounted vertically in the gravity direction where the bottom is parallel to the floor.



Front View **Side View**

3-2-4 Installation Conditions

- The panel environmental (such as operating temperature and humidity) conditions must abide by the environmental specifications stated in this manual. Operating the Amplifier outside of these conditions may result in malfunction.

Dimension	Distance
Operating ambient temperature	0 to 55℃
Operating ambient humidity	90% maximum (with no condensation)
Operating atmosphere	Without corrosive gases
Operating altitude	1,000 m maximum

- When the Amplifiers are installed side by side in a closed space, such as a drawer, the ambient temperature may increase due to heat dissipation from each unit. Use a fan or air conditioner to sustain the specified operating ambient temperature.
- Operating in an environment where temperature rise is minimal is recommended to maintain a high level of reliability.
- The use of the Amplifier continuously in a hot environment shortens its lifetime. Use a fan or air conditioner to maintain the ambient temperature at or below 40°C.
- Install the Amplifier on a vertical metal surface.
- Remove any surface paint to provide electrical conductivity.
- Apply conductive plating when using a custom mounting bracket.
- The recommended tightening screw for mounting the Amplifier is 1.2 Nm.
- Make sure that the threaded portion of the mounting screws has sufficient depth to withstand the weight of the Amplifier and recommended torque.

3-2-5 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 Amplifier.

- Place a cover over the Amplifier or take other preventative measures to keep foreign objects, such as drill filings, out of the Amplifier during installation. Be sure to remove the cover after installation is complete. If the cover is left on during operation, heat dissipation from the Amplifier is blocked, which may result in malfunction.

4

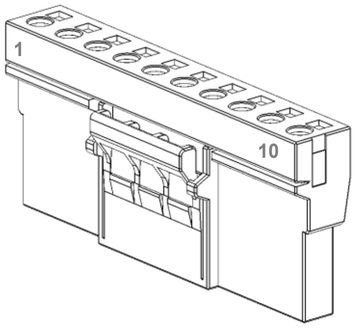
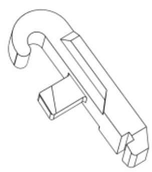
Wiring

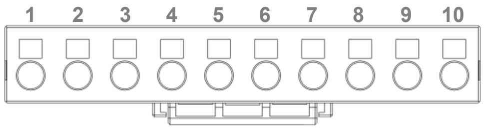
This section describes the wiring methods of the CK3A Direct PWM Amplifier.

4-1 Main Circuit Power and Shunt (CN1)

4-1-1 Connector Pinout

CK3A-G305L

Mating Connector	Spring Opener
10-pin receptacle TE Connectivity PN 1-2134249-0 ODT PN 040-000006 	TE Connectivity PN 1981045-1 ODT PN 100-000032 



Pin	Symbol	Description and Specifications
1	L1	**Main circuit power supply input.** • 3-phase 170 – 252VAC 50/60Hz across L1-L2-L3 • 1-phase 85 – 252VAC 50/60Hz across L1-L2, leave L3 floating (open) • 1-phase 44 - 60VDC across L1-L2, leave L3 floating (open)
2	L2	
3	L3	
4	NC	Not connected (leave floating)
5	NC	
6	B1	**Shunt resistor terminals.** • For internal shunt resistor, short-circuit B2-B3, leave B1 floating • For external shunt resistor, connect resistor between B1-B2, leave B3 floating
7	B2	
8	B3	
9	LV1	**Low voltage operation.** • Short-circuit LV1-LV2 for 44 - 60VDC operation[*1] • Leave floating (open) for 1- or 3-phase AC operation
10	LV2	

*1. Must also set ADC Strobe Word for low voltage mode in the controller.

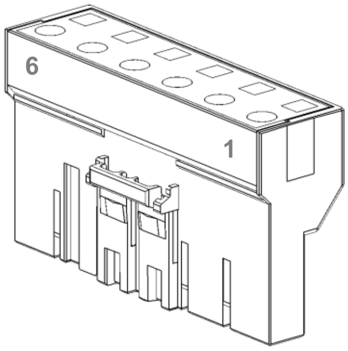
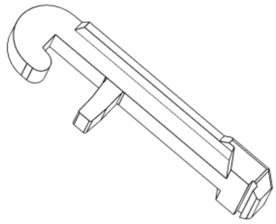
Precautions for Safe Use

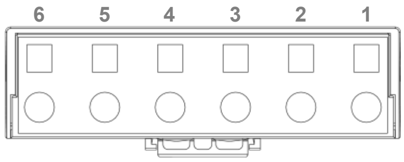
Do not apply AC power if the low voltage short-circuit wire is installed.

Additional Information

Connector is shipped with short-circuit wire B2-B3 installed for internal shunt resistor.

CK3A-G310L

Mating Connector		Spring Opener
6-pin receptacle TE Connectivity PN 6-2260663-1 ODT PN 016-000006		TE Connectivity PN 2260828-1 ODT PN 100-000033 



Pin	Symbol	Description and Specifications
1	L1	**Main circuit power supply input.** • 3-phase 170 – 252VAC 50/60Hz across L1-L2-L3 • 1-phase 85 – 252VAC 50/60Hz across L1-L2, leave L3 floating (open)
2	L2	
3	L3	
4	B3	**Shunt resistor terminals.** • For internal shunt resistor, short-circuit B2-B3, leave B1 floating (open) • For external shunt resistor, connect to B1-B2, leave B3 floating (open)
5	B2	
6	B1	

Additional Information

Connector is shipped with short-circuit wire B2-B3 installed for internal shunt resistor.

4-1-2 Wire size

Precautions for Correct Use

Use cable wires rated to 600VAC or higher for single or three-phase AC main circuit power.

The recommended wire gauges for the main circuit power supply, internal resistor short-circuit wire, and low voltage mode short-circuit wire are shown below.

Item	CK3A-G305L	CK3A-G310L
Main circuit power and internal resistor short-circuit wire	AWG 18 – 14 0.75 – 2.0 mm^2	AWG 12 – 8 3.3 – 8.4 mm^2
Protective earth ⏚	AWG 12, 2.5 mm^2	
Screw size	M4	
Tightening torque	1.2 Nm	

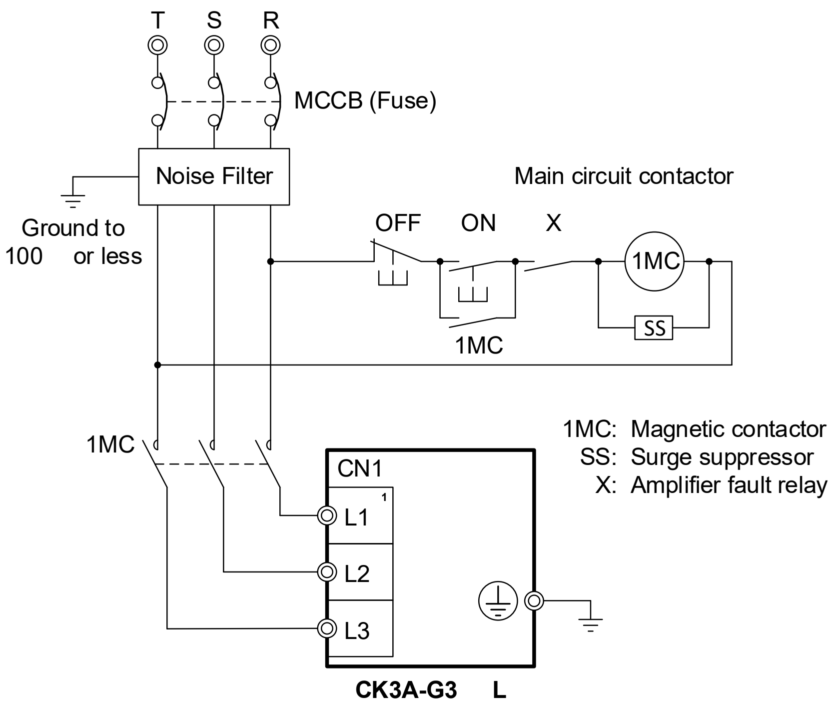
Note For the external shunt resistor, use the built-in wires provided with the resistor.

4-1-3 Wiring Examples

Three-Phase AC

The Amplifier is designed to operate with three-phase AC main circuit power input. The following diagram depicts this wiring scheme.

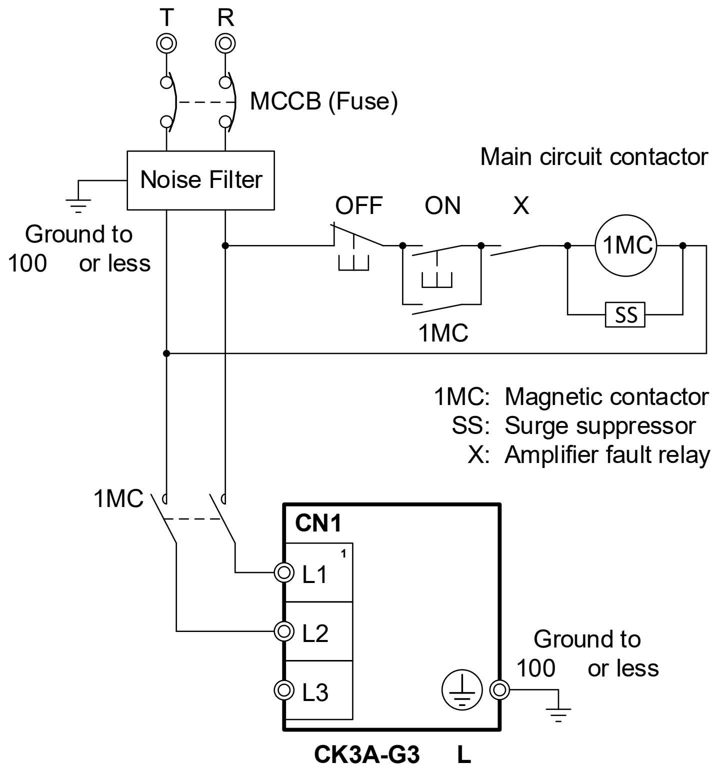
CK3A-G3□□L Three-phase Main Circuit Input



Single-Phase AC

The Amplifier can operate with single-phase AC main circuit power input. The following diagram depicts this wiring scheme.

CK3A-G3□□L Single-phase Main Circuit Input



DC Input (Low Voltage Mode)

The CK3A-G305L can operate with low voltage (44 - 60VDC) main circuit power input. The following diagram depicts this wiring scheme.

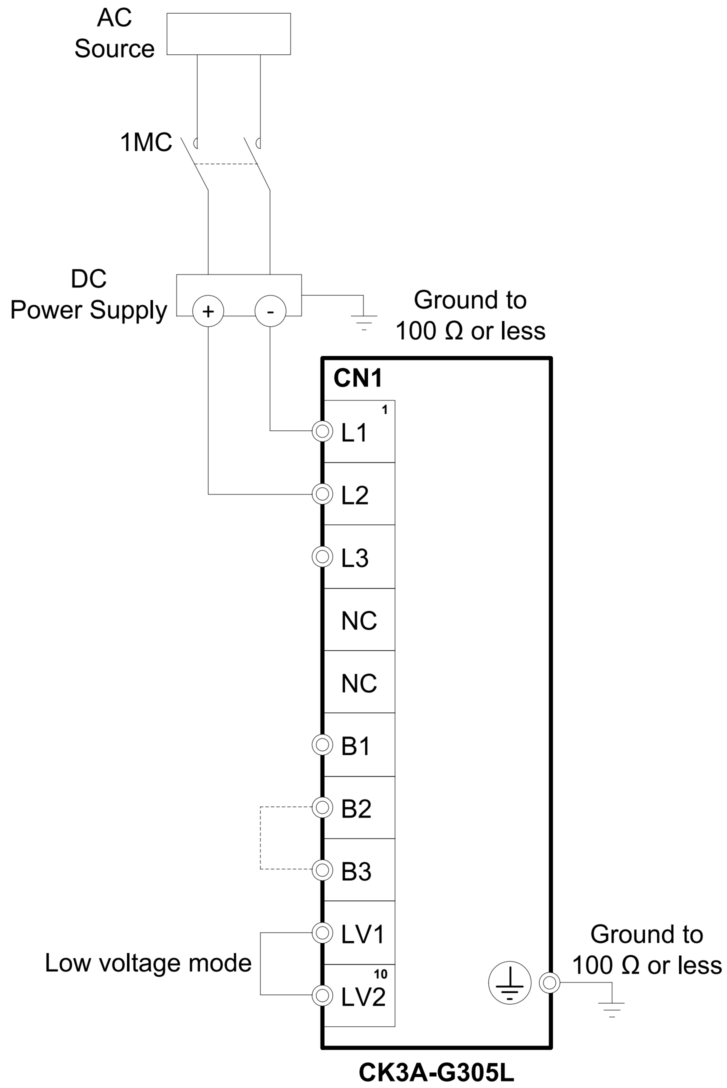
Precautions for Safe Use

Do not apply AC power if the low voltage short-circuit wire is installed.

Precautions for Correct Use

The ADC Strobe Word setting in the controller must be set accordingly for successful low voltage operation.

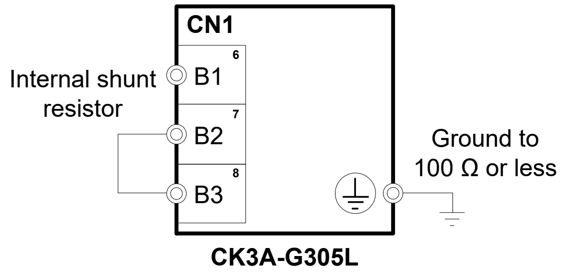
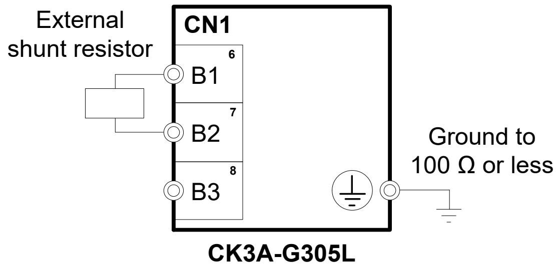
CK3A-G305L DC Main Circuit Input



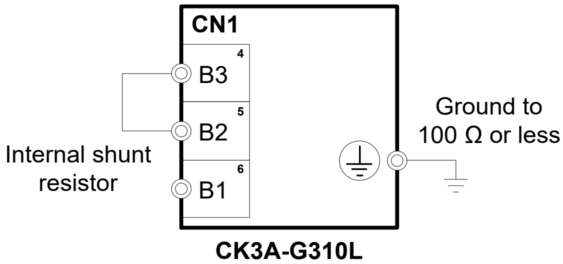
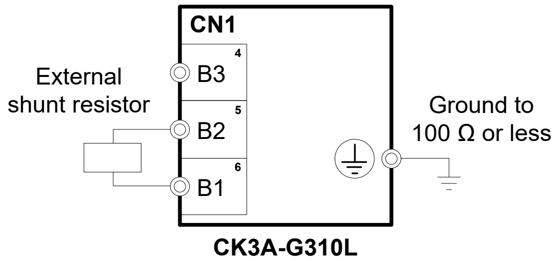
Shunt Resistor Wiring

The following diagrams show the internal and external shunt resistor wiring configurations.

● CK3A-G305L

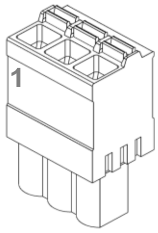
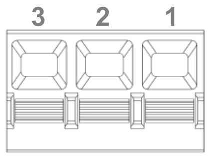
Internal Shunt Resistor	External Shunt Resistor
	

● CK3A-G310L

Internal Shunt Resistor	External Shunt Resistor
	

4-2 Logic Power Supply (CN2)

4-2-1 Connector Pinout

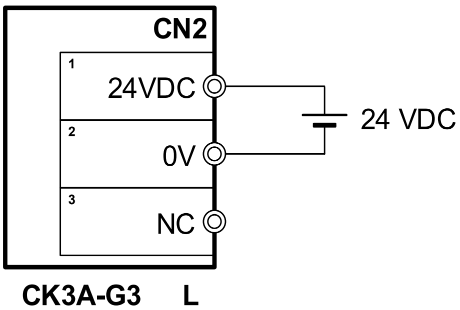
Mating Connector		
3-pin terminal block Phoenix Contact PN 1754571 ODT PN 016-175457-1P3		

Pin	Symbol	Description and Specifications
1	24 VDC	**Logic power supply input.** • 22.0 – 26.4VDC • Current consumption1.5A (2.5A in-rush for 5 msec)
2	0 V	
3	NC	Not connected (leave floating)

4-2-2 Wire size

The recommended wire gauge for the logic power supply input is AWG 20 – 16, 0.5 – 1.5 mm^2 .

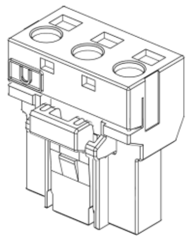
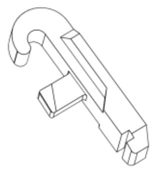
4-2-3 Wiring Example

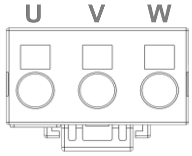


4-3 Motor Connection (CN3)

4-3-1 Connector Pinout

CK3A-G305L

Mating Connector	Spring Opener
3-pin receptacle TE Connectivity PN 3-2229794-1 ODT PN 017-000002 	TE Connectivity PN 1981045-1 ODT PN 100-000032 



Pin	Symbol	Description and Specifications
1	U	**Motor phases output terminals.** • For brushless Motor, connect U, V, and W • For brushed Motor, connect U and W, leave V floating (open)
2	V	
3	W	

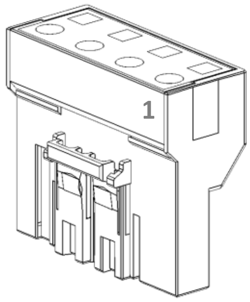
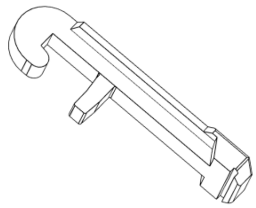
Precautions for Correct Use

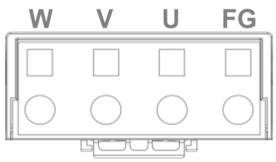
Connect the Motor frame ground to the protective earth ⏚ mountings of the Amplifier.

Precautions for Safe Use

The Amplifier does not error if the Controller is sending Motor commands while the Motor connector is unplugged. Do NOT plug the Motor connector until you make sure that the Controller is not sending Motor commands or main power is disconnected.

CK3A-G310L

Mating Connector		Spring Opener
4-pin receptacle TE Connectivity PN 4-2260663-1 ODT PN 017-000001		TE Connectivity PN 2260828-1 ODT PN 100-000033 



Pin	Symbol	Description and Specifications
1	FG	**Motor phases output terminals.** • FG is frame ground, tie Motor frame ground wire to this terminal • For 3-phase brushless Motor, connect U, V, and W • For 2-phase brushed Motor, connect U and W, and leave V floating (open)
2	U	
3	V	
4	W	

4-3-2 Wire Size

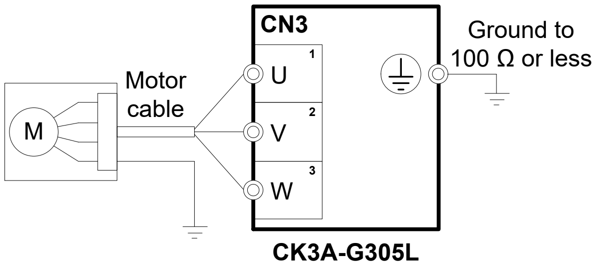
The recommended wire gauges for the Motor phase terminals and frame grounds are shown below.

Item	CK3A-G305L	CK3A-G310L
Main circuit power and short-circuit wire	AWG 18 – 14 0.75 – 2.0 mm^2	AWG 14 – 8 2.0 – 8.4 mm^2
Protective earth ⏚	AWG 12, 2.5 mm^2	
Screw size	M4	
Tightening torque	1.2 Nm	

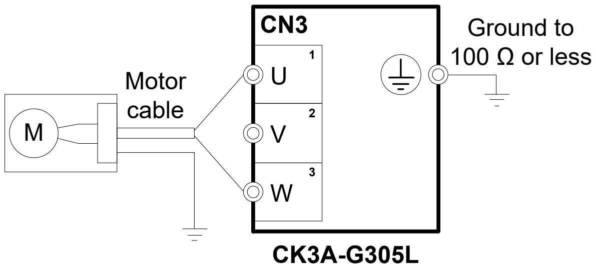
4-3-3 Wiring Examples

CK3A-G305L

- **Brushless Motor**

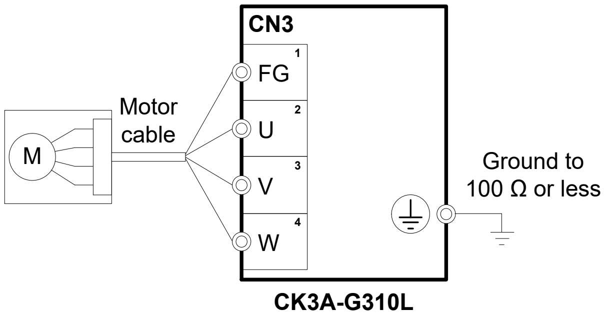


- **Brushed Motor**

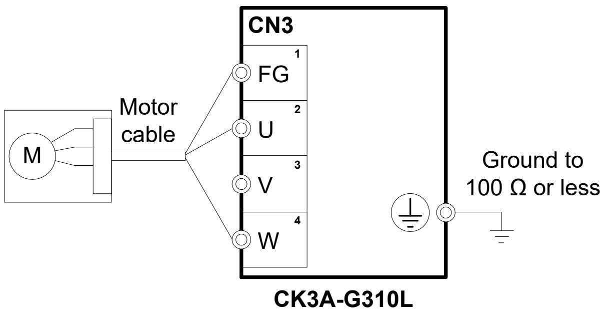


CK3A-G310L

- **Brushless Motor**

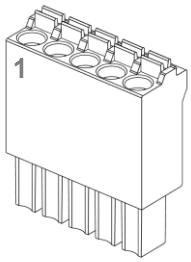
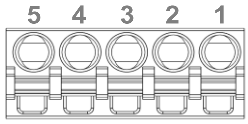


- **Brushed Motor**



4-4 Safe Torque OFF STO (CN4)

4-4-1 Connector Pinout

Mating Connector		
5-pin terminal block Phoenix Contact PN 1745920 ODT PN 040-000005		

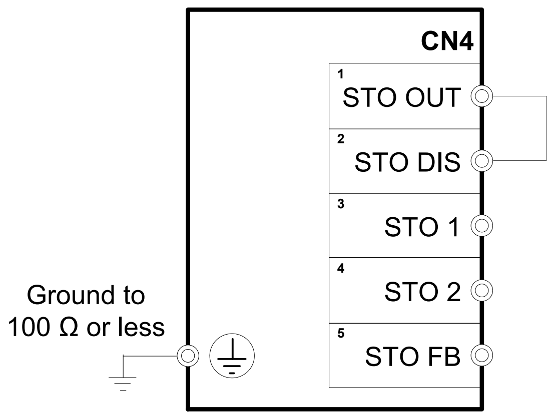
Pin	Symbol	Description and Specifications
1	STO OUT	**Safe Torque Off STO control terminals.** • To disable the STO function; Short-circuit STO OUT and STO DIS, leave all other terminals floating (open). • To use STO input 1 only; Wire relay inputs between STO IN1 and STO OUT, tie STO IN2 to STO OUT, leave STO DIS floating (open). • To use STO input 2 only; Wire relay inputs between STO IN2 and STO OUT, tie STO IN1 to STO OUT, leave STO DIS floating (open). • To use both STO inputs 1 and 2; Wire relay input 1 between STO IN1 and STO OUT, wire relay input 2 between STO IN2 and STO OUT, leave STO DIS floating (open).
2	STO DIS	
3	STO IN1	
4	STO IN2	
5	STO FB	Safe Torque OFF STO status terminal. This terminal is a PNP type output reflecting the STO status.

4-4-2 Wire Size

The recommended wire gauge for the STO functions is AWG 20 – 16, 0.5 – 1.5 mm^2 .

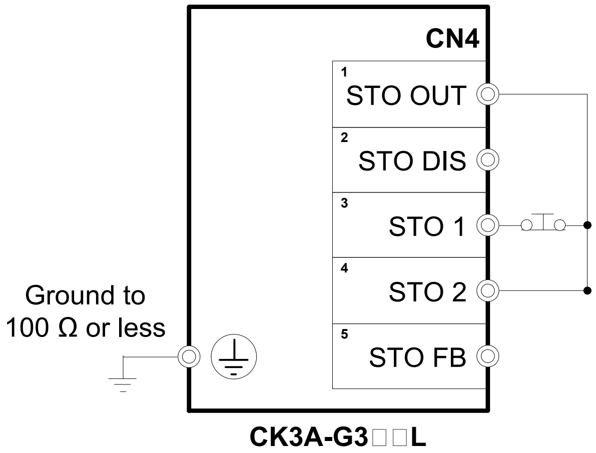
4-4-3 Wiring Examples

Disabling the STO

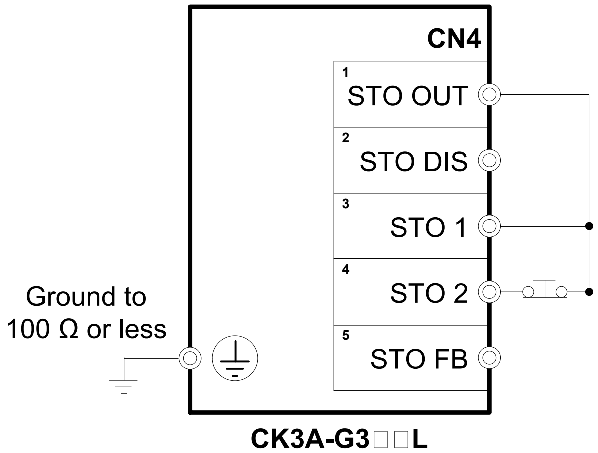


CK3A-G3□□L

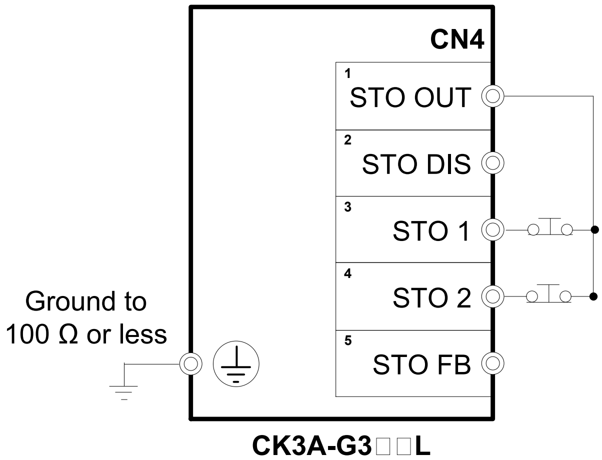
Using STO 1 Only



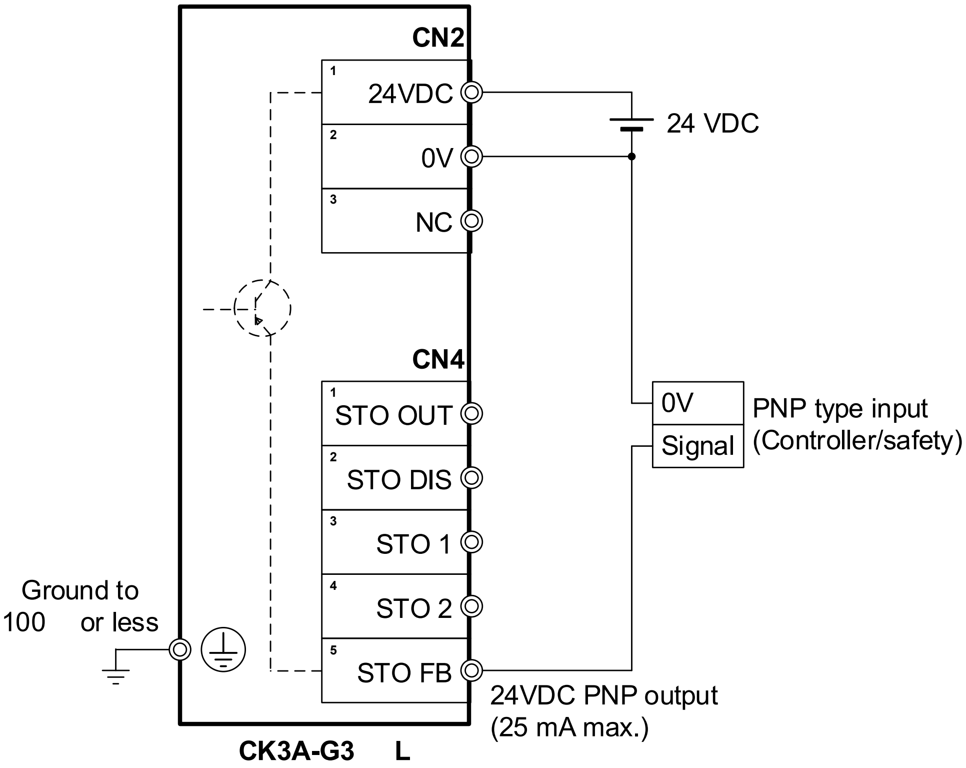
Using STO 2 Only



Using Both STO 1 and STO 2



STO Feedback



4-5 Direct PWM Connection (CN5)

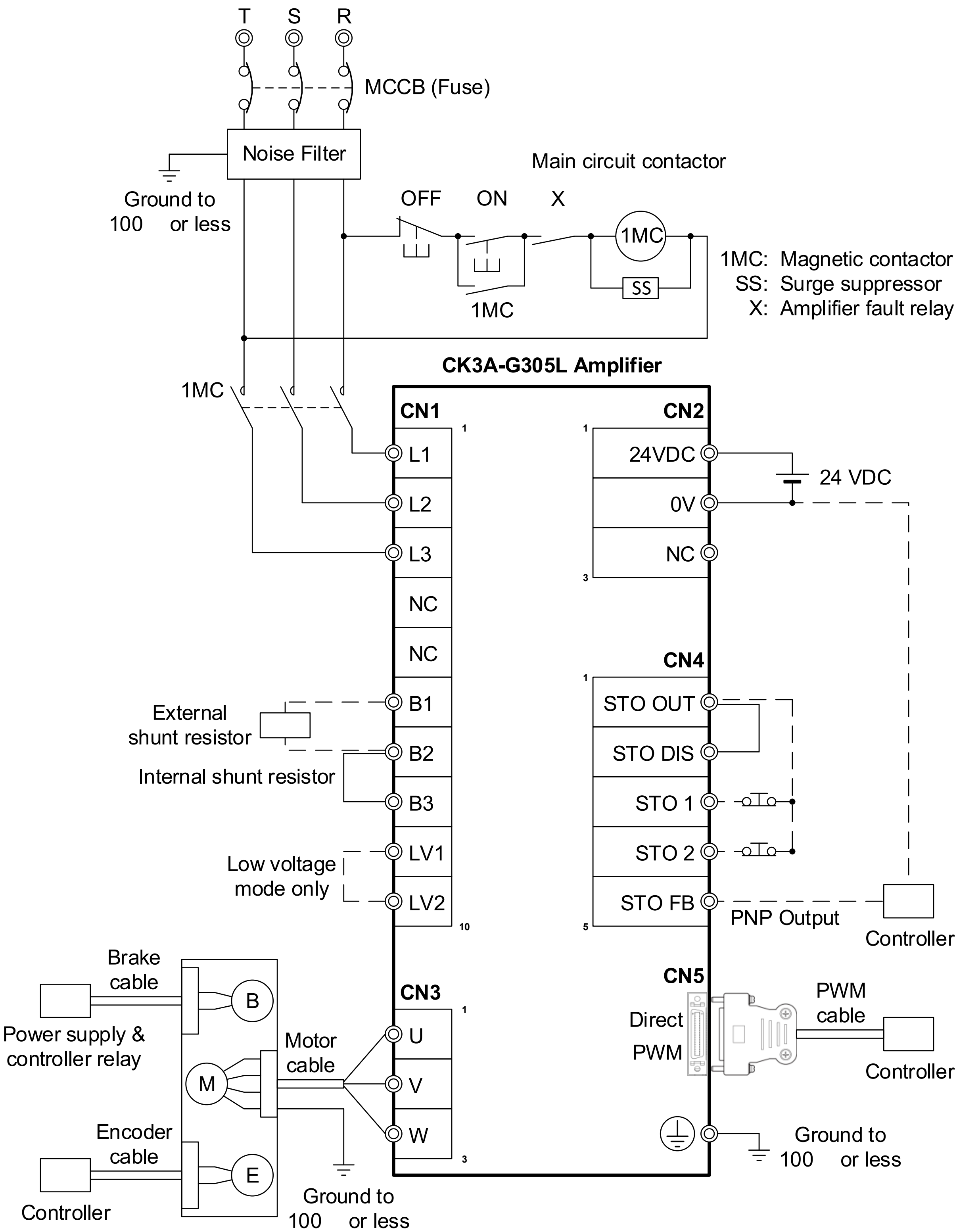
4-5-1 Connector Pinout

Pin	Symbol	Description	Function[*1]
1	PHACLK1+	Phase clock	Input
2	N.C.	No connect	-
3	ADCCLK1+	A/D converter clock	Input
4	ADCSTB1+	A/D converter strobe	Input
5	ADCDAT1A+	Phase A current data	Output
6	ADCDAT1B+	Phase B current data	Output
7	AENA1+	Amplifier enable	Input
8	FAULT1+	Amplifier fault	Output
9	PWMATOP1+	Phase A top cmd.	Input
10	PWMABOT1+	Phase A bottom cmd.	Input
11	PWMBTOP1+	Phase B top cmd.	Input
12	PWMBBOT1+	Phase B bottom cmd.	Input
13	PWMCTOP1+	Phase C top cmd.	Input
14	PWMCBOT1+	Phase C bottom cmd.	Input
15	0V	Reference voltage	Common
16	+5V	Digital power	Input
17	PWMDTOP1+	Phase D top cmd.	Input
18	PWMDBOT1+	Phase D bottom cmd.	Input
19	PHACLK1-	Phase clock	Input
20	N.C.	No connect	-
21	ADCCLK1-	A/D converter clock	Input
22	ADCSTB1-	A/D converter strobe	Input
23	ADCDAT1A-	Phase A current data	Output
24	ADCDAT1B-	Phase B current data	Output
25	AENA1-	Amplifier enable	Input
26	FAULT1-	Amplifier fault	Output
27	PWMATOP1-	Phase A top cmd.	Input
28	PWMABOT1-	Phase A bottom cmd.	Input
29	PWMBTOP1-	Phase B top cmd.	Input
30	PWMBBOT1-	Phase B bottom cmd.	Input
31	PWMCTOP1-	Phase C top cmd.	Input
32	PWMCBOT1-	Phase C bottom cmd.	Input
33	0V	Reference voltage	Common
34	+5V	Digital power	Input
35	PWMDTOP1-	Phase D top cmd.	Input
36	PWMDBOT1-	Phase D bottom cmd.	Input

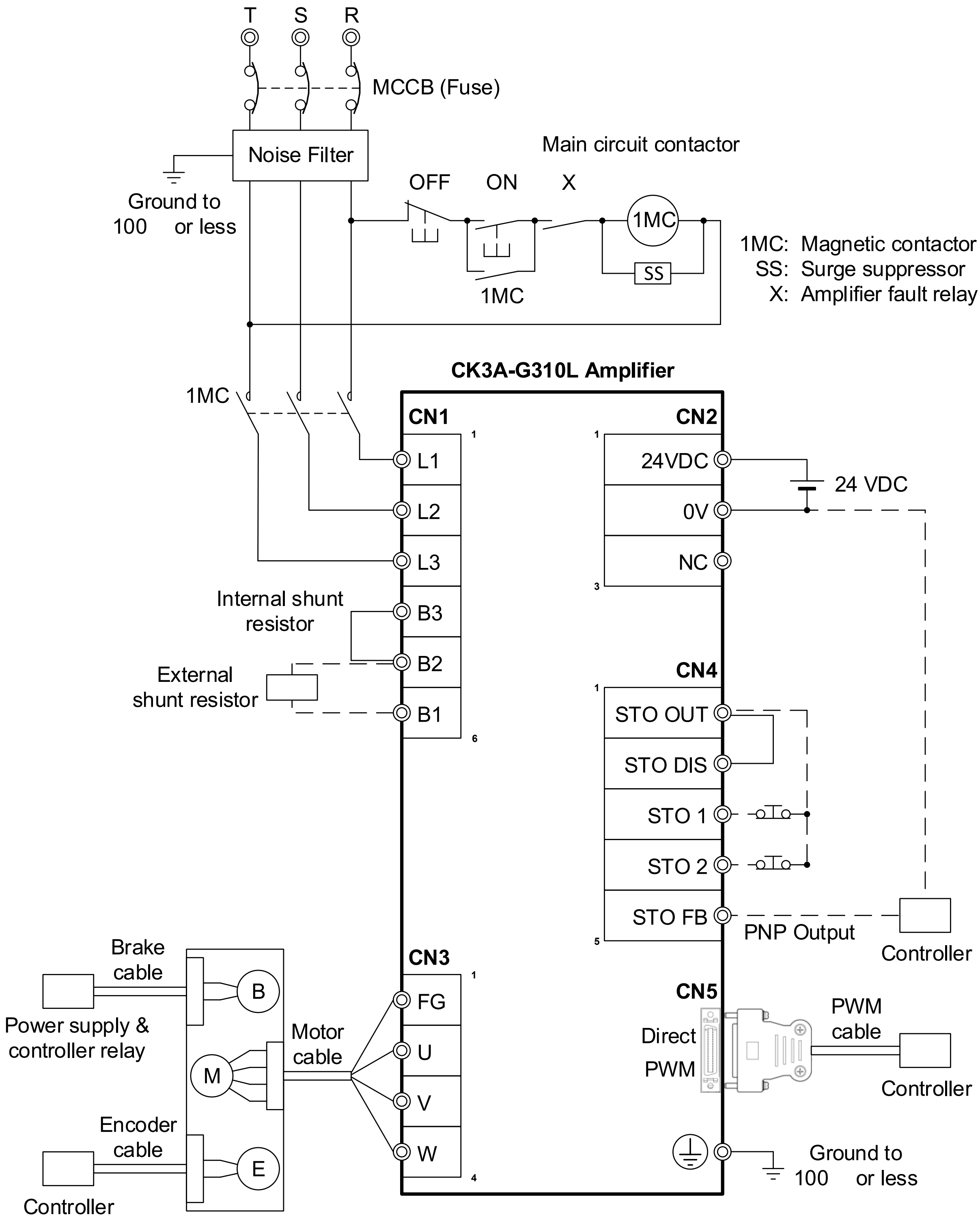
*1. From Amplifier perspective.

4-6 Overall Wiring Diagram Examples

4-6-1 CK3A-G305L



4-6-2 CK3A-G310L



4-6-3 Amplifier Fault Relay

If it is desired to disconnect the main power supply in case of an Amplifier fault, then the output signal that controls the relay coil must come from the Controller. This can be done in a PLC that mirrors the corresponding motor amplifier fault bit.

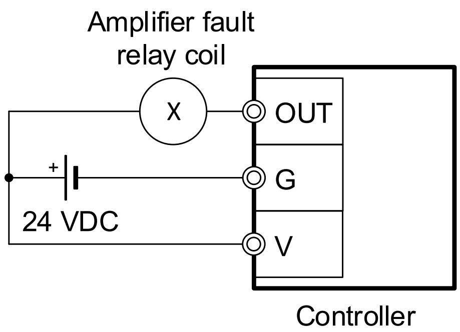
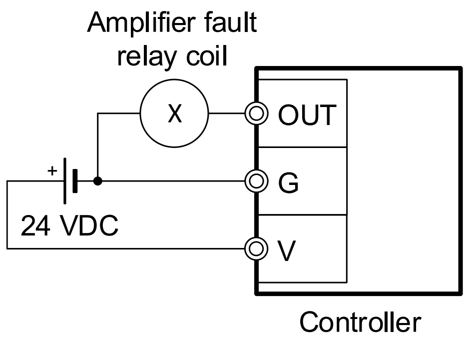
Sample PLC

```
PTR DigitalOutput1->Gate3[0].GpioData[0].16

OPEN PLC SamplePLC
DigitalOutput1 = Motor[1].AmpFault
CLOSE
```

Additional Information

The code line in the above PLC can be merged with an existing active PLC. It does not have to be its own PLC.

Wiring Example

NPN type output	PNP type output
	

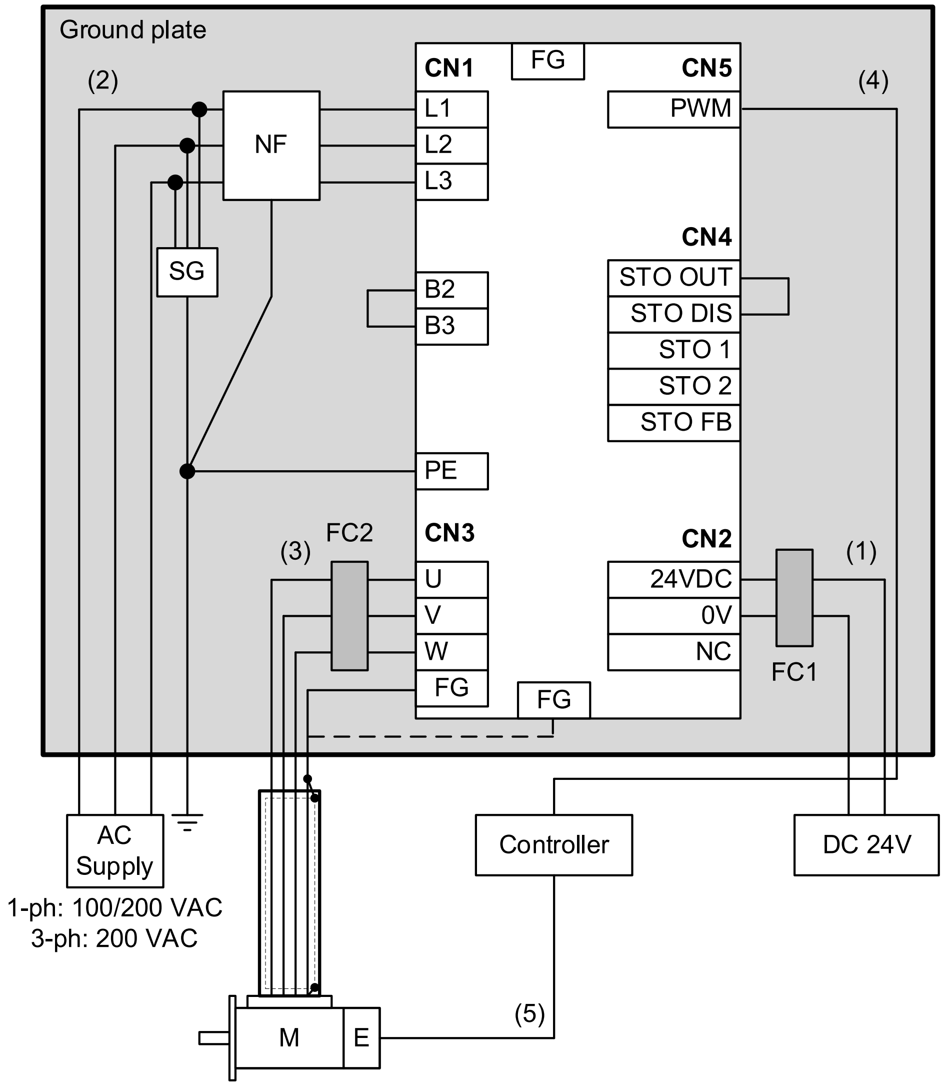
4-7 Wiring Conforming to EMC Directives

The CK3A Direct PWM Amplifiers conform to the EMC Directives (EN 61800-3) under the wiring conditions described in this section.
The following conditions are determined so that the CK3A series can conform to EMC Directives. When the products are installed in the equipment, the customer must perform the check to confirm that the overall machine conforms to EMC Directives.

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

- Install the Amplifier on the ground plate.
- Install a noise filter and lightening surge-absorbing element (surge absorber) on the power line.
- Use braided-shield cables for the I/O signals and encoder.
- Tinned soft steel wires must be used for the shields.
- Ground the shield of each cable.

4-7-1 Peripheral Equipment Connection Example



Note For single-phase inputs, connect between L1 and L2.
Note For CK3A-G305L, FG does not exist on Motor connector, use FG screw terminal instead.

- Provide single-point grounding of the ground plate for unit frame grounding as shown in the above diagram.
- Use a protective earth wire with a minimum thickness of 2.5 mm^2 and arrange the wiring so that the protective earth wire is as short as possible.
- Install a surge absorber and noise filter near the main circuit connector CN1 of the Amplifier.

Component Details

Symbol	Name	Manufacturer	Model	Amplifier Model	Remarks
SG	Surge Absorber	Soshin Electric Co., Ltd.	LT-C12G801WS	CK3A-G3□□L	1-ph 100/200VAC
			LT-C32G801WS		3-ph 200VAC
NF	Noise Filter	OMRON	R88A-FI1S105	CK3A-G305L	1-ph 100/200VAC
			R88A-FI1S208		3-ph 200VAC
		Schaffner	FN-series	CK3A-G310L	1-ph 100VAC
		OMRON	R88A-FI1S116		1-ph 200VAC
			R88A-FI1S216		3-ph 200VAC
FC1	Ferrite core	NEC TOKIN	ESD-SR-250	CK3A-G3□□L	-
FC2	Ferrite core	SEIWA ELECTRIC MFG	E04SR301334		-
M	Motor	-	-	-	-
E	Encoder	-	-	-	-
-	Controller	OMRON	CK3M-series	CK3A-G3□□L	W/ Direct PWM
			UMAC-series		W/ Direct PWM

Note Consult with third party manufacturer(s) or OMRON sales representative for detailed specifications.

Cable Information

No.	Interface	Ferrite core	Max. Length	Shielded
(1)	Logic power supply cable	2 turns	3 m	No
(2)	Main circuit power supply cable	-	3 m	No
(3)	Motor cable	2 turns	20 m	Yes
(4)	Direct PWM cable	-	3.6 m	-
(5)	Encoder cable	-	-	Yes

4-8 Selecting Peripheral Components

4-8-1 Main Circuit AC Power Supply

The main circuit AC power supply must be chosen in compliance with the specifications listed in the Specifications section of this manual.

4-8-2 Main Circuit DC Power Supply (Low Voltage Mode)

The main circuit DC power supply must be chosen in compliance with the specifications listed in the Specifications section of this manual. The following, are suggested (48VDC) power supplies.

Power Supply	Manufacturer
S8VK-G series	OMRON
S8FS-G series	OMRON
S8FS-C series	OMRON

Precautions for Correct Use

The choice of the DC power supply rating depends on the application requirements.

- Number of Amplifiers using the same power supply (total effective load)
- Transient, or acceleration current (Amps @48VDC) needed
- Steady state, idle or constant speed current (Amps @48VDC) needed

4-8-3 Logic Power Supply

The logic power supply must be chosen in compliance with the specifications listed in the Specifications section of this manual. The following, are suggested (24VDC) power supplies.

Power Supply	Manufacturer
S8VK-G series	OMRON
S8VK-S series	OMRON

4-8-4 Noise Filter

In AC voltage environment, line noise filters eliminate electromagnetic noise in a bi-directional manner (from and into the system). T type filters are not recommended. PI type filters are recommended.

- The noise filter should be mounted on the same panel as the drive and power source.
- The noise filter should be mounted as close as possible to the power source.
- The noise filter should be mounted as close as possible to incoming cabinet power.
- The noise filter must be chosen in compliance with the main power input specifications listed in the Specifications section of this manual.

The following, are suggested noise filters.

Amplifier Model	Voltage Input	Noise Filter	Manufacturer
CK3A-G305L	1-ph 100VAC	R88A-FI1S105	OMRON
	1-ph 200VAC	R88A-FI1S105	OMRON
	3-ph 200VAC	R88A-FI1S208	OMRON
CK3A-G310L	1-ph 100VAC	FN series	Schaffner
	1-ph 200VAC	R88A-FI1S116	OMRON
	3-ph 200VAC	R88A-FI1S216	OMRON

Note Contact your local OMRON sales representative for detailed specifications.

4-8-5 Voltage Surge Absorber

Surge absorbers protect from lightning surge voltage and abnormal voltage in the power input line. The following, are suggested surge absorbers.

Voltage Input	Surge Absorber	Manufacturer
1-ph 100VAC	LT-C12G801WS	Soshin Electric Co., Ltd.
1-ph 200VAC	LT-C12G801WS	Soshin Electric Co., Ltd.
3-ph 200VAC	LT-C32G801WS	Soshin Electric Co., Ltd.

Note Contact manufacturer for detailed specifications.

4-8-6 Molded Case Circuit Breaker (MCCB)

Select a molded case circuit breaker based on the maximum input current.

- The momentary maximum output of the Amplifier is approximately two times as much as the rated output, and the maximum output duration is two seconds.

  Therefore, select a molded case circuit breaker that can operate 10 seconds or more at 200% of the rated current.
- Select a molded case circuit breaker with a rated current larger than the sum of the effective load current (when multiple Amplifiers are used).
- When you select a molded case circuit breaker, remember to add the current consumption of other devices.

Recommended Fuse

Be sure to use a fuse which is a UL-listed product with LISTED and UL mark. Use copper wiring with a temperature rating of 75°C or higher. The following table shows recommended fuse.

Amplifier Model	Main Circuit Supply	Recommended Fuse
CK3A-G305L	3-Phase (170 – 252 VAC)	UL CLASS RK5 10 A
	1-Phase (85 – 252 VAC)	UL CLASS RK5 15 A
	1-Phase (44 – 60 VDC)	UL CLASS RK5 10 A
CK3A-G310L	3-Phase (170 – 252 VAC)	UL CLASS RK5 15 A
	1-Phase (85 – 252 VAC)	UL CLASS RK5 25 A

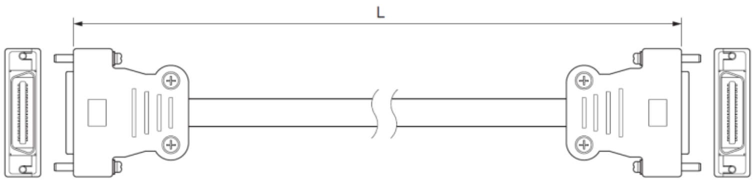
4-8-7 External Shunt Resistor

The external shunt resistor must be chosen in compliance with the specifications listed in the Specifications section of this manual. The following, are recommended external shunt resistors.

Amplifier Model	External Shunt Resistor	Manufacturer	Specifications
CK3A-G305L	R88A-RR30020	OMRON	60W, 20Ω
CK3A-G310L	R88A-RR30017	OMRON	60W, 17Ω

Note Contact your local OMRON sales representative for detailed specifications.

4-8-8 Direct PWM Cable



OMRON Model	Delta Tau Model	Length (L)
CK3W-CAAD009A	CABPWM-2 (200-602739-036x)	900 mm (36in)
CK3W-CAAD018A	CABPWM-4 (200-602739-072x)	1.8 m (72in)
CK3W-CAAD036A	CABPWM-6 (200-602739-144x)	3.6 m (144in)

4-8-9 Compatible Motors

The CK3A-series product line is capable of interfacing to a wide variety of motors. The Amplifier can control almost any type of three-phase Motor, including AC/DC brushless (synchronous) servo rotary or linear, and AC Induction (asynchronous) motors. Additionally, the amplifier can control permanent magnet DC brush motors such as a voice-coil actuator (using two of the amplifier’s three phases).

Motor sizing is outside of the scope of this manual. Motor sizing depends on a variety of application needs, and it is typically performed using proper sizing tools.

Having chosen Motor specifications and candidates, a quick comparison between the specification sheet of the Motor and that of the Amplifier reveals compatibility. Essentially, there are four main specification parameters to consider, shown in the table below.

Item	CK3A-G305L	CK3A-G310L
Rated voltage (main circuit input)	110 - 240VAC (85 - 252VAC)*1	
Rated current	Max. 5 A_{rms}	Max. 10 A_{rms}
Peak current	Max. 10 A_{rms} for 2sec	Max. 20 A_{rms} for 2 sec
Motor Inductance	Min. 1 mH	

*1. Rated values are outside of parentheses. Values in parentheses indicate the acceptable range.

Rated Voltage

Voltage is directly proportional to speed.

● Motor rated voltage less than or equal to the supply voltage

If the Motor rated voltage is less than or equal to the main circuit supply voltage, then it is possible to achieve the Motor rated speed (no issue here).

Precautions for Correct Use

- In this case, the Motor[x].PwmSf setting in the Controller should be equal to:
  16,384 * (Motor Voltage / Supply voltage).
- It is possible to increase by 10% if necessary to account for losses.

● Motor rated voltage greater than the supply voltage

If the Motor rated voltage is greater than the (maximum) main circuit supply voltage, then the Motor will function but the rated speed may not be achieved. In this case, consider whether the Motor rated speed is necessary for the application or not. If the rated speed is not necessary then the Motor can be compatible. If the rated speed is necessary, then the Motor is not compatible.

Additional Information

- Fundamentally, sufficient supply voltage and adequate current loop tuning should allow achieving Motor rated/top speeds. However, in certain applications, this may not be possible due to limitations such as feedback device, or excessive load.
- In general, the maximum Motor speed can be determined by input voltage line-to-line divided by the Motor’s back EMF constant. It is wise to de-rate this a little for proper servo applications.

Rated and Peak Current

Current is directly proportional to torque.

- Rated current typically correlates to steady state position holding or constant speed motion.
- Peak current typically correlates to transient (acceleration) motion.

● Motor rated current less than or equal to the Amplifier rated current

If the Motor rated current is less than or equal to the Amplifier rated current, then the Motor can be compatible (no issue here) and could achieve optimal steady state motion performance.

Precautions for Safe Use

I2T protection in the Controller must be set correctly to avoid burning the Motor in this case.

● Motor rated current greater than Amplifier rated current

If the Motor rated current is greater than the Amplifier rated current, then the Motor will function but it may not achieve optimal torque at steady state or constant motion.

Additional Information

It is fairly common to use slightly over-rated Motor in motion applications. In many cases, only a small percentage of the rated current is used during position holding or constant speed.

● Motor peak Current less than or equal to the Amplifier peak current

If the Motor peak current is less than or equal to the Amplifier peak current, then the Motor can be compatible (no issue here) and could achieve optimal transient (acceleration) motion performance.

Precautions for Safe Use

I2T protection in the Controller must be set correctly to avoid burning the Motor in this case.

● Motor peak current greater than the Amplifier peak current

If the Motor peak current is greater than the Amplifier peak current, then the Motor will function but it may not achieve optimal torque at transient (acceleration) motion.
In this case, consider whether top accelerations are necessary for the application or not. If top accelerations are not necessary, then the Motor can be compatible. If top accelerations are necessary, then the Motor is not compatible.

Additional Information

For time at peak, remember that maximum acceleration (peak current draw) profiles in motion applications are typically less than 500 milliseconds.

Motor Inductance

PWM outputs require significant Motor inductance to turn the on-off voltage signals into relatively smooth current flow with small ripple. Typically, Motor inductance of servomotors is 1 to 15 mH. The CK3A-series can drive this range easily.

For lower-inductance motors (below 1mH), problems could occur due to PWM switching where large ripple currents flow through the Motor, causing excessive energy waste and heating. If an application requires a Motor of less than 1mH, external inductors can be added to increase the inductance.

Motors with inductance in excess of 15mH can still be driven, but are slow to react and typically are out of the range of high performance servomotors.

5

Software Configuration

This section describes the software configuration of the CK3A Direct PWM Amplifier.

5-1 Outline of Software Configuration

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

5-1-1 Controller Amplifier Interface

The CK3A Direct PWM Amplifier interfaces with the controller via PWM cable. It does not communicate with the PC directly, and it does not contain any configuration settings.

Using a structure element in the Controller called ADC Strobe Word (explained in a later section), it is possible to control the Amplifier functions, and specify which status bits and data information are reported to the Controller.

5-1-2 Summary

The following topics are discussed in the Software Configuration section.

Topic	Content
Basic Configuration	Setup elements, including the “default” setting of the ADC Strobe Word allowing the basic configuration of the Amplifier
Details of the ADC Strobe Word	Explanation of the ADC Strobe Word including: • Control fields of Amplifier functions • Control fields of status bits • Control fields of data reporting
Details of the Amplifier Functions	Description and examples of how to: • Clear display errors • Configure the dynamic brake • Control the fan operation mode • Configure the discharge mode (residual energy at main power removal) • Set up for low voltage operation
Details of the ADC Registers	List of registers where the status bits and data information reside
Details of the Status Bits	List and addresses of all status bits
Details of the Data Information	List and addresses of data information
CK3A Sample PLCs	Sample code for 4 and 8 channels that allows the user to easily control the Amplifier(s) functions, read status bits and data information.

Additional Information

The Amplifier functions, status bits and data reporting are not necessary for the basic (default) operation of the CK3A Amplifier. The need of their use is dictated by the application.

5-1-3 Intended Users

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

5-1-4 Intended Hardware

The Software Configuration section pertains to the CK3M and UMAC series of the Power PMAC Controllers using the following axis cards.

- CK3M AX□3□3□
- ACC-24E3 3-4002A5□5□-□□□-□□□

5-1-5 Gate3 Alias Names

In the IDE script environment, the structure elements associated with CK3WAX[i], and ACC-24E3[i] are hardware alias names. They are the same as Gate3[i]. For simplicity, we will use the Gate3[i] nomenclature in all the examples of the Software Configuration section.

5-2 Basic Configuration

This section describes the minimum necessary settings that allow the basic configuration of the CK3A Direct PWM Amplifier.

Additional Information

- Motor and encoder setup are not discussed in this section. Refer to the Appendices section for that purpose.
- The settings discussed in this section must be saved to the Power PMAC memory (and IDE project) for proper startup of the CK3A Direct PWM Amplifier after a power cycle.

5-2-1 PWM Frequency

For the proper operation of the CK3A Direct PWM Amplifier, the PWM frequency must be in the [8 – 20] KHz range. The PWM frequency is dictated by the Power PMAC phase frequency (setting of Gate3[i].PhaseFreq) and PWM frequency (setting of Gate3[i].Chan[j].PwmFreqMult) which have the following relationship:

$$\text{PWM Frequency} = \frac{\text{Gate3[i].Chan[j].PwmFreqMult} + 1}{2} \times \text{Phase Frequency}$$

5-2-2 Gate3 Settings

Gate3[i].AdcAmpStrobe = $901001
Gate3[i].AdcAmpHeaderBits = 4
Gate3[i].AdcAmpClockDiv = 5

Precautions for Correct Use

If one of the four Gate3[i] amplifiers is a CK3A-G305L model that will be used in low voltage (48VDC) mode, set Gate3[i].AdcAmpStrobe = $B01001.

5-2-3 Channel Settings

CK3A-G305L

Gate3[i].Chan[j].PwmDeadTime = 2 / 0.0533
Gate3[i].Chan[j].PackInData = 0
Gate3[i].Chan[j].PackOutData = 0

CK3A-G310L

Gate3[i].Chan[j].PwmDeadTime = 3.1 / 0.0533
Gate3[i].Chan[j].PackInData = 0
Gate3[i].Chan[j].PackOutData = 0

5-2-4 Motor Settings

Motor[x].AdcMask = $FFFF0000
Motor[x].CurrentNullPeriod = 0

For PMAC I2T settings, the maximum ADC value for each model is shown below.

Model	Max ADC
CK3A-G305L	15.735
CK3A-G310L	31.470

Precautions for Safe Use

PMAC I2T settings, including Motor[x].I2tSet, Motor[x].I2tTrip and Motor[x].MaxDac must be set up correctly. Especially if the Motor is lower rated than the Amplifier. Not doing so could risk burning the Motor.

5-3 Details of the ADC Strobe Word

This section describes the details of the ADC Strobe Word which software settings allow the user to:

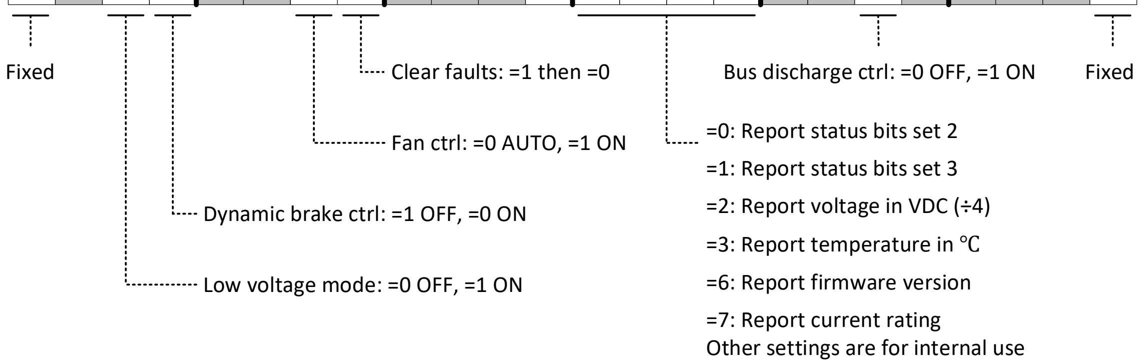
- Control the Amplifier functions
- Access the Amplifier status bits
- Access the Amplifier data information

The structure element of the ADC Strobe Word is Gate3[i].AdcAmpStrobe.

5-3-1 Overview

In the IDE script environment, the ADC Strobe Word is a 24-bit element comprised of the following fields related to the CK3A Direct PWM Amplifier:

Hex	9				0				1				0				0				1			
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	1	0	0	1	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1



5-3-2 Explicit Addresses

The Gate3[i].AdcAmpStrobe structure element cannot be used for bitwise (individual bit) mapping. If necessary, explicit addressing must be used instead.

Following, is a list of those addresses for the first eight Gate3[i] indexes, typically corresponding to the first 32 Amplifier channels.

Index i	Structure Element	Channels	Address
0	Gate3[0].AdcAmpStrobe	1 – 4	**$90020C**
1	Gate3[1].AdcAmpStrobe	5 – 8	**$90420C**
2	Gate3[2].AdcAmpStrobe	9 – 12	**$90820C**
3	Gate3[3].AdcAmpStrobe	13 – 16	**$90C20C**
4	Gate3[4].AdcAmpStrobe	17 – 20	**$91020C**
5	Gate3[5].AdcAmpStrobe	21 – 24	**$91420C**
6	Gate3[6].AdcAmpStrobe	25 – 28	**$91820C**
7	Gate3[7].AdcAmpStrobe	29 – 32	**$91C20C**

Additional Information

- Dollar ($) sign indicates hexadecimal value in PMAC script language.
- The explicit address is found by subtracting Sys.piom from Gate3[i].AdcAmpStrobe.a and reading the result in hexadecimal.

5-4 Details of the Amplifier Functions

This section describes the Amplifier functions which can be set from the Controller.

5-4-1 Features

The Amplifier functions, controlled by the ADC Strobe Word, allow the user to:

- Set the bus discharge mode
- Clear (reset) display errors
- Set the dynamic brake mode
- Set the fan operation mode
- Set low voltage mode

Additional Information

- The Amplifier functions are global to the four Amplifier channels on the same Gate3[i].
- The examples in this section can be used to find the desired setting of the ADC Strobe Word, which should then be saved in the IDE project file or configuration for subsequent power-ups.

5-4-2 Bus Discharge Control

Bit #5 of the ADC Strobe Word controls the discharge mode of the residual energy in the Amplifier when the main circuit power is removed.

= 0 Discharge OFF (Recommended)

If bit #5 of the ADC Strobe Word is set to 0, and when the main circuit power is removed, the Amplifier dissipates the residual energy slowly. This mode is recommended because of the short time allowed between main power cycles.

Precautions for Correct Use

In this mode, you should allow a minimum of 10 seconds between main power cycles.

Additional Information

Discharge of the residual energy (< 36VDC) in this mode takes approximately 5 minutes.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for bus discharge OFF from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFFFDF) | $0
```

= 1 Discharge ON

If bit #5 of the ADC Strobe Word is set to 1, and when the main circuit power is removed, the Amplifier discharges the residual energy quickly into the internal or external shunt resistor.

Precautions for Safe Use

If a Shunt resistor is not connected, the Amplifier will remain charged.

- The user must wait a minimum of 1 minute between main circuit power cycles for a successful full discharge every time.
- This function is primarily useful for repair if the unit needs to be opened and worked on safely.
- This function may be useful for some systems where safety mandates that all energy must be removed from the machine equipment during an emergency stop or maintenance.

Precautions for Correct Use

In this mode:

- The user must wait a minimum of 1 minute between main circuit power cycles.
- A Shunt Resistor (internal or external) must be connected for successful operation.

Additional Information

Discharge of the residual energy (< 36VDC) in this mode takes approximately 2 seconds.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for bus discharge ON from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFFFDF) | $20
```

5-4-3 Clear Fault

Bit #16 of the ADC Strobe Word clears the Amplifier display errors after fault condition has been removed.

Additional Information

- Clearing Soft start (A2), ADC offset (A9), and Short Circuit (AC) faults requires a logic power cycle of the Amplifier.
- This function only clears the Amplifier display errors. It does not clear the Amplifier fault (Motor[x].AmpFault) status in PMAC.
- There is no generic command to clear Amplifier fault status in PMAC, it is automatically cleared if the fault condition has been removed and the Motor is enabled.

Clearing Faults Operation

To clear display faults on any of the four Gate3[i] Amplifiers, set bit #16 of the ADC Strobe Word to 1 for a minimum of 10 milliseconds, then set it back to 0 for normal operation.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word bit #16 to 1 then 0 from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FEFFFF) | $10000
```

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FEFFFF) | $0
```

5-4-4 Fan Control

Bit #17 of the ADC Strobe Word controls the Amplifier fan operation.

= 0 Fan Auto

If bit #17 of the ADC Strobe Word is set to 0, the fan is set to automatic mode for all four Gate3[i] Amplifiers. It is ON when the power module temperature reaches 35℃, and OFF when it goes back down to 32℃.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for fan Auto mode from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FDFFFF) | $0
```

= 1 Fan Always ON

If bit #17 of the ADC Strobe Word is set to 1, the fan is always ON for all four Gate3[i] Amplifiers.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for fan always ON from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FDFFFF) | $20000
```

5-4-5 Dynamic Brake Control

Bit #20 of the ADC Strobe Word controls the dynamic brake function.

Precautions for Safe Use

- The dynamic brake function is intended for the stop at the time of an error and therefore it has a short-time rating. Do not use it to stop motion in normal operation.
- Do not drive the Motor by the use of an external drive source when the power supply is OFF and the dynamic brake function is ON.
- As a general guideline, allow a minimum of 1 minute between dynamic brake uses.
- Do NOT rely on the dynamic brake to stop a vertical Motor without counter-balance.
- The dynamic brake setting does not take effect if the Amplifier is enabled. The Amplifier enable signal must be OFF before setting the dynamic brake function ON/OFF.

The dynamic brake function, when it is ON, prevents the Motor from coasting freely after a KILL (servo OFF) event. This situation applies to the following.

- Motor killed (servo OFF) by user command or PMAC function such as fatal following error
- Amplifier error, including main circuit power OFF, or logic power OFF

= 0 Dynamic Brake ON

If bit #20 of the ADC Strobe Word is set to 0, dynamic brake is ON for all four Gate3[i] Amplifiers.

Precautions for Correct Use

- If the dynamic brake function is ON, the user must enable the Amplifier and wait a minimum of 50 milliseconds before commanding motion.
- If dynamic braking has been performed (after Motor KILL while in motion), the user must wait a minimum of 500 milliseconds before re-enabling the Amplifier.
- If the dynamic brake function is ON and the Motor is in motion, the user should issue a JOG/ first to decelerate the Motor to a stop before issuing a KILL (servo OFF) command.
- Do not toggle the dynamic brake setting ON/OFF when the Amplifier is enabled.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for dynamic brake function ON from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $EFFFFF) | $0
```

= 1 Dynamic Brake OFF

If bit #20 of the ADC Strobe Word is set to 1, dynamic brake is OFF for all four Gate3[i] Amplifiers.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for dynamic brake function OFF from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $EFFFFF) | $100000
```

5-4-6 Low Voltage (LV) Mode

Bit #21 of the ADC Strobe Word specifies whether the Amplifier will be used with low voltage (44 - 60 VDC) main circuit power supply or not.

- This mode is only available for the CK3A-G305L models.
- For the CK3A-G310L models, using this feature requires factory modification.
- If at least one of the four Gate3[i] Amplifiers will be used in the low voltage range, then the ADC Strobe Word must be set for low voltage mode because it is necessary. The other Amplifiers on the same Gate3[i] can still be used with AC input.

Additional Information

When low voltage mode is set, and main power (e.g. 48VDC) is applied, the Amplifier displays L on the left 7-segment display.

= 0 Normal Mode

If bit#21 of the ADC Strobe Word is set to 0, then none of the four Gate3[i] Amplifiers can operate in the low voltage range.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for normal mode from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $DFFFFF) | $0
```

= 1 Low Voltage Mode

If bit#21 of the ADC Strobe Word is set to 1, then any of the four Gate3[i] Amplifiers can operate in the 44 – 60VDC main power input range.

The following example, using Gate3 index 0 typically tied to Amplifier channels 1 – 4, shows a quick way to set the ADC Strobe Word for low voltage mode from the IDE terminal window:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $DFFFFF) | $200000
```

Precautions for Correct Use

The low voltage short-circuit wire on the main circuit power connector (CN1) must be installed for proper operation with low voltage range main power.

5-5 Details of the ADC Registers

This section describes the details of the ADC registers.

For each channel of the Power PMAC Controller (typically connected to one Amplifier), there are two ADC registers defined as follows.

Name	Structure Element
ADC **A**	Gate3[i].Chan[j].AdcAmp[**0**]
ADC **B**	Gate3[i].Chan[j].AdcAmp[**1**]

5-5-1 Information in ADC A

ADC A is a 32-bit register conveying the following information.

BIT(s)	Information in ADC A	ADC Strobe Word setting
7:0	Internal use	-
15:8	**Status bits set 1**	**Any “correct” strobe word setting**
31:16	Motor phase U current feedback	Any “correct” strobe word setting

5-5-2 Information in ADC B

ADC B is a 32-bit register conveying the following information.

BIT(s)	Information In ADC B	ADC Strobe Word setting
7:0	Internal use	-
15:08	**Status bits set 2**	**Bits [11:08] = 0**
	Status bits set 3	**Bits [11:08] = 1**
	DC bus voltage	**Bits [11:08] = 2**
	Power module temperature	**Bits [11:08] = 3**
	Amplifier firmware	**Bits [11:08] = 6**
	Amplifier current rating	**Bits [11:08] = 7**
31:16	Motor phase V current feedback	Any “correct” strobe word setting

Additional Information

All other settings of bits [11:08] report data used internally.

5-5-3 ADC Registers Explicit Addresses

The Gate3[i].Chan[j].AdcAmp[k] structure elements cannot be used for bitwise (individual bit) mapping. Explicit addressing must be used instead. Following, is a list of those addresses for the first eight Gate3[i] indexes, typically corresponding to the first 32 Amplifier channels:

	Ch.	ADC A Structure element	Address	ADC B Structure element	Address
Gate3[0]	1	Gate3[0].Chan[0].AdcAmp[0]	**$900020**	Gate3[0].Chan[0].AdcAmp[1]	**$900024**
	2	Gate3[0].Chan[1].AdcAmp[0]	**$9000A0**	Gate3[0].Chan[1].AdcAmp[1]	**$9000A4**
	3	Gate3[0].Chan[2].AdcAmp[0]	**$900120**	Gate3[0].Chan[2].AdcAmp[1]	**$900124**
	4	Gate3[0].Chan[3].AdcAmp[0]	**$9001A0**	Gate3[0].Chan[3].AdcAmp[1]	**$9001A4**
Gate3[1]	5	Gate3[1].Chan[0].AdcAmp[0]	**$904020**	Gate3[1].Chan[0].AdcAmp[1]	**$904024**
	6	Gate3[1].Chan[1].AdcAmp[0]	**$9040A0**	Gate3[1].Chan[1].AdcAmp[1]	**$9040A4**
	7	Gate3[1].Chan[2].AdcAmp[0]	**$904120**	Gate3[1].Chan[2].AdcAmp[1]	**$904124**
	8	Gate3[1].Chan[3].AdcAmp[0]	**$9041A0**	Gate3[1].Chan[3].AdcAmp[1]	**$9041A4**
Gate3[2]	9	Gate3[2].Chan[0].AdcAmp[0]	**$908020**	Gate3[2].Chan[0].AdcAmp[1]	**$908024**
	10	Gate3[2].Chan[1].AdcAmp[0]	**$9080A0**	Gate3[2].Chan[1].AdcAmp[1]	**$9080A4**
	11	Gate3[2].Chan[2].AdcAmp[0]	**$908120**	Gate3[2].Chan[2].AdcAmp[1]	**$908124**
	12	Gate3[2].Chan[3].AdcAmp[0]	**$9081A0**	Gate3[2].Chan[3].AdcAmp[1]	**$9081A4**
Gate3[3]	13	Gate3[3].Chan[0].AdcAmp[0]	**$90C020**	Gate3[3].Chan[0].AdcAmp[1]	**$90C024**
	14	Gate3[3].Chan[1].AdcAmp[0]	**$90C0A0**	Gate3[3].Chan[1].AdcAmp[1]	**$90C0A4**
	15	Gate3[3].Chan[2].AdcAmp[0]	**$90C120**	Gate3[3].Chan[2].AdcAmp[1]	**$90C124**
	16	Gate3[3].Chan[3].AdcAmp[0]	**$90C1A0**	Gate3[3].Chan[3].AdcAmp[1]	**$90C1A4**
Gate3[4]	17	Gate3[4].Chan[0].AdcAmp[0]	**$910020**	Gate3[4].Chan[0].AdcAmp[1]	**$910024**
	18	Gate3[4].Chan[1].AdcAmp[0]	**$9100A0**	Gate3[4].Chan[1].AdcAmp[1]	**$9100A4**
	19	Gate3[4].Chan[2].AdcAmp[0]	**$910120**	Gate3[4].Chan[2].AdcAmp[1]	**$910124**
	20	Gate3[4].Chan[3].AdcAmp[0]	**$9101A0**	Gate3[4].Chan[3].AdcAmp[1]	**$9101A4**
Gate3[5]	21	Gate3[5].Chan[0].AdcAmp[0]	**$914020**	Gate3[5].Chan[0].AdcAmp[1]	**$914024**
	22	Gate3[5].Chan[1].AdcAmp[0]	**$9140A0**	Gate3[5].Chan[1].AdcAmp[1]	**$9140A4**
	23	Gate3[5].Chan[2].AdcAmp[0]	**$914120**	Gate3[5].Chan[2].AdcAmp[1]	**$914124**
	24	Gate3[5].Chan[3].AdcAmp[0]	**$9141A0**	Gate3[5].Chan[3].AdcAmp[1]	**$9141A4**
Gate3[6]	25	Gate3[6].Chan[0].AdcAmp[0]	**$918020**	Gate3[6].Chan[0].AdcAmp[1]	**$918024**
	26	Gate3[6].Chan[1].AdcAmp[0]	**$9180A0**	Gate3[6].Chan[1].AdcAmp[1]	**$9180A4**
	27	Gate3[6].Chan[2].AdcAmp[0]	**$918120**	Gate3[6].Chan[2].AdcAmp[1]	**$918124**
	28	Gate3[6].Chan[3].AdcAmp[0]	**$9181A0**	Gate3[6].Chan[3].AdcAmp[1]	**$9181A4**
Gate3[7]	29	Gate3[7].Chan[0].AdcAmp[0]	**$91C020**	Gate3[7].Chan[0].AdcAmp[1]	**$91C024**
	30	Gate3[7].Chan[1].AdcAmp[0]	**$91C0A0**	Gate3[7].Chan[1].AdcAmp[1]	**$91C0A4**
	31	Gate3[7].Chan[2].AdcAmp[0]	**$91C120**	Gate3[7].Chan[2].AdcAmp[1]	**$91C124**
	32	Gate3[7].Chan[3].AdcAmp[0]	**$91C1A0**	Gate3[7].Chan[3].AdcAmp[1]	**$91C1A4**

Additional Information

- Dollar ($) sign indicates hexadecimal value in PMAC script language.
- The explicit address is found by subtracting Sys.piom from Gate3[i].Chan[j].AdcAmp[k].a and reading the result in hexadecimal.

5-6 Details of the Status Bits

This section describes the details of the three sets of status bits which can be reported by the CK3A Amplifier to the Controller.

5-6-1 Status Bits Set 1

For any “correct” setting of the ADC Strobe Word, the following status bits can be found in **ADC A**, **Gate3[i].Chan[j].AdcAmp[0]**:

Bit #	Item	Specification	Fault/Display Code
8	Main power	=0 no fault, =1 fault	A1
9	Soft start	=0 no fault, =1 fault	A2
10	STO status	=0 not triggered, =1 triggered	A6
11	Shunt short	=0 no fault, =1 fault	A3
12	Over voltage	=0 no fault, =1 fault	A5
13	I2T fault	=0 no fault, =1 fault	A8
14	Short circuit	=0 no fault, =1 fault	AC
15	Overload	=0 no fault, =1 fault	AL

Direct mapping access example using Gate3 index 0 channel 0 typically tied to Amplifier/Motor #1:

```
// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1
PTR Ck3a1PwrFlt          ->U.IO:$900020.8.1
PTR Ck3a1SoftStartFlt    ->U.IO:$900020.9.1
PTR Ck3a1StoStatus       ->U.IO:$900020.10.1
PTR Ck3a1ShuntShortFlt   ->U.IO:$900020.11.1
PTR Ck3a1OverVoltFlt     ->U.IO:$900020.12.1
PTR Ck3a1I2tFlt          ->U.IO:$900020.13.1
PTR Ck3a1ShortFlt        ->U.IO:$900020.14.1
PTR Ck3a1OverLoadFlt     ->U.IO:$900020.15.1
```

5-6-2 Status Bits Set 2

If bits [11:08] of the ADC Strobe Word are set to 0, the Amplifier reports status bits set 2 in **ADC B**, **Gate3[i].Chan[j].AdcAmp[1]**:

Bit #	Item	Specification	Fault/Display Code
8	Power input status	=1 AC power applied, =0 Removed	A1
10	Over temperature	=0 no fault, =1 fault	A7
11	Shunt overload	=0 no fault, =1 fault	A4
12	PWM frequency fault	=0 no fault, =1 fault	P1
13	PWM command fault	=0 no fault, =1 fault	P2
14	Amplifier enabled status	=0 not enabled, =1 enabled	0 (when enabled)

Additional Information

Other bit values are reserved for internal use.

Direct mapping access example using Gate3 index 0 channel 0 typically tied to Amplifier/Motor #1:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $0
```

```
// CH. 1, Gate3[0].Chan[0].AdcAmp[1] STATUS SET 2
PTR Ck3a1PwrStatus        ->U.IO:$900024.8.1
PTR Ck3a1OverTempFlt      ->U.IO:$900024.10.1
PTR Ck3a1ShuntOverLFlt    ->U.IO:$900024.11.1
PTR Ck3a1PwmFreqFlt       ->U.IO:$900024.12.1
PTR Ck3a1PwmCmdFlt        ->U.IO:$900024.13.1
PTR Ck3a1AmpEna           ->U.IO:$900024.14.1
```

5-6-3 Status Bits Set 3

If bits [11:08] of the ADC Strobe Word are equal to 1, the Amplifier reports status bits set 3 in **ADC B**, **Gate3[i].Chan[j].AdcAmp[1]**:

Bit #	Item	Specification	Fault/Display Code
10	I2T integrator status	=0 OFF, =1 ON	-
12	ADC offset	=0 no fault, =1 fault	A9
14	Amplifier Ready	=0 ready, =1 not ready	-

Additional Information

- Other bit values are reserved for internal use.
- The Amplifier Ready bit is set to 1 during initializing of the Amplifier (about 6 seconds). It is set to 0 when the boot up is complete indicating that the Amplifier is ready.

Direct mapping access example using Gate3 index 0 channel 0 typically tied to Amplifier/Motor #1:

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $100
```

```
// CH. 1, Gate3[0].Chan[0].AdcAmp[1] STATUS SET 3
PTR Ck3a1I2tint       ->U.IO:$900024.10.1
PTR Ck3a1AdcOffsetFlt ->U.IO:$900024.12.1
PTR Ck3a1Ready        ->U.IO:$900024.14.1
```

5-7 Details of the Amplifier Data

This section describes the details of the data which can be reported by the CK3A Amplifier to the Controller.

5-7-1 Overview and Examples

Bits [11:08] of the ADC Strobe Word also specify which Amplifier data is reported to the Controller. The reported data is found in ADC B bits [15:08].

Bit # [11:08] Setting	Amplifier Data Reported
= 2	Bus voltage (divided by 4) in VDC
= 3	Amplifier power module temperature in ℃
= 6	Amplifier Firmware version
= 7	Amplifier current rating in Amperes (5 for CK3A-G305L, 10 for CK3A-G310L)

The following, are examples for quick access from the IDE terminal window. Using Gate3 index 0 channel 0 typically tied to Amplifier/Motor #1.

Global definitions

```
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
```

DC Bus Voltage

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $200
```

```
Ck3a1DcBus = ((Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8) * 4
```

Power Module Temperature

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $300
```

```
Ck3a1Temp = (Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8
```

Amplifier Firmware Version

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $600
```

```
Ck3a1FwVer = (Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8
```

Current Rating of the Amplifier

```
Gate3[0].AdcAmpStrobe = (Gate3[0].AdcAmpStrobe & $FFF0FF) | $700
```

```
Ck3a1CurRating = (Gate3[0].Chan[0].AdcAmp[1] & $FF00) >> 8
```

5-8 CK3A Sample PLCs

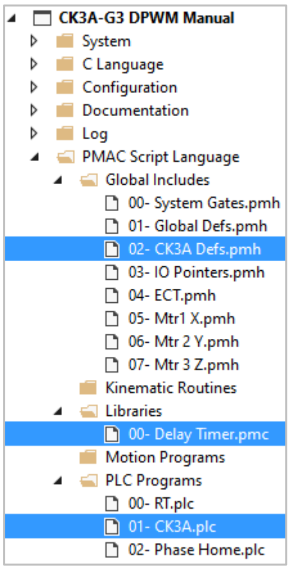
This section provides a Power PMAC script list of suggested variables, pointer definitions, and sample PLC programs for 4- and 8- motors allowing the user to easily:

- Control the Amplifier functions
- Read the Amplifier data
- Read the Amplifier status bits

5-8-1 Implementing the Sample PLC

The following steps are necessary to implement the sample PLC in this section successfully:

1. Make sure that the delay timer subroutine exists in the IDE project.
2. Make sure that the used Sys.Udata[i] structures do not conflict with the existing application.
3. Copy the **CK3A definitions** into a file under **Global Includes** section of the IDE project.
4. Copy the **CK3A PLC** into a file under **PLC programs** of the IDE project.
5. Enable the CK3A PLC on power-up (optional).
6. Make sure the download is successful (without compilation errors).



Additional Information

The name of the file in the IDE project tree is not important for operation.

5-8-2 Delay Timer Subroutine

The delay timer subroutine is used to introduce time delay in Power PMAC script PLC programs. This delay timer subroutine exists in most IDE project templates. It is shown here for reference.

```
OPEN SUBPROG DelayTimer
SUB: sec (DelayTimeSec)
LOCAL EndTimeSec
EndTimeSec = Sys.Time + DelayTimeSec
WHILE (EndTimeSec > Sys.Time){}
RETURN

SUB: msec (DelayTimeMsec)
LOCAL EndTimeMsec
EndTimeMsec = Sys.Time + DelayTimeMsec * 0.001
WHILE (EndTimeMsec > Sys.Time){}
RETURN
CLOSE
```

5-8-3 Explanation of the User-Shared Memory

Status bits set 1 is reported in the ADC A register of a given channel. These bits can be accessed at all time without any special setting of the ADC Strobe Word.

However, status bits sets 2, and 3 share the same location of the ADC B register. Bits [11:08] of the ADC Strobe settings dictate which set is currently being reported:

- Status set 2 is reported if bits [11:08] of the ADC Strobe Word are equal to 0
- Status set 3 is reported if bits [11:08] of the ADC Strobe Word are equal to 1

For this reason, we will use “mirror words” as holding registers for each time bits [11:08] setting is changed in the PLC (between 0 and 1).

The “unsigned” user-shared memory registers Sys.Udata[i] are best for this type of procedure. The registers used in the sample code in this section are defined as follows.

Status Set 2	Sys.Udata[i]	Address	Status Set 3	Sys.Udata[i]	Address
Channel 1	**Sys.Udata[256000]**	**$FA000**	Channel 1	**Sys.Udata[256040]**	**$FA0A0**
Channel 2	**Sys.Udata[256001]**	**$FA004**	Channel 2	**Sys.Udata[256041]**	**$FA0A4**
Channel 3	**Sys.Udata[256002]**	**$FA008**	Channel 3	**Sys.Udata[256042]**	**$FA0A8**
Channel 4	**Sys.Udata[256003]**	**$FA00C**	Channel 4	**Sys.Udata[256043]**	**$FA0AC**
Channel 5	**Sys.Udata[256004]**	**$FA010**	Channel 5	**Sys.Udata[256044]**	**$FA0B0**
Channel 6	**Sys.Udata[256005]**	**$FA014**	Channel 6	**Sys.Udata[256045]**	**$FA0B4**
Channel 7	**Sys.Udata[256006]**	**$FA018**	Channel 7	**Sys.Udata[256046]**	**$FA0B8**
Channel 8	**Sys.Udata[256007]**	**$FA01C**	Channel 8	**Sys.Udata[256047]**	**$FA0BC**

The proposed method in the PLC is as follows.

- Set bits [11:08] of the ADC Strobe Word = 0
- Copy status set 2 to Sys.Udata[25600x]
- Set bits [11:08] of the ADC Strobe Word = 1
- Copy status set 3 to Sys.Udata[25604x]

Mapping individual bits to Sys.Udata[i] registers will then reflect the status bits of each set.

Additional Information

- Make sure that these Sys.Udata[i] registers are not currently used in the application.
- Care must be taken if the register indexes are changed.
- The explicit address of the “unsigned” user shared memory registers can be found by subtracting sys.pushm from sys.Udata[i].a and reading the result in hexadecimal.

5-8-4 CK3A Sample Definitions 4-Motors

```
// GATE3[0], CH. 1 – 4 AMPLIFIER FUNCTIONS ====================== //
#define Gate30WpDis   Gate3[0].WpKey = $AAAAAAAA

GLOBAL Ck3a0Clrf, Ck3a0FanCtrl, Ck3a0BusDisCtrl, Ck3a0DbCtrl

PTR Gate30DataCtrl        ->U.IO:$90020C.16.4
PTR Ck3a0BusDisState      ->U.IO:$90020C.13.1
PTR Ck3a0ClrfBit          ->U.IO:$90020C.24.1
PTR Ck3a0FanState         ->U.IO:$90020C.25.1
PTR Ck3a0DbState          ->U.IO:$90020C.28.1

// CH. 1 – 4 ADC B REGISTERS [15:08] ============================ //
PTR Ck3a1AdcBData         ->U.IO:$900024.8.8  // Gate3[0].Chan[0].AdcAmp[1]
PTR Ck3a2AdcBData         ->U.IO:$9000A4.8.8  // Gate3[0].Chan[1].AdcAmp[1]
PTR Ck3a3AdcBData         ->U.IO:$900124.8.8  // Gate3[0].Chan[2].AdcAmp[1]
PTR Ck3a4AdcBData         ->U.IO:$9001A4.8.8  // Gate3[0].Chan[3].AdcAmp[1]

// CH. 1 – 4 AMPLIFIER DATA ===================================== //
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
GLOBAL Ck3a2DcBus, Ck3a2Temp, Ck3a2FwVer, Ck3a2CurRating
GLOBAL Ck3a3DcBus, Ck3a3Temp, Ck3a3FwVer, Ck3a3CurRating
GLOBAL Ck3a4DcBus, Ck3a4Temp, Ck3a4FwVer, Ck3a4CurRating

// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a1PwrFlt           ->U.IO:$900020.8.1
PTR Ck3a1SoftStartFlt     ->U.IO:$900020.9.1
PTR Ck3a1StoStatus        ->U.IO:$900020.10.1
PTR Ck3a1ShuntShortFlt    ->U.IO:$900020.11.1
PTR Ck3a1OverVoltFlt      ->U.IO:$900020.12.1
PTR Ck3a1I2tFlt           ->U.IO:$900020.13.1
PTR Ck3a1ShortFlt         ->U.IO:$900020.14.1
PTR Ck3a1OverLoadFlt      ->U.IO:$900020.15.1

// CH. 2, Gate3[0].Chan[1].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a2PwrFlt           ->U.IO:$9000A0.8.1
PTR Ck3a2SoftStartFlt     ->U.IO:$9000A0.9.1
PTR Ck3a2StoStatus        ->U.IO:$9000A0.10.1
PTR Ck3a2ShuntShortFlt    ->U.IO:$9000A0.11.1
PTR Ck3a2OverVoltFlt      ->U.IO:$9000A0.12.1
PTR Ck3a2I2tFlt           ->U.IO:$9000A0.13.1
PTR Ck3a2ShortFlt         ->U.IO:$9000A0.14.1
PTR Ck3a2OverLoadFlt      ->U.IO:$9000A0.15.1

// CH. 3, Gate3[0].Chan[2].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a3PwrFlt           ->U.IO:$900120.8.1
PTR Ck3a3SoftStartFlt     ->U.IO:$900120.9.1
PTR Ck3a3StoStatus        ->U.IO:$900120.10.1
PTR Ck3a3ShuntShortFlt    ->U.IO:$900120.11.1
PTR Ck3a3OverVoltFlt      ->U.IO:$900120.12.1
PTR Ck3a3I2tFlt           ->U.IO:$900120.13.1
PTR Ck3a3ShortFlt         ->U.IO:$900120.14.1
PTR Ck3a3OverLoadFlt      ->U.IO:$900120.15.1
```

```
// CH. 4, Gate3[0].Chan[3].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a4PwrFlt             ->U.IO:$9001A0.8.1
PTR Ck3a4SoftStartFlt       ->U.IO:$9001A0.9.1
PTR Ck3a4StoStatus          ->U.IO:$9001A0.10.1
PTR Ck3a4ShuntShortFlt      ->U.IO:$9001A0.11.1
PTR Ck3a4OverVoltFlt        ->U.IO:$9001A0.12.1
PTR Ck3a4I2tFlt             ->U.IO:$9001A0.13.1
PTR Ck3a4ShortFlt           ->U.IO:$9001A0.14.1
PTR Ck3a4OverLoadFlt        ->U.IO:$9001A0.15.1

// CH. 1, SYS.UDATA[256000] STATUS SET 2 ======================== //
PTR Ck3a1PwrStatus          ->U.USER:$FA000.0.1
PTR Ck3a1OverTempFlt        ->U.USER:$FA000.2.1
PTR Ck3a1ShuntOverLFlt      ->U.USER:$FA000.3.1
PTR Ck3a1PwmFreqFlt         ->U.USER:$FA000.4.1
PTR Ck3a1PwmCmdFlt          ->U.USER:$FA000.5.1
PTR Ck3a1AmpEna             ->U.USER:$FA000.6.1

// CH. 2, SYS.UDATA[256001] STATUS SET 2 ======================== //
PTR Ck3a2PwrStatus          ->U.USER:$FA004.0.1
PTR Ck3a2OverTempFlt        ->U.USER:$FA004.2.1
PTR Ck3a2ShuntOverLFlt      ->U.USER:$FA004.3.1
PTR Ck3a2PwmFreqFlt         ->U.USER:$FA004.4.1
PTR Ck3a2PwmCmdFlt          ->U.USER:$FA004.5.1
PTR Ck3a2AmpEna             ->U.USER:$FA004.6.1

// CH. 3, SYS.UDATA[256002] STATUS SET 2 ======================== //
PTR Ck3a3PwrStatus          ->U.USER:$FA008.0.1
PTR Ck3a3OverTempFlt        ->U.USER:$FA008.2.1
PTR Ck3a3ShuntOverLFlt      ->U.USER:$FA008.3.1
PTR Ck3a3PwmFreqFlt         ->U.USER:$FA008.4.1
PTR Ck3a3PwmCmdFlt          ->U.USER:$FA008.5.1
PTR Ck3a3AmpEna             ->U.USER:$FA008.6.1

// CH. 4, SYS.UDATA[256003] STATUS SET 2 ======================== //
PTR Ck3a4PwrStatus          ->U.USER:$FA00C.0.1
PTR Ck3a4OverTempFlt        ->U.USER:$FA00C.2.1
PTR Ck3a4ShuntOverLFlt      ->U.USER:$FA00C.3.1
PTR Ck3a4PwmFreqFlt         ->U.USER:$FA00C.4.1
PTR Ck3a4PwmCmdFlt          ->U.USER:$FA00C.5.1
PTR Ck3a4AmpEna             ->U.USER:$FA00C.6.1

// CH. 1, SYS.UDATA[256040] STATUS SET 3 ======================== //
PTR Ck3a1I2tint             ->U.USER:$FA0A0.2.1
PTR Ck3a1AdcOffsetFlt       ->U.USER:$FA0A0.4.1
PTR Ck3a1Ready              ->U.USER:$FA0A0.6.1

// CH. 2, SYS.UDATA[256041] STATUS SET 3 ======================== //
PTR Ck3a2I2tint             ->U.USER:$FA0A4.2.1
PTR Ck3a2AdcOffsetFlt       ->U.USER:$FA0A4.4.1
PTR Ck3a2Ready              ->U.USER:$FA0A4.6.1

// CH. 3, SYS.UDATA[256042] STATUS SET 3 ======================== //
PTR Ck3a3I2tint             ->U.USER:$FA0A8.2.1
PTR Ck3a3AdcOffsetFlt       ->U.USER:$FA0A8.4.1
PTR Ck3a3Ready              ->U.USER:$FA0A8.6.1

// CH. 4, SYS.UDATA[256043] STATUS SET 3 ======================== //
PTR Ck3a4I2tint             ->U.USER:$FA0AC.2.1
PTR Ck3a4AdcOffsetFlt       ->U.USER:$FA0AC.4.1
PTR Ck3a4Ready              ->U.USER:$FA0AC.6.1
```

5-8-5 CK3A Sample PLC 4-Motors

```
OPEN PLC Ck3aPLC
// CH. 1 - 4 STATUS BITS SET 2
Gate30WpDis Gate30DataCtrl = 0
CALL DelayTimer.msec(1)
Sys.Udata[256000] = Ck3a1AdcBData
Sys.Udata[256001] = Ck3a2AdcBData
Sys.Udata[256002] = Ck3a3AdcBData
Sys.Udata[256003] = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 STATUS BITS SET 3
Gate30WpDis Gate30DataCtrl = 1
CALL DelayTimer.msec(1)
Sys.Udata[256040] = Ck3a1AdcBData
Sys.Udata[256041] = Ck3a2AdcBData
Sys.Udata[256042] = Ck3a3AdcBData
Sys.Udata[256043] = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 DC BUS VOLTAGE
Gate30WpDis Gate30DataCtrl = 2
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1DcBus = Ck3a1AdcBData * 4
IF(Ck3a2Ready == 0) Ck3a2DcBus = Ck3a2AdcBData * 4
IF(Ck3a3Ready == 0) Ck3a3DcBus = Ck3a3AdcBData * 4
IF(Ck3a4Ready == 0) Ck3a4DcBus = Ck3a4AdcBData * 4
CALL DelayTimer.msec(1)

// CH. 1 - 4 TEMPERATURE
Gate30WpDis Gate30DataCtrl = 3
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1Temp = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2Temp = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3Temp = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4Temp = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 4 FIRMWARE VERSION
Gate30WpDis Gate30DataCtrl = 6
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1FwVer = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2FwVer = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3FwVer = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4FwVer = Ck3a4AdcBData
CALL  DelayTimer.msec(1)

// CH. 1 - 4 CURRENT RATING
Gate30WpDis Gate30DataCtrl = 7
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1CurRating = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2CurRating = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3CurRating = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4CurRating = Ck3a4AdcBData
CALL DelayTimer.msec(1)

// GATE3[0] CH. 1-4 BUS DISCHARGE CONTROL
IF(Ck3a0BusDisCtrl == 1 && Ck3a0BusDisState == 0) Gate30WpDis Ck3a0BusDisState = 1
IF(Ck3a0BusDisCtrl == 0 && Ck3a0BusDisState == 1) Gate30WpDis Ck3a0BusDisState = 0
```

```
// GATE3[0] CH. 1-4 FAN CTRL
IF(Ck3a0FanCtrl == 1 && Ck3a0FanState == 0) Gate30WpDis Ck3a0FanState = 1
IF(Ck3a0FanCtrl == 0 && Ck3a0FanState == 1) Gate30WpDis Ck3a0FanState = 0

// GATE3[0] CH1-4 DYNAMIC BRAKE CONTROL
IF(Ck3a0DbCtrl == 1 && Ck3a0DbState == 1) Gate30WpDis Ck3a0DbState = 0
IF(Ck3a0DbCtrl == 0 && Ck3a0DbState == 0) Gate30WpDis Ck3a0DbState = 1

// GATE3[0] CH. 1-4 CLEAR FAULT(S)
IF(Ck3a0Clrf == 1)
{
 Gate30WpDis Ck3a0ClrfBit = 1
 CALL DelayTimer.msec(10)
 Gate30WpDis Ck3a0ClrfBit = 0
 Ck3a0Clrf = 0
}

// CH. 1 LOGIC POWER OFF?
IF(Ck3a1Ready == 1) Ck3a1DcBus,4 = 0
IF(Ck3a2Ready == 1) Ck3a2DcBus,4 = 0
IF(Ck3a3Ready == 1) Ck3a3DcBus,4 = 0
IF(Ck3a4Ready == 1) Ck3a4DcBus,4 = 0
CLOSE
```

5-8-6 CK3A Sample Definitions 8-Motors

```
// GATE3[0], CH. 1 - 8 AMPLIFIER FUNCTIONS ====================== //
#define Gate30WpDis  Gate3[0].WpKey = $AAAAAAAA
#define Gate31WpDis  Gate3[1].WpKey = $AAAAAAAA

GLOBAL Ck3a0Clrf, Ck3a0FanCtrl, Ck3a0BusDisCtrl, Ck3a0DbCtrl
GLOBAL Ck3a1Clrf, Ck3a1FanCtrl, Ck3a1BusDisCtrl, Ck3a1DbCtrl

PTR Gate30DataCtrl    ->U.IO:$90020C.16.4
PTR Ck3a0BusDisState  ->U.IO:$90020C.13.1
PTR Ck3a0ClrfBit      ->U.IO:$90020C.24.1
PTR Ck3a0FanState     ->U.IO:$90020C.25.1
PTR Ck3a0DbState      ->U.IO:$90020C.28.1

PTR Gate31DataCtrl    ->U.IO:$90420C.16.4
PTR Ck3a1BusDisState  ->U.IO:$90420C.13.1
PTR Ck3a1ClrfBit      ->U.IO:$90420C.24.1
PTR Ck3a1FanState     ->U.IO:$90420C.25.1
PTR Ck3a1DbState      ->U.IO:$90420C.28.1

// CH. 1 - 8 ADC B REGISTERS [15:08] ============================ //
PTR Ck3a1AdcBData     ->U.IO:$900024.8.8  // Gate3[0].Chan[0].AdcAmp[1]
PTR Ck3a2AdcBData     ->U.IO:$9000A4.8.8  // Gate3[0].Chan[1].AdcAmp[1]
PTR Ck3a3AdcBData     ->U.IO:$900124.8.8  // Gate3[0].Chan[2].AdcAmp[1]
PTR Ck3a4AdcBData     ->U.IO:$9001A4.8.8  // Gate3[0].Chan[3].AdcAmp[1]

PTR Ck3a5AdcBData     ->U.IO:$904024.8.8  // Gate3[1].Chan[0].AdcAmp[1]
PTR Ck3a6AdcBData     ->U.IO:$9040A4.8.8  // Gate3[1].Chan[1].AdcAmp[1]
PTR Ck3a7AdcBData     ->U.IO:$904124.8.8  // Gate3[1].Chan[2].AdcAmp[1]
PTR Ck3a8AdcBData     ->U.IO:$9041A4.8.8  // Gate3[1].Chan[3].AdcAmp[1]

// CH. 1 - 8 AMPLIFIER DATA ===================================== //
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
GLOBAL Ck3a2DcBus, Ck3a2Temp, Ck3a2FwVer, Ck3a2CurRating
GLOBAL Ck3a3DcBus, Ck3a3Temp, Ck3a3FwVer, Ck3a3CurRating
GLOBAL Ck3a4DcBus, Ck3a4Temp, Ck3a4FwVer, Ck3a4CurRating

GLOBAL Ck3a5DcBus, Ck3a5Temp, Ck3a5FwVer, Ck3a5CurRating
GLOBAL Ck3a6DcBus, Ck3a6Temp, Ck3a6FwVer, Ck3a6CurRating
GLOBAL Ck3a7DcBus, Ck3a7Temp, Ck3a7FwVer, Ck3a7CurRating
GLOBAL Ck3a8DcBus, Ck3a8Temp, Ck3a8FwVer, Ck3a8CurRating

// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a1PwrFlt           ->U.IO:$900020.8.1
PTR Ck3a1SoftStartFlt     ->U.IO:$900020.9.1
PTR Ck3a1StoStatus        ->U.IO:$900020.10.1
PTR Ck3a1ShuntShortFlt    ->U.IO:$900020.11.1
PTR Ck3a1OverVoltFlt      ->U.IO:$900020.12.1
PTR Ck3a1I2tFlt           ->U.IO:$900020.13.1
PTR Ck3a1ShortFlt         ->U.IO:$900020.14.1
PTR Ck3a1OverLoadFlt      ->U.IO:$900020.15.1

// CH. 2, Gate3[0].Chan[1].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a2PwrFlt           ->U.IO:$9000A0.8.1
PTR Ck3a2SoftStartFlt     ->U.IO:$9000A0.9.1
PTR Ck3a2StoStatus        ->U.IO:$9000A0.10.1
PTR Ck3a2ShuntShortFlt    ->U.IO:$9000A0.11.1
PTR Ck3a2OverVoltFlt      ->U.IO:$9000A0.12.1
PTR Ck3a2I2tFlt           ->U.IO:$9000A0.13.1
PTR Ck3a2ShortFlt         ->U.IO:$9000A0.14.1
PTR Ck3a2OverLoadFlt      ->U.IO:$9000A0.15.1
```

```
// CH. 3, Gate3[0].Chan[2].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a3PwrFlt           ->U.IO:$900120.8.1
PTR Ck3a3SoftStartFlt     ->U.IO:$900120.9.1
PTR Ck3a3StoStatus        ->U.IO:$900120.10.1
PTR Ck3a3ShuntShortFlt    ->U.IO:$900120.11.1
PTR Ck3a3OverVoltFlt      ->U.IO:$900120.12.1
PTR Ck3a3I2tFlt           ->U.IO:$900120.13.1
PTR Ck3a3ShortFlt         ->U.IO:$900120.14.1
PTR Ck3a3OverLoadFlt      ->U.IO:$900120.15.1

// CH. 4, Gate3[0].Chan[3].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a4PwrFlt           ->U.IO:$9001A0.8.1
PTR Ck3a4SoftStartFlt     ->U.IO:$9001A0.9.1
PTR Ck3a4StoStatus        ->U.IO:$9001A0.10.1
PTR Ck3a4ShuntShortFlt    ->U.IO:$9001A0.11.1
PTR Ck3a4OverVoltFlt      ->U.IO:$9001A0.12.1
PTR Ck3a4I2tFlt           ->U.IO:$9001A0.13.1
PTR Ck3a4ShortFlt         ->U.IO:$9001A0.14.1
PTR Ck3a4OverLoadFlt      ->U.IO:$9001A0.15.1

// CH. 5, Gate3[1].Chan[0].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a5PwrFlt           ->U.IO:$904020.8.1
PTR Ck3a5SoftStartFlt     ->U.IO:$904020.9.1
PTR Ck3a5StoStatus        ->U.IO:$904020.10.1
PTR Ck3a5ShuntShortFlt    ->U.IO:$904020.11.1
PTR Ck3a5OverVoltFlt      ->U.IO:$904020.12.1
PTR Ck3a5I2tFlt           ->U.IO:$904020.13.1
PTR Ck3a5ShortFlt         ->U.IO:$904020.14.1
PTR Ck3a5OverLoadFlt      ->U.IO:$904020.15.1

// CH. 6, Gate3[1].Chan[1].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a6PwrFlt           ->U.IO:$9040A0.8.1
PTR Ck3a6SoftStartFlt     ->U.IO:$9040A0.9.1
PTR Ck3a6StoStatus        ->U.IO:$9040A0.10.1
PTR Ck3a6ShuntShortFlt    ->U.IO:$9040A0.11.1
PTR Ck3a6OverVoltFlt      ->U.IO:$9040A0.12.1
PTR Ck3a6I2tFlt           ->U.IO:$9040A0.13.1
PTR Ck3a6ShortFlt         ->U.IO:$9040A0.14.1
PTR Ck3a6OverLoadFlt      ->U.IO:$9040A0.15.1

// CH. 7, Gate3[1].Chan[2].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a7PwrFlt           ->U.IO:$904120.8.1
PTR Ck3a7SoftStartFlt     ->U.IO:$904120.9.1
PTR Ck3a7StoStatus        ->U.IO:$904120.10.1
PTR Ck3a7ShuntShortFlt    ->U.IO:$904120.11.1
PTR Ck3a7OverVoltFlt      ->U.IO:$904120.12.1
PTR Ck3a7I2tFlt           ->U.IO:$904120.13.1
PTR Ck3a7ShortFlt         ->U.IO:$904120.14.1
PTR Ck3a7OverLoadFlt      ->U.IO:$904120.15.1

// CH. 8, Gate3[1].Chan[3].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a8PwrFlt           ->U.IO:$9041A0.8.1
PTR Ck3a8SoftStartFlt     ->U.IO:$9041A0.9.1
PTR Ck3a8StoStatus        ->U.IO:$9041A0.10.1
PTR Ck3a8ShuntShortFlt    ->U.IO:$9041A0.11.1
PTR Ck3a8OverVoltFlt      ->U.IO:$9041A0.12.1
PTR Ck3a8I2tFlt           ->U.IO:$9041A0.13.1
PTR Ck3a8ShortFlt         ->U.IO:$9041A0.14.1
PTR Ck3a8OverLoadFlt      ->U.IO:$9041A0.15.1
```

```
// CH. 1, SYS.UDATA[256000] STATUS SET 2 ======================== //
PTR Ck3a1PwrStatus        ->U.USER:$FA000.0.1
PTR Ck3a1OverTempFlt      ->U.USER:$FA000.2.1
PTR Ck3a1ShuntOverLFlt    ->U.USER:$FA000.3.1
PTR Ck3a1PwmFreqFlt       ->U.USER:$FA000.4.1
PTR Ck3a1PwmCmdFlt        ->U.USER:$FA000.5.1
PTR Ck3a1AmpEna           ->U.USER:$FA000.6.1

// CH. 2, SYS.UDATA[256001] STATUS SET 2 ======================== //
PTR Ck3a2PwrStatus        ->U.USER:$FA004.0.1
PTR Ck3a2OverTempFlt      ->U.USER:$FA004.2.1
PTR Ck3a2ShuntOverLFlt    ->U.USER:$FA004.3.1
PTR Ck3a2PwmFreqFlt       ->U.USER:$FA004.4.1
PTR Ck3a2PwmCmdFlt        ->U.USER:$FA004.5.1
PTR Ck3a2AmpEna           ->U.USER:$FA004.6.1

// CH. 3, SYS.UDATA[256002] STATUS SET 2 ======================== //
PTR Ck3a3PwrStatus        ->U.USER:$FA008.0.1
PTR Ck3a3OverTempFlt      ->U.USER:$FA008.2.1
PTR Ck3a3ShuntOverLFlt    ->U.USER:$FA008.3.1
PTR Ck3a3PwmFreqFlt       ->U.USER:$FA008.4.1
PTR Ck3a3PwmCmdFlt        ->U.USER:$FA008.5.1
PTR Ck3a3AmpEna           ->U.USER:$FA008.6.1

// CH. 4, SYS.UDATA[256003] STATUS SET 2 ======================== //
PTR Ck3a4PwrStatus        ->U.USER:$FA00C.0.1
PTR Ck3a4OverTempFlt      ->U.USER:$FA00C.2.1
PTR Ck3a4ShuntOverLFlt    ->U.USER:$FA00C.3.1
PTR Ck3a4PwmFreqFlt       ->U.USER:$FA00C.4.1
PTR Ck3a4PwmCmdFlt        ->U.USER:$FA00C.5.1
PTR Ck3a4AmpEna           ->U.USER:$FA00C.6.1

// CH. 5, SYS.UDATA[256004] STATUS SET 2 ======================== //
PTR Ck3a5PwrStatus        ->U.USER:$FA010.0.1
PTR Ck3a5OverTempFlt      ->U.USER:$FA010.2.1
PTR Ck3a5ShuntOverLFlt    ->U.USER:$FA010.3.1
PTR Ck3a5PwmFreqFlt       ->U.USER:$FA010.4.1
PTR Ck3a5PwmCmdFlt        ->U.USER:$FA010.5.1
PTR Ck3a5AmpEna           ->U.USER:$FA010.6.1

// CH. 6, SYS.UDATA[256005] STATUS SET 2 ======================== //
PTR Ck3a6PwrStatus        ->U.USER:$FA014.0.1
PTR Ck3a6OverTempFlt      ->U.USER:$FA014.2.1
PTR Ck3a6ShuntOverLFlt    ->U.USER:$FA014.3.1
PTR Ck3a6PwmFreqFlt       ->U.USER:$FA014.4.1
PTR Ck3a6PwmCmdFlt        ->U.USER:$FA014.5.1
PTR Ck3a6AmpEna           ->U.USER:$FA014.6.1

// CH. 7, SYS.UDATA[256006] STATUS SET 2 ======================== //
PTR Ck3a7PwrStatus        ->U.USER:$FA018.0.1
PTR Ck3a7OverTempFlt      ->U.USER:$FA018.2.1
PTR Ck3a7ShuntOverLFlt    ->U.USER:$FA018.3.1
PTR Ck3a7PwmFreqFlt       ->U.USER:$FA018.4.1
PTR Ck3a7PwmCmdFlt        ->U.USER:$FA018.5.1
PTR Ck3a7AmpEna           ->U.USER:$FA018.6.1

// CH. 8, SYS.UDATA[256007] STATUS SET 2 ======================== //
PTR Ck3a8PwrStatus        ->U.USER:$FA01C.0.1
PTR Ck3a8OverTempFlt      ->U.USER:$FA01C.2.1
PTR Ck3a8ShuntOverLFlt    ->U.USER:$FA01C.3.1
PTR Ck3a8PwmFreqFlt       ->U.USER:$FA01C.4.1
PTR Ck3a8PwmCmdFlt        ->U.USER:$FA01C.5.1
PTR Ck3a8AmpEna           ->U.USER:$FA01C.6.1
```

```
// CH. 1, SYS.UDATA[256040] STATUS SET 3 ======================== //
PTR Ck3a1I2tint          ->U.USER:$FA0A0.2.1
PTR Ck3a1AdcOffsetFlt    ->U.USER:$FA0A0.4.1
PTR Ck3a1Ready           ->U.USER:$FA0A0.6.1

// CH. 2, SYS.UDATA[256041] STATUS SET 3 ======================== //
PTR Ck3a2I2tint          ->U.USER:$FA0A4.2.1
PTR Ck3a2AdcOffsetFlt    ->U.USER:$FA0A4.4.1
PTR Ck3a2Ready           ->U.USER:$FA0A4.6.1

// CH. 3, SYS.UDATA[256042] STATUS SET 3 ======================== //
PTR Ck3a3I2tint          ->U.USER:$FA0A8.2.1
PTR Ck3a3AdcOffsetFlt    ->U.USER:$FA0A8.4.1
PTR Ck3a3Ready           ->U.USER:$FA0A8.6.1

// CH. 4, SYS.UDATA[256043] STATUS SET 3 ======================== //
PTR Ck3a4I2tint          ->U.USER:$FA0AC.2.1
PTR Ck3a4AdcOffsetFlt    ->U.USER:$FA0AC.4.1
PTR Ck3a4Ready           ->U.USER:$FA0AC.6.1

// CH. 5, SYS.UDATA[256044] STATUS SET 3 ======================== //
PTR Ck3a5I2tint          ->U.USER:$FA0B0.2.1
PTR Ck3a5AdcOffsetFlt    ->U.USER:$FA0B0.4.1
PTR Ck3a5Ready           ->U.USER:$FA0B0.6.1

// CH. 6, SYS.UDATA[256045] STATUS SET 3 ======================== //
PTR Ck3a6I2tint          ->U.USER:$FA0B4.2.1
PTR Ck3a6AdcOffsetFlt    ->U.USER:$FA0B4.4.1
PTR Ck3a6Ready           ->U.USER:$FA0B4.6.1

// CH. 7, SYS.UDATA[256046] STATUS SET 3 ======================== //
PTR Ck3a7I2tint          ->U.USER:$FA0B8.2.1
PTR Ck3a7AdcOffsetFlt    ->U.USER:$FA0B8.4.1
PTR Ck3a7Ready           ->U.USER:$FA0B8.6.1

// CH. 8, SYS.UDATA[256047] STATUS SET 3 ======================== //
PTR Ck3a8I2tint          ->U.USER:$FA0BC.2.1
PTR Ck3a8AdcOffsetFlt    ->U.USER:$FA0BC.4.1
PTR Ck3a8Ready           ->U.USER:$FA0BC.6.1
```

5-8-7 CK3A Sample PLC 8-Motors

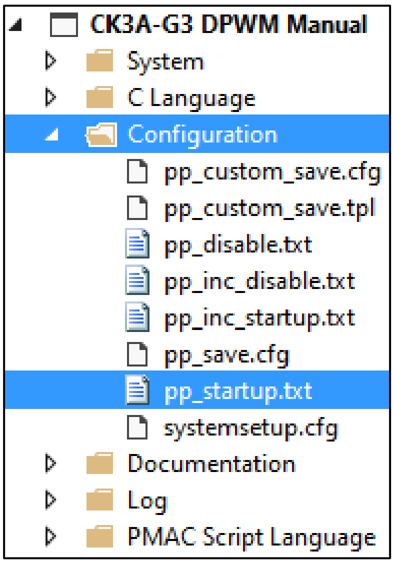
```
OPEN PLC Ck3aPLC
// CH. 1 - 8 STATUS BITS SET 2
Gate30WpDis Gate30DataCtrl = 0
Gate31WpDis Gate31DataCtrl = 0
CALL DelayTimer.msec(1)
Sys.Udata[256000] = Ck3a1AdcBData
Sys.Udata[256001] = Ck3a2AdcBData
Sys.Udata[256002] = Ck3a3AdcBData
Sys.Udata[256003] = Ck3a4AdcBData
Sys.Udata[256004] = Ck3a5AdcBData
Sys.Udata[256005] = Ck3a6AdcBData
Sys.Udata[256006] = Ck3a7AdcBData
Sys.Udata[256007] = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 8 STATUS BITS SET 3
Gate30WpDis Gate30DataCtrl = 1
Gate31WpDis Gate31DataCtrl = 1
CALL DelayTimer.msec(1)
Sys.Udata[256040] = Ck3a1AdcBData
Sys.Udata[256041] = Ck3a2AdcBData
Sys.Udata[256042] = Ck3a3AdcBData
Sys.Udata[256043] = Ck3a4AdcBData
Sys.Udata[256044] = Ck3a5AdcBData
Sys.Udata[256045] = Ck3a6AdcBData
Sys.Udata[256046] = Ck3a7AdcBData
Sys.Udata[256047] = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 8 DC BUS VOLTAGE
Gate30WpDis Gate30DataCtrl = 2
Gate31WpDis Gate31DataCtrl = 2
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1DcBus = Ck3a1AdcBData * 4
IF(Ck3a2Ready == 0) Ck3a2DcBus = Ck3a2AdcBData * 4
IF(Ck3a3Ready == 0) Ck3a3DcBus = Ck3a3AdcBData * 4
IF(Ck3a4Ready == 0) Ck3a4DcBus = Ck3a4AdcBData * 4
IF(Ck3a5Ready == 0) Ck3a5DcBus = Ck3a5AdcBData * 4
IF(Ck3a6Ready == 0) Ck3a6DcBus = Ck3a6AdcBData * 4
IF(Ck3a7Ready == 0) Ck3a7DcBus = Ck3a7AdcBData * 4
IF(Ck3a8Ready == 0) Ck3a8DcBus = Ck3a8AdcBData * 4
CALL DelayTimer.msec(1)

// CH. 1 - 8 TEMPERATURE
Gate30WpDis Gate30DataCtrl = 3
Gate31WpDis Gate31DataCtrl = 3
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1Temp = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2Temp = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3Temp = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4Temp = Ck3a4AdcBData
IF(Ck3a5Ready == 0) Ck3a5Temp = Ck3a5AdcBData
IF(Ck3a6Ready == 0) Ck3a6Temp = Ck3a6AdcBData
IF(Ck3a7Ready == 0) Ck3a7Temp = Ck3a7AdcBData
IF(Ck3a8Ready == 0) Ck3a8Temp = Ck3a8AdcBData
CALL DelayTimer.msec(1)
```

```
// CH. 1 - 8 FIRMWARE VERSION
Gate30WpDis Gate30DataCtrl = 6
Gate31WpDis Gate31DataCtrl = 6
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1FwVer = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2FwVer = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3FwVer = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4FwVer = Ck3a4AdcBData
IF(Ck3a5Ready == 0) Ck3a5FwVer = Ck3a5AdcBData
IF(Ck3a6Ready == 0) Ck3a6FwVer = Ck3a6AdcBData
IF(Ck3a7Ready == 0) Ck3a7FwVer = Ck3a7AdcBData
IF(Ck3a8Ready == 0) Ck3a8FwVer = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// CH. 1 - 8 CURRENT RATING
Gate30WpDis Gate30DataCtrl = 7
Gate31WpDis Gate31DataCtrl = 7
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1CurRating = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2CurRating = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3CurRating = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4CurRating = Ck3a4AdcBData
IF(Ck3a5Ready == 0) Ck3a5CurRating = Ck3a5AdcBData
IF(Ck3a6Ready == 0) Ck3a6CurRating = Ck3a6AdcBData
IF(Ck3a7Ready == 0) Ck3a7CurRating = Ck3a7AdcBData
IF(Ck3a8Ready == 0) Ck3a8CurRating = Ck3a8AdcBData
CALL DelayTimer.msec(1)

// GATE3[0] CH. 1-4 BUS DISCHARGE CONTROL
IF(Ck3a0BusDisCtrl == 1 && Ck3a0BusDisState == 0) Gate30WpDis Ck3a0BusDisState = 1
IF(Ck3a0BusDisCtrl == 0 && Ck3a0BusDisState == 1) Gate30WpDis Ck3a0BusDisState = 0
// GATE3[0] CH. 1-4 FAN CTRL
IF(Ck3a0FanCtrl == 1 && Ck3a0FanState == 0) Gate30WpDis Ck3a0FanState = 1
IF(Ck3a0FanCtrl == 0 && Ck3a0FanState == 1) Gate30WpDis Ck3a0FanState = 0
// GATE3[0] CH1-4 DYNAMIC BRAKE CONTROL
IF(Ck3a0DbCtrl == 1 && Ck3a0DbState == 1) Gate30WpDis Ck3a0DbState = 0
IF(Ck3a0DbCtrl == 0 && Ck3a0DbState == 0) Gate30WpDis Ck3a0DbState = 1

// GATE3[1] CH. 5-8 BUS DISCHARGE CONTROL
IF(Ck3a1BusDisCtrl == 1 && Ck3a1BusDisState == 0) Gate31WpDis Ck3a1BusDisState = 1
IF(Ck3a1BusDisCtrl == 0 && Ck3a1BusDisState == 1) Gate31WpDis Ck3a1BusDisState = 0
// GATE3[1] CH. 5-8 FAN CTRL
IF(Ck3a1FanCtrl == 1 && Ck3a1FanState == 0) Gate31WpDis Ck3a1FanState = 1
IF(Ck3a1FanCtrl == 0 && Ck3a1FanState == 1) Gate31WpDis Ck3a1FanState = 0
// GATE3[1] CH5-8 DYNAMIC BRAKE CONTROL
IF(Ck3a1DbCtrl == 1 && Ck3a1DbState == 1) Gate31WpDis Ck3a1DbState = 0
IF(Ck3a1DbCtrl == 0 && Ck3a1DbState == 0) Gate31WpDis Ck3a1DbState = 1

// GATE3[0] CH. 1-4 CLEAR FAULT(S)
IF(Ck3a0Clrf == 1)
{
 Gate30WpDis Ck3a0ClrfBit = 1
 CALL DelayTimer.msec(10)
 Gate30WpDis Ck3a0ClrfBit = 0
 Ck3a0Clrf = 0
}
```

```
// GATE3[1] CH. 5-8 CLEAR FAULT(S)
IF(Ck3a1Clrf == 1)
{
 Gate31WpDis Ck3a1ClrfBit = 1
 CALL DelayTimer.msec(10)
 Gate31WpDis Ck3a1ClrfBit = 0
 Ck3a1Clrf = 0
}

// CH. 1 - 8 LOGIC POWER OFF?
IF(Ck3a1Ready == 1) Ck3a1DcBus,4 = 0
IF(Ck3a2Ready == 1) Ck3a2DcBus,4 = 0
IF(Ck3a3Ready == 1) Ck3a3DcBus,4 = 0
IF(Ck3a4Ready == 1) Ck3a4DcBus,4 = 0
IF(Ck3a5Ready == 1) Ck3a5DcBus,4 = 0
IF(Ck3a6Ready == 1) Ck3a6DcBus,4 = 0
IF(Ck3a7Ready == 1) Ck3a7DcBus,4 = 0
IF(Ck3a8Ready == 1) Ck3a8DcBus,4 = 0
CLOSE
```

5-8-8 Enabling the CK3A PLC on Power-up

Enabling the CK3A PLC on power-up can be done using the startup.txt file under Configuration in the IDE project.



Additional Information

The ENABLE PLC command refers to the name of the PLC, not the file.

5-8-9 Using the CK3A PLC

Functions control

The suggested Amplifier functions structure looks like **Ck3a{i}{function name}** where:

- {i} is the Gate3 index
- {function name} is the desired function

Following, is a summary of the possible commands:

Precautions for Correct Use

The desired ADC Strobe Word resulting from function changes must be updated and saved in the main project file or configuration to apply in subsequent power cycles.

Ch.	Clear Errors	Dynamic Brake Control	Fan Control	Bus Discharge Control
Gate3[**0**] Ch. 1 – 4	Ck3a**0**Clrf=1	Ck3a**0**DbCtrl=1 for ON Ck3a**0**DbCtrl=0 for OFF	Ck3a**0**FanCtrl=1 ON Ck3a**0**FanCtrl=0 Auto	Ck3a**0**BusDisCtrl=1 ON Ck3a**0**BusDisCtrl=0 OFF
Gate3[**1**] Ch. 5 – 8	Ck3a**1**Clrf=1	Ck3a**1**DbCtrl=1 for ON Ck3a**1**DbCtrl=0 for OFF	Ck3a**1**FanCtrl=1 ON Ck3a**1**FanCtrl=0 Auto	Ck3a**1**BusDisCtrl=1 ON Ck3a**1**BusDisCtrl=0 OFF

Additional Information

The Ck3a[i]Clrf parameter is set back to 0 automatically by the PLC.

Precautions for Correct Use

Do not toggle the dynamic brake setting ON/OFF when the Amplifier is enabled.

Reading Data

The suggested Amplifier data structure looks like **Ck3a{x}{data name}** where:

- {x} is the Motor/Amplifier number
- {data name} is the desired data to be read

These variables can be placed in the IDE watch window for monitoring, used in PLC programs, or operator interface display. Following, are read examples.

Channel 1

Watch

Command/Query	Response
Ck3a1DcBus	294
Ck3a1Temp	25
Ck3a1FwVer	60
Ck3a1CurRating	5

Watch

Command/Query	Response
Ck3a1DcBus,4	294 23 60 5

Channels 1 – 4

Watch

Command/Query	Response
Ck3a1DcBus,4,4	294 298 300 294
Ck3a1Temp,4,4	23 28 25 24
Ck3a1FwVer,4,4	60 60 60 60
Ck3a1CurRating,4,4	5 5 5 10

Watch

Command/Query	Response
Ck3a1DcBus,4	294 23 60 5
Ck3a2DcBus,4	298 28 60 5
Ck3a3DcBus,4	300 25 60 5
Ck3a4DcBus,4	294 24 60 10

Additional Information

- The **{variable},{n}** syntax reports the next n consecutive variables.
- The **{variable},{n},{m}** syntax reports the next n variables separated by m variables.

Reading Status bits

The suggested Amplifier status structure looks like **Ck3a {x} {status name}** where:

- {x} is the Motor/Amplifier number
- {Status name} is the desired status to be read

These variables can be placed in the IDE watch window for monitoring, used in PLC programs, or operator interface display. Following, are read examples.

Channel 1

Watch

Command/Query	Response	
Ck3a1PwrFlt	0	Status Set 1
Ck3a1SoftStartFlt	0	
Ck3a1StoStatus	0	
Ck3a1ShuntShortFlt	0	
Ck3a1OverVoltFlt	0	
Ck3a1I2tFlt	0	
Ck3a1ShortFlt	0	
Ck3a1OverLoadFlt	0	
Ck3a1PwrStatus	1	Status Set 2
Ck3a1OverTempFlt	0	
Ck3a1ShuntOverLFlt	0	
Ck3a1PwmFreqFlt	0	
Ck3a1PwmCmdFlt	0	
Ck3a1AmpEna	1	
Ck3a1I2tint	0	Status Set 3
Ck3a1AdcOffsetFlt	0	
Ck3a1Ready	0	

Watch

Command/Query	Response	
Ck3a1PwrFlt,8	0 0 0 0 0 0 0 0	Set 1
Ck3a1PwrStatus,6	1 0 0 0 0 1	Set 2
Ck3a1I2tint,3	0 0 0	Set 3

Channels 1 – 4

Watch

Command/Query	Response	
Ck3a1PwrFlt,8	0 0 0 0 0 0 0 0	Ch. 1
Ck3a1PwrStatus,6	1 0 0 0 0 1	
Ck3a1I2tint,3	0 0 0	
Ck3a2PwrFlt,8	0 0 0 0 0 0 0 0	Ch. 2
Ck3a2PwrStatus,6	1 0 0 0 0 1	
Ck3a2I2tint,3	0 0 0	
Ck3a3PwrFlt,8	0 0 0 0 0 0 0 0	Ch. 3
Ck3a3PwrStatus,6	1 0 0 0 0 1	
Ck3a3I2tint,3	0 0 0	
Ck3a4PwrFlt,8	0 0 0 0 0 0 0 0	Ch. 4
Ck3a4PwrStatus,6	1 0 0 0 0 1	
Ck3a4I2tint,3	0 0 0	

Additional Information

The **{variable}{,n}** syntax reports the next n consecutive variables.

6

Troubleshooting

This section describes the LED indications, error display codes, and troubleshooting methods of the CK3A Direct PWM Amplifier.

6-1 LED Indicators

The following LED indicators are located on the front Amplifier. Below, is a description of their color designation and description.

Name	Indicator	Status	Color	Description
Logic Power	PWR	Lit	Green	24V logic power applied and within range (must be between 22 – 26.4VDC)
	PWR	Not lit	-	24V logic power not applied, or not within range (Amplifier OFF)
Amplifier Enabled	ENA	Lit	Green	Amplifier is enabled (e.g. OUT or JOG PMAC command, servo ON)
	ENA	Not lit	-	Amplifier is disabled (KILL PMAC command, servo OFF, or Amplifier OFF)
STO	STO	Lit	Green	Power drivers energized, normal operation (can servo Motor)
	STO	Lit	Red	Power drivers not energized, STO triggered or not disabled (cannot servo Motor)
	STO	Not lit	-	Amplifier is OFF, or internal failure (not normal if Amplifier is ON)
Shunt	SHU	Lit	Green	Shunt operation is active (chasing energy, or braking) (usually blinks for very short time during high load deceleration)
	SHU	Not lit	-	Shunt operation is not active (normal operation, no issue)
Bus	BUS	Lit	Red	Bus link charged (main circuit power applied, capacitors charged)
	BUS	Dim	Red	Bus link is dissipating energy (main circuit power removed)
	BUS	Very dim	-	Finished discharging, logic power is ON (always residual ~20V) Or Amplifier OFF for less than ~5 min
	BUS	Not lit	-	Amplifier has been OFF for longer than ~5 min

Additional Information

- The intensity of the BUS LED is directly proportional to the voltage level in the DC link
- If the logic power input is less than 22.0VDC or greater than 26.4VDC, the Amplifier logic power will turn OFF, and the STO LED will be lit (dim/red)

6-2 Display Codes

This section describes the error codes on the dual 7-segment display.

6-2-1 Normal Mode Operation

In normal mode operation, the Amplifier displays the following code.

Display	Fault	Description
	No fault	Amplifier OFF
-	No fault	Amplifier not enabled (Motor KILLED, servo OFF)
0.	No fault	Amplifier enabled (PMAC command OUT or closed loop like JOG, servo ON)
L -	No fault	Low voltage mode set (strobe word setting and LV short-circuit wire installed on CN1) Amplifier not enabled (Motor KILLED, servo OFF)
	Warning	If main power supply is AC and voltage drops to 80 – 100VDC (Amplifier is NOT disabled)
L0.	No fault	Low voltage mode set (strobe word setting and LV short-circuit wire installed on CN1) Amplifier enabled (PMAC command OUT or closed loop like JOG, servo ON)

6-2-2 Error Codes

In the event of a fault, the Amplifier displays error codes and turns OFF the power driver circuit.

Display	Fault	Description	Reset
P1	**PWM Frequency**	The detected PWM frequency is out of the permissible range [8 – 20] KHz	Clear Fault
P2	**PWM Command**	The PWM command stayed ON for too long causing the IPM transistors to conduct continuously	Clear Fault
P3	**ADC Clock**	The detected ADC clock frequency is out of the permissible range [2.450 – 6.250] MHz	Auto resets

Display	Fault	Description	Reset
A1	**Main Power Circuit**	The main circuit power input (AC or DC) is not present or removed	Auto resets
A2	**Soft Start Overload**	Too many main circuit power cycles without enough time in between	Logic power cycle
A3	**Shunt Short Circuit**	Short circuit detected in shunt circuitry	Clear Fault
A4	**Shunt Overload**	Exceeded maximum continuous time (2 seconds) of shunt or braking	Clear Fault
A5	**Over Voltage**	Internal bus voltage exceeded threshold (403VDC)	Clear Fault
A6	**STO input**	STO disable short-circuit wire not installed or STO input(s) removed	Enable/ Clear Fault
A7	**Over Temperature**	Power module temperature exceeded threshold (80℃)	Clear Fault
A8	**I2T**	Time allowed at 200% nominal output exceeded threshold (2 seconds)	Clear Fault
A9	**ADC Offset**	Current ADC offsets greater than allowed value (Amplifier firmware)	Logic power cycle
AC	**Short Circuit**	Time allowed at maximum power module output exceeded threshold (3 microseconds)	Clear Fault
AL	**Overload**	Time allowed at maximum specified output of Amplifier exceeded threshold (6 milliseconds)	Clear Fault

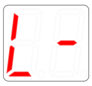
Precautions for Correct Use

- Allow 5 minutes to cool down after Soft Start A2 fault.
- Soft start (A2), ADC offset (A9), and short circuit (AC) faults require logic power cycle to clear display errors.
- Display errors can only be cleared when the fault condition is removed.

Additional Information

- Some display errors are reset automatically as soon as the fault condition is removed.
- Some errors can be cleared by enabling the Amplifier or clearing fault using the ADC Strobe Word.

6-2-3 Troubleshooting Error Codes

Display	Error and Troubleshooting
	**Amplifier OFF** • Check logic power supply and wiring, measure 24VDC input, make sure it is between 22.0 - 26.4VDC
	**Under Voltage Warning (if AC main power supply input)** • If in low voltage mode, this is normal indication • If main power supply is AC and voltage drops to 80 – 100VDC, warning (Amplifier is NOT disabled) • Check main circuit (AC) power supply and wiring
	**Main Power Supply Fault** • Check main circuit (AC or DC) power supply and wiring • If low voltage mode set, make sure low voltage short-circuit wire on CN1 is installed • If low voltage mode set, make sure ADC Strobe Word bit#21 (LV Mode) is set to 1
	**Soft Start Overload** • Reduce number of main circuit power cycling in machine operation • If discharge mode is OFF, make sure to allow minimum 10 seconds between main power cycles • If discharge mode is ON, make sure to allow minimum 1 minute between main power cycles • Allow 5 minutes to cool down after fault
	**Shunt Short Circuit** • If using internal shunt, make sure B2-B3 short-circuit (CN1) is correct wire gauge and installed properly • If using external shunt, check wiring connection to CN1 • Measure external shunt resistance, make sure it is not infinite • Make sure external shunt wires are not shorting with other wires in the panel
	**Shunt Overload** • A4 is also displayed when over voltage A5 fault occurs (this is normal) • If using external resistor, make sure the specifications comply with the recommended shunt resistor • Use the recommended shunt resistor • Measure shunt resistance, make sure it is not infinite • Try reducing Motor deceleration profile
	**Over Voltage** • If using internal shunt, make sure B2-B3 short-circuit (CN1) is correct wire gauge and installed properly • If using external shunt, use recommended resistor, check connection to CN1 • Measure shunt resistance, make sure it is not lower than expected/specification • Read reported voltage in the IDE, make sure it is within expected range • Try reducing Motor deceleration profile

Display	Error and Troubleshooting
A6	**STO Input** • The STO error display A6 is latched until cleared by ADC Strobe Word or Amplifier enabled (servo ON) • If not using the STO input, make sure STO disable short-circuit between CN4.1 and CN4.2 is installed • If using the STO input, make sure CN4.2 is open (floating) • If using the STO input, verify the wiring, measure 24VDC on CN4.1 • If there is no 24VDC on CN4.1, make sure logic power is ON • If using the STO input, measure 24VDC on STO1, and STO2 inputs • Check STO LED change when the CN4 connector is plugged/unplugged
A7	**Over Temperature** • Make sure operation environment is within specifications described in this manual • Make sure cabinet and mounting clearances are within specifications described in this manual • Make sure that the fan is not clogged with dust or debris • Check if the fan is operational, visual, feel air blowing • Toggle fan ON/OFF from IDE
A8	**I2T Integrated Current Fault** • Check Motor wiring for loose connection or intermittent short circuits • Make sure Motor is not pushing against hard surface constantly • If Motor has a brake, make sure it is disengaged during motion • If using CK3A-G305L, consider using CK3A-G310L for more power • Try reducing Motor acceleration/deceleration profile
A9	**ADC Offset** • Clear fault using ADC Strobe Word • Cycle logic power
AC	**Short Circuit** • Make sure not to toggle the dynamic brake ON/OFF setting if the Amplifier is enabled (servo ON) • Make sure Motor wiring is correct and using proper wire gauge • Check that none of the Motor leads are shorted or tied to ground • Measure Motor phase-phase resistances and make sure they are consistent • Try reducing Motor acceleration/deceleration profiles
AL	**Overload** • If the dynamic brake function is ON, do not KILL (servo OFF) Motor before issuing JOG/ and Motor comes to a stop • If the dynamic brake function is ON, make sure to wait a minimum of 50 milliseconds after enabling the Amplifier before commanding motion • If the dynamic brake function is ON, after the Motor is KILLED (servo OFF), make sure to wait a minimum of 500 milliseconds before servo-ing the Motor again • Make sure Motor wiring is using proper wire gauge and wired correctly • Check that the Motor chassis is grounded and does not have potential voltage • Measure Motor phase-phase resistances and make sure they are consistent • Make sure that the main circuit connection is using proper wire gauge and wired correctly

Display	Error and Troubleshooting
	**PWM frequency out of range** • Make sure the PWM frequency setting in the Controller is within range [8 – 20] KHz • Read PWM frequency reported by Amplifier in IDE software • Check PWM connection, makes sure connector is plugged in tightly • Try plugging the PWM cable in another servo channel (to see if it is a bad Controller card) • Try replacing the PWM cable • Check Motor wiring for loose connection or intermittent short circuits
	**PWM command (saturation) ON for too long** • Make sure the PWM frequency setting in the Controller is within range [8 – 20] KHz • Make sure that Motor[x].PwmSf is set correctly • Try reducing Motor acceleration profile
	**ADC clock out of range** • Make sure the ADC clock frequency setting in the Controller is within range [2.450 – 6.250] MHz • Check PWM cable and (loose) connection • Try plugging the PWM cable in another servo channel (to see if it is a bad Controller card) • Try replacing the PWM cable

Additional Information

If a fault condition persists and re-occurs after performing all the troubleshooting steps described above, then replace the Amplifier.

Precautions for Safe Use

The CK3A Direct PWM Amplifier does not error if a phase is lost in three-phase operation because it is designed to operate with single- or three-phase AC main power interchangeably.

6-2-4 Fault Thresholds

Item		Error	Threshold
Over voltage (A5)		A5	403 VDC
Shunt turn ON		-	390 VDC
Shunt turn OFF		-	370 VDC
Under Voltage Warning		L-	If AC main power supply and voltage is between 80-100VDC
Under voltage (A1) – Normal AC mode		A1	80 VDC
Under voltage (A1) – Low voltage mode		A1	43 VDC
Over temperature trip (A7)		A7	80 ℃
Over temperature release		-	70 ℃
Automatic fan turn ON		-	35 ℃
Automatic fan turn OFF		-	32 ℃
Overload (AL)	CK3A-G305L	AL	> 6 msec @ 15.55 A_{peak}
	CK3A-G310L		> 6 msec @ 28.85 A_{peak}
I2T integration start	CK3A-G305L	-	5.1 A_{rms}
	CK3A-G310L		10.1 A_{rms}
I2T trip time	CK3A-G305L	A8	> 2 seconds @ 5 A_{rms}
	CK3A-G310L		> 2 seconds @ 10 A_{rms}

6-2-5 Amplifier Operation Troubleshooting

Issue	Possible Root Cause	Checks and Measures
Error on 7-segment display	• Fault condition occurred • Amplifier damage	• Refer to Troubleshooting Error Codes section, remove fault condition, and reset display error
STO LED not lit	• Amplifier damage	
BUS LED not lit	• Main circuit supply is OFF • Amplifier damage / burned LED	• Refer to A1 error code troubleshooting
Amplifier does not turn ON (blank 7-segment display, PWR LED not lit)	• Logic power not supplied • Logic power out of range • Wire gauge too thin • Wiring • Amplifier damage	• Power supply is within specifications • Power supply is operational • Wiring uses the correct wire gauge • Wiring is correct
ENA LED not turning ON	• Not enabled (servo OFF) • Fault in Controller • Bad PWM cable • Fault condition occurred • Amplifier damage / burned LED	• Enable Amplifier (servo ON) • Motor[x].AmpEna must be 1 in PMAC • Replace PWM cable • Refer to Troubleshooting Error Codes section, remove fault condition, and reset display error
Bus does not dissipate in discharge mode	• Internal shunt resistor short-circuit wire not installed • External shunt resistor not installed • External shunt resistor used is not within specifications • ADC Strobe Word setting for discharge ON mode not set • Logic power removed too fast after main power removal	• Make sure CN1 short-circuit wire for internal shunt is using proper gauge, and wired correctly • Make sure external shunt resistor is installed and wired correctly • Use recommended Omron external shunt or shunt with correct specifications • Set ADC Strobe Word correctly for discharge ON, check status bit in ADC registers • After removing main power, wait minimum 5 seconds before removing logic power

Additional Information

When the Amplifier is ON, the STO LED must always be lit either red or green.

A

Appendices

This section contains appendices pertaining to the CK3A Direct PWM Amplifier.

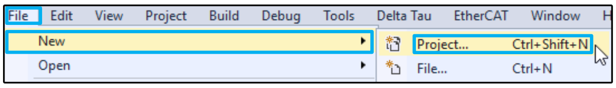
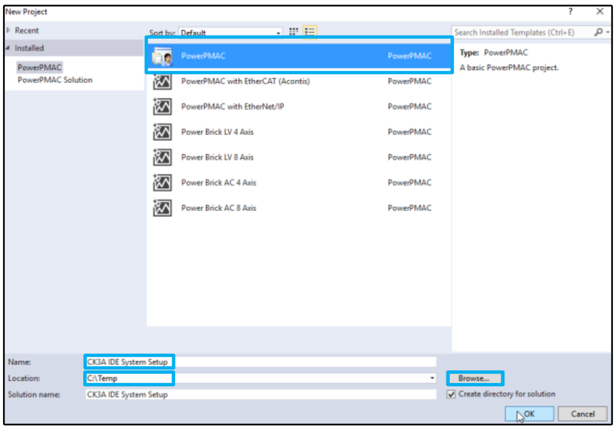
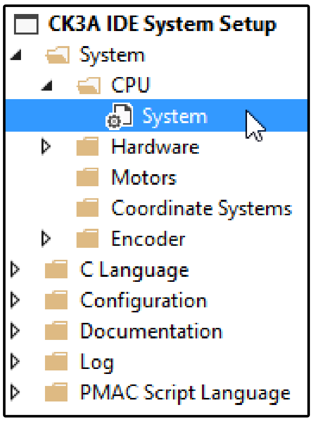
A-1 Automatic Motor Setup Using IDE

This section describes the steps for setting up a brushless servo Motor with the CK3A Direct PWM Amplifier using the IDE System Setup tool.

Configuration

Item	Notes
Controller	This example is applicable to CK3M AX, or UMAC ACC-24E3
Amplifier	CK3A-G305L (110VAC supply voltage)
Motor	Rotary Servo Motor, 200VAC, 2.4/10.2A, 5 pole pairs
Encoder	20-bit single-turn absolute (Panasonic protocol)

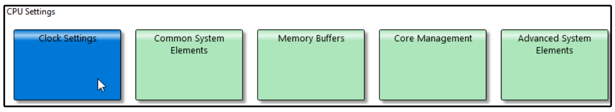
A-1-1 IDE System Setup

Steps	Illustration
Create a new IDE project • File >>> New >>> Project… • Choose template • Assign (type) project name • Choose (browse) PC folder location	 
The previous step creates a project tree in the IDE solution explorer window. • Double click on “System”	

Steps	Illustration

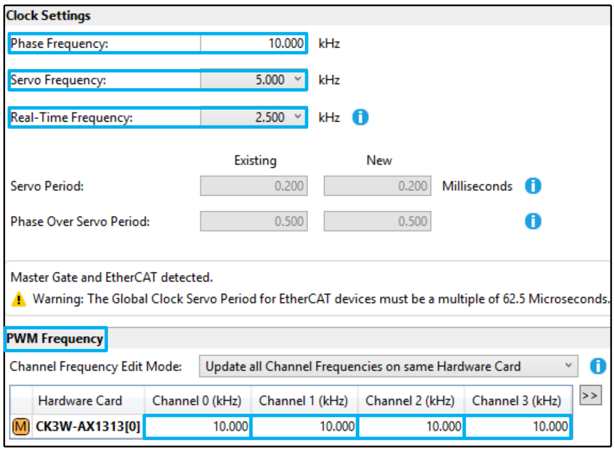
CPU (system and global) settings

- Click on “Clock Settings”

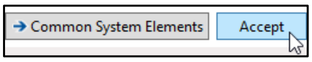


Clock Settings

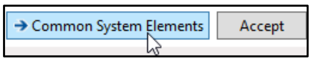
- Set phase frequency
- Set servo frequency
- Set real-time frequency
- Set PWM frequency.
  Must be [8–20KHz] for CK3A channel



- “Accept” when finished

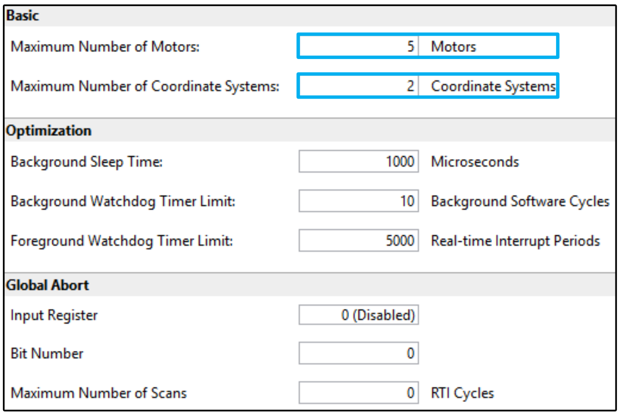


- “Common System Elements” for next step



Common System Elements

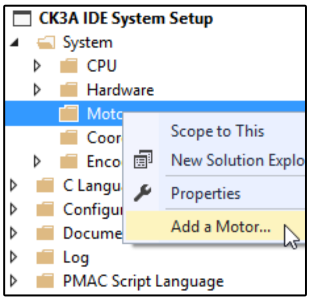
- Set highest No. of Motor + 1
  (to include Motor #0)
- Set highest No. of Coordinate System number + 1
  (to include Coordinate system 0)
- “Accept” when finished



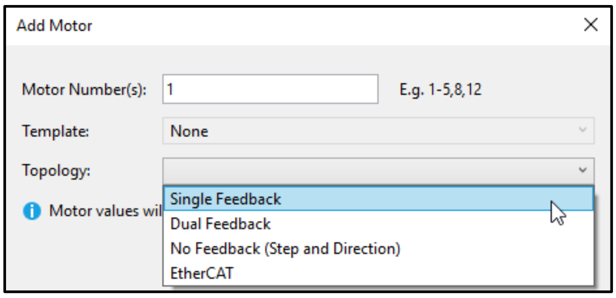
- Close System window, finished for now

Add a Motor

- Right click Motor folder in project tree
- Click “Add a Motor”



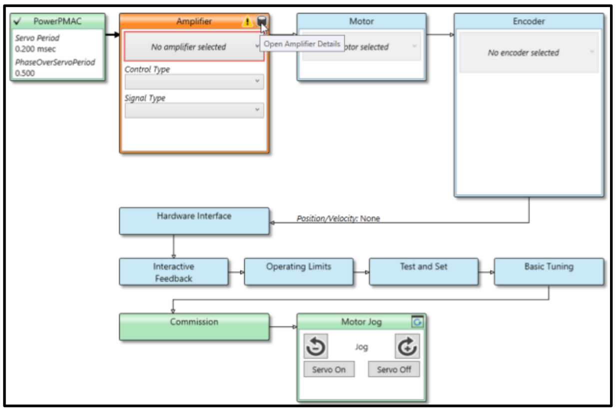
- Choose Motor number
- Choose topology (single feedback in this example)



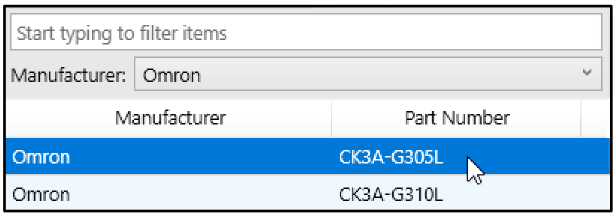
Steps	Illustration

Specify Amplifier

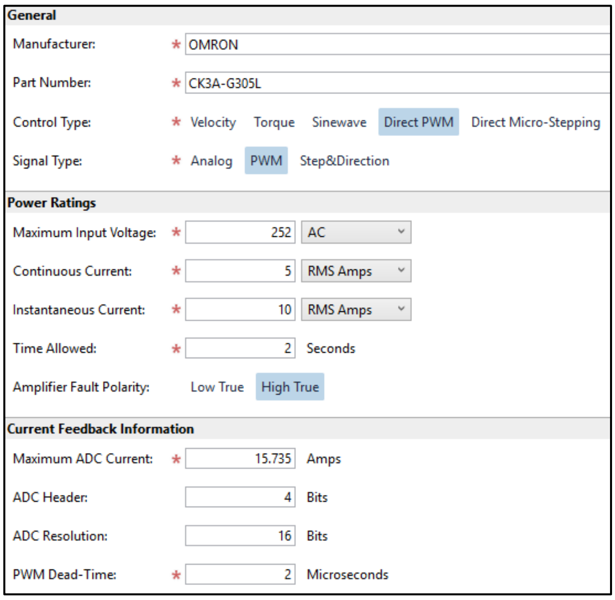
- Click on Amplifier details (database icon), or select default, or previously saved Amplifier from database (pulldown menu)



- Choose Manufacturer (OMRON and Delta Tau are defaults)
- Under OMRON, choose CK3A model

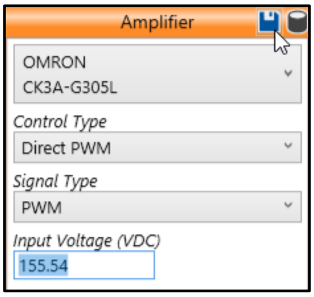


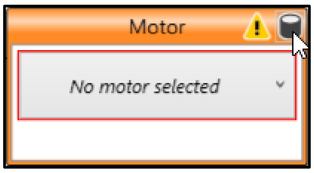
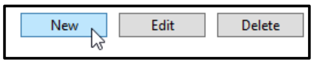
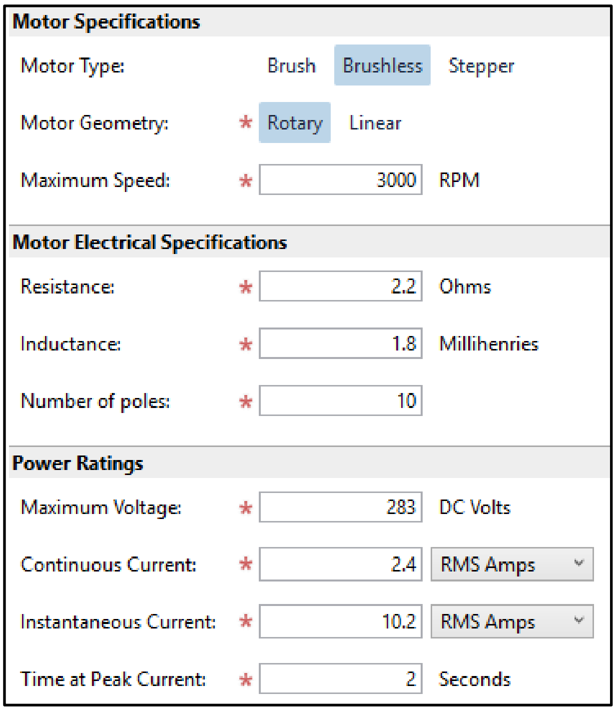
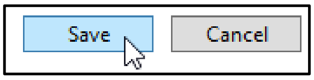
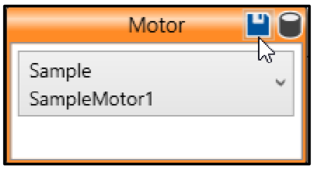
- Click on Select Amplifier.
  No need to make any changes here



- "Select Amplifier"

- Enter input (supply) voltage
  110VAC = ~155.54VDC in this example
- Click save (floppy disk icon)

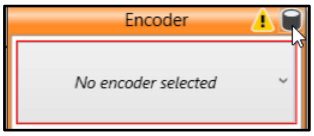


Steps	Illustration
Specify Motor • Click on Motor details (database icon), or select previously saved Motor from database (pulldown menu)	
• Click on "New" for new Motor	
• Enter "New" Motor information	
• "Save" to database	
• "Select Motor"	
• "Save" (database icon)	

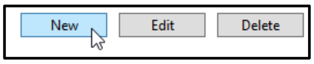
Steps	Illustration

Specify Encoder

- Click on encoder details (database icon), or select previously saved encoder from database (pulldown menu)

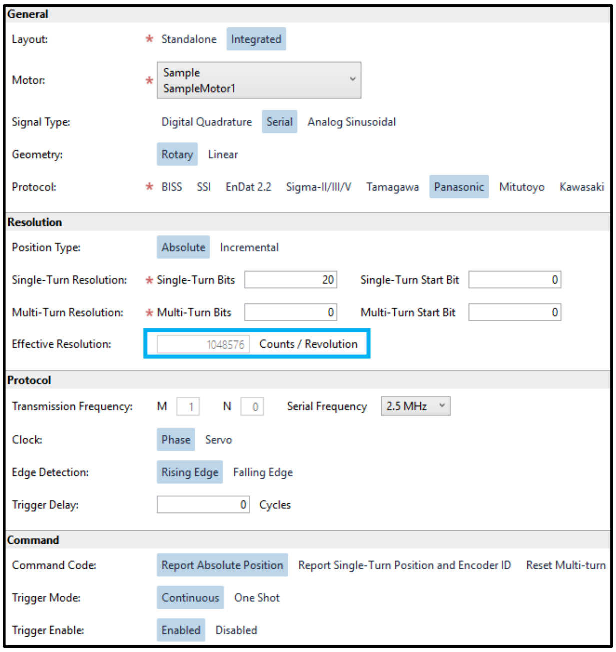


- Click on "New" for new Encoder

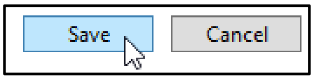


- Enter new Encoder information.

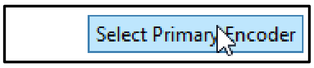
Note effective resolution result (for user units calculation in the next step)



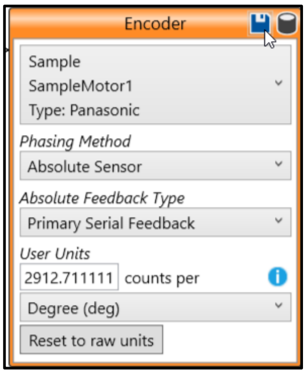
- "Save" to database



- "Select Primary Encoder"



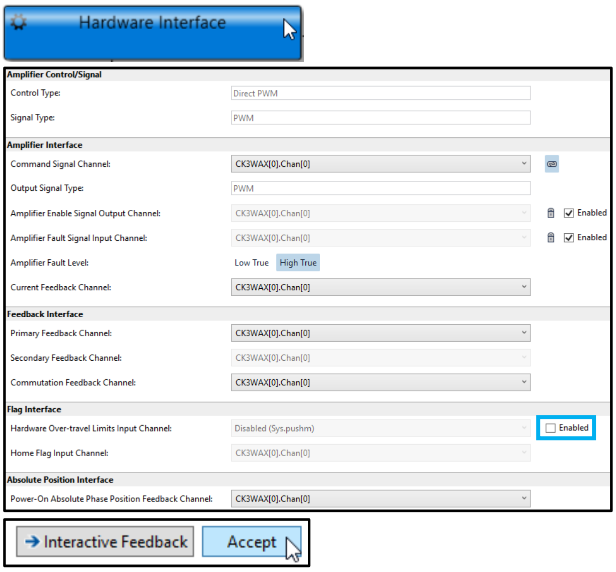
- Choose phasing method
- Choose feedback device for phasing method (if applicable)
- Choose axis engineering user units (e.g. degrees)
- Enter No. of counts per user unit.
  = 1048576/360 for degrees in this example
- Save when finished (floppy disk icon)



Steps	Illustration

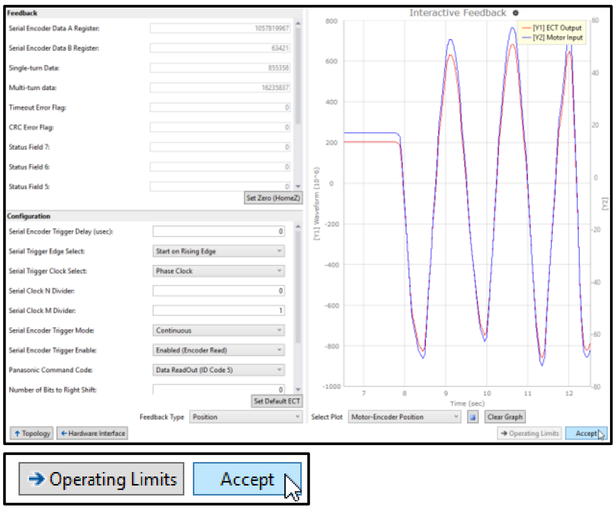
Hardware Interface

- Click on “Hardware Interface”
- Typically, the one item to consider in this page is the over-travel limits. They should be disabled if not wired.
- “Accept”
- “Interactive Feedback” for next step



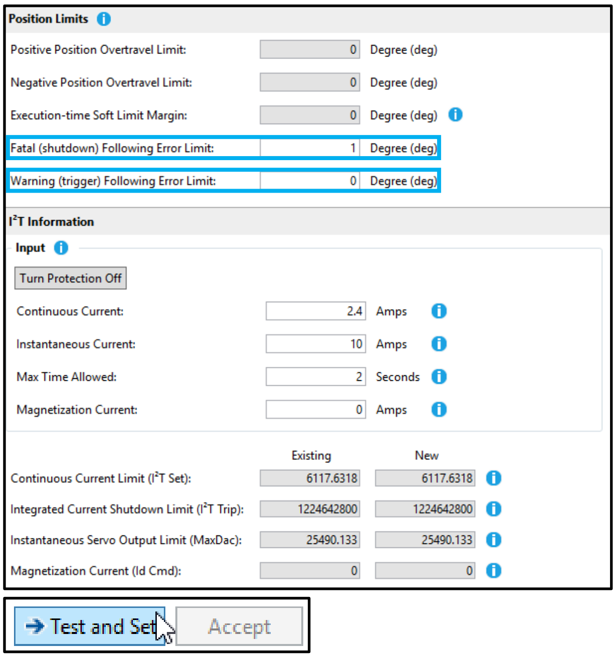
Interactive Feedback

- This page is useful for verifying encoder feedback. Typically, if possible and safe, Move Motor by hand and monitor position change.
- Generally, no changes are made on this page.
- “Accept”
- “Operating Limits” for next step



Operating Limits

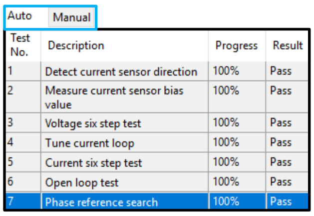
- Typically, the items to consider on this page are the “Fatal Following Error Limit” and “Warning Following Error Limit”.
- I2T settings are computed automatically
- It is NEVER recommended to turn I2T protection OFF, especially if the Motor is lower rated than the CK3A.
- “Accept”
- “Test and Set” for next step

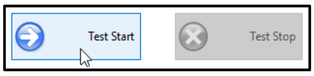


Steps	Illustration

Test and Set

- These tests will move the Motor, often at high rate of acceleration. Caution must be taken. Motor must be mounted and fixed firmly.
- User can choose “Auto” or “Manual”
- “Auto” will perform listed tests sequentially
- If a test fails during “Auto” mode, it automatically switches “Manual”.
- In “Manual” mode, the user can choose whichever test to perform.

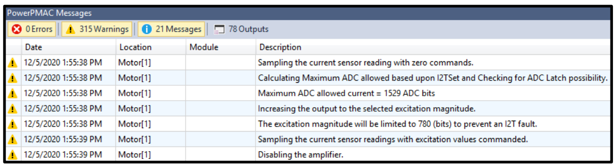




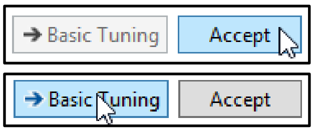
- During the tests, the structure element settings are shown in the output window. This could be useful in troubleshooting exercises

Step No.	PowerPMAC Command	Value
3	Motor[1].PhaseOffset	683
3	Motor[1].PwmSf	13337
3	Motor[1].PhasePosSf	2048/(4096 * 1 * 209715.200000)

- Additionally, more detailed messages can be monitored in the Power PMAC Messages window. This could be useful in troubleshooting exercises

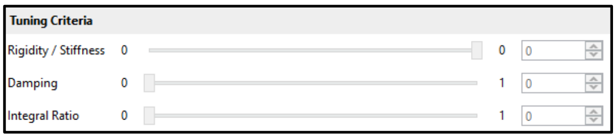


- All tests must pass before continuing
- “Accept”
- “Basic Tuning”

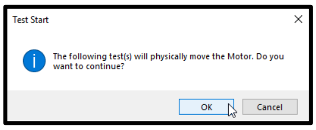


Basic (Auto) Tuning

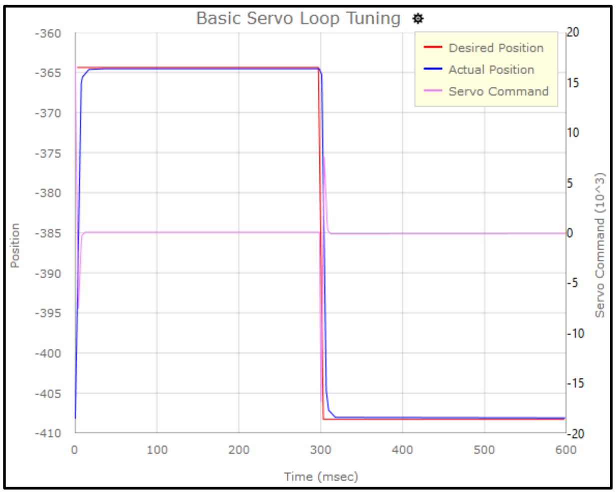
- This is an “Auto-Tuning” tool without initial user input
- “Start Tuning” to initiate the process



- During this identification and tuning process, the Motor will move, often at high rate of acceleration. Caution must be taken. Motor must be mounted and fixed firmly.



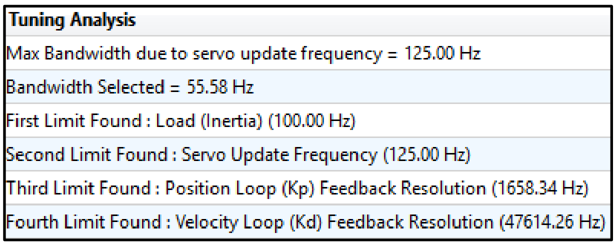
- The result shows a plot of a step move response



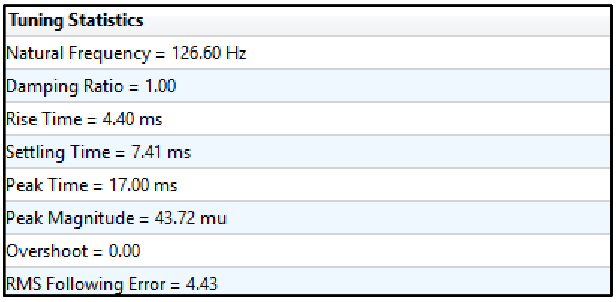
Steps	Illustration

- The result shows criteria chosen by the Auto-tuning algorithm. It is possible to modify these parameters (e.g. 0 integral ratio) and “Re-Tune”

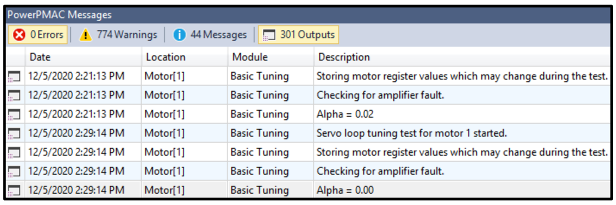
- The result shows a tuning analysis. This explains the choice of bandwidth and some of the system limitations



- The result shows statistics of the step move response

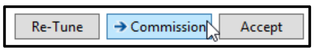


- Note The Power PMAC Messages Outputs window provides information during the Auto-tuning process



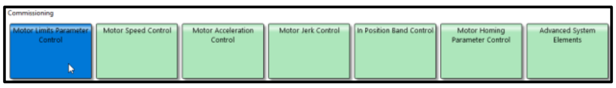
- When finished, “Accept”

- “Commission” for next step



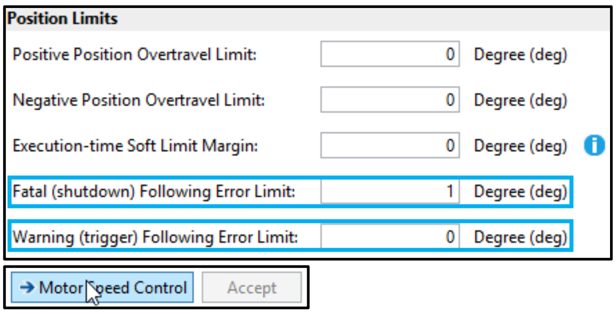
Commissioning

- “Motor Limits Parameter control” to start the commissioning



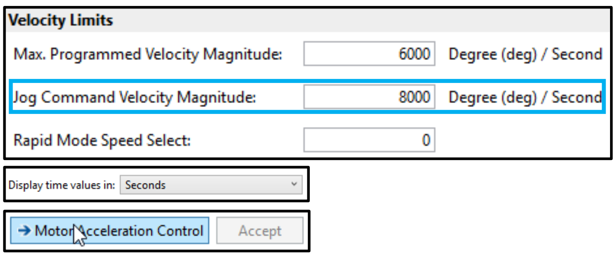
Position Limits

- Typically, confirming fatal, and warning following error limits is the main consideration here
- Positive and negative soft can be set for absolute encoders after establishing absolute homing parameters. For incremental encoders, the soft limits are typically set after performing a homing routine.
- “Accept” and “Motor Speed Control” for next item



Velocity Limits

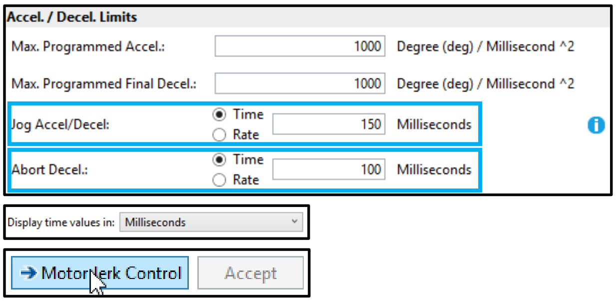
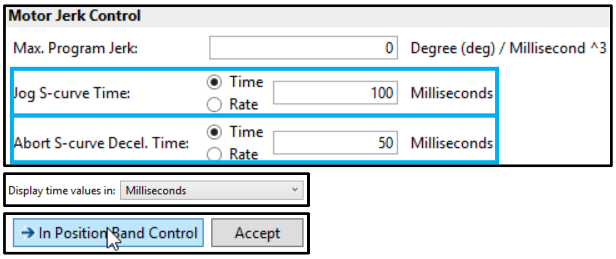
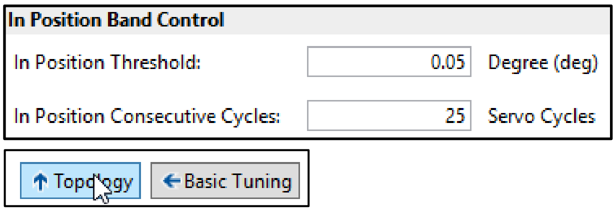
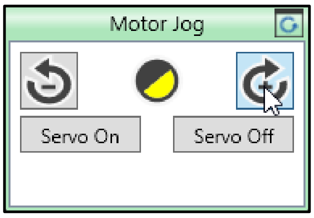
- Generally, the main consideration here is Jog speed (magnitude). Max programmed velocity is usually configured for motion programs in conjunction with coordinate system settings.
- User input time unit is selectable.



A-1 Automatic Motor Setup Using IDE

A

A-1-1 IDE System Setup

Steps	Illustration
Acceleration/Deceleration Limits • Generally, the main consideration here is Jog and Abort linear acceleration/deceleration time or rate Max programmed accel/decel are usually configured for motion programs in conjunction with coordinate system settings • “Accept” and “Motor Jerk Control” for next item	
Motor Jerk Control • Generally, the main consideration here is Jog and Abort linear acceleration/deceleration time or rate Max programmed accel/decel are usually configured for motion programs in conjunction with coordinate system settings • “Accept” and “In Position Band Control” for next item	
Motor In Position Band Control • The “In Position Threshold” defines the following error “window” held for “In Position Consecutive Cycles” amount of time before the Motor[x].InPos bit is set to true • This concludes the essential commissioning • “Topology” to exit	
Motor Jog • This control panel is an optional test (Jog) run	

A-2 Power PMAC3 Manual Motor Setup

This section describes the step-by-step procedure for manually setting up a brushless Motor with a Power PMAC Controller with Gate3, ACC-84E, and CK3A Direct PWM Amplifier. The procedure steps are shown below.

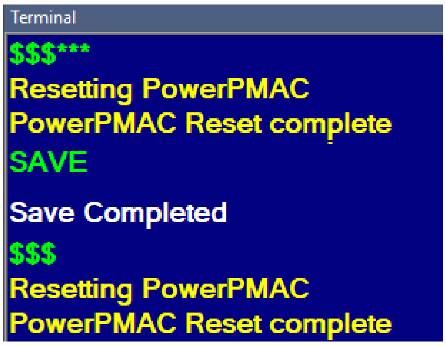
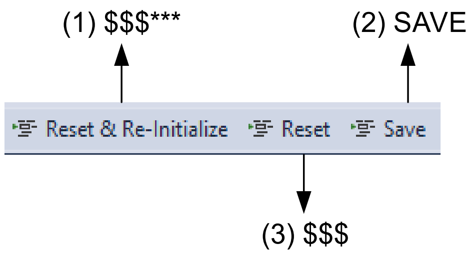
1. Creating an IDE project
2. Basic optimization, system and gates settings
3. Encoder on-going position, verification
4. Motor setup and commissioning

A-2-1 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 $$$***, followed by a SAVE, and a normal reset $$$.

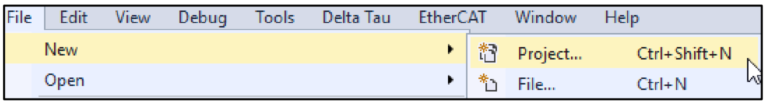
This can be done either from the terminal window, or using the IDE toolbar shortcuts.

Terminal Window	IDE Toolbar
	

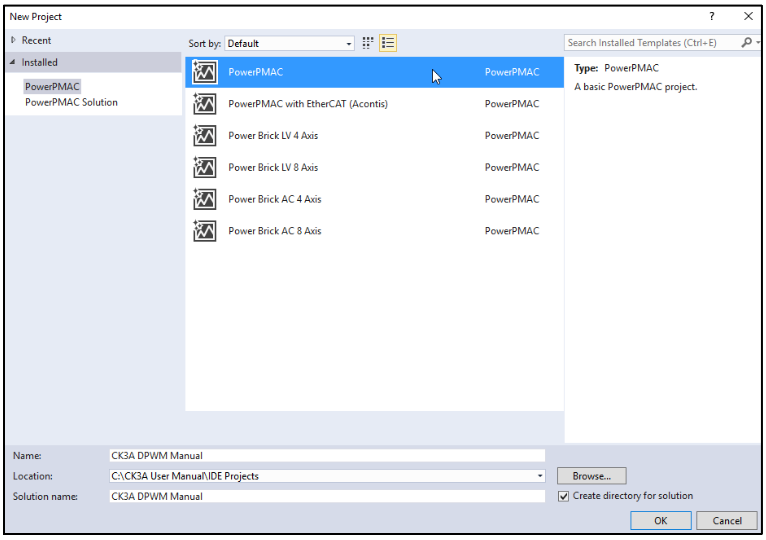
New IDE Project

Creating a new project is done from the file menu.

- File > New > Project



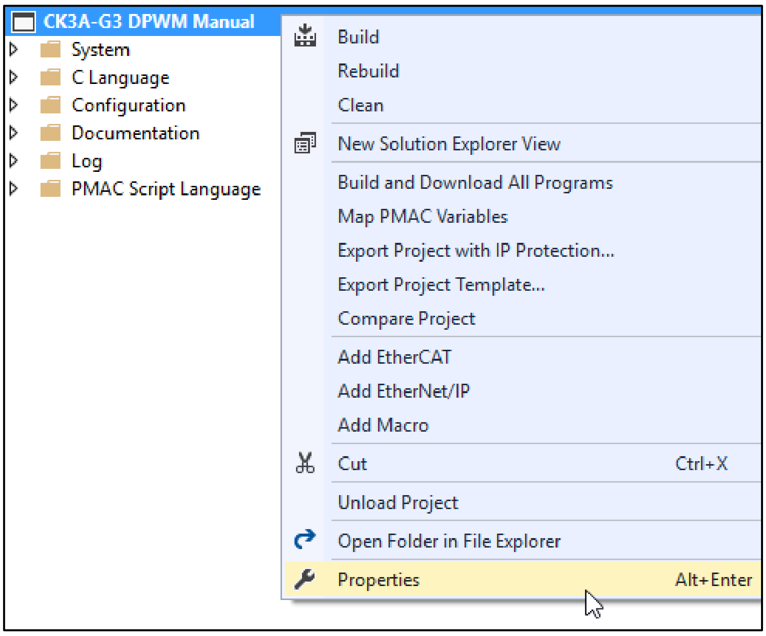
- Choose template
- Choose project location
- Enter project name

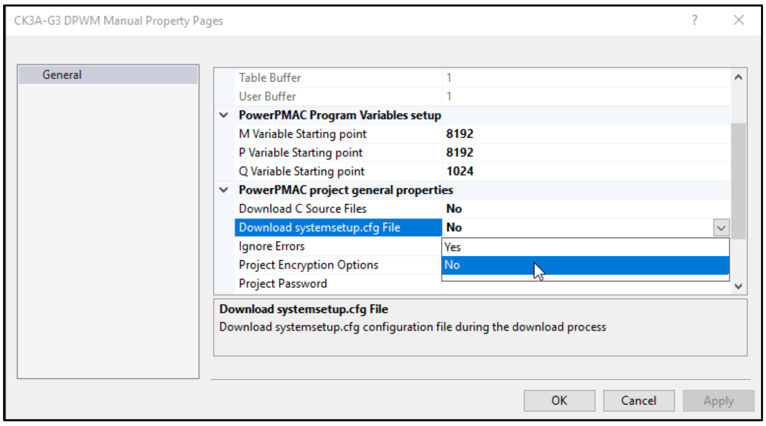


Systemsetup.cfg

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

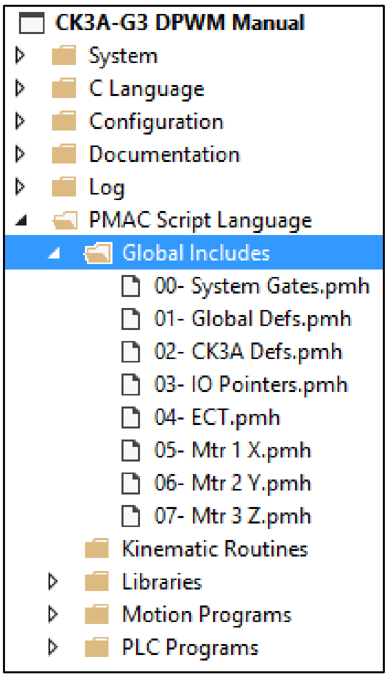
- Right-click on project name
- Properties
- Download Systemsetup.cfg File > NO > OK

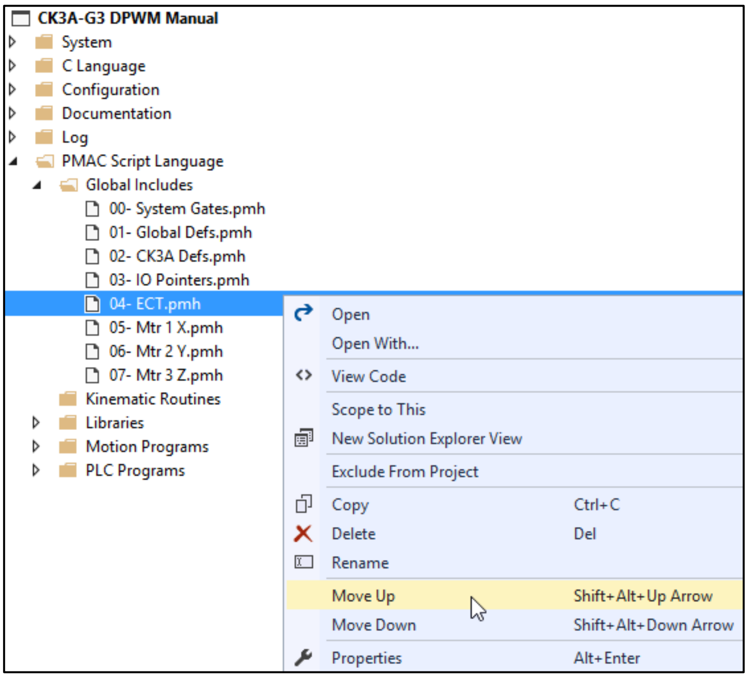




Project Organization

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, sorted out (moved up and down) etc… Refer to the IDE Manual to learn about these manipulations. One suggested assortment is shown below.



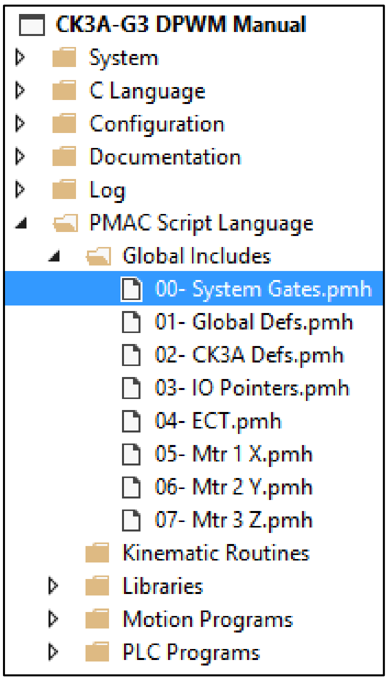


Below, is a brief description of these suggested files.

File	Typical Content	Content Example
System Gates	System structure elements	Sys.MaxMotors
	Gate3 structure elements	Gate3[0].PhaseFreq
	Channel structure elements	Gate3[0].Chan[0].PwmFreqMult
Global Defs	User-defined variables	GLOBAL MyVariable
CK3A Defs	CK3A specific Variables	GLOBAL Ck3a1DcBus
IO Pointers	Digital I/O pointers	PTR Input1->Gate3[0].GpioData[0].0.1
	Analog I/O pointers	PTR AdcIn1->S:IO:$900040.16.16
ECT	Encoder conversion table	EncTable[1].type
Mtr1 X	Motor structure elements	Motor[1].ServoCtrl

A-2-2 Basic Optimization, System and Gates Settings

This section describes basic optimization, clock frequencies, and CK3A-specific structure elements.




```
Sys.WpKey = $AAAAAAAA
Sys.MaxCoords = 2
Sys.MaxMotors = 5

Gate3[0].PhaseFreq = 10000
Gate3[0].ServoClockDiv = 1
Sys.RtIntPeriod = 1
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)

Gate3[0].Chan[0].PwmFreqMult = 1
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PwmDeadTime = 2 / 0.0533
Gate3[0].AdcAmpStrobe = $901001
Gate3[0].AdcAmpClockDiv = 5
Gate3[0].AdcAmpHeaderBits = 4

Gate3[0].Chan[0].PackInData = 0
Gate3[0].Chan[1].PackInData = 0
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0
Gate3[0].Chan[1].PackOutData = 0
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0
```

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, = 2 (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)
Gate3[0].PhaseFreq	**Phase clock frequency in [Hz]** Example, = 10000 (10KHz)
Gate3[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: = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq
Sys.PhaseOverServoPeriod	**Ratio of phase to servo** Necessary for correct frequencies setup. It can be set as fixed expression: = 1 / (Gate3[0].ServoClockDiv + 1)
Gate3[i].Chan[j].PwmFreqMult	**PWM frequency multiplier (defines PWM frequency)** PWM frequency = (PwmFreqMult + 1) * PhaseFreq / 2 Example, = 1 (10 KHz – CK3A range is 8 – 20 KHz)
Gate3[i].Chan[j].PwmDeadTime	**PWM deadtime** Always = 2 / 0.0533 (2 µsec for CK3A-G305L) Always = 3 / 0.0533 (3 µsec for CK3A-G310L)
Gate3[i].AdcAmpStrobe	**Amplifier ADC strobe word (CK3A specific)** Example, = $901001 (described in detail in this manual)
Gate3[i].AdcAmpClockDiv	**Amplifier A/DC converter clock frequency (CK3A specific)** Permissible Gate3 settings for CK3A: = 5 (3.125 MHz), = 4 (6.250 MHz)

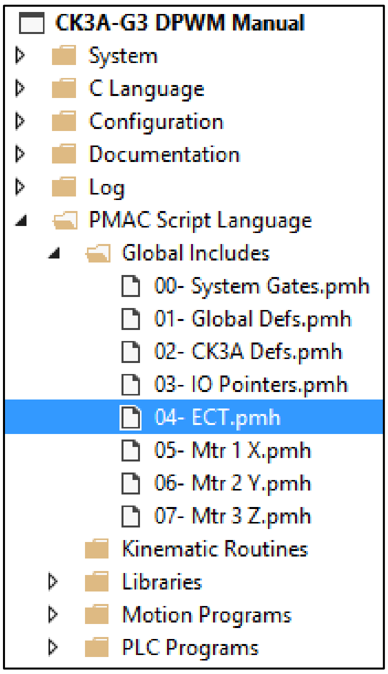
Gate3[i].AdcAmpHeaderBits	**Amplifier ADC number of header bits (CK3A specific)** Always = 4 for CK3A
Gate3[i].Chan[j].PackInData	**ADC input pack data enable** Always = 0 for CK3A
Gate3[i].Chan[j].PackOutData	**ADC output pack data enable** Always = 0 for CK3A

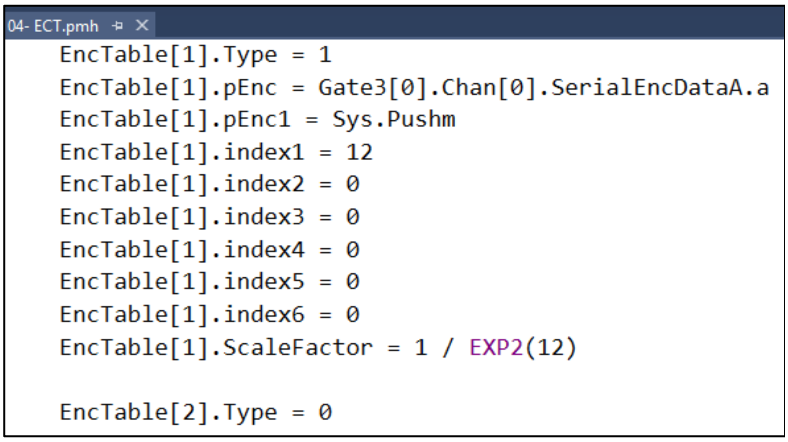
Additional Information

Detailed descriptions can be found in the Power PMAC Software Reference Manual.

A-2-3 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	= Gate3[i].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].ScaleFactor	= 1 / 256

ECT Entry for Standard Analog Sinusoidal x16384 (UMAC and CK3M)

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].ScaleFactor	= 1

● Other UMAC (ACC-24E3) necessary settings for this type of processing

Structure Element	Setting
Motor[x].EncType	= 6
Gate3[i].Chan[j].AtanEna	= 1

● Other CK3M (CK3W-AX2323□) necessary settings for this type of processing

Structure Element	Setting
Motor[x].EncType	= 6
Gate3[i].Chan[j].AtanEna	= 1
Gate3[i].EncClockDiv	= 3
Gate3[i].AdcEncClockDiv	= 3
Gate3[i].AdcEncCtrl	= $3FFFC000

ECT Entry for Analog Sinusoidal ACI x65536 (UMAC)

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].ServoCapt.a
EncTable[n].pEnc1	= Gate3[i].Chan[j].AtanSumOfSqr.a
EncTable[n].ScaleFactor	= 1

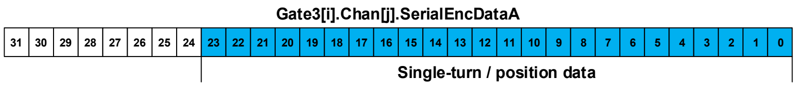
● Other necessary settings for this type of processing

Structure Element	Setting
Motor[x].EncType	= 7
Gate3[i].Chan[j].AtanEna	= 1
Gate3[i].AdcEncHeaderBits	= 0
Gate3[i].AdcEncStrobe	= $800000

ECT Entry for Serial w/ Gate3 (CK3W-AX and ACC-24E3)

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= Gate3[i].Chan[j].SerialEncDataA.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].Index1	= No. of data bits to shift left to bit #31 + index2. This is the 2nd shift operation. In example diagram below, = 8
EncTable[n].Index2	= No. 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
Gate3[i].Chan[j].SerialEncEna	= 1
Gate3[i].SerialEncCtrl	Refer to the Power PMAC Software Reference Manual
Gate3[i].Chan[j].SerialEncCmd	Refer to the Power PMAC Software Reference Manual

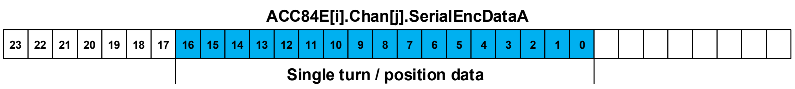
Additional Information

After setting up Gate3[i].SerialEncCtrl and Gate3[i].Chan[j].SerialEncCmd successfully, and Gate3[i].Chan[j].SerialEncEna=1 raw serial encoder data can be seen in Gate3[i].Chan[j].SerialEncDataA, and Gate3[i].Chan[j].SerialEncDataB. Depending on the protocol, some information may reside in Gate3[i].Chan[j].SerialEncDataC, or Gate3[i].Chan[j].SerialEncDataD.

ECT Entry for Serial w/ ACC-84E (UMAC)

Structure Element	Typical Setting
EncTable[n].Type	= 1
EncTable[n].pEnc	= ACC84E[i].Chan[j].SerialEncDataA.a
EncTable[n].pEnc1	= Sys.pushm
EncTable[n].Index1	= No. 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 + No. 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
ACC84E[i].SerialEncCtrl	Refer to the Power PMAC Software Reference Manual
ACC84E[i].Chan[0].SerialEncCmd	Refer to the Power PMAC Software Reference Manual

Additional Information

After setting up ACC84E[i].SerialEncCtrl and ACC84E[i].Chan[j].SerialEncCmd successfully, raw serial encoder data can be seen in ACC84E[i].Chan[j].SerialEncDataA, and ACC84E[i].Chan[j].SerialEncDataB. Depending on the protocol, some information may reside in ACC84E[i].Chan[j].SerialEncDataC, or ACC84E[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 as well as 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	Gate3[i].Chan[j].ServoCapt
Analog sinusoidal	Gate3[i].Chan[j].ServoCapt
Analog sinusoidal ACI	Gate3[i].Chan[j].ServoCapt
Serial w/ Gate3	Gate3[i].Chan[j].SerialEncDataA Gate3[i].Chan[j].SerialEncDataB (does not always have data)
Serial w/ ACC-84E	ACC84E[i].Chan[j].SerialEncDataA ACC84E[i].Chan[j].SerialEncDataB (does not always have data)

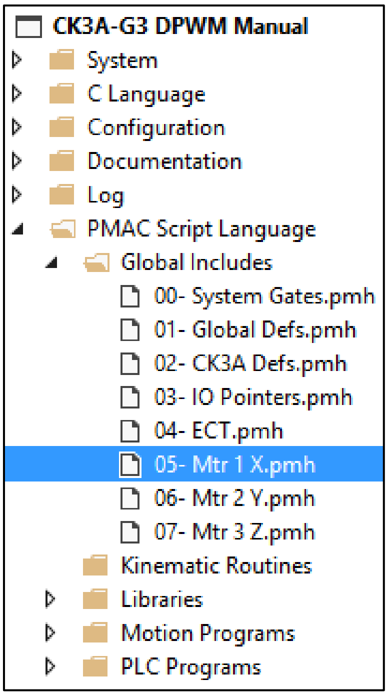
- **Reporting the Correct Distance**

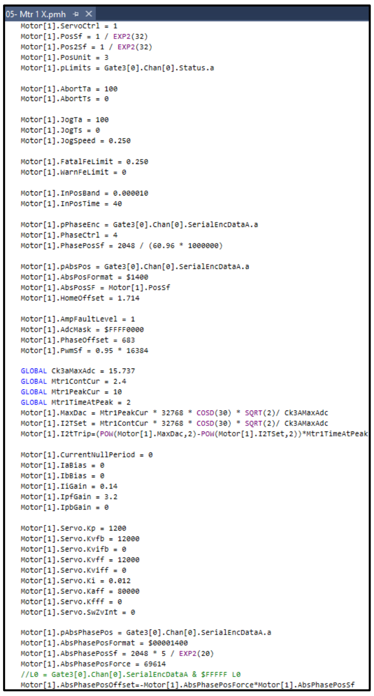
The user can verify if the feedback device is counting correctly by moving the Motor a known amount and recording the elapsed distance shown in the IDE position window. In some cases, the #nHMZ (where n is the Motor number) command can be used to zero the position display.

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-2-4 Motor Setup and Commissioning

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





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.

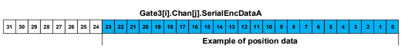
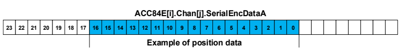
Common Amplifier and Direct PWM Control Elements

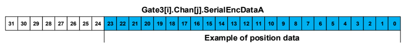
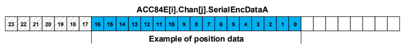
Structure Element	Description
Motor[x].AmpFaultLevel	**Amplifier fault logical state** Always = 1 for CK3A
Motor[x].AdcMask	**Specifies current feedback resolution (in bits)** Always = $FFFF0000 for CK3A
Motor[x].PhaseOffset	**Angular distance between Motor phases** Typical, = 683
Motor[x].PwmSf	**PWM 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 in the description below.

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 x4)
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 = Gate3[i].Chan[j].PhaseCapt.a Analog sinusoidal = Gate3[i].Chan[j].PhaseCapt.a Analog sinusoidal ACI = Gate3[i].Chan[j].PhaseCapt.a Serial w/ Gate3 = Gate3[i].Chan[j].SerialEncDataA.a Serial w/ ACC-84E = ACC84E[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-84E = 8 in this example (typical w/ ACC-84E) 

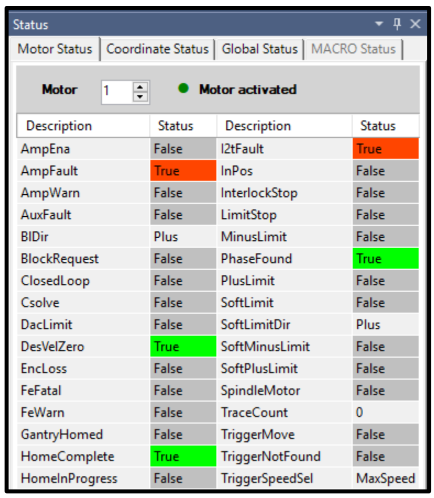
Structure Element	Description and Examples
Motor[x].PhaseEncLeftShift	**Number of bits to shift phase-position source data left (to bit #31)** Digital quadrature = 0 Analog sinusoidal = 0 Analog sinusoidal ACI = 0 Serial w/ Gate3 = 8 in this example  Serial w/ ACC-84E = 15 in this example (7 data bits + 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-84E	Rotary Motor = 2048 * NoOfPolePairs / ($2^{Motor[x].PhaseEncLeftShift + ST}$) Linear Motor = 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.

The CK3A Amplifier has its own built-in I2T protection to protect it from overheating, this is not configurable. 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, 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 following table is a reminder of the CK3A Amplifier specifications.

Specification	CK3A-G305L	CK3A-G310L
Maximum RMS current	5 A_{rms}	10 A_{rms}
Rated RMS current	10 A_{rms}	20 A_{rms}
Current scaling (max. ADC)	15.735 A_{peak}	31.470 A_{peak}
Time allowed at maximum current	2 sec	2 sec

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

Additional Information

The CK3A Amplifier rated and maximum current specifications are given in RMS. If the Motor current specifications are given in "peak", they must be divided by √2.

Assuming the following definitions, below are the expressions for setting up PMAC I2T.

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
Ck3aMaxAdc	Amplifier current scaling (defined in amperes peak)

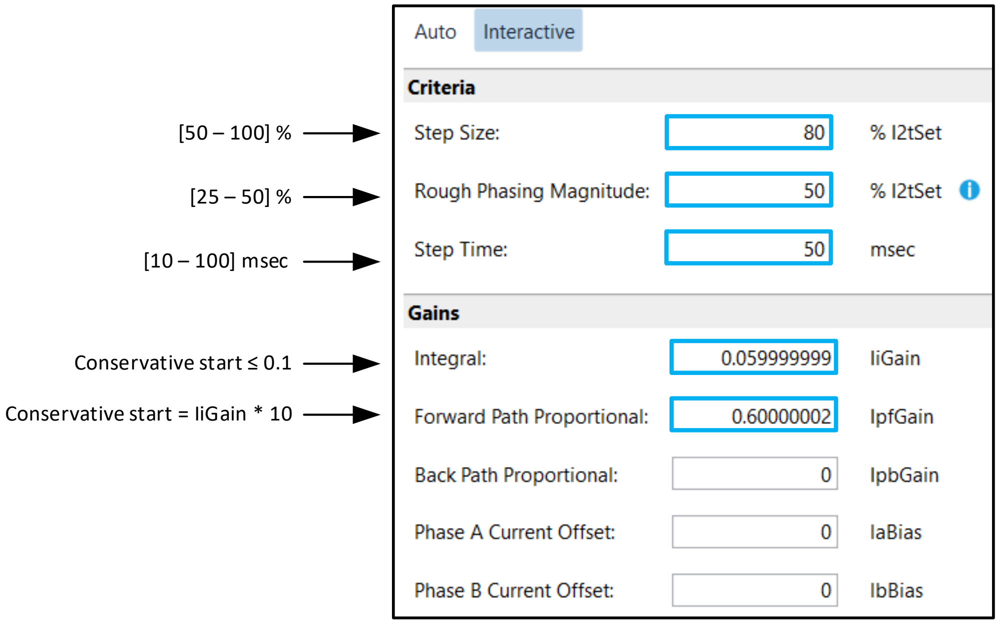
Structure Element	Description and Example
Motor[x].MaxDac	**Maximum servo output** = MaxRmsCur * 32768 * SQRT(2) * COSD(30) / Ck3aMaxAdc
Motor[x].I2tSet	**Continuous current limit** = RatedRmsCur * 32768 * SQRT(2) * COSD(30) / Ck3aMaxAdc
Motor[x].I2tTrip	**Maximum (shutdown) current limit** = ((Motor[x].MaxDac * 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.



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

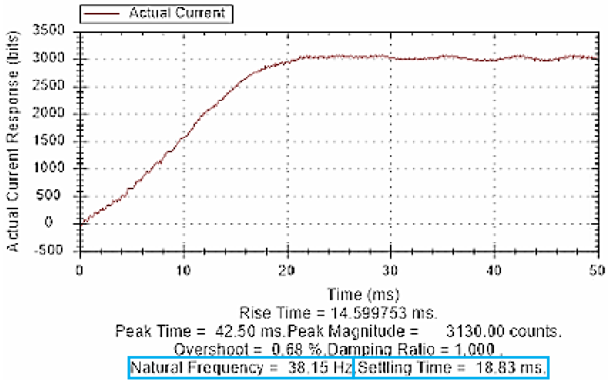
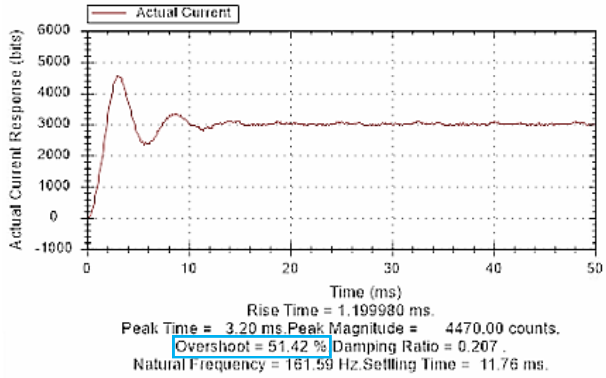
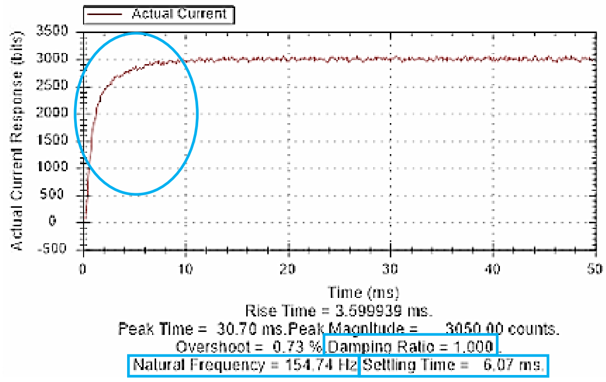
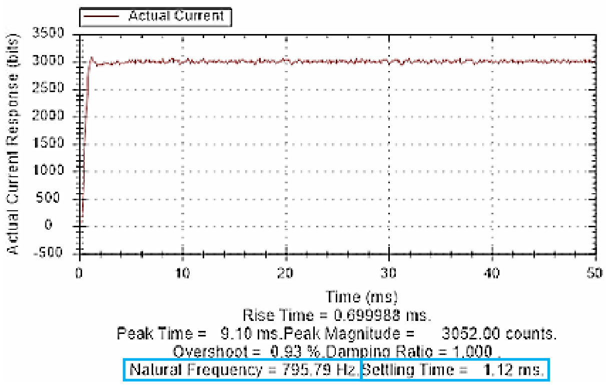
Structure Element	Description and Example
Motor[x].IiGain	**Current loop integral gain** Conservative start, = 0.1 or less
Motor[x].IpfGain	**Current loop forward-path proportional gain** Conservative start, = Motor[x].IiGain * 10
Motor[x].IpbGain	**Current loop back-path proportional gain** Not used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

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.

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

x designates Motor number.

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

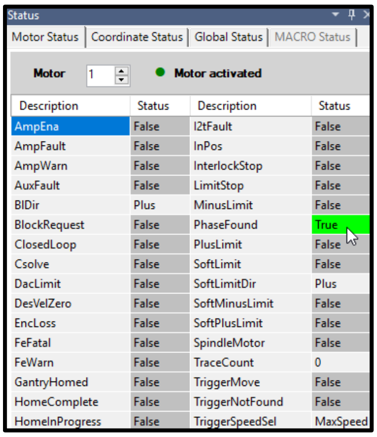
The automatic stepper phasing method is one of two phase referencing routines built into the Power PMAC firmware. It requires the setting of the following structure elements.

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

- Increase Motor[x].PhaseFindingTime

- Reverse encoder decode (for incremental encoders)
  Gate3[i].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

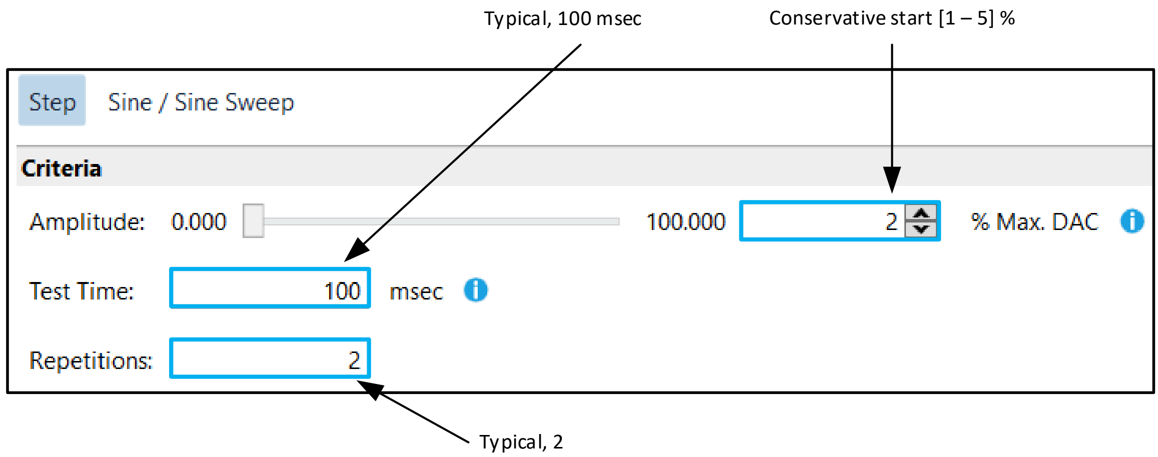
Precautions for Safe Use

- 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.

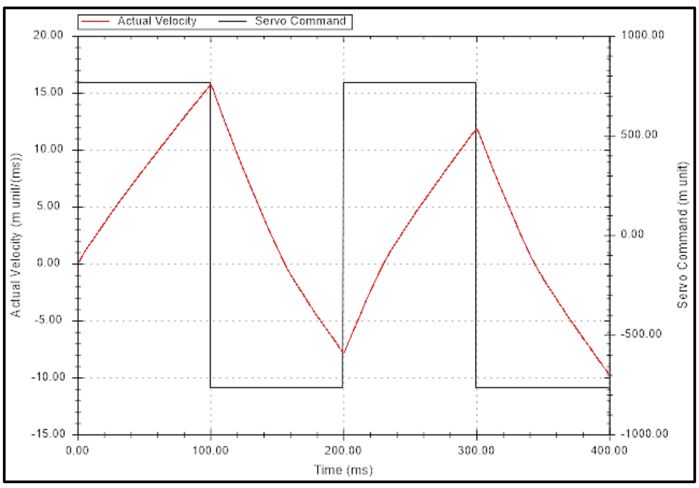
The open loop test is a critical step in the setup of a commutated “brushless” Motor. Primarily, it provides verification for the following functions.

- 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.

  - Always remember #xK kill Motor x (servo OFF)
  - #xOUT1, is the Motor moving in the positive direction?
  - #x OUT-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.

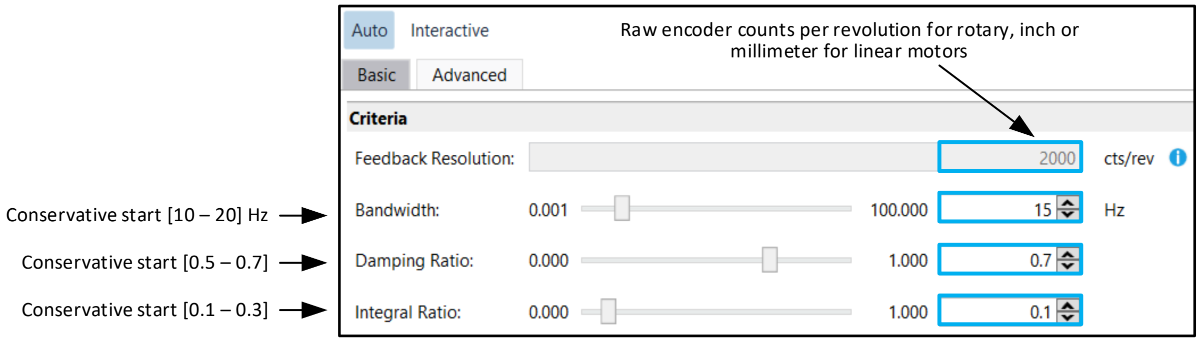
  - Incorrect commutation cycle size, Motor[x].PhasePosSf setting
  - Initial phasing was not performed successfully
  - Reversed incremental encoder direction, Gate3[i].Chan[j].EncCtrl = 7 <-> 3
    (does not apply to serial encoders)
  - Serial encoder direction does not match Amplifier output. In this 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

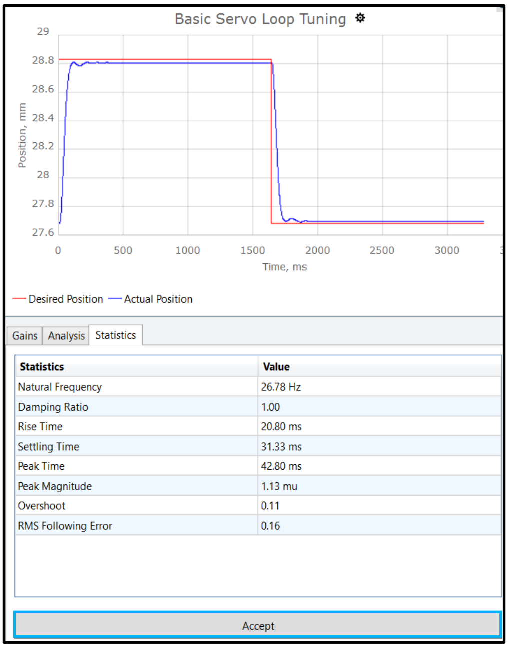
Precautions for Safe Use

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

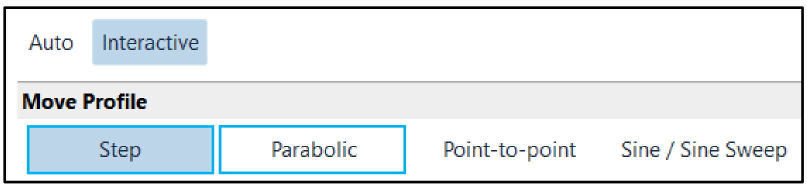
The basic auto tune tool can be used to obtain initial gains.



Successful auto-tuning shows a step response and statistics.

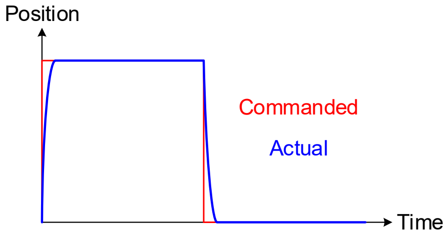


The gains generated by the auto-tuning tool can be accepted or interactive fine tuning can be performed using the most common step and parabolic tuning moves.



● Step Move (Basic) Tuning

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

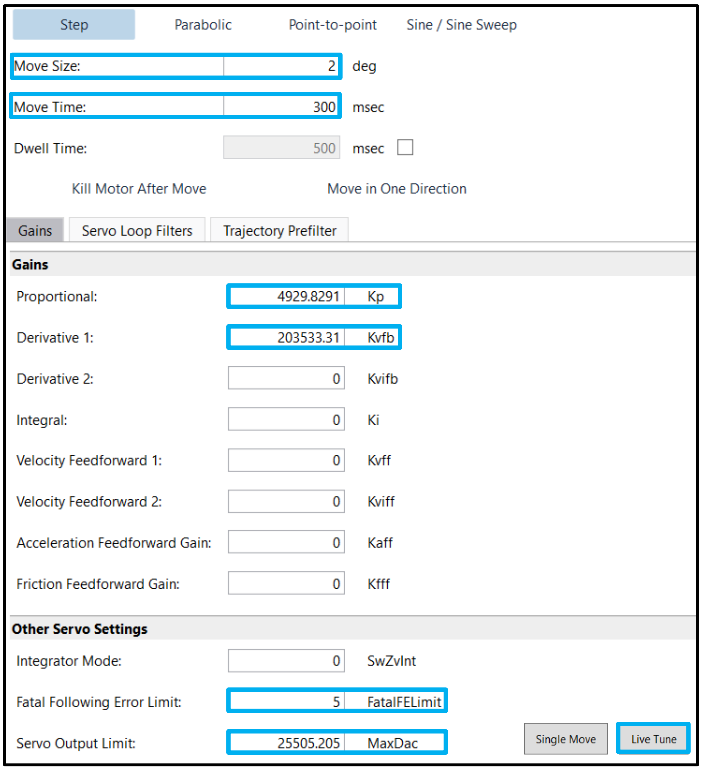


That is, by optimizing the following structure element gains.

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

Preparing for the step move tuning:

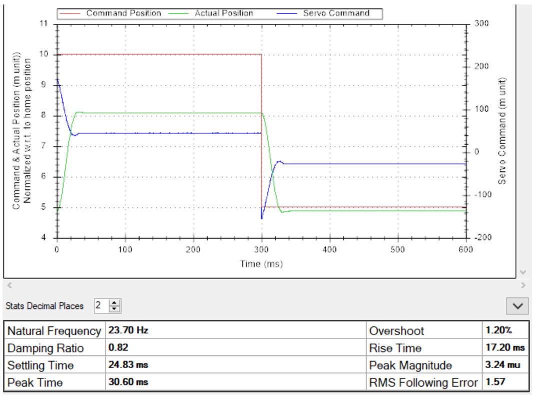
- Make sure all other gains are set to zero
- Choose move size (that moves the load)
  Rotary Motor typical range [1 – 5 deg]
  Linear Motor typical range [0.1 – 0.5 mm]
- Choose move time, typical range [300 – 2000 msec]
- The fatal following error must be greater than the step move. It can be slightly 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



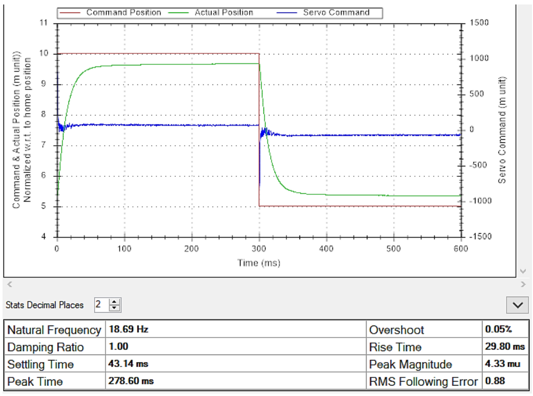
Additional Information

Try (single) step move at first. If safe and stable, use the Live Tune feature, then change gains on the fly while monitoring response changes on the plot.

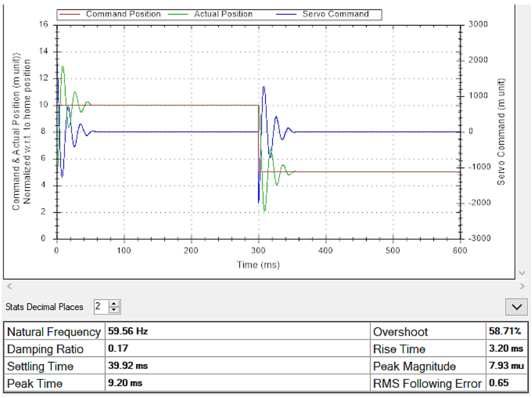
Below, are general guidelines for step move tuning.

<table>
<tr><th>Step Response</th><th>Observation</th><th>Counter-measure</th></tr>
<tr><td>
</td><td>

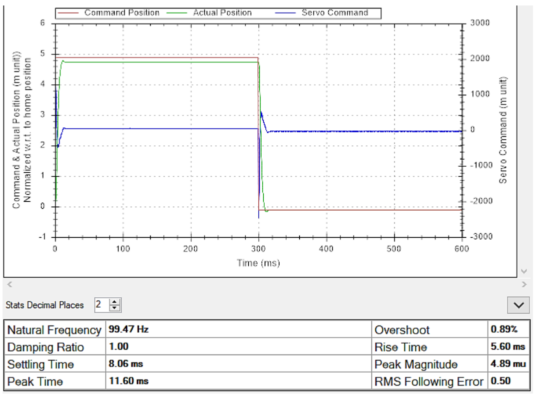
- “Sluggish” response. Motor may feel loose.
- Too little servo “effort” command. About 1% of Motor[x].MaxDac in this example.
- Relatively long settling time

</td><td>Increase Motor[x].Servo.Kp and Motor[x].Servo.Kvfb (proportionally)</td></tr>
<tr><td>
</td><td>

- “Over-damped” response

</td><td>Increase Motor[x].Servo.Kp and or decrease Motor[x].Servo.Kvfb</td></tr>
<tr><td>
</td><td>

- “Under-damped” response
- Too much overshoot

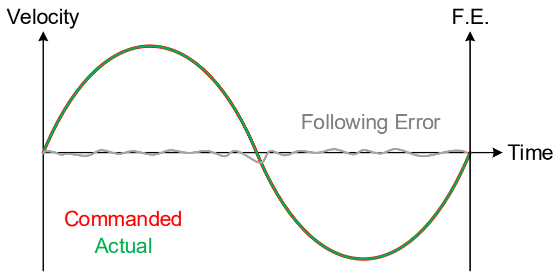
</td><td>Increase Motor[x].Servo.Kvfb</td></tr>
<tr><td>
</td><td>

- “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)

</td><td>Good step move response</td></tr>
</table>

Parabolic Move (Basic) Tuning

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

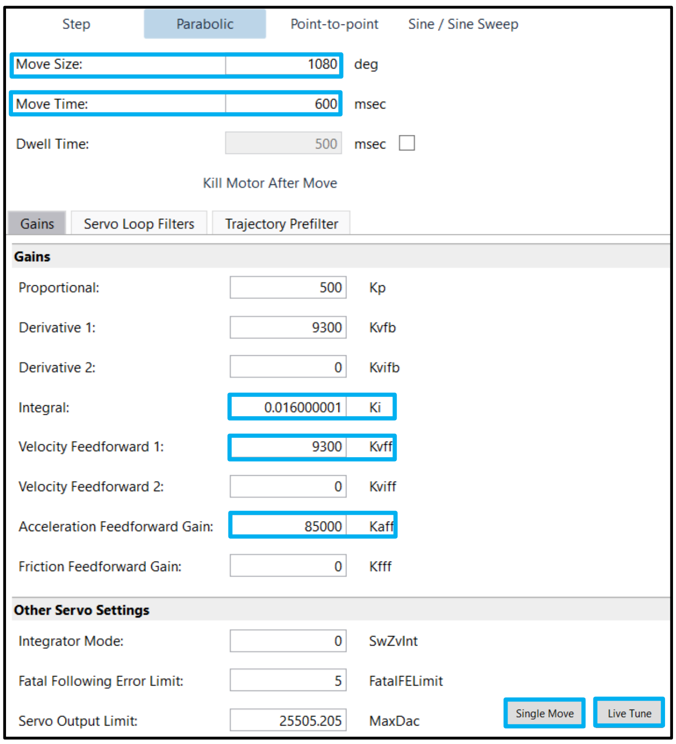


That is, by optimizing the following structure element gains.

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

Preparing for the parabolic move tuning:

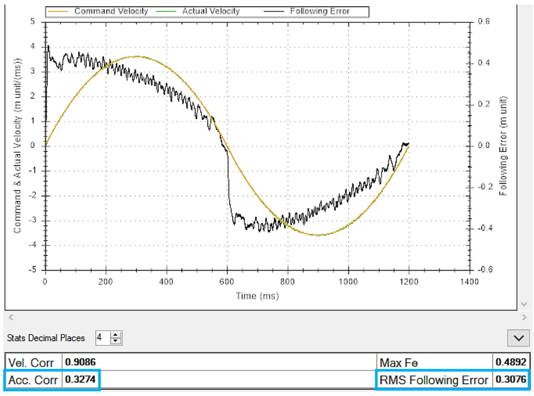
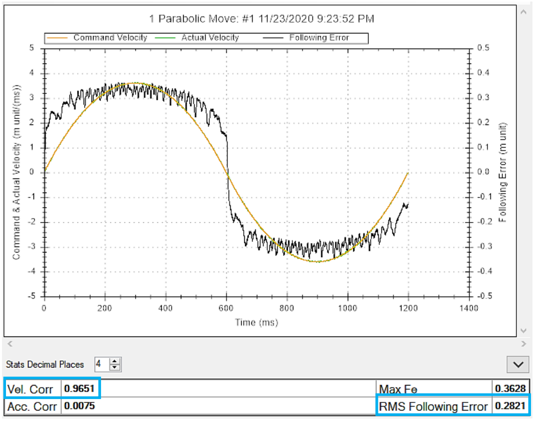
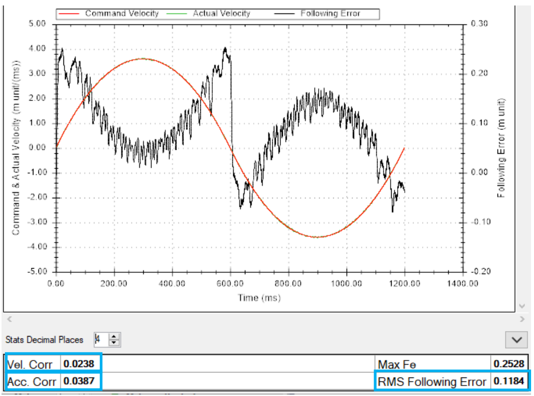
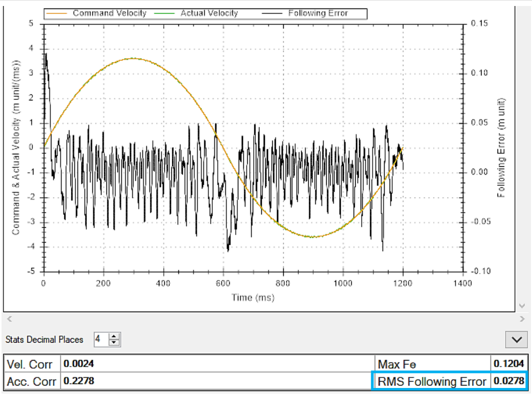
- Copy Motor[x].Servo.Kvfb into Motor[x].Servo.Kvff (these two gains are almost always equal)
- Choose move size (that moves the load)
  Rotary Motor typical range [1 – 4 revolutions]
  Linear Motor typical range [10 – 80 mm]
- Choose move time, typical range [600 – 2000 msec]
- Typically, it is desirable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



Additional Information

Try (single) parabolic move at first. If safe and stable, use the Live Tune feature (PA-33), then change gains on the fly while monitoring the change on the plot.

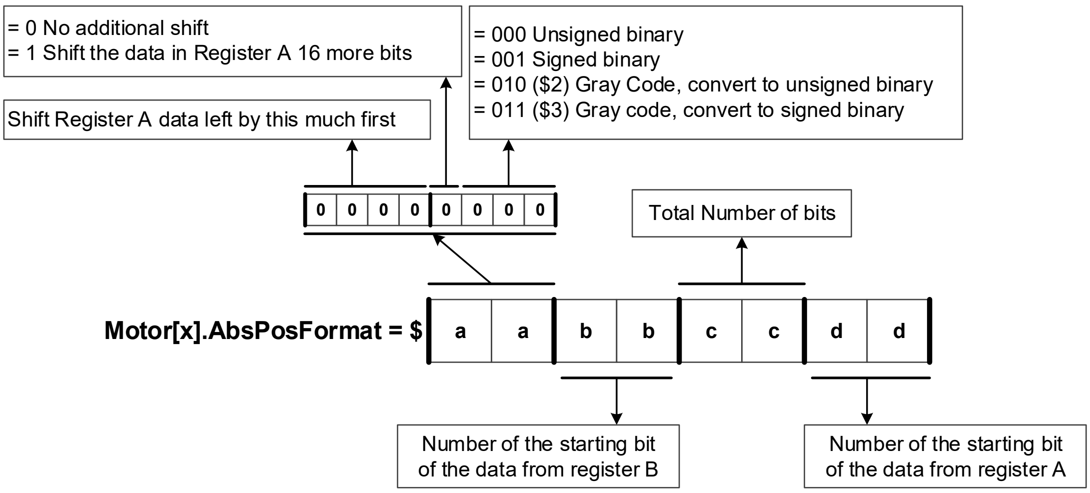
It is recommended to optimize Motor[x].Servo.Kaff first before Motor[x].Servo.Ki. Below, are general guidelines for parabolic move tuning.

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. 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, = Gate3[i].Chan[j].SerialEncDataA.a Typically, for ACC-84E, = ACC84E[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 Gate3[i].Chan[j].SerialEncCmd, or ACC84E[i].Chan[j].SerialEncCmd.

<table>
<tr><th>Example</th><th>Motor[x].AbsPosFormat Setting</th></tr>
<tr><td>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
Gate3[i].Chan[j].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

• ACC-84E
ACC84E[i].Chan[j].SerialEncDataA
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
<td>• Gate3
00: unsigned binary — 17 bits
Motor[x].AbsPosFormat = $ 0 0 0 0 1 1 0 0
Serial data B: none 00 — Serial data A start at bit 00

• ACC-84E
00: unsigned binary — 17 bits
Motor[x].AbsPosFormat = $ 0 0 0 0 1 1 0 8
Serial data B: none — Serial data A start at bit 8</td></tr>
<tr><td>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
Gate3[i].Chan[j].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

• ACC-84E
ACC84E[i].Chan[j].SerialEncDataA
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
<td>• Gate3
00: unsigned binary — 20 bits
Motor[x].AbsPosFormat = $ 0 0 0 0 1 4 0 4
Serial data B: none 00 — Serial data A start at bit 4

• ACC-84E
00: unsigned binary — 20 bits
Motor[x].AbsPosFormat = $ 0 0 0 0 1 4 0 C
Serial data B: none — Serial data A start at bit 12</td></tr>
<tr><td>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
Gate3[i].Chan[j].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
Gate3[i].Chan[j].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

• ACC-84E
ACC84E[i].Chan[j].SerialEncDataA
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
ACC84E[i].Chan[j].SerialEncDataB
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0</td>
<td>• Gate3
00: unsigned binary — 36 bits
Motor[x].AbsPosFormat = $ 0 0 0 0 2 4 0 0
Serial data B start at bit 0 — Serial data A start at bit 0

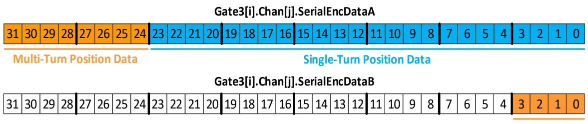
• ACC-84E
00: unsigned binary — 36 bits
Motor[x].AbsPosFormat = $ 0 0 0 8 2 4 0 8
Serial data B start at bit 8 — Serial data A start at bit 8</td></tr>
<tr><td>A 29-bit binary absolute serial encoder with 17 bits of single-turn and 12 bits of multi-turn position data starting at bit #0 of serial data A register and continuously extending to bit #28.

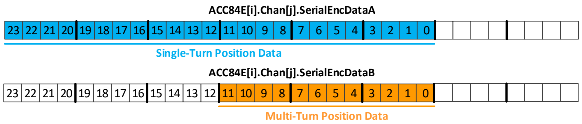
• Gate3
Gate3[i].Chan[j].SerialEncDataA
31 30 29 28 27 26 25 24 23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Multi-Turn Position Data — Single-Turn Position Data

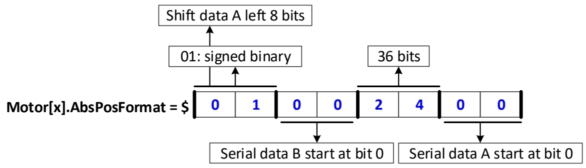
• ACC-84E
ACC84E[i].Chan[j].SerialEncDataA
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Multi-Turn Position Data — Single-Turn Position Data
ACC84E[i].Chan[j].SerialEncDataB
23 22 21 20 19 18 17 16 15 14 13 12 11 10 9 8 7 6 5 4 3 2 1 0
Multi-Turn Position Data</td>
<td>• Gate3
01: signed binary — 29 bits
Motor[x].AbsPosFormat = $ 0 1 0 0 1 D 0 0
Serial data B, none — Serial data A start at bit 0

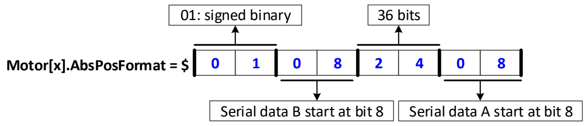
• ACC-84E
01: signed binary — 29 bits
Motor[x].AbsPosFormat = $ 0 1 0 8 1 D 0 8
Serial data B start at bit 8 — Serial data A start at bit 8</td></tr>
</table>

<table>
<tr><th>Example</th><th>Motor[x].AbsPosFormat Setting</th></tr>
<tr><td>
A 36-bit binary absolute serial encoder with 24 bits of single-turn position data, and 12 bits of multi-turn position data.

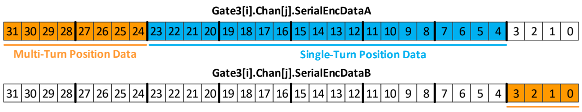
• Gate3


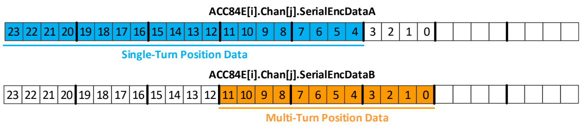
• ACC-84E


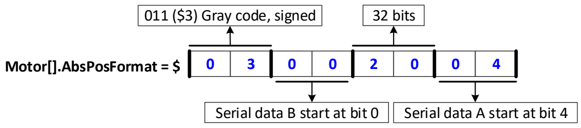
</td><td>
• Gate3


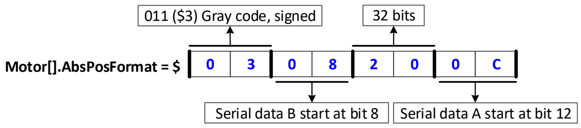
• ACC-84E


</td></tr>
<tr><td>
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-84E


Note If gray code is already set up in global command word, then it should not be set up here.
</td><td>
• Gate3


• ACC-84E


</td></tr>
</table>

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

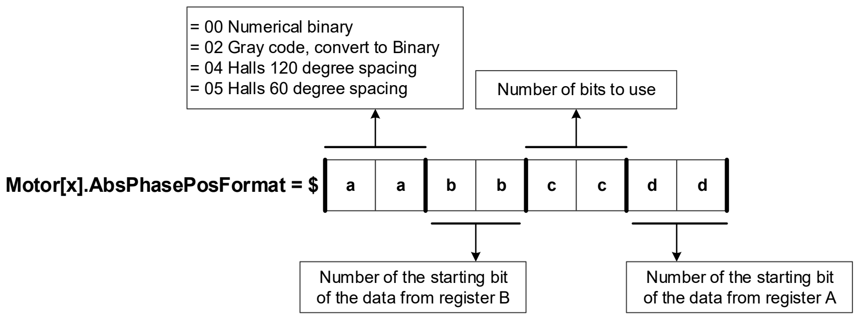
- Motor must have been successfully phased prior to setting up absolute phase reference.
- Motor[x].PhaseFindingDac, and Motor[x].PhaseFindingTime must be equal to zero.

This section describes the setup of absolute phase (commutation) reference for Motors with Hall effect sensors and absolute serial encoder. 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, =Gate3[i].Chan[j].Status.a For absolute serial encoders, =Gate3[i].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.

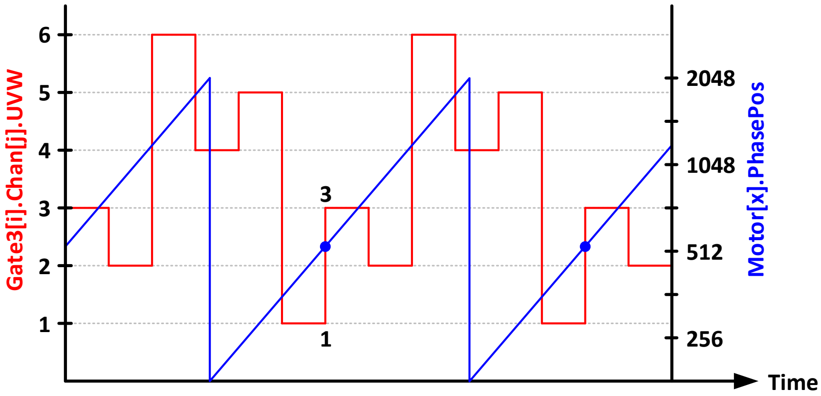
● Absolute Phase Reference Using Hall Effect Sensors

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 Gate3[i].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 (most significant bit) 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.

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	Read Motor[x].AbsPhasePosForce	={value}, refer to the examples below
6	Motor[x].IbBias = 0 #xK	Reset Motor[x].IbBias, and kill Motor

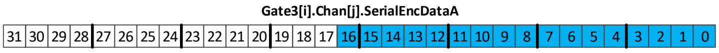
x designates Motor number.

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.

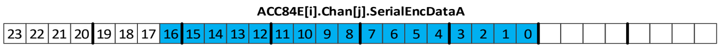
Ex.	Absolute Phase Reference Settings
1	An absolute serial encoder with 17 bits of binary single-turn (or linear scale), and no multi-turn, position data located in the lower fields of serial data A register.

- Gate3



Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17}
Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm
Motor[x].pAbsPhasePos = Gate3[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = $00001100
Motor[x].AbsPhasePosForce = value of {**Gate3[i].Chan[j].SerialEncDataA & $1FFFF**} during test
Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

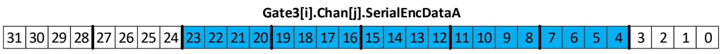
- ACC-84E



Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17}
Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm
Motor[x].pAbsPhasePos = ACC84E[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = $00001108
Motor[x].AbsPhasePosForce = value of {**ACC84E[i].Chan[j].SerialEncDataA & $1FFFF**} during test
Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

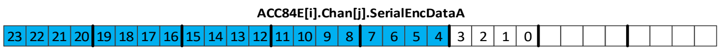
Ex.	Absolute Phase Reference Settings
2	An absolute serial encoder with 20 bits of binary single-turn (or linear scale), and no multi-turn, position data starting at bit #4 of serial data A register.

- Gate3

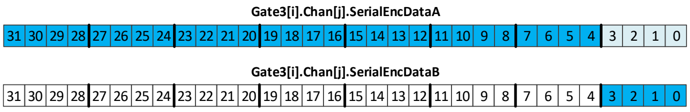
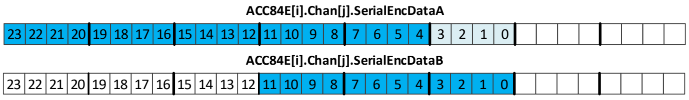
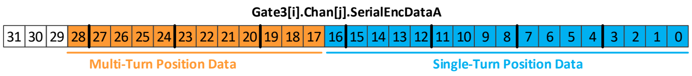
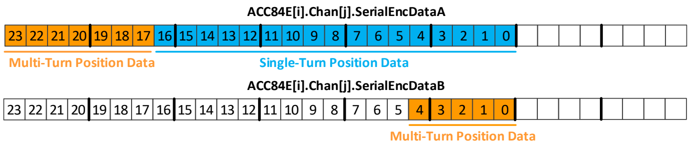


Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{20}
Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm
Motor[x].pAbsPhasePos = Gate3[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = $00001404
Motor[x].AbsPhasePosForce = value of {**(Gate3[i].Chan[j].SerialEncDataA & $FFFFF0) >> 4**} during test
Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

- ACC-84E



Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{20}
Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm / ECLmm
Motor[x].pAbsPhasePos = ACC84E[i].Chan[j].SerialEncDataA.a
Motor[x].AbsPhasePosFormat = $0000140C
Motor[x].AbsPhasePosForce =value of {**(ACC84E[i].Chan[j].SerialEncDataA & $FFFFF0) >> 4**}during test
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 = Gate3[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00002004 (ignore lower 4 bits, use upper 32 bits to include MSB) Motor[x].AbsPhasePosForce = value of {**Gate3[i].Chan[j].SerialEncDataA>>4+(Gate3[i].Chan[j].SerialEncDataB&$F)<<28**} during test Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf • ACC-84E  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{32} Linear: Motor[x].AbsPhasePosSf = 2048 * RESmm * 2^4 / ECLmm Motor[x].pAbsPhasePos = ACC84E[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $0008200C (ignore lower 4 bits, use upper 32 bits to include MSB) Motor[x].AbsPhasePosForce = value of {**ACC84E[i].Chan[j].SerialEncDataA>>4+(ACC84E[i].Chan[j].SerialEncDataB&$FFF)<<20**} during test 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 = Gate3[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001100 (use only 17-bit single-turn data in register A) Motor[x].AbsPhasePosForce = value of {**Gate3[i].Chan[j].SerialEncDataA & $1FFFF**} during test Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf • ACC-84E  Rotary: Motor[x].AbsPhasePosSf = 2048 * NoOfPolePairs / 2^{17} Motor[x].pAbsPhasePos = ACC84E[i].Chan[j].SerialEncDataA.a Motor[x].AbsPhasePosFormat = $00001108 (use only 17-bit single-turn data in register A) Motor[x].AbsPhasePosForce = value of {**ACC84E[i].Chan[j].SerialEncDataA & $1FFFF**} during test Motor[x].AbsPhasePosOffset = –Motor[x].AbsPhasePosForce * Motor[x].AbsPhasePosSf

A-2-5 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
Gate3[i].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-3 Gate3 “Script” Motor Setup Samples

Brushless Motor setup samples w/ Power PMAC ACC-24E3, ACC-84E, and CK3M AX units are provided in this section for reference.

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 Gate3, index 0, channel 0.

A-3-1 Rotary Servo Motor w/ Quadrature Encoder

Hardware Configuration

Item	Notes
Controller	This sample is applicable to UMAC ACC-24E3, and CK3M AX
Amplifier	CK3A-G310L (three-phase 208VAC main supply voltage)
Motor	Rotary servo Motor, 200VAC, 11/23A, 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)

Gate3[0].PhaseFreq = 10000                                              10 KHZ PHASE FREQUENCY
Gate3[0].ServoClockDiv = 1                                              5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                     2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq    TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)             TYPICAL CALCULATION

Gate3[0].Chan[0].PwmFreqMult = 1                                        CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PackInData = 0                                         CH1-4 IN DATA
Gate3[0].Chan[1].PackInData = 0                                         MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0                                        CH1-4 OUT DATA
Gate3[0].Chan[1].PackOutData = 0                                        MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

Gate3[0].AdcAmpStrobe = $901001                                         CH1-4 STROBE WORD (FOR CK3A)
Gate3[0].AdcAmpClockDiv = 5                                             CH1-4 3.125 MHZ ADC CLOCK
Gate3[0].AdcAmpHeaderBits = 4                                           CH1-4 MUST BE 4 (FOR CK3A)
Gate3[0].Chan[0].PwmDeadTime = 3 / 0.0533                               CH1 MIN. 3µSEC FOR G310
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                    TYPICAL QUADRATURE ENCODER ENTRY
EncTable[1].pEnc = Gate3[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)
```

Motor Setup and Commissioning

```
GLOBAL Ck3a1MaxAdc = 31.470                                                 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.

Gate3[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 = Gate3[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 = Gate3[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)
Motor[1].IpbGain = 0                                                        ...

Motor[1].Servo.Kp = 0                                                       POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                                     (MUST TUNE)
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
```

A-3-2 OMRON G5 Servo Motor

Hardware Configuration

Item	Notes
Controller	This setup sample is applicable to CK3M AX, or UMAC ACC-24E3
Amplifier	CK3A-G305L (110VAC main supply voltage)
Motor	G5 Servo Motor R88M-K40030H-S2, 200VAC, 2.4/10.2A, 5 pole pairs
Encoder	G5 20-bit absolute per 1 revolution (after initial mark)

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)

Gate3[0].PhaseFreq = 10000                                                10 KHZ PHASE FREQUENCY
Gate3[0].ServoClockDiv = 1                                                5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                       2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)               TYPICAL CALCULATION

Gate3[0].Chan[0].PwmFreqMult = 1                                          CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PackInData = 0                                           CH1-4 IN DATA
Gate3[0].Chan[1].PackInData = 0                                           MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0                                          CH1-4 OUT DATA
Gate3[0].Chan[1].PackOutData = 0                                          MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

Gate3[0].AdcAmpStrobe = $901001                                           CH1-4 STROBE WORD (FOR CK3A)
Gate3[0].AdcAmpClockDiv = 5                                               CH1-4 3.125 MHZ ADC CLOCK
Gate3[0].AdcAmpHeaderBits = 4                                             CH1-4 MUST BE 4 (FOR CK3A)
Gate3[0].Chan[0].PwmDeadTime = 2 / 0.0533                                 CH1 MIN. 2µSEC FOR G305

Gate3[0].SerialEncCtrl = $01000008                                        CH1-4 SERIAL ENC. CTRL 20-BIT G5
Gate3[0].Chan[0].SerialEncCmd = $521000                                   CH1 SERIAL ENC. CMD
Gate3[0].Chan[0].SerialEncEna = 1                                         CH1 SERIAL ENC. ENABLE
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                          32-BIT READ
EncTable[1].pEnc = Gate3[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 12                                       12-BIT LEFT SHIFT TO MSB FOR ROLLOVER
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(EncTable[1].index1)        1/2^INDEX1 SCALE TO LSB
EncTable[1].MaxDelta = 0

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

Motor Setup and Commissioning

```
GLOBAL Ck3a1MaxAdc = 15.735                                   CK3A MAX ADC
GLOBAL Mtr1SingleTurn = 20                                    SINGLE TURN ST BITS
GLOBAL Mtr1CtsPerRev = EXP2(Mtr1SingleTurn)                   COUNTS PER REV. =2^ST
GLOBAL Mtr1PolePairs = 5                                      NUMBER OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 10                                     MAX RMS CUR.(CK3A<MTR)
GLOBAL Mtr1RatedRmsCur = 2.5                                  RATED RMS CUR.(MTR<CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2                                   TIME ALLOWED AT MAX CUR.

Motor[1].ServoCtrl = 1                                        ACTIVATE CHANNEL
Motor[1].PosUnit = 11                                         IDE WINDOW UNITS (DEGREES)
Motor[1].PosSf = 360 / Mtr1CtsPerRev                          SCALE POSITION TO DEG.
Motor[1].Pos2Sf = 360 / Mtr1CtsPerRev                         SCALE VELOCITY TO DEG.
Motor[1].pLimits = Gate3[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 = 3                                         JOG SPEED, 3 DEG/MSEC

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

Motor[1].InPosBand = 0.050                                    INPOS BAND, [DEG]
Motor[1].InPosTime = 5 / Sys.ServoPeriod                      INPOS TIME, 5 MSEC

Motor[1].pPhaseEnc = Gate3[0].Chan[0].SerialEncDataA.a        SERIAL ENCODER
Motor[1].PhaseCtrl = 4                                        TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 12                               SHIFT LEFT TO MSB
Motor[1].PhaseEncRightShift = 0                               NO RIGHT SHIFT
Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / EXP2(32)         =2048*POLE PAIRS/2^(ST+LeftShift)

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 = 0.144                                       CURRENT LOOP GAINS
Motor[1].IpfGain = 2.7                                        ...
Motor[1].IpbGain = 0                                          ...

Motor[1].Servo.Kp = 1200                                      POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 13000                                   ...
Motor[1].Servo.Kvifb = 0                                      ...
Motor[1].Servo.Kvff = 13000                                   ...
Motor[1].Servo.Kviff = 0                                      ...
Motor[1].Servo.Ki = 0.016                                     ...
Motor[1].Servo.Kaff = 80000                                   ...
Motor[1].Servo.Kfff = 0                                       ...
Motor[1].Servo.SwZvInt = 0                                    ...

Motor[1].pAbsPos = Gate3[0].Chan[0].SerialEncDataA.a          TYPICAL FOR SERIAL ABS. ENC.
Motor[1].AbsPosFormat = $1400                                 UNSIGNED ST BITS
Motor[1].AbsPosSF = Motor[1].PosSf                            SAME AS POSITION SCALE
Motor[1].HomeOffset = 0                                       USER CONFIGURABLE POS. OFFSET

Motor[1].pAbsPhasePos = Gate3[0].Chan[0].SerialEncDataA.a     TYPICAL FOR ABS. SERIAL ENC.
Motor[1].AbsPhasePosFormat = $1400                            USE ST DATA FOR ABS. PHASE
Motor[1].AbsPhasePosSf = 2048 * Mtr1PolePairs / Mtr1CtsPerRev                =2048*MOTOR POLE PAIRS/2^ST
//Motor[1].AbsPhasePosForce = Gate3[0].Chan[0].SerialEncDataA & $FFFFF       MASK ST BITS, TEST VALUE
Motor[1].AbsPhasePosForce = 69614                             STORE TEST VALUE
Motor[1].AbsPhasePosOffset =-Motor[1].AbsPhasePosForce * Motor[1].AbsPhasePosSf   TYPICAL SETTING
Motor[1].PowerOnMode = 0                                      NO POWER ON ABS READS
```

Special PLC for this Motor/Encoder Assembly

The first time this G5 encoder is supplied with power, it requires a special procedure to move to an index mark before the absolute position (and phase reference) can be read. The following is a sample PLC that performs this procedure.

```
GLOBAL PhaseHomeFlag = 0

OPEN PLC PhaseHomePLC
IF(PhaseHomeFlag == 1)
{
 LOCAL EncStatusWord
 LOCAL StepAngle = 1
 LOCAL StepMag = 1000

 // READ ENCODER STATUS WORD
 EncStatusWord = (Gate3[0].Chan[0].SerialEncDataA & $F00000) >> 20
 CALL DelayTimer.msec(10)

 // NOT INITIALIZED?
 IF(EncStatusWord > 0)
 {
     WHILE(EncStatusWord > 0 && Motor[1].AmpFault == 0)
     {
         IF(Motor[1].AmpEna == 0)
         {
             COUT 1:0
             WHILE(Motor[1].AmpEna == 0 && Motor[1].AmpFault == 0){}
         }
         Motor[1].IaBias = StepMag * SIND(StepAngle + 120)
         Motor[1].IbBias = StepMag * SIND(StepAngle)
         EncStatusWord = (Gate3[0].Chan[0].SerialEncDataA & $F00000) >> 20
         StepAngle++
     }
     CALL DelayTimer.msec(50)

     Motor[1].IaBias = 0
     Motor[1].IbBias = 0
     KILL 1
     CALL DelayTimer.msec(50)
 }

 // ABSOLUTE PHASE READ
 Motor[1].PhaseFindingStep = 1
 CALL DelayTimer.msec(100)
 WHILE(Motor[1].PhaseFound == 0){}

 // ABSOLUTE POSITION READ
 HOMEZ 1
 CALL DelayTimer.msec(100)
 WHILE(Motor[1].HomeComplete == 0){}

 PhaseHomeFlag = 0
}
CLOSE
```

A-3-3 OMRON 1S Servo Motor (ACC-84E)

Hardware Configuration

Item	Notes
Controller	This setup sample is applicable to UMAC ACC-24E3 w/ ACC-84E (1S protocol)
Amplifier	CK3A-G305L (110VAC main supply voltage)
Motor	1S Servo Motor R88M-1M10030S, 100VAC, 1.5/4.7A
Encoder	1S serial absolute 23-bit single-turn, 16-bit multi-turn

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)

Gate3[0].PhaseFreq = 10000                                             10 KHZ PHASE FREQUENCY
Gate3[0].ServoClockDiv = 1                                             5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                    2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq   TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)            TYPICAL CALCULATION

Gate3[0].Chan[0].PwmFreqMult = 1                                       CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PackInData = 0                                        CH1-4 IN DATA
Gate3[0].Chan[1].PackInData = 0                                        MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0                                       CH1-4 OUT DATA
Gate3[0].Chan[1].PackOutData = 0                                       MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

Gate3[0].AdcAmpStrobe = $901001                                        CH1-4 STROBE WORD (FOR CK3A)
Gate3[0].AdcAmpClockDiv = 5                                            CH1-4 3.125 MHZ ADC CLOCK
Gate3[0].AdcAmpHeaderBits = 4                                          CH1-4 MUST BE 4 (FOR CK3A)
Gate3[0].Chan[0].PwmDeadTime = 2 / 0.0533                              CH1 MIN. 2µSEC FOR G305

ACC84E[0].SerialEncCtrl = $E                                           ACC-84E CH1-4 FOR 1S
ACC84E[0].Chan[0].SerialEncCmd = $1400                                 ACC-84E CH1 SERIAL ENC. CMD FOR 1S
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                                   32-BIT READ
EncTable[1].pEnc = Gate3[0].Chan[0].SerialEncDataA.a
EncTable[1].pEnc1 = Sys.Pushm
EncTable[1].index1 = 8                                                 ACC-84 (LEFT SHIFT)
EncTable[1].index2 = 8                                                 ACC-84 (RIGHT SHIFT)
EncTable[1].index3 = 0
EncTable[1].index4 = 0
EncTable[1].index5 = 0
EncTable[1].index6 = 0
EncTable[1].ScaleFactor = 1 / EXP2(EncTable[1].index1)                 1/2^INDEX1 SCALE TO LSB
EncTable[1].MaxDelta = 0

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

Motor Setup and Commissioning

```
GLOBAL Ck3a1MaxAdc = 15.735                                   CK3A MAX ADC
GLOBAL Mtr1SingleTurn = 23                                    SINGLE TURN ST BITS
GLOBAL Mtr1MultiTurn = 16                                     MULTI TURN MT BITS
GLOBAL Mtr1CtsPerRev = EXP2(Mtr1SingleTurn)                   COUNTS PER REV. =2^ST
GLOBAL Mtr1PolePairs = 5                                      NUMBER OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 4.7                                    MAX RMS CUR.(MTR<CK3A)
GLOBAL Mtr1RatedRmsCur = 1.5                                  RATED RMS CUR.(MTR<CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2                                   TIME ALLOWED AT MAX CUR.

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 = Gate3[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 = 3                                         JOG SPEED, 3 DEG/MSEC

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

Motor[1].InPosBand = 0.050                                    INPOS BAND, [DEG]
Motor[1].InPosTime = 5 / Sys.ServoPeriod                      INPOS TIME, 5 MSEC

Motor[1].pPhaseEnc = ACC84E[0].Chan[0].SerialEncDataA.a       ACC-84 SERIAL ENCODER
Motor[1].PhaseCtrl = 4                                        TYPICAL FOR BRUSHLESS MOTOR
Motor[1].PhaseEncLeftShift = 9                                SHIFT LEFT TO MSB
Motor[1].PhaseEncRightShift = 8                               TYPICAL ACC-84
Motor[1].PhasePosSf = 2048 * Mtr1PolePairs / EXP2(32)         =2048*POLE PAIRS/2^(ST+LeftShift)

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)
Motor[1].IpbGain = 0                                          ...

Motor[1].Servo.Kp = 0                                         POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                       (MUST TUNE)
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 = ACC84E[0].Chan[0].SerialEncDataA.a         TYPICAL FOR SERIAL ABS. ENC.
Motor[1].AbsPosFormat = $01082708                             SIGNED ST+MT BITS
Motor[1].AbsPosSF = Motor[1].PosSf                            SAME AS POSITION SCALE
Motor[1].HomeOffset = 0                                       USER CONFIGURABLE POS. OFFSET

Motor[1].pAbsPhasePos = ACC84E[0].Chan[0].SerialEncDataA.a    TYPICAL FOR ABS. SERIAL ENC.
Motor[1].AbsPhasePosFormat = $1708                            USE ST DATA FOR ABS. PHASE
Motor[1].AbsPhasePosSf = 2048 * Mtr1PolePairs / Mtr1CtsPerRev        =2048*MOTOR POLE PAIRS/2^ST
//Motor[1].AbsPhasePosForce = ACC84E[0].Chan[0].SerialEncDataA & $7FFFFF   MASK ST BITS, TEST VALUE
Motor[1].AbsPhasePosForce = 1269945                           STORE TEST VALUE
Motor[1].AbsPhasePosOffset =-Motor[1].AbsPhasePosForce * Motor[1].AbsPhasePosSf   TYPICAL SETTING
Motor[1].PowerOnMode = $6                                     POWER-ON ABS. PHASE AND POS
```

A-3-4 Linear Servo Motor w/ Sinusoidal Encoder

Hardware Configuration

Item	Notes
Controller	This sample is applicable to UMAC ACC-24E3 (standard sinusoidal), and CK3M AX (sinusoidal)
Amplifier	CK3A-G310L (three-phase 208VAC main supply voltage)
Motor	Linear servo Motor, 200VAC, 2.6/8.3A, 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)

Gate3[0].PhaseFreq = 10000                                         10 KHZ PHASE FREQUENCY
Gate3[0].ServoClockDiv = 1                                         5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq   TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)        TYPICAL CALCULATION

Gate3[0].Chan[0].PwmFreqMult = 1                                   CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PackInData = 0                                    CH1-4 IN DATA
Gate3[0].Chan[1].PackInData = 0                                    MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0                                   CH1-4 OUT DATA
Gate3[0].Chan[1].PackOutData = 0                                   MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

Gate3[0].AdcAmpStrobe = $901001                                    CH1-4 STROBE WORD (FOR CK3A)
Gate3[0].AdcAmpClockDiv = 5                                        CH1-4 3.125 MHZ ADC CLOCK
Gate3[0].AdcAmpHeaderBits = 4                                      CH1-4 MUST BE 4 (FOR CK3A)
Gate3[0].Chan[0].PwmDeadTime = 3 / 0.0533                          CH1 MIN. 3µSEC FOR G310

Gate3[0].EncClockDiv = 3                                           ENC. SAMPLING 12.5MHZ
Gate3[0].AdcEncClockDiv = 3                                        ENC. ADC SAMPLING 12.5MHZ
//Gate3[0].AdcEncCtrl = $3FFFC000                                  UNCOMMENT FOR CK3M AX
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                  TYPICAL SINUSOIDAL ENCODER ENTRY
EncTable[1].pEnc = Gate3[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
EncTable[1].MaxDelta = 0

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

Motor Setup and Commissioning

```
GLOBAL Ck3a1MaxAdc = 31.470                              // CK3A MAX ADC
GLOBAL Mtr1Res = 0.020                                   // RESOLUTION, ENC. PITCH [mm]
GLOBAL Mtr1CtsPerMm = 16384 / 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].EncType = 6                                     // TYPICAL FOR SINUSOIDAL
Gate3[0].Chan[0].AtanEna = 1                             // TYPICAL FOR SINUSOIDAL
Gate3[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 = Gate3[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 = Gate3[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 * 16384) // 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)
Motor[1].IpbGain = 0                                     // ...

Motor[1].Servo.Kp = 0                                    // POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                  // (MUST TUNE)
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 = Gate3[0].Chan[0].Status.a        // HALLS ABS. PHASING
Motor[1].AbsPhasePosFormat = $400030C
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
```

A-3-5 Linear Servo Motor w/ BiSS Encoder (ACC-84E)

Hardware Configuration

Item	Notes
Controller	This sample is applicable to UMAC ACC-24E3 w/ ACC-84E(BiSS)
Amplifier	CK3A-G310L (three-phase 208VAC main supply voltage)
Motor	Linear servo Motor, 200VAC, 2.6/8.3A, 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)

Gate3[0].PhaseFreq = 10000                                           10 KHZ PHASE FREQUENCY
Gate3[0].ServoClockDiv = 1                                           5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                                                  2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 1000 * (Gate3[0].ServoClockDiv + 1) / Gate3[0].PhaseFreq   TYPICAL CALCULATION
Sys.PhaseOverServoPeriod = 1 / (Gate3[0].ServoClockDiv + 1)          TYPICAL CALCULATION

Gate3[0].Chan[0].PwmFreqMult = 1                                     CH1-4 10 KHZ PWM FREQUENCY
Gate3[0].Chan[1].PwmFreqMult = 1
Gate3[0].Chan[2].PwmFreqMult = 1
Gate3[0].Chan[3].PwmFreqMult = 1

Gate3[0].Chan[0].PackInData = 0                                      CH1-4 IN DATA
Gate3[0].Chan[1].PackInData = 0                                      MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackInData = 0
Gate3[0].Chan[3].PackInData = 0

Gate3[0].Chan[0].PackOutData = 0                                     CH1-4 OUT DATA
Gate3[0].Chan[1].PackOutData = 0                                     MUST BE 0 FOR CK3A
Gate3[0].Chan[2].PackOutData = 0
Gate3[0].Chan[3].PackOutData = 0

Gate3[0].AdcAmpStrobe = $901001                                      CH1-4 STROBE WORD (FOR CK3A)
Gate3[0].AdcAmpClockDiv = 5                                          CH1-4 3.125 MHZ ADC CLOCK
Gate3[0].AdcAmpHeaderBits = 4                                        CH1-4 MUST BE 4 (FOR CK3A)
Gate3[0].Chan[0].PwmDeadTime = 3 / 0.0533                            CH1 MIN. 3μSEC FOR G310

ACC84E[0].SerialEncCtrl = $31000B                                    SERIAL ENC. CTRL (BISS)
ACC84E[0].Chan[0].SerialEncCmd = $2114A0                             SERIAL ENC. COMMAND (32 BITS)
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 1                                     TYPICAL SERIAL ACC-84 ENTRY
EncTable[1].pEnc = ACC84E[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)
```

Motor Setup and Commissioning

```
GLOBAL Ck3a1MaxAdc = 31.470                          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 = Gate3[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 = ACC84E[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)
Motor[1].IpbGain = 0                                 ...

Motor[1].Servo.Kp = 0                                POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                              (MUST TUNE)
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 = ACC84E[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 = ACC84E[0].Chan[0].SerialEncDataA.a    TYPICAL FOR ABS. SERIAL ENC.
Motor[1].AbsPhasePosFormat = $00082008               USE 32 BITS FOR ABS. PHASE
Motor[1].AbsPhasePosSf = 2048 * Mtr1Res / Mtr1Ecl    TYPICAL FOR ABS. SERIAL ENC.
//Motor[1].AbsPhasePosForce =                        READ 32 BITS, TEST VALUE
//ACC84E[0].Chan[0].SerialEncDataA+(ACC84E[0].Chan[0].SerialEncDataB&$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 AND POS ON POWER-UP
```

A-4 Gate1 “Script” Motor Setup Samples

Brushless Motor setup samples w/ Gate1 (UMAC ACC-24E2) are provided in this section for reference.

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 Gate1, index 4, channel 0 for the Power PMAC section, and $78200 base address for the Turbo PMAC section.

A-4-1 Power PMAC with Gate1 (ACC-24E2)

Hardware Configuration

Item	Notes
Controller	This sample is applicable to Power PMAC UMAC with ACC-24E2
Amplifier	CK3A-G305L (single-phase 110VAC main supply voltage)
Motor	Rotary servo Motor, 48VDC, 3/9A, 2 pole pairs
Encoder	Digital quadrature, 500-line (2,000 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)

Gate1[4].PwmPeriod = 2948                       20 KHZ MAX PHASE FREQUENCY
Gate1[4].PhaseClockDiv = 1                      10 KHZ PHASE & PWM FREQUENCY
Gate1[4].ServoClockDiv = 1                      5 KHZ SERVO FREQUENCY
Sys.RtIntPeriod = 1                             2.5 KHZ RTI FREQUENCY
Sys.ServoPeriod = 0.2                           SERVO PERIOD [MSEC]
Sys.PhaseOverServoPeriod = 0.5                  PHASE OVER SERVO RATIO

Gate1[4].AdcStrobe = $901001                    CH1-4 STROBE WORD (FOR CK3A)
Gate1[4].HardwareClockCtrl = 2258               CH1-4 2.4576 MHZ ADC CLOCK (DEFAULT)
Gate1[4].Chan[0].PwmDeadTime = 2 / 0.135        CH1-4 2µSEC FOR G310
```

Encoder Conversion Table (ECT)

```
EncTable[1].Type = 3                            TYPICAL QUADRATURE ENCODER ENTRY
EncTable[1].pEnc = Gate1[4].Chan[0].ServoCapt.a
EncTable[1].pEnc1 = Gate1 [4].Chan[0].TimeBetweenCts.a
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 / 512
EncTable[1].MaxDelta = 0

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

Motor Setup and Commissioning

```
GLOBAL Ck3a1MaxAdc = 15.735                                       CK3A MAX ADC
GLOBAL Mtr1CtsPerRev = 2000                                       COUNTS PER REVOLUTION
GLOBAL Mtr1PolePairs = 4                                          NO. OF POLE PAIRS
GLOBAL Mtr1MaxRmsCur = 9                                          MAX RMS CUR.(MTR>CK3A)
GLOBAL Mtr1RatedRmsCur = 3                                        RATED RMS CUR.(MTR>CK3A)
GLOBAL Mtr1TimeAtMaxCur = 2                                       TIME ALLOWED AT MAX CUR.

Gate1[4].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 = Gate1[4].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 = 3                                             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 = 0.01                                         INPOS BAND, 0.05 DEGREES
Motor[1].InPosTime = 20 / Sys.ServoPeriod                         INPOS TIME, 20 MSEC

Motor[1].pPhaseEnc = Gate1[4].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 =-Gate1[4].PwmPeriod * 48 / 155                    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)
Motor[1].IpbGain = 0                                              ...

Motor[1].Servo.Kp = 0                                             POSITION SERVO LOOP GAINS
Motor[1].Servo.Kvfb = 0                                           (MUST TUNE)
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 = 4000 / (2 * Sys.ServoPeriod * (Sys.RtIntPeriod + 1))   STEPPER PHASING (4 SECONDS)
Motor[1].PhaseFindingDac = 0.5 * Motor[1].I2tSet                  50% OF I2TSET
Motor[1].AbsPhasePosOffset = 2048 / 5                             TYPICAL SETTING
Motor[1].PowerOnMode = 0                                          NO ABS. PHASE ON POWER-UP
```

A-4-2 Turbo PMAC with Gate1 (ACC-24E2)

Hardware Configuration

Item	Notes
Controller	This sample is applicable to Turbo PMAC UMAC with ACC-24E2
Amplifier	CK3A-G305L (single-phase 110VAC main supply voltage)
Motor	Rotary servo Motor, 48VDC, 3/9A, 2 pole pairs
Encoder	Digital quadrature, 500-line (2,000 counts per revolution with x4 decode)

System Gates

```
END GAT                                          END GATHER
DEL GAT                                          DELETE GATHER BUFFER
CLOSE                                            CLOSE ANY OPEN BUFFER

I7200 = 2947                                     20 KHZ MAX PHASE FREQUENCY
I7201 = 1                                        10 KHZ PHASE & PWM FREQUENCY
I7202 = 1                                        5 KHZ SERVO FREQUENCY
I10 = 1677368                                    SERVO INTERRUPT TIME

I7206 = $901001                                  CH1-4 STROBE WORD (FOR CK3A)
I7203 = 2258                                     CH1-4 2.4576 MHZ ADC CLOCK (DEFAULT)
I7204 = 2 / 0.135                                CH1-4 2µSEC FOR G310

#define MaxPhaseFreq    P8000 ; Max Phase Clock [KHz]
#define PWMClk          P8001 ; PWM Clock [KHz]
#define PhaseClk        P8002 ; Phase Clock [KHz]
#define ServoClk        P8003 ; Servo Clock [KHz]
MaxPhaseFreq = 117964.8 / (2 * I7200 + 3)
PWMClk = 117964.8 / (4 * I7200 + 6)
PhaseClk = MaxPhaseFreq / (I7201 + 1)
ServoClk = PhaseClk / (I7202 + 1)
```

Encoder Conversion Table (ECT)

```
I8000 = $78200                                   TYPICAL QUADRATURE ENCODER ENTRY
I8001 = 0                                        END OF ECT (FOR EFFICIENCY)
```

Motor Setup and Commissioning

```
I7210 = 3                                                              ENC. COUNTING DIRECTION
I100 = 1                                                               SERVOCTRL ENABLE
I101 = 1                                                               PHASECTRL ENABLE
I124 = $800001                                                         FLAG CONTROL

I184 = $FFFF00                                                         ADC MASK (FOR CK3A)
I172 = 683                                                             PHASE OFFSET, TYPICAL FOR BRUSHLESS
I166 = I7200 * 48 / 155                                                PWM SCALE FACTOR, MOTOR<SUPPLY VOLTAGE

I170 = 2                                                               NO. OF POLE PAIRS
I171 = 2000                                                            COUNTS PER REVOLUTION

I119 = 0.5                                                             BASIC MOTION SETTINGS
I120 = 100                                                             ...
I121 = 0
I122 = 32
I123 = 16
I115 = 0.5
I111 = 55 * 16
I112 = 0
I128 = 1 * 16

#define ContCurrent     3                                              I2T CALCULATION
#define PeakCurrent     9
#define MaxADC          15.735
#define I2TOnTime       2
I157 = INT(32767 * (ContCurrent * 1.414 / MaxADC) * COS(30))
I169 = INT(32767 * (PeakCurrent * 1.414 / MaxADC) * COS(30))
I158 = INT((I169*I169-I157*I157)*ServoClk*1000*I2TOnTime/(32767*32767))

I182 = $78216                                                          CURRENT LOOP ADDRESS
I129 = 0
I179 = 0
I161 = 0                                                               TUNE CURRENT LOOP
I162 = 0
I176 = 0

I130 = 0
I131 = 0                                                               TUNE POSITION LOOP
I132 = 0
I133 = 0
I134 = 0
I135 = 0
I136 = 0
I137 = 0
I138 = 0
I139 = 0
I168 = 0

I180 = 6                                                               4-GUESS PHASING METHOD
I173 = 1000
I174 = 60
```

A-5 CK3A Sample PLCs for Gate1

A-5-1 Power PMAC with Gate1

CK3A Definitions 4-Motors

```
// GATE3[0], CH. 1 – 4 AMPLIFIER FUNCTIONS ====================== //
GLOBAL Ck3a0Clrf, Ck3a0FanCtrl, Ck3a0BusDisCtrl, Ck3a0DbCtrl
PTR Gate30DataCtrl      ->U.IO:$6000B0.16.4
PTR Ck3a0BusDisState    ->U.IO:$6000B0.13.1
PTR Ck3a0ClrfBit        ->U.IO:$6000B0.24.1
PTR Ck3a0FanState       ->U.IO:$6000B0.25.1
PTR Ck3a0DbState        ->U.IO:$6000B0.28.1

// CH. 1 – 4 ADC B REGISTERS [15:08] ============================ //
PTR Ck3a1AdcBData ->U.IO:$600018.8.8 // Gate3[0].Chan[0].AdcAmp[1]
PTR Ck3a2AdcBData ->U.IO:$600058.8.8 // Gate3[0].Chan[1].AdcAmp[1]
PTR Ck3a3AdcBData ->U.IO:$600098.8.8 // Gate3[0].Chan[2].AdcAmp[1]
PTR Ck3a4AdcBData ->U.IO:$6000D8.8.8 // Gate3[0].Chan[3].AdcAmp[1]

// CH. 1 – 4 AMPLIFIER DATA ===================================== //
GLOBAL Ck3a1DcBus, Ck3a1Temp, Ck3a1FwVer, Ck3a1CurRating
GLOBAL Ck3a2DcBus, Ck3a2Temp, Ck3a2FwVer, Ck3a2CurRating
GLOBAL Ck3a3DcBus, Ck3a3Temp, Ck3a3FwVer, Ck3a3CurRating
GLOBAL Ck3a4DcBus, Ck3a4Temp, Ck3a4FwVer, Ck3a4CurRating

// CH. 1, Gate3[0].Chan[0].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a1PwrFlt         ->U.IO:$600014.8.1
PTR Ck3a1SoftStartFlt   ->U.IO:$600014.9.1
PTR Ck3a1StoStatus      ->U.IO:$600014.10.1
PTR Ck3a1ShuntShortFlt  ->U.IO:$600014.11.1
PTR Ck3a1OverVoltFlt    ->U.IO:$600014.12.1
PTR Ck3a1I2tFlt         ->U.IO:$600014.13.1
PTR Ck3a1ShortFlt       ->U.IO:$600014.14.1
PTR Ck3a1OverLoadFlt    ->U.IO:$600014.15.1

// CH. 2, Gate3[0].Chan[1].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a2PwrFlt         ->U.IO:$600054.8.1
PTR Ck3a2SoftStartFlt   ->U.IO:$600054.9.1
PTR Ck3a2StoStatus      ->U.IO:$600054.10.1
PTR Ck3a2ShuntShortFlt  ->U.IO:$600054.11.1
PTR Ck3a2OverVoltFlt    ->U.IO:$600054.12.1
PTR Ck3a2I2tFlt         ->U.IO:$600054.13.1
PTR Ck3a2ShortFlt       ->U.IO:$600054.14.1
PTR Ck3a2OverLoadFlt    ->U.IO:$600054.15.1

// CH. 3, Gate3[0].Chan[2].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a3PwrFlt         ->U.IO:$600094.8.1
PTR Ck3a3SoftStartFlt   ->U.IO:$600094.9.1
PTR Ck3a3StoStatus      ->U.IO:$600094.10.1
PTR Ck3a3ShuntShortFlt  ->U.IO:$600094.11.1
PTR Ck3a3OverVoltFlt    ->U.IO:$600094.12.1
PTR Ck3a3I2tFlt         ->U.IO:$600094.13.1
PTR Ck3a3ShortFlt       ->U.IO:$600094.14.1
PTR Ck3a3OverLoadFlt    ->U.IO:$600094.15.1

// CH. 4, Gate3[0].Chan[3].AdcAmp[0], STATUS SET 1 ============== //
PTR Ck3a4PwrFlt         ->U.IO:$6000D4.8.1
PTR Ck3a4SoftStartFlt   ->U.IO:$6000D4.9.1
PTR Ck3a4StoStatus      ->U.IO:$6000D4.10.1
PTR Ck3a4ShuntShortFlt  ->U.IO:$6000D4.11.1
PTR Ck3a4OverVoltFlt    ->U.IO:$6000D4.12.1
PTR Ck3a4I2tFlt         ->U.IO:$6000D4.13.1
PTR Ck3a4ShortFlt       ->U.IO:$6000D4.14.1
PTR Ck3a4OverLoadFlt    ->U.IO:$6000D4.15.1

// CH. 1, SYS.UDATA[256000] STATUS SET 2 ======================== //
PTR Ck3a1PwrStatus      ->U.USER:$FA000.0.1
PTR Ck3a1OverTempFlt    ->U.USER:$FA000.2.1
PTR Ck3a1ShuntOverLFlt  ->U.USER:$FA000.3.1
PTR Ck3a1PwmFreqFlt     ->U.USER:$FA000.4.1
PTR Ck3a1PwmCmdFlt      ->U.USER:$FA000.5.1
PTR Ck3a1AmpEna         ->U.USER:$FA000.6.1
```

```
$6000B0 is Gate1[i].AdcStrobe,
the address is found as
follows:
Gate1[i].AdcStrobe.a - Sys.piom

$600xx8 is ADC B
Gate1[i].Chan[j].Adc[1], the
addresses are found as follows:
Gate1[i].Chan[j].Adc[1].a -
Sys.piom

$600014 is ADC A
Gate1[i].Chan[j].Adc[0], the
addresses are found as follows:
Gate1[i].Chan[j].Adc[0].a -
Sys.piom

$600054 is ADC A
Gate1[i].Chan[j].Adc[0], the
addresses are found as follows:
Gate1[i].Chan[j].Adc[0].a -
Sys.piom

$600094 is ADC A
Gate1[i].Chan[j].Adc[0], the
addresses are found as follows:
Gate1[i].Chan[j].Adc[0].a -
Sys.piom

$6000D4 is ADC A
Gate1[i].Chan[j].Adc[0], the
addresses are found as follows:
Gate1[i].Chan[j].Adc[0].a -
Sys.piom

$FA000 is Sys.Udata[256000],
the address can be found as
follows:
Sys.Udata[i].a – Sys.pushm
```

```
// CH. 2, SYS.UDATA[256001] STATUS SET 2 ======================== //
PTR Ck3a2PwrStatus      ->U.USER:$FA004.0.1
PTR Ck3a2OverTempFlt    ->U.USER:$FA004.2.1
PTR Ck3a2ShuntOverLFlt ->U.USER:$FA004.3.1
PTR Ck3a2PwmFreqFlt     ->U.USER:$FA004.4.1
PTR Ck3a2PwmCmdFlt      ->U.USER:$FA004.5.1
PTR Ck3a2AmpEna         ->U.USER:$FA004.6.1

// CH. 3, SYS.UDATA[256002] STATUS SET 2 ======================== //
PTR Ck3a3PwrStatus      ->U.USER:$FA008.0.1
PTR Ck3a3OverTempFlt    ->U.USER:$FA008.2.1
PTR Ck3a3ShuntOverLFlt ->U.USER:$FA008.3.1
PTR Ck3a3PwmFreqFlt     ->U.USER:$FA008.4.1
PTR Ck3a3PwmCmdFlt      ->U.USER:$FA008.5.1
PTR Ck3a3AmpEna         ->U.USER:$FA008.6.1

// CH. 4, SYS.UDATA[256003] STATUS SET 2 ======================== //
PTR Ck3a4PwrStatus      ->U.USER:$FA00C.0.1
PTR Ck3a4OverTempFlt    ->U.USER:$FA00C.2.1
PTR Ck3a4ShuntOverLFlt ->U.USER:$FA00C.3.1
PTR Ck3a4PwmFreqFlt     ->U.USER:$FA00C.4.1
PTR Ck3a4PwmCmdFlt      ->U.USER:$FA00C.5.1
PTR Ck3a4AmpEna         ->U.USER:$FA00C.6.1

// CH. 1, SYS.UDATA[256040] STATUS SET 3 ======================== //
PTR Ck3a1I2tint         ->U.USER:$FA0A0.2.1
PTR Ck3a1AdcOffsetFlt   ->U.USER:$FA0A0.4.1
PTR Ck3a1Ready          ->U.USER:$FA0A0.6.1

// CH. 2, SYS.UDATA[256041] STATUS SET 3 ======================== //
PTR Ck3a2I2tint         ->U.USER:$FA0A4.2.1
PTR Ck3a2AdcOffsetFlt   ->U.USER:$FA0A4.4.1
PTR Ck3a2Ready          ->U.USER:$FA0A4.6.1

// CH. 3, SYS.UDATA[256042] STATUS SET 3 ======================== //
PTR Ck3a3I2tint         ->U.USER:$FA0A8.2.1
PTR Ck3a3AdcOffsetFlt   ->U.USER:$FA0A8.4.1
PTR Ck3a3Ready          ->U.USER:$FA0A8.6.1

// CH. 4, SYS.UDATA[256043] STATUS SET 3 ======================== //
PTR Ck3a4I2tint         ->U.USER:$FA0AC.2.1
PTR Ck3a4AdcOffsetFlt   ->U.USER:$FA0AC.4.1
PTR Ck3a4Ready          ->U.USER:$FA0AC.6.1
```

```
$FA004 is Sys.Udata[256001],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm

$FA008 is Sys.Udata[256002],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm

$FA00C is Sys.Udata[256003],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm

$FA0A0 is Sys.Udata[256040],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm

$FA0A4 is Sys.Udata[256041],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm

$FA0A8 is Sys.Udata[256042],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm

$FA0AC is Sys.Udata[256043],
the address can be found as follows:
Sys.Udata[i].a - Sys.pushm
```

CK3A Sample PLC 4-Motors

```
OPEN PLC Ck3aPLC
// CH. 1-4 STATUS BITS SET 2
Gate30DataCtrl = 0
CALL DelayTimer.msec(1)
Sys.Udata[256000] = Ck3a1AdcBData
Sys.Udata[256001] = Ck3a2AdcBData
Sys.Udata[256002] = Ck3a3AdcBData
Sys.Udata[256003] = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 STATUS BITS SET 3
Gate30DataCtrl = 1
CALL DelayTimer.msec(1)
Sys.Udata[256040] = Ck3a1AdcBData
Sys.Udata[256041] = Ck3a2AdcBData
Sys.Udata[256042] = Ck3a3AdcBData
Sys.Udata[256043] = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 DC BUS VOLTAGE
Gate30DataCtrl = 2
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1DcBus = Ck3a1AdcBData * 4
IF(Ck3a2Ready == 0) Ck3a2DcBus = Ck3a2AdcBData * 4
IF(Ck3a3Ready == 0) Ck3a3DcBus = Ck3a3AdcBData * 4
IF(Ck3a4Ready == 0) Ck3a4DcBus = Ck3a4AdcBData * 4
CALL DelayTimer.msec(1)
// CH. 1-4 TEMPERATURE
Gate30DataCtrl = 3
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1Temp = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2Temp = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3Temp = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4Temp = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 FIRMWARE VERSION
Gate30DataCtrl = 6
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1FwVer = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2FwVer = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3FwVer = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4FwVer = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// CH. 1-4 CURRENT RATING
Gate30DataCtrl = 7
CALL DelayTimer.msec(1)
IF(Ck3a1Ready == 0) Ck3a1CurRating = Ck3a1AdcBData
IF(Ck3a2Ready == 0) Ck3a2CurRating = Ck3a2AdcBData
IF(Ck3a3Ready == 0) Ck3a3CurRating = Ck3a3AdcBData
IF(Ck3a4Ready == 0) Ck3a4CurRating = Ck3a4AdcBData
CALL DelayTimer.msec(1)
// GATE3[0] CH. 1-4 BUS DISCHARGE CONTROL
IF(Ck3a0BusDisCtrl == 1 && Ck3a0BusDisState == 0) Ck3a0BusDisState = 1
IF(Ck3a0BusDisCtrl == 0 && Ck3a0BusDisState == 1) Ck3a0BusDisState = 0
// GATE3[0] CH. 1-4 FAN CTRL
IF(Ck3a0FanCtrl == 1 && Ck3a0FanState == 0) Ck3a0FanState = 1
IF(Ck3a0FanCtrl == 0 && Ck3a0FanState == 1) Ck3a0FanState = 0
// GATE3[0] CH1-4 DYNAMIC BRAKE CONTROL
IF(Ck3a0DbCtrl == 1 && Ck3a0DbState == 1) Ck3a0DbState = 0
IF(Ck3a0DbCtrl == 0 && Ck3a0DbState == 0) Ck3a0DbState = 1
// GATE3[0] CH. 1-4 CLEAR FAULT(S)
IF(Ck3a0Clrf == 1)
{
     Ck3a0ClrfBit = 1
     CALL DelayTimer.msec(10)
     Ck3a0ClrfBit = 0
     Ck3a0Clrf = 0
}
// CH. 1-4 LOGIC POWER OFF?
IF(Ck3a1Ready == 1) Ck3a1DcBus,4 = 0
IF(Ck3a2Ready == 1) Ck3a2DcBus,4 = 0
IF(Ck3a3Ready == 1) Ck3a3DcBus,4 = 0
IF(Ck3a4Ready == 1) Ck3a4DcBus,4 = 0
CLOSE
```

A-5-2 Turbo PMAC with Gate1

CK3A Definitions 4-Motors

```
// CH. 1-4 AMPLIFIER FUNCTIONS
#define Ck3a0Clrf                P1000
#define Ck3a0BusDisCtrl          P1001
#define Ck3a0FanCtrl             P1002
#define Ck3a0DbCtrl              P1003

#define Ck3a0BusDisState         P1004
#define Ck3a0FanState            P1005
#define Ck3a0DbState             P1006

// CH. 1-4 ADC B DATA REGISTERS
#define Ck3a1AdcBData            M1000
#define Ck3a2AdcBData            M1001
#define Ck3a3AdcBData            M1002
#define Ck3a4AdcBData            M1003

Ck3a1AdcBData->Y:$078206,0,8
Ck3a2AdcBData->Y:$07820E,0,8
Ck3a3AdcBData->Y:$078216,0,8
Ck3a4AdcBData->Y:$07821E,0,8

// CH. 1-4 DATA
#define Ck3a1DcBus               P1010
#define Ck3a1Temp                P1011
#define Ck3a1FwVer               P1012
#define Ck3a1CurRating           P1013

#define Ck3a2DcBus               P1014
#define Ck3a2Temp                P1015
#define Ck3a2FwVer               P1016
#define Ck3a2CurRating           P1017

#define Ck3a3DcBus               P1018
#define Ck3a3Temp                P1019
#define Ck3a3FwVer               P1020
#define Ck3a3CurRating           P1021

#define Ck3a4DcBus               P1022
#define Ck3a4Temp                P1023
#define Ck3a4FwVer               P1024
#define Ck3a4CurRating           P1025

// Ch. 1-4 STATUS BITS SET 1, ADC A (ANY STROBE)
#define Ck3a1PwrFlt              M1004
#define Ck3a1SoftStartFlt        M1005
#define Ck3a1StoStatus           M1006
#define Ck3a1ShuntShortFlt       M1007
#define Ck3a1OverVoltFlt         M1008
#define Ck3a1I2tFlt              M1009
#define Ck3a1ShortFlt            M1010
#define Ck3a1OverLoadFlt         M1011
Ck3a1PwrFlt->Y:$78205,0,1
Ck3a1SoftStartFlt->Y:$78205,1,1
Ck3a1StoStatus->Y:$78205,2,1
Ck3a1ShuntShortFlt->Y:$78205,3,1
Ck3a1OverVoltFlt->Y:$78205,4,1
Ck3a1I2tFlt->Y:$78205,5,1
Ck3a1ShortFlt->Y:$78205,6,1
Ck3a1OverLoadFlt->Y:$78205,7,1

#define Ck3a2PwrFlt              M1012
#define Ck3a2SoftStartFlt        M1013
#define Ck3a2StoStatus           M1014
#define Ck3a2ShuntShortFlt       M1015
#define Ck3a2OverVoltFlt         M1016
#define Ck3a2I2tFlt              M1017
#define Ck3a2ShortFlt            M1018
#define Ck3a2OverLoadFlt         M1019
Ck3a2PwrFlt->Y:$7820D,0,1
Ck3a2SoftStartFlt->Y:$7820D,1,1
Ck3a2StoStatus->Y:$7820D,2,1
Ck3a2ShuntShortFlt->Y:$7820D,3,1
Ck3a2OverVoltFlt->Y:$7820D,4,1
Ck3a2I2tFlt->Y:$7820D,5,1
Ck3a2ShortFlt->Y:$7820D,6,1
Ck3a2OverLoadFlt->Y:$7820D,7,1
```

```
#define Ck3a3PwrFlt            M1020
#define Ck3a3SoftStartFlt      M1021
#define Ck3a3StoStatus         M1022
#define Ck3a3ShuntShortFlt     M1023
#define Ck3a3OverVoltFlt       M1024
#define Ck3a3I2tFlt            M1025
#define Ck3a3ShortFlt          M1026
#define Ck3a3OverLoadFlt       M1027
Ck3a3PwrFlt->Y:$78215,0,1
Ck3a3SoftStartFlt->Y:$78215,1,1
Ck3a3StoStatus->Y:$78215,2,1
Ck3a3ShuntShortFlt->Y:$78215,3,1
Ck3a3OverVoltFlt->Y:$78215,4,1
Ck3a3I2tFlt->Y:$78215,5,1
Ck3a3ShortFlt->Y:$78215,6,1
Ck3a3OverLoadFlt->Y:$78215,7,1

#define Ck3a4PwrFlt            M1028
#define Ck3a4SoftStartFlt      M1029
#define Ck3a4StoStatus         M1030
#define Ck3a4ShuntShortFlt     M1031
#define Ck3a4OverVoltFlt       M1032
#define Ck3a4I2tFlt            M1033
#define Ck3a4ShortFlt          M1034
#define Ck3a4OverLoadFlt       M1035
Ck3a4PwrFlt->Y:$7821D,0,1
Ck3a4SoftStartFlt->Y:$7821D,1,1
Ck3a4StoStatus->Y:$7821D,2,1
Ck3a4ShuntShortFlt->Y:$7821D,3,1
Ck3a4OverVoltFlt->Y:$7821D,4,1
Ck3a4I2tFlt->Y:$7821D,5,1
Ck3a4ShortFlt->Y:$7821D,6,1
Ck3a4OverLoadFlt->Y:$7821D,7,1

// Ch. 1-4 STATUS BITS SET 2, ADC B (STROBE [11:8] = 0)
#define Ck3a1PwrStatus         M1036
#define Ck3a1OverTempFlt       M1037
#define Ck3a1ShuntOverLFlt     M1038
#define Ck3a1PwmFreqFlt        M1039
#define Ck3a1PwmCmdFlt         M1040
#define Ck3a1AmpEna            M1041
Ck3a1PwrStatus->X:$10F0,0,1
Ck3a1OverTempFlt->X:$10F0,2,1
Ck3a1ShuntOverLFlt->X:$10F0,3,1
Ck3a1PwmFreqFlt->X:$10F0,4,1
Ck3a1PwmCmdFlt->X:$10F0,5,1
Ck3a1AmpEna->X:$10F0,6,1

#define Ck3a2PwrStatus         M1042
#define Ck3a2OverTempFlt       M1043
#define Ck3a2ShuntOverLFlt     M1044
#define Ck3a2PwmFreqFlt        M1045
#define Ck3a2PwmCmdFlt         M1046
#define Ck3a2AmpEna            M1047
Ck3a2PwrStatus->X:$10F1,0,1
Ck3a2OverTempFlt->X:$10F1,2,1
Ck3a2ShuntOverLFlt->X:$10F1,3,1
Ck3a2PwmFreqFlt->X:$10F1,4,1
Ck3a2PwmCmdFlt->X:$10F1,5,1
Ck3a2AmpEna->X:$10F1,6,1

#define Ck3a3PwrStatus         M1048
#define Ck3a3OverTempFlt       M1049
#define Ck3a3ShuntOverLFlt     M1050
#define Ck3a3PwmFreqFlt        M1051
#define Ck3a3PwmCmdFlt         M1052
#define Ck3a3AmpEna            M1053
Ck3a3PwrStatus->X:$10F2,0,1
Ck3a3OverTempFlt->X:$10F2,2,1
Ck3a3ShuntOverLFlt->X:$10F2,3,1
Ck3a3PwmFreqFlt->X:$10F2,4,1
Ck3a3PwmCmdFlt->X:$10F2,5,1
Ck3a3AmpEna->X:$10F2,6,1
```

```
#define Ck3a4PwrStatus          M1054
#define Ck3a4OverTempFlt        M1055
#define Ck3a4ShuntOverLFlt      M1056
#define Ck3a4PwmFreqFlt         M1057
#define Ck3a4PwmCmdFlt          M1058
#define Ck3a4AmpEna             M1059
Ck3a4PwrStatus->X:$10F3,0,1
Ck3a4OverTempFlt->X:$10F3,2,1
Ck3a4ShuntOverLFlt->X:$10F3,3,1
Ck3a4PwmFreqFlt->X:$10F3,4,1
Ck3a4PwmCmdFlt->X:$10F3,5,1
Ck3a4AmpEna->X:$10F3,6,1

// Ch. 1-4 STATUS BITS SET 3, ADC B (STROBE [11:8] = 1)
#define Ck3a1I2tint             M1060
#define Ck3a1AdcOffsetFlt       M1061
#define Ck3a1Ready              M1062
Ck3a1I2tint->Y:$10F0,2,1
Ck3a1AdcOffsetFlt->Y:$10F0,4,1
Ck3a1Ready->Y:$10F0,6,1

#define Ck3a2I2tint             M1063
#define Ck3a2AdcOffsetFlt       M1064
#define Ck3a2Ready              M1065
Ck3a2I2tint->Y:$10F1,2,1
Ck3a2AdcOffsetFlt->Y:$10F1,4,1
Ck3a2Ready->Y:$10F1,6,1

#define Ck3a3I2tint             M1066
#define Ck3a3AdcOffsetFlt       M1067
#define Ck3a3Ready              M1068
Ck3a3I2tint->Y:$10F2,2,1
Ck3a3AdcOffsetFlt->Y:$10F2,4,1
Ck3a3Ready->Y:$10F2,6,1

#define Ck3a4I2tint             M1069
#define Ck3a4AdcOffsetFlt       M1070
#define Ck3a4Ready              M1071
Ck3a4I2tint->Y:$10F3,2,1
Ck3a4AdcOffsetFlt->Y:$10F3,4,1
Ck3a4Ready->Y:$10F3,6,1

// CH. 1-4 ADC B MIRROR WORDS
#define Ch1Mirror2              M1072
#define Ch1Mirror3              M1073
Ch1Mirror2->X:$10F0,0,8
Ch1Mirror3->Y:$10F0,0,8

#define Ch2Mirror2              M1074
#define Ch2Mirror3              M1075
Ch2Mirror2->X:$10F1,0,8
Ch2Mirror3->Y:$10F1,0,8

#define Ch3Mirror2              M1076
#define Ch3Mirror3              M1077
Ch3Mirror2->X:$10F2,0,8
Ch3Mirror3->Y:$10F2,0,8

#define Ch4Mirror2              M1078
#define Ch4Mirror3              M1079
Ch4Mirror2->X:$10F3,0,8
Ch4Mirror3->Y:$10F3,0,8
```

CK3A Sample PLC 4-Motors

```
// MOTORS 1-4 CK3A PLC
OPEN PLC 1 CLEAR
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
// STATUS BITS SET 2
I7206 = (I7206 & $FFF0FF) | $0
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
Ch1Mirror2 = Ck3a1AdcBData
Ch2Mirror2 = Ck3a2AdcBData
Ch3Mirror2 = Ck3a3AdcBData
Ch4Mirror2 = Ck3a4AdcBData
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// STATUS BITS SET 3
I7206 = (I7206 & $FFF0FF) | $100
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
Ch1Mirror3 = Ck3a1AdcBData
Ch2Mirror3 = Ck3a2AdcBData
Ch3Mirror3 = Ck3a3AdcBData
Ch4Mirror3 = Ck3a4AdcBData
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// DC BUS VOLTAGE
I7206 = (I7206 & $FFF0FF) | $200
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1DcBus = Ck3a1AdcBData * 4 ENDIF
IF(Ck3a2Ready = 0) Ck3a2DcBus = Ck3a2AdcBData * 4 ENDIF
IF(Ck3a3Ready = 0) Ck3a3DcBus = Ck3a3AdcBData * 4 ENDIF
IF(Ck3a4Ready = 0) Ck3a4DcBus = Ck3a4AdcBData * 4 ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// TEMPERATURE
I7206 = (I7206 & $FFF0FF) | $300
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1Temp = Ck3a1AdcBData ENDIF
IF(Ck3a2Ready = 0) Ck3a2Temp = Ck3a2AdcBData ENDIF
IF(Ck3a3Ready = 0) Ck3a3Temp = Ck3a3AdcBData ENDIF
IF(Ck3a4Ready = 0) Ck3a4Temp = Ck3a4AdcBData ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// FIRMWARE VERSION
I7206 = (I7206 & $FFF0FF) | $600
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1FwVer = Ck3a1AdcBData ENDIF
IF(Ck3a2Ready = 0) Ck3a2FwVer = Ck3a2AdcBData ENDIF
IF(Ck3a3Ready = 0) Ck3a3FwVer = Ck3a3AdcBData ENDIF
IF(Ck3a4Ready = 0) Ck3a4FwVer = Ck3a4AdcBData ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// CURRENT RATING
I7206 = (I7206 & $FFF0FF) | $700
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a1Ready = 0) Ck3a1CurRating = Ck3a1AdcBData ENDIF
IF(Ck3a2Ready = 0) Ck3a2CurRating = Ck3a2AdcBData ENDIF
IF(Ck3a3Ready = 0) Ck3a3CurRating = Ck3a3AdcBData ENDIF
IF(Ck3a4Ready = 0) Ck3a4CurRating = Ck3a4AdcBData ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// BUS DISCHARGE CONTROL
Ck3a0BusDisState = (I7206 & $20) / 32
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a0BusDisCtrl = 1 AND Ck3a0BusDisState = 0)
    I7206 = (I7206 & $FFFFDF) | $20
ENDIF
IF(Ck3a0BusDisCtrl = 0 AND Ck3a0BusDisState = 1)
    I7206 = (I7206 & $FFFFDF) | $0
ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// FAN CONTROL
Ck3a0FanState = (I7206 & $20000) / 131072
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a0FanCtrl = 1 AND Ck3a0FanState = 0)
    I7206 = (I7206 & $FDFFFF) | $20000
ENDIF
IF(Ck3a0FanCtrl = 0 AND Ck3a0FanState = 1)
    I7206 = (I7206 & $FDFFFF) | $0
ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
```

```
// DYNAMIC BRAKE CONTROL
Ck3a0DbState = (I7206 & $100000) / 1048576
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
IF(Ck3a0DbCtrl = 1 AND Ck3a0DbState = 1)
     I7206 = (I7206 & $EFFFFF) | $0
ENDIF
IF(Ck3a0DbCtrl = 0 AND Ck3a0DbState = 0)
     I7206 = (I7206 & $EFFFFF) | $100000
ENDIF
I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW

// CLEAR FAULT
IF(Ck3a0Clrf = 1)
     I7206 = (I7206 & $FEFFFF) | $10000
     I5111 = 10 * 8388608 / I10 WHILE (I5111 > 0) ENDW
     I7206 = (I7206 & $FEFFFF) | $0
     I5111 = 1 * 8388608 / I10 WHILE (I5111 > 0) ENDW
     Ck3a0Clrf = 0
ENDIF

// NOT READY (POWER OFF)?
IF(Ck3a1Ready = 1)
     Ck3a1DcBus = 0
     Ck3a1Temp = 0
     Ck3a1FwVer = 0
     Ck3a1CurRating = 0
ENDIF

IF(Ck3a2Ready = 1)
     Ck3a2DcBus = 0
     Ck3a2Temp = 0
     Ck3a2FwVer = 0
     Ck3a2CurRating = 0
ENDIF

IF(Ck3a3Ready = 1)
     Ck3a3DcBus = 0
     Ck3a3Temp = 0
     Ck3a3FwVer = 0
     Ck3a3CurRating = 0
ENDIF

IF(Ck3a4Ready = 1)
     Ck3a4DcBus = 0
     Ck3a4Temp = 0
     Ck3a4FwVer = 0
     Ck3a4CurRating = 0
ENDIF
CLOSE
```

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.
No. 438A Alexandra Road # 05-05/08 (Lobby 2),
Alexandra Technopark,
Singapore 119967
Tel: (65) 6835-3011/Fax: (65) 6835-2711

OMRON (CHINA) CO., LTD.
Room 2211, Bank of China Tower,
200 Yin Cheng Zhong Road,
PuDong New Area, Shanghai, 200120, China
Tel: (86) 21-5037-2222/Fax: (86) 21-5037-2200

Authorized Distributor:

Cat. No. O050-E1-01