ElectroCraft

PRO Series IMD

Programmable Integrated Motor-Drives

Programmable Integrated Motor Drives

ElectroCraft Document Number A11268

Technical Reference

© ElectroCraft 2014

ELECTROCRAFT

PRO Series IMD
Technical Reference

ElectroCraft Document Number
A11268

ElectroCraft
4480 Varsity Drive, Suite G
Ann Arbor, MI 48108

www.electrocraft.com

Read This First

While ElectroCraft believes that the information and guidance given in this manual is correct, all parties must rely upon their own skill and judgment when making use of it. ElectroCraft does not assume any liability to anyone for any loss or damage caused by any error or omission in the work, whether such error or omission is the result of negligence or any other cause. Any and all such liability is disclaimed.

All rights reserved. No part or parts of this document may be reproduced or transmitted in any form or by any means, electrical or mechanical including photocopying, recording or by any information-retrieval system without permission in writing from ElectroCraft, Inc.

The information in this document is subject to change without notice.

About This Manual

This book is a technical reference manual for the **PRO Series** programmable integrated motor-drive.

PRO Series IMD

In order to operate the **PRO Series IMD**, you need to perform the following 3 steps:

- ❑ **Step 1 Hardware installation**
- ❑ **Step 2 IMD setup** using the ElectroCraft **PROconfig** software for IMD commissioning
- ❑ **Step 3 Motion programming** using one of the options:
  - ❑ A **CANopen master**[1]
  - ❑ The IMD's **built-in motion controller** executing an ElectroCraft Motion PROgramming Language (**MPL**) program developed using ElectroCraft **MotionPRO Suite** software
  - ❑ A **MPL_LIB motion library for PCs** (Windows or Linux)
  - ❑ A **MPL_LIB motion library for PLCs**
  - ❑ A **distributed control** approach which combines the above options, like for example a host calling motion functions programmed on the IMDs in MPL

This manual covers **Step 1** in detail. It describes the **PRO Series IMD** hardware including the technical data, the connectors and the wiring diagrams needed for installation. The manual also presents an overview of the following steps, and includes the scaling factors between the real SI units and the IMD internal units. For detailed information regarding the next steps, refer to the related documentation.

Notational Conventions

This document uses the following conventions:

- **PRO Series IMD** – all products described in this manual
- **IU units** – Internal units of the IMD
- **SI units** – International standard units (meter for length, seconds for time, etc.)
- **MPL** – ElectroCraft Motion PROgramming Language
- **MPLCAN** – ElectroCraft protocol for exchanging MPL commands via CAN-bus

[1] when PRO Series IMD is set in CANopen mode

Related Documentation

Help Screens within the PROconfig software – describes how to use **PROconfig** to quickly setup any ElectroCraft PRO Series IMD for your application using only 2 dialogue boxes. The output of PROconfig is a set of setup data that can be downloaded into the IMD EEPROM or saved on a PC file. At power-on, the IMD is initialized with the setup data read from its EEPROM. With PROconfig it is also possible to retrieve the complete setup information from an IMD previously programmed. PROconfig is part of the ElectroCraft Motion PRO Suite. Motion PRO Suite is available as part of a PRO Series IMD Evaluation Kit. Please contact ElectroCraft or your local ElectroCraft sales representative for more information on obtaining MotionPRO Suite or an evaluation kit.

PRO Series CANOpen Programming Manual **(*Document No. A11226)*** – explains how to program the PRO Series family of programmable IMDs using **CANopen** protocol and describes the associated object dictionary for **CiA 301 v.4.2** application layer and communication profile, **CiA WD 305 v.2.2.13** layer settings services and protocols and **CiA DSP 402 v3.0** device profile for drives and motion control now included in IEC 61800-7-1 Annex A, IEC 61800-7-201 and IEC 61800-7-301 standards

Motion Programming using ElectroCraft MotionPRO Suite (Document **No. A11229*)*** – describes how to use the MotionPRO Suite to create motion programs using the ElectroCraft Motion PROgramming Language (MPL). The MotionPRO Suite platform includes **PROconfig** for the drive/motor setup, and a **Motion Wizard** for the motion programming. The Motion Wizard provides a simple, graphical way of creating motion programs and automatically generates all the MPL instructions. *With MotionPRO Suite you can fully benefit from a key advantage of ElectroCraft IMDs – their capability to execute complex moves without requiring an external motion controller, thanks to their built-in motion controller.* Motion PRO Suite is available as part of a PRO Series IMD Evaluation Kit. Please contact ElectroCraft or your local ElectroCraft sales representative for more information on obtaining MotionPRO Suite or an evaluation kit.

MPL_LIB v2.0 **(Document No. A11230)** – explains how to program in **C, C++,C#, Visual Basic or Delphi Pascal** a motion application for the ElectroCraft programmable IMDs using ElectroCraft Document Number A11230 motion control library for PCs. The MPL_LIB includes ready-to-run examples that can be executed on **Windows** or **Linux** (x86 and x64).

PRO Series and MPL_LIB_LabVIEW Compatibility **(Document No. A11231)** – explains how to program in **LabVIEW** a motion application for the ElectroCraft programmable IMDs using MPL_LIB_Labview v2.0 motion control library for PCs. The MPL_LIB_LabVIEW includes over 40 ready-to-run examples.

PRO Series and PLC Siemens Series S7-300 or S7-400 **(Document No. A11232)** – explains how to program in a PLC **Siemens series S7-300 or S7-400** a motion application for the ElectroCraft programmable IMDs using MPL_LIB_S7 motion control library. The MPL_LIB_S7 library is **IEC61131-3 compatible**.

PRO Series and PLC Omron Series MPL_LIB_CJ1 **(Document No. A11233)** – explains how to program in a PLC **Omron series CJ1** a motion application for the ElectroCraft programmable IMDs using MPL_LIB_CJ1 motion control library for PLCs. The MPL_LIB_CJ1 library is **IEC61131-3 compatible**.

MPL_LIB_X20 **(Document No. A11234)** – explains how to program in a PLC **B&R series X20** a motion application for the ElectroCraft programmable IMDs using MPL_LIB_X20 motion control library for PLCs. The MPL_LIB_X20 library is **IEC61131-3 compatible**.

***ElectroCAN* (Document No. A11235)** – presents ElectroCAN protocol – an extension of the CANopen communication profile used for MPL commands

If you Need Assistance …

If you want to …	Contact ElectroCraft at …
Visit ElectroCraft online	World Wide Web: www.electrocraft.com
Receive general information or assistance (see Note)	World Wide Web: www.electrocraft.com Email: drivesuport@electrocraft.com
Ask questions about product operation or report suspected problems (see Note)	Tel : +1 734.662-7771 Email: drivesuport@electrocraft.com
Make suggestions about, or report errors in documentation (see Note)	Mail: ElectroCraft Michigan 4480 Varsity Drive, Suite G Ann Arbor, MI 48108 USA

Table of Contents

This page is empty.

1. Safety information

Read carefully the information presented in this chapter before carrying out the IMD installation and setup! It is imperative to implement the safety instructions listed hereunder.

This information is intended to protect you, the IMD and the accompanying equipment during the product operation. Incorrect handling of the IMD can lead to personal injury or material damage.

Only qualified personnel may install, set up, operate and maintain the IMD. A “qualified person” has the knowledge and authorization to perform tasks such as transporting, assembling, installing, commissioning and operating IMDs.

The following safety symbols are used in this manual:

WARNING!	***SIGNALS A DANGER THAT MIGHT CAUSE BODILY INJURY TO THE OPERATOR. MAY INCLUDE INSTRUCTIONS TO PREVENT THIS SITUATION***

CAUTION!	***SIGNALS A DANGER FOR THE IMD, WHICH MIGHT DAMAGE THE PRODUCT OR OTHER EQUIPMENT. MAY INCLUDE INSTRUCTIONS TO AVOID THIS SITUATION***

1.1. Warnings

WARNING!	***TO AVOID ELECTRIC ARCING AND HAZARDS, NEVER PLUG / UNPLUG THE PRO Series IMD FROM IT’S SOCKET WHILE THE POWER SUPPLIES ARE ON !***

WARNING!	**THE IMD MAY HAVE HOT SURFACES DURING OPERATION.**

WARNING!	**DURING IMD OPERATION, THE CONTROLLED MOTOR WILL MOVE. KEEP AWAY FROM ALL MOVING PARTS TO AVOID INJURY**

1.2. Cautions

CAUTION!	**THE POWER SUPPLIES CONNECTED TO THE IMD MUST COMPLY WITH THE PARAMETERS SPECIFIED IN THIS DOCUMENT**

CAUTION!	**TROUBLESHOOTING AND SERVICING ARE PERMITTED ONLY FOR PERSONNEL AUTHORISED BY ELECTROCRAFT**

2. Product Overview

2.1. Introduction

The **PRO Series IMD** is a family of fully digital programmable servo motor-drives, based on the latest DSP technology and they offer unprecedented performance combined with an embedded motion controller.

All IMDs perform position, speed or torque control and work in single, multi-axis or stand-alone configurations. Thanks to the embedded motion controller, the PRO Series IMDs combine motion control and PLC functionality in a single compact unit and are capable to execute complex moves without requiring intervention of an external motion controller. Using the high-level ElectroCraft Motion PROgramming Language (**MPL**) the following operations can be executed directly at drive level:

- Setting various motion modes (profiles, PVT, PT, electronic gearing[1] or camming[1], etc.)
- Changing the motion modes and/or the motion parameters
- Executing homing sequences
- Controlling the program flow through:
  - Conditional jumps and calls of MPL functions
  - MPL interrupts generated on pre-defined or programmable conditions (protections triggered, transitions on limit switch or capture inputs, etc.)
  - Waits for programmed events to occur
- Handling of digital I/O and analog input signals
- Executing arithmetic and logic operations
- Performing data transfers between axes
- Controlling motion of an axis from another axis via motion commands sent between axes
- Sending commands to a group of axes (multicast). This includes the possibility to start simultaneously motion sequences on all the axes from the group
- Synchronizing all the axes from a network

By implementing motion sequences directly at IMD level you can distribute the intelligence between the master and the IMDs in complex multi-axis applications, reducing both the development time and the overall communication requirements. For example, instead of trying to command each movement of an axis, you can program the IMDs using MPL to execute complex motion tasks and inform the master when these tasks are done. Thus, for each axis control the master job may be reduced to: calling MPL functions stored in the IMD EEPROM and waiting for a message, which confirms the MPL functions execution completion.

All PRO Series IMDs are equipped with a serial RS232 and a CAN 2.0B interface and can be set (hardware, via a jumper) to operate in 2 modes:

- **CANopen**
- **MPLCAN**

When **CANopen** mode is selected, the PRO Series IMD conforms to **CiA 301 v4.2** application layer and communication profile, **CiA WD 305 v2.2.13** and **CiA DSP 402 v3.0** device profile for drives and motion control, now included in IEC 61800-7-1 Annex A, IEC 61800-7-201 and IEC 61800-7-301

[1] Available if the master axis sends its position via a communication channel

standards. In this mode, the PRO Series IMD may be controlled via a CANopen master. Additionally, PRO Series IMD offers a CANopen master the option to call motion sequences, written in MPL and stored in the IMD EEPROM, using manufacturer specific objects (see for details par. 5.3).

When **MPL CAN** mode is selected, the PRO Series IMD behaves as standard ElectroCraft programmable drive and conforms to ElectroCraft protocol for exchanging MPL commands via CAN-bus. When the MPLCAN protocol is used, it is not mandatory to have a master. Any PRO Series IMD can be set to operate standalone, and may play the role of a master to coordinate both network communication/synchronization and the motion application via MPL commands sent directly to the other IMDs.

When higher level coordination is needed, apart from a CANopen master, the PRO Series IMD IMDs can also be controlled via a PC or a PLC using one of the **MPL_LIB** motion libraries.

For PRO Series IMD commissioning **PROconfig** or **MotionPRO Developer** PC applications may be used.

PROconfig is a subset of MotionPRO Suite that includes only the drive setup part. The output of PROconfig is a set of setup data that can be downloaded into the IMD EEPROM or saved on a PC file. At power-on, the IMD is initialized with the setup data read from its EEPROM. With PROconfig it is also possible to retrieve the complete setup information from an IMD previously programmed. PROconfig shall be used for IMD setup in all cases where the motion commands are sent exclusively from a master. Hence neither the PRO Series IMD MPL programming capability nor the IMD camming mode are used.

MotionPRO Suite platform includes PROconfig for the IMD setup, and a **Motion Editor** for the motion programming. The Motion Editor provides a simple way of creating motion programs and automatically generates all the MPL instructions. *With MotionPRO Suite you can fully benefit from a key advantage of ElectroCraft IMDs – their capability to execute complex moves without requiring an external motion controller, thanks to their built-in motion controller*s. MotionPRO Developer, shall be used to program motion sequences in MPL. This is the PRO Series IMD typical operation mode when MPLCAN protocol is selected. MotionPRO Developer shall also be used with CANopen protocol, if the user wants to call MPL functions stored in the IMD EEPROM or to use the camming mode. With camming mode, MotionPRO Developer offers the possibility to quickly download and test a cam profile and also to create a **.sw** file (see par. 5.2.4) with the cam data. The .sw file can be afterwards stored in a master and downloaded to the IMD, wherever needed.

2.2. Key Features

- Fully integrated digital servo motor-drive suitable for the control of brushless and step motors
- Very compact design
- Sinusoidal (FOC) or trapezoidal (Hall-based) control of brushless motors
- Open or closed-loop control of 2 phase steppers
- Various modes of operation, including: torque, speed or position control; position or speed profiles, external analog reference or sent via a communication bus
- ElectroCraft Motion PROgramming Language (MPL) instruction set for the definition and execution of motion sequences
- Standalone operation with stored motion sequences
- Communication:
  - RS-232 serial up to 115kbits/s
  - CAN-Bus up to 1Mbit/s
- Digital and analog I/Os:
  - 5 digital inputs: 9-36 V, programmable polarity: sourcing/NPN: Enable, 2 Limit switches and 2 general-purpose
  - 2 digital outputs: 9-36 V, 0.5 A, sinking/NPN open-collector
  - 1 analog input: 12 bit, 0-5V: Reference, Feedback or general purpose

- Feedback devices provided:
  - Incremental quadrature encoder with BLDC and closed loop stepper models
  - Digital Hall sensors with BLDC models
- Various motion programming modes:
  - Position profiles with trapezoidal or S-curve speed shape
  - Position, Velocity, Time (PVT) 3rd order interpolation
  - Position, Time (PT) 1st order interpolation
  - Electronic gearing and camming[1]
  - 35 Homing modes
- hardware axis ID selection[2]
- Two operation modes selectable by jumper:
  - **CANopen** – conforming with **CiA 301 v4.2, CiA WD 305 v2.2.13** and **CiA DSP 402 v3.0**
  - **MPLCAN** – programmable drive conforming with ElectroCraft protocol for exchanging MPL commands via CAN-bus
- 2.5K $\times$ 16 internal SRAM memory
- 4K $\times$ 16 E^2ROM to store MPL programs and data
- PWM switching frequency up to 100kHz
- Motor supply: 11-48V
- Logic supply: 9-36V.Separate supply is optional
- Operating ambient temperature: 0-40°C (over 40°C with de-rating)
- Protection:
  - Under-voltage
  - Over-current
  - Over-temperature
  - Communication error
  - Control error

[1] Available if the master axis sends its position via a communication channel
[2] 6 hardware addresses in CANopen mode or 7 in MPLCAN mode

2.3. Electrical Specifications

All parameters measured under the following conditions (unless otherwise specified):

T_{amb} = 0…40°C, V_{LOG} = 24 V_{DC}; V_{MOT} = 48V_{DC}; Supplies start-up / shutdown sequence: -*any*-
Load current (sinusoidal amplitude / continuous BLDC, DC, stepper) = 8A

2.3.1. Operating Conditions

		Min.	Typ.	Max.	Units
Ambient temperature[1]		0		+40	ºC
Ambient humidity	Non-condensing	0		90	%Rh
Altitude	Altitude (referenced to sea level)	-500		3	Km

2.3.2. Storage Conditions

		Min.	Typ.	Max.	Units
Ambient temperature		-40		+85	ºC
Ambient humidity	Non-condensing	0		100	%Rh
Ambient Pressure		0.65		1.07	atm
ESD capability (Human body model)	Not powered; applies to any accessible part			±0.5	kV
	Original packaging			±15	kV

2.3.3. Mechanical Mounting

Airflow		natural convection[2], closed box			

2.3.4. Environmental Characteristics

		Min.	Typ.	Max.	Units
Power dissipation	Idle (no load)		2.8		W
Efficiency			98		%
Cleaning agents	Dry cleaning is recommended	Only Water- or Alcohol- based			
Protection degree	According to IEC60529, UL508	IP20			-

[1] Operating temperature can be extended up to **+65ºC** with reduced current and power ratings.
[2] It is recommended to mount the PRO Series IMD on a metallic support using the provided mounting holes, for better reliability and reduced de-rating due to heat dissipation

2.3.5. Logic Supply Input (+V_{LOG})

			Min.	Typ.	Max.	Units
Supply voltage	Nominal values		9		36	V_{DC}
	Absolute maximum values, drive operating but outside guaranteed parameters		6.5		39	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms) †		0		+45	V
Supply current	No Load on Digital Outputs	+V_{LOG} = 9V		190	320	mA
		+V_{LOG} = 12V		145	220	
		+V_{LOG} = 24V		100	150	
		+V_{LOG} = 40V		70	125	

2.3.6. Motor Supply Input (+V_{MOT})

		Min.	Typ.	Max.	Units
Supply voltage	Nominal values	11		50	V_{DC}
	Absolute maximum values, drive operating but outside guaranteed parameters	9		52	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms) †	-1		TBD	V
Supply current	Idle		1	5	mA
	Operating	-20	±8	+20	A
	Absolute maximum value, short-circuit condition (duration ≤ 10ms) †			26	A

2.3.7. Digital Inputs (IN0, IN1, IN2/LSP, IN3/LSN, IN4/Enable)

		Min.	Typ.	Max.	Units
Mode compliance		NPN			
Default state	Input floating (wiring disconnected)	Logic HIGH			
Input voltage	Logic "LOW"		0	1.6	V
	Logic "HIGH"	1.8	24	39	
	Floating voltage (not connected)		3.1		
	Absolute maximum, continuous	-10		+39	
	Absolute maximum, surge (duration ≤ 1S) †	-20		+40	
Input current	Logic "LOW"; Pulled to GND		0	0	mA
	Logic "HIGH"		2.9	3.4	
Input frequency		0		150	KHz
Minimum pulse width		3.3			µS
ESD protection	Human body model	±5			KV

2.3.8. Digital Outputs (OUT0, OUT1, OUT2/Error, OUT3/ Ready)

			Min.	Typ.	Max.	Units
Mode compliance	All outputs (OUT0, OUT1, OUT2/Error, OUT3/Ready)		TTL / CMOS / Open-collector / NPN 24V			
	Ready, Error Same as above + LVTTL (3.3V)		Ready, Error Same as above + LVTTL (3.3V)			
Default state	Not supplied (+V_{LOG} floating or to GND)		High-Z (floating)			
	Immediately OUT0, OUT1, OUT5 Logic “HIGH”	Immediately OUT0, OUT1, OUT5 Logic “HIGH”	Immediately OUT0, OUT1, OUT5 Logic “HIGH”			
		after power-up OUT2/Error, OUT3/ Ready Logic “LOW”	after power-up OUT2/Error, OUT3/ Ready Logic “LOW”			
	Normal operation	OUT0, OUT1, OUT2/Error, OUT5	Logic “HIGH”			
		OUT3/Ready	Logic “LOW”			
Output voltage	Logic “LOW”; output current = 0.5A		2.9	3	3.3	V
	Logic “HIGH”; output current = 0, no load	OUT2/Error, OUT3/ Ready	4	4.5	5	
		OUT0, OUT1				
	Logic “HIGH”, external load to +V_{LOG}			V_{LOG}		
	Absolute maximum, continuous		-0.5		V_{LOG} + 0.5	
	Absolute maximum, surge (duration ≤ 1S) †		-1		V_{LOG} + 1	
Output current	Logic “LOW”, sink current, continuous				0.5	A
	Logic “LOW”, sink current, pulse ≤ 5 sec.				1	A
	Logic “HIGH”, source current; external load to GND; V_{OUT} >= 2.0V	OUT2/Error, OUT3/ Ready			2	mA
		OUT0, OUT1			4	mA
	Logic “HIGH”, leakage current; external load to +V_{LOG}; V_{OUT} = V_{LOG} max = 40V			0.1	0.2	mA
Minimum pulse width			150			µS
ESD protection	Human body model		±5			KV

2.3.9. Analog 0…5V Input (REF)

		Min.	Typ.	Max.	Units
Input voltage	Operational range	0		4.95	V
	Absolute maximum values, continuous	-12		+18	
	Absolute maximum, surge (duration ≤ 1S) †			±36	
Input impedance	To GND		30		KΩ
Resolution		12			bits
Integral linearity				±2	bits
Offset error			±2	±10	bits
Gain error			±1%	±3%	% FS[1]
Bandwidth (-3dB)	Depending on software settings	0		1	KHz
ESD protection	Human body model	±5			KV

2.3.10. RS-232

		Min.	Typ.	Max.	Units
Standards compliance		TIA/EIA-232-C			
Bit rate	Depending on software settings	9600		115200	Baud
Short-circuit protection	232TX short to GND	Guaranteed			
ESD protection	Human body model	±2			KV

2.3.11. CAN-Bus

		Min.	Typ.	Max.	Units
Standards compliance		ISO11898, CiA 301v4.2, CiA WD 305 v2.2.13, CiA DSP402v3.0			
Bit rate	Depending on software settings	125		1000	Kbps
Bus length	1Mbps			25	m
	500Kbps			100	
	≤ 250Kbps			250	
Number of CAN nodes/drives				125	-
Termination resistor	Between CAN-Hi, CAN-Lo	none on-board			
Node addressing	Hardware: external resistor between AXISID (P2-2) and GND (P2-7)	4, 8, 12, 13, 14, 15 & LSS non-configured (CANopen); 4, 8, 12, 13, 14, 15 & 255 (TMLCAN)			
	Software	1 ÷ 127 (CANopen); 1- 255 (TMLCAN)			
ESD protection	Human body model	±15			KV

[1] "FS" stands for "Full Scale"

3. Step 1. Hardware Installation

3.1. Mechanical Mounting

The PRO Series IMD is intended to be mounted vertically or horizontally on a metallic support using the provided mounting holes and the recommended mating connectors, as specified in Appendix A. Fixing the PRO Series IMD onto a support using the provided mounting holes is strongly recommended to avoid vibration and shock problems.

3.1.1. Mounting

Fixing the PRO Series IMD onto a support using the provided mounting holes is strongly recommended to avoid vibration and shock problems.

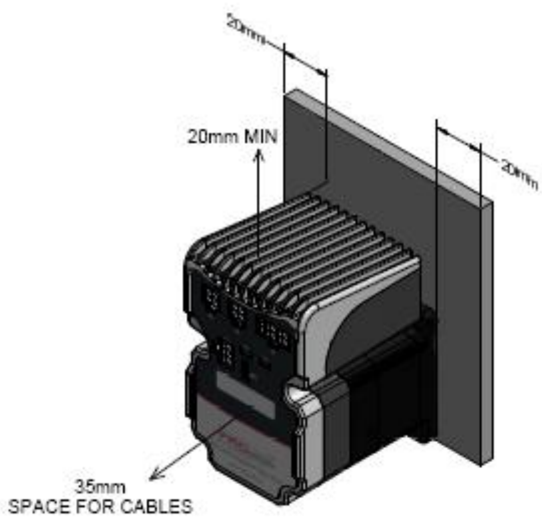


Figure 3.1 *Recommended spacing for mounted PRO Series IMD*

The PRO Series IMD drive(s) can be cooled by natural convection. The support can be mounted horizontally or vertically. In both cases, the air temperature must not exceed the limits indicated in section 2.3.4.

3.2. Mating Connectors

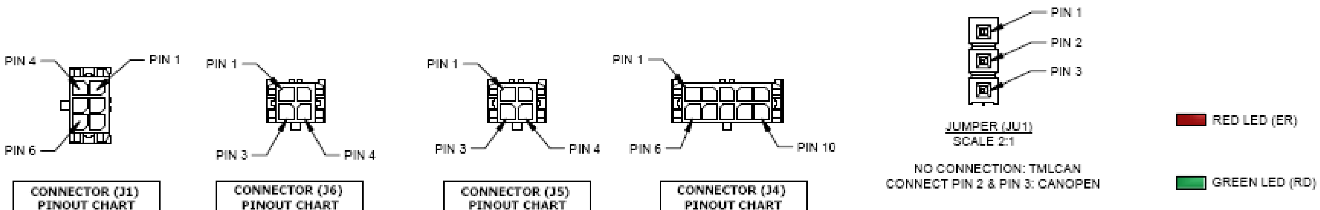
Connector	Description	Manufacturer	Part Number	Wire Gauge	Insulation Diameter
J1	MICROFIT RECEPTACLE HOUSING, 2x3 WAY	MOLEX	43025-0600	AWG 20..24	1.85 mm max.
J5,J6	MICROFIT RECEPTACLE HOUSING, 2x2 WAY	MOLEX	43025-0400	AWG 20..24	1.85 mm max.
J4	MICROFIT RECEPTACLE HOUSING, 2x5 WAY	MOLEX	43025-1000	AWG 20..24	1.85 mm max.
J1,J4,J5,J6	CRIMP PIN, MICROFIT, 5A	MOLEX	43030-0007	AWG 20..24	1.85 mm max.

3.3. Connectors and Connection Diagrams

3.3.1. Connector Layout

Figure 3.2: PRO Series IMD size 17 and size 23/24 connector layout.

3.3.2. Connector Pinout



CONNECTOR (J1) PINOUT CHART	
PIN NO.	FUNCTION
1	232RX
2	Vlog
3	Vmot
4	232TX
5	GND
6	GND

CONNECTOR (J6) PINOUT CHART	
PIN NO.	FUNCTION
1	Vlogic
2	GND
3	CAN H
4	CAN L

CONNECTOR (J5) PINOUT CHART	
PIN NO.	FUNCTION
1	Vlogic
2	GND
3	CAN H
4	CAN L

CONNECTOR (J4) PINOUT CHART	
PIN NO.	FUNCTION
1	STATUS
2	N/C
3	GPoØ
4	GPiØ
5	LSN
6	ENABLE
7	GND
8	REF
9	GPi1
10	LSP

Figure 3.3: PRO Series IMD connector pinouts and LED indicators.

3.3.3. J1 Power supply input connector pinout

	Pin	Name	Type	Description
Connector description				
J1	**1**	**232RX**	**I**	RS-232 Data Reception
	2	**$+V_{LOG}$**	**I**	Positive terminal of the logic supply: 9 to $36V_{DC}$ / If not connected, the logic supply is automatically routed from J1 pin 2[1]
	3	**$+V_{MOT}$**	**I**	Positive terminal of the motor supply: 9 to $36V_{DC}$. Feeds the positive terminal of the logic supply if J4 pin 7 not connected separately
	4	**232TX**	O	RS-232 Data Transmission
	5	**GND**	-	Negative return (ground) of the power supplies $+V_{MOT}$ and $+V_{LOG}$ and RS-232 pins
	6	**GND**	-	Negative return (ground) of the power supplies $+V_{MOT}$ and $+V_{LOG}$ and RS-232 pins

3.3.4. J4 signal inputs and outputs connector pinout

	Pin	Name	Type	Description
Connector description				
J4	**1**	**GPO5**	**O**	5-36V 0.5A, general-purpose digital NPN output
	2	**AXISID**	**I**	Connect resistor between AXISID and GND to set axis ID in hardware
	3	**GPO0**	**O**	5-36V 0.5A, general-purpose digital NPN output
	4	**GPI0**	**I**	5-36V general-purpose digital PNP input
	5	**LSN**	**I**	5-36V digital input. Negative limit switch input
	6	**ENABLE**	**I**	5-36V digital input. Drive enable input
	7	**GND**	-	Return ground for I/O pins
	8	**REF**	**I**	Analog input, 12-bit, 0-5V. Used to read an analog position, speed or torque reference, or used as general purpose analogue input
	9	**GPI1**	**I**	5-36V general-purpose digital PNP input
	10	**LSP**	**I**	5-36V digital PNP input. Positive limit switch input

3.3.5. J5, J6 CAN connectors pinout

	Pin	Name	Type	Description
Connector description				
J5, J6	**1**	**$+V_{LOG}$**	**O**	Positive terminal of the logic supply: 9 to $36V_{DC}$
	2	**GND**	-	Return ground for CAN-Bus
	3	**Can-Hi**	**I/O**	CAN-Bus positive line (dominant high)
	4	**Can-Lo**	**I/O**	CAN-Bus negative line (dominant low)

1 In case $+V_{LOG}$ (J4 pin7) is not connected, the digital outputs and inputs will not be operational.

3.3.6. Digital I/O Connection

3.3.7. 5V Digital NPN I/O Connection

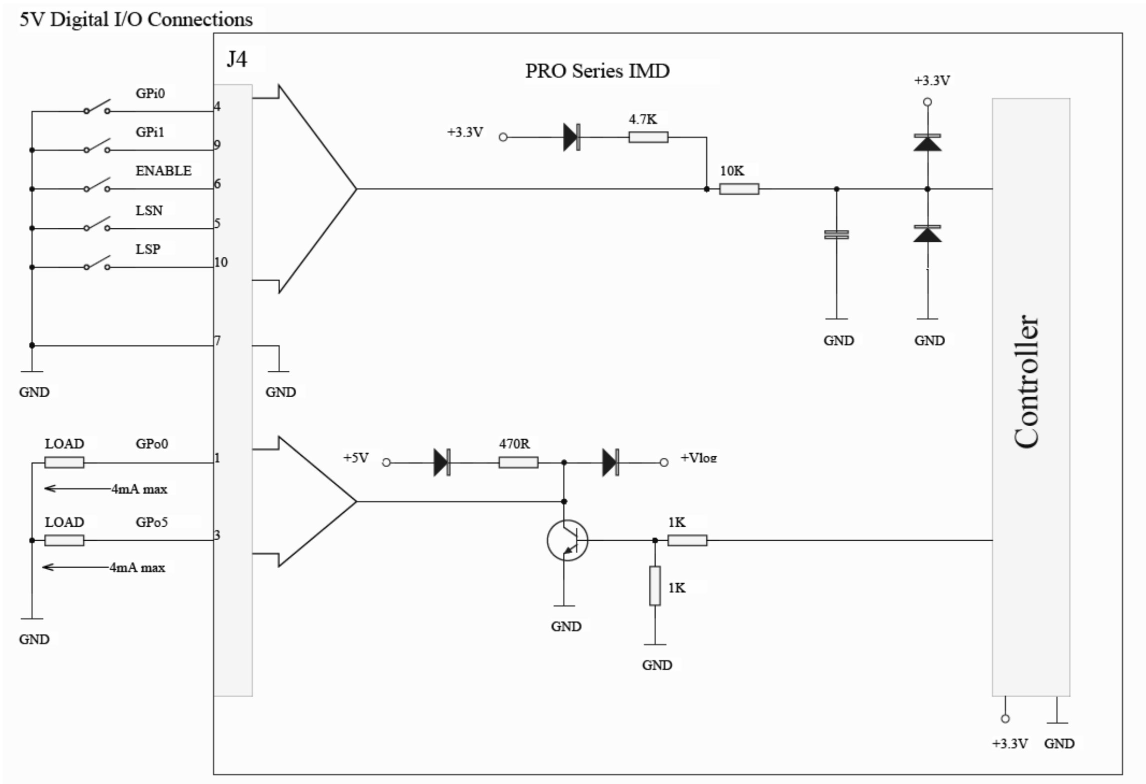


Figure 3.2. *5V Digital I/O connection*

3.3.8. Vlog Digital NPN I/O Connection

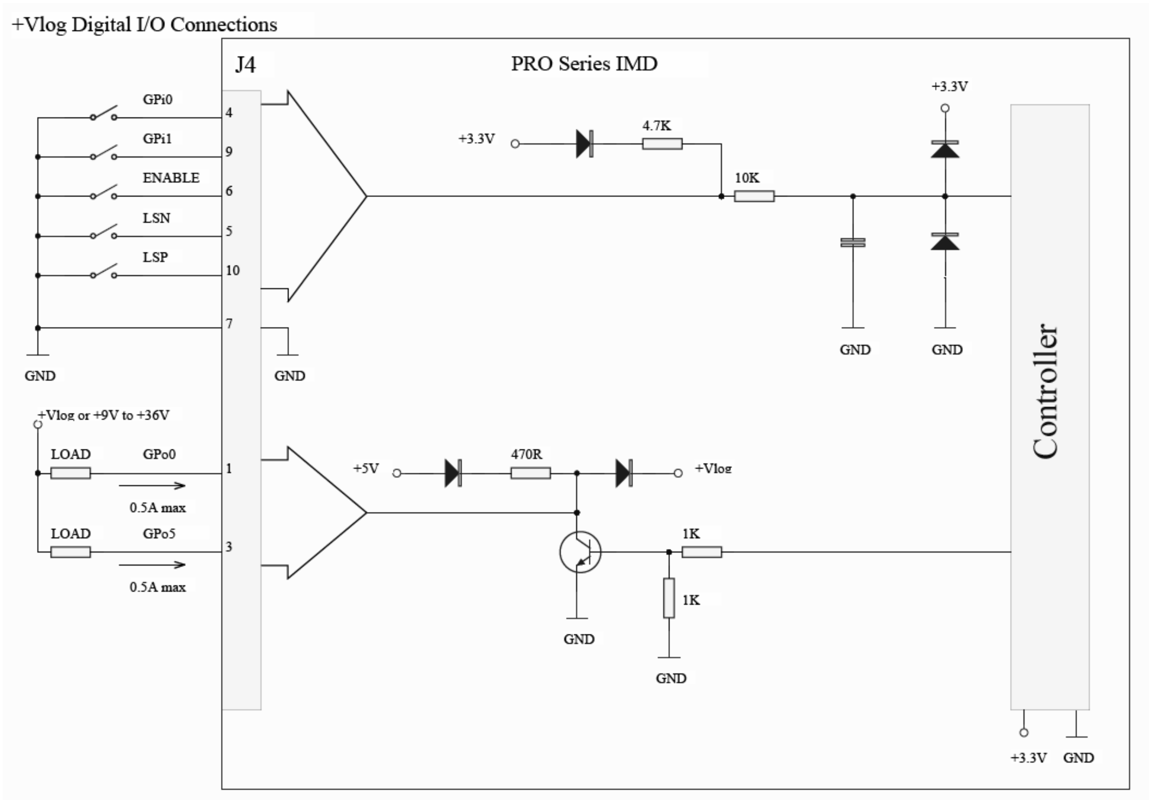


Figure 3.9. *+Vlog Digital I/O connection*

Remarks:

1. *The inputs are compatible with TTL(5V), LVTTL(3.3V), CMOS and open collector outputs*
2. *The outputs are compatible with TTL (5V) inputs*
3. *The output loads can be individually and independently connected to +5V or to GND.*

3.3.9. Analog Inputs Connection

3.3.9.1 0-5V Input Range

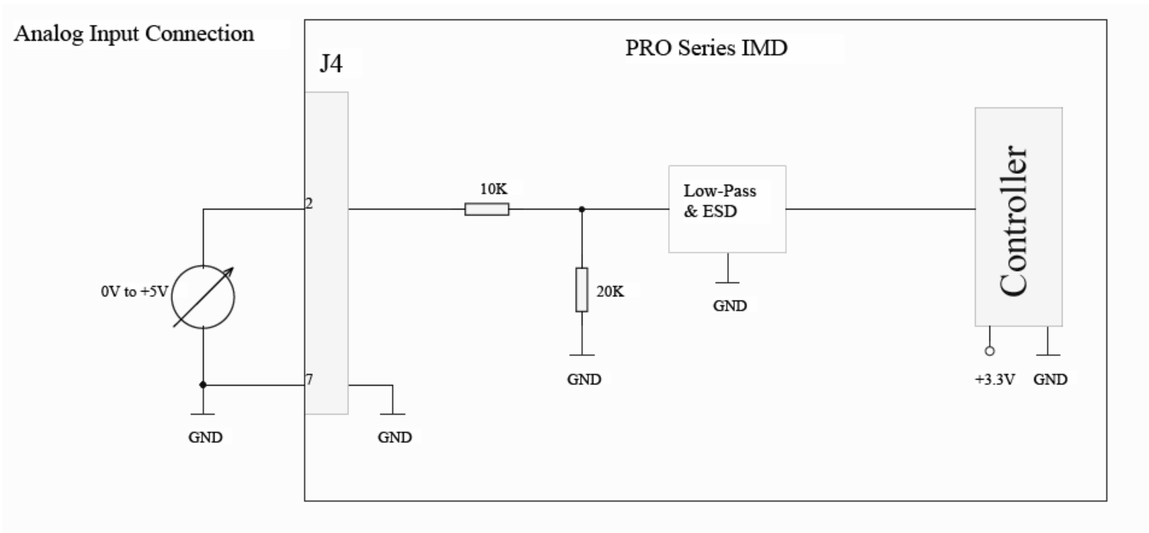


Figure 3.10. *Analog inputs connection*

3.3.9.2 Recommendation for wiring

a) If the analog signal source is single-ended, use a 2-wire twisted shielded cable as follows: 1st wire connects the live signal to the drive input; 2nd wire connects the source ground to the drive ground; shield will be connected to the drive ground terminal.

b) If the analog signal source is differential and the signal source ground is isolated from the drive GND, use a 2-wire twisted shielded cable as follows: 1st wire connects the source plus (positive, in-phase) to the drive analog input; 2nd wire connects the source minus (negative, out-of-phase) to the drive ground (GND). Shield is connected only at the drive side, to the drive GND, and is left unconnected at the source side.

c) If the analog signal source is differential and the signal source ground is common with the drive GND, use a 2-wire shielded cable as follows: 1st wire connects the source plus (positive, in-phase) to the drive analog input; 2nd wire connects the source ground to the drive ground (GND); shield is connected only at the drive side, to the drive GND, and is left unconnected at the source side. The source minus (negative, out-of-phase) output remains unconnected.

3.3.10. Power Supply Connection

3.3.10.1 Supply Connection

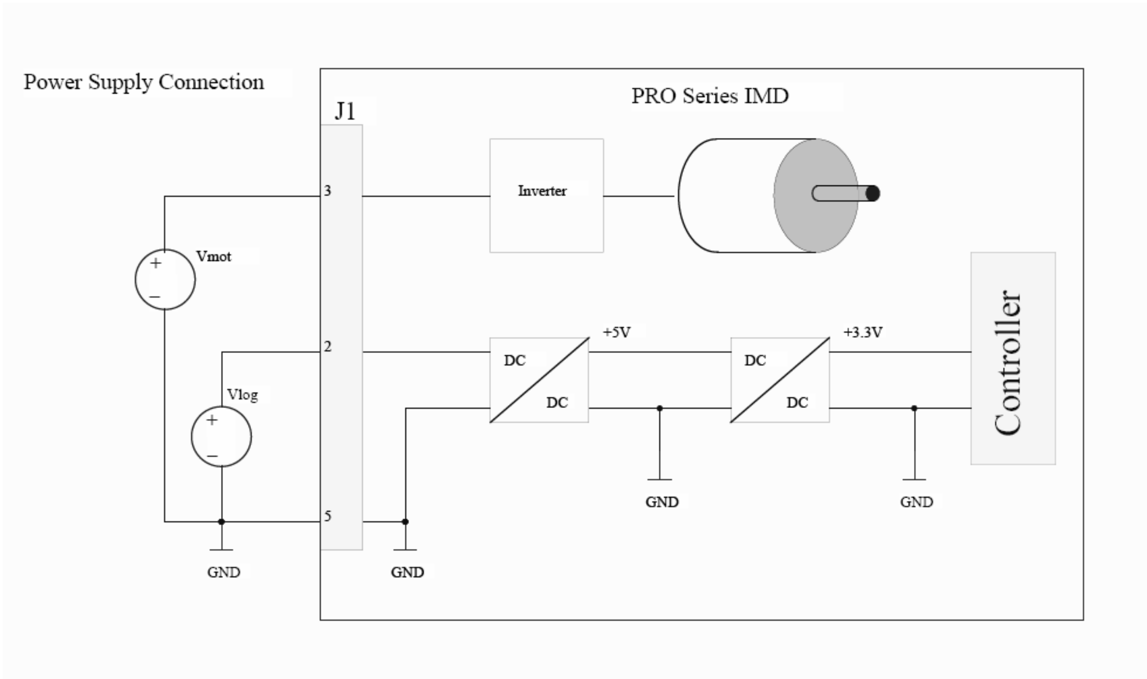


Figure 3.3. *Supply connection*

3.3.10.2 Recommendations for Supply Wiring

The PRO Series IMD always requires two supply voltages: V_{log} and V_{mot}.
The logic power supply can be connected to either J1 pin 2, J5 pin 1 or J6 pin 1. All three input pins are connected together inside the PRO Series IMD.
Use short, thick wires between the PRO Series IMD and the motor power supply. Connect power supply wires to all the indicated pins. If the wires are longer than 2 meters, use twisted wires for the supply and ground return. For wires longer than 20 meters, add a capacitor of at least 4,700µF (rated at an appropriate voltage) right on the terminals of the PRO Series IMD.
It is recommended to connect the negative motor supply return (GND) to the Earth protection near the power supply terminals.

3.3.10.3 Recommendations to limit over-voltage during braking

During abrupt motion brakes or reversals the regenerative energy is injected into the motor power supply. This may cause an increase of the motor supply voltage (depending on the power supply characteristics). If the voltage bypasses 54V, the drive over-voltage protection is triggered and the drive power stage is disabled. In order to avoid this situation a capacitor big enough to absorb the overall energy flowing back to the supply may be added. The capacitor must be rated to a voltage equal or bigger than the maximum expected over-voltage and can be sized with the formula:

$$C \geq \frac{2 \times E_M}{U_{MAX}^2 - U_{NOM}^2}$$

where:

U_{MAX} = 54V is the over-voltage protection limit
U_{NOM} is the nominal motor supply voltage
E_M = the overall energy flowing back to the supply in Joules. In case of a rotary motor and load, E_M can be computed with the formula:

$$E_M = \underbrace{\frac{1}{2}(J_M + J_L)\varpi_M^2}_{\textit{Kinetic energy}} + \underbrace{(m_M + m_L)g(h_{initial} - h_{final})}_{\textit{Potential energy}} - \underbrace{3I_M^2 R_{Ph} t_d}_{\textit{Copper losses}} - \underbrace{\frac{t_d \varpi_M}{2} T_F}_{\textit{Friction losses}}$$

where:

- J_M – total rotor inertia [kgm^2]
- J_L – total load inertia as seen at motor shaft after transmission [kgm^2]
- ϖ_M – motor angular speed before deceleration [rad/s]
- m_M – motor mass [kg] – when motor is moving in a non-horizontal plane
- m_L – load mass [kg] – when load is moving in a non-horizontal plane
- g – gravitational acceleration i.e. 9.8 [m/s^2]
- $h_{\mathbf{initial}}$ – initial system altitude [m]
- $h_{\mathbf{final}}$ – final system altitude [m]
- I_M – motor current during deceleration [A_{RMS}/phase]
- R_{Ph} – motor phase resistance [Ω]
- $t_{\mathbf{d}}$ – time to decelerate [s]
- T_F – total friction torque as seen at motor shaft [Nm] – includes load and transmission

In case of a linear motor and load, the motor inertia J_M and the load inertia J_L will be replaced by the motor mass and the load mass measured in [kg], the angular speed ϖ_M will become linear speed measured in [m/s] and the friction torque T_F will become friction force measured in [N].

3.3.11. Serial RS-232 connection

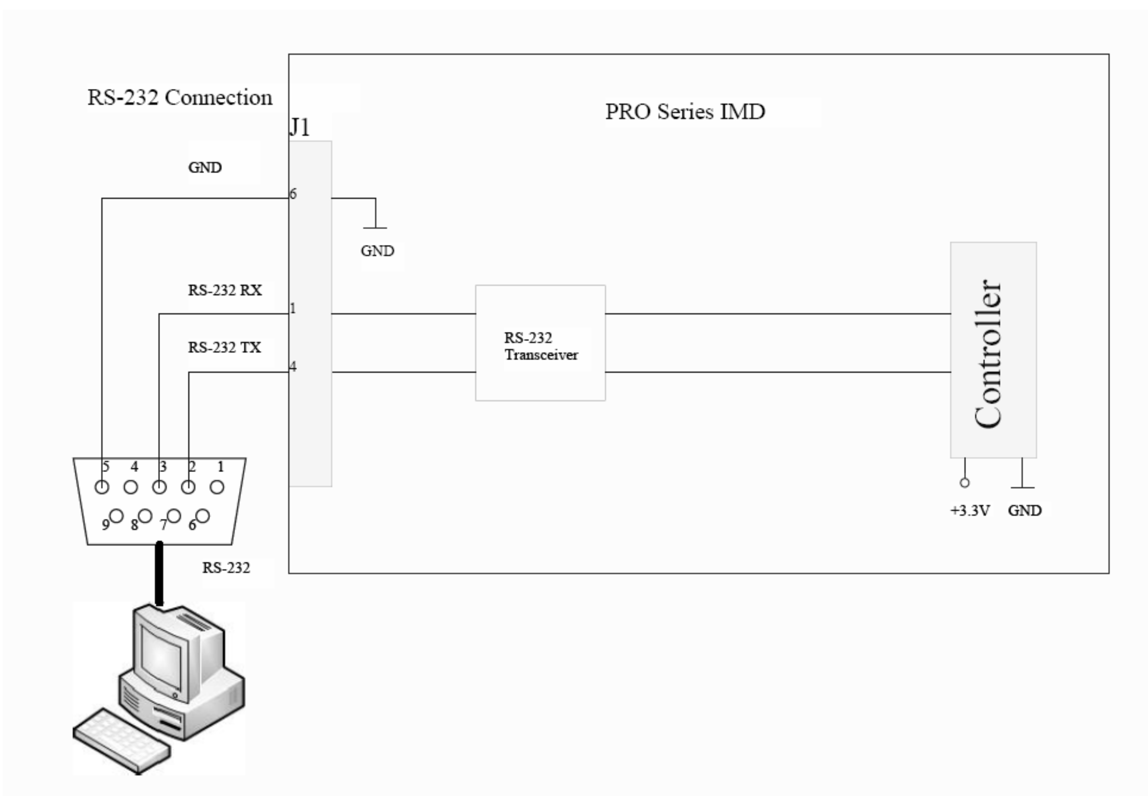


Figure 3.4. *Serial RS-232 connection*

3.3.11.1 Recommendation for wiring

a) If you build the serial cable, you can use a 3-wire shielded cable with shield connected to BOTH ends. Do not use the shield as GND. The ground wire (pin 5 or 6 of J1) must be included inside the shield, like the 232Rx and 232Tx signals

b) Always power-off all the PRO Series IMD supplies before inserting/removing the RS-232 serial connector

c) Do not rely on an earthed PC to provide the PRO Series IMD GND connection! The drive must be earthed through a separate circuit. Most communication problems are caused by the lack of such connection

3.3.12. CAN-bus connection

3.3.12.1 CAN connection

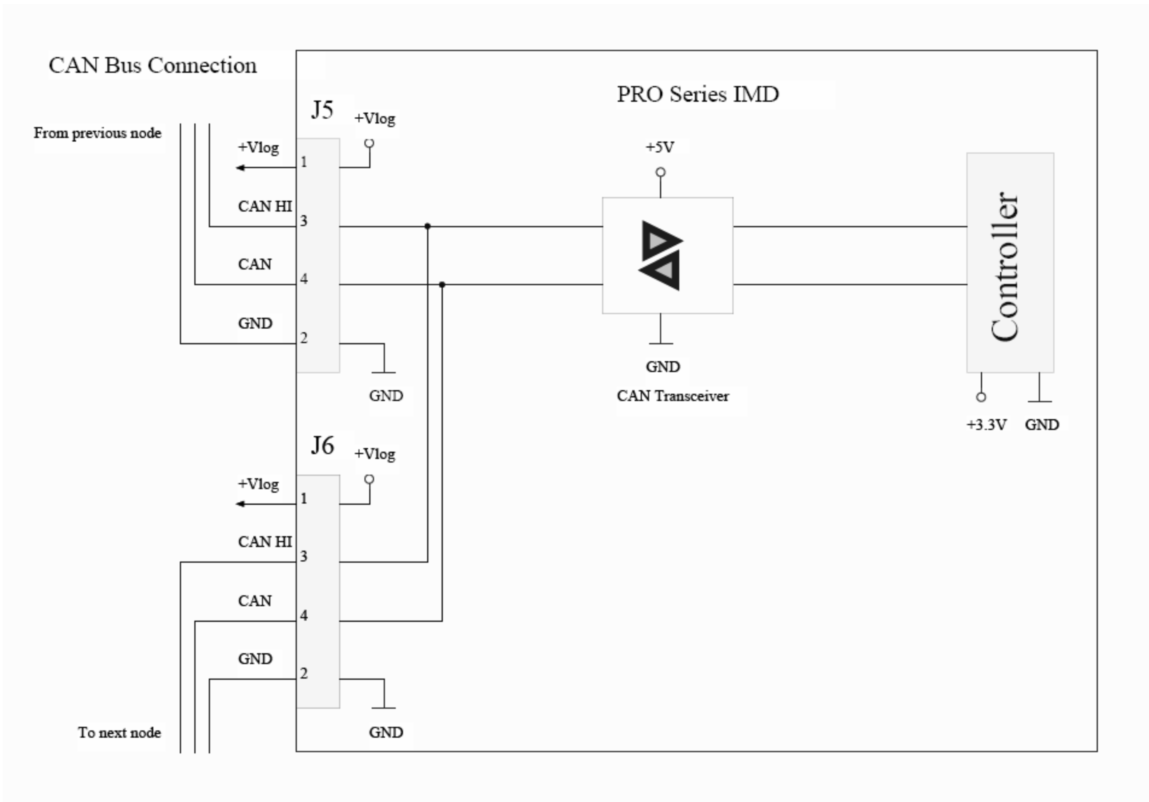


Figure 3.5. *CAN connection*

Remarks:

1. *The CAN network requires a 120-Ohm terminator. This is not included in the drive. Figure 3.6 shows how to connect it on your network.*
2. *CAN signals are not insulated from other PRO Series IMD circuits.*

3.3.12.2 Recommendation for wiring

a) Build CAN network using cables with twisted wires (2 wires/pair), with CAN-Hi twisted together with CAN-Lo. It is recommended but not mandatory to use a shielded cable. If so, connect the shield to GND. The cable impedance must be 105 ... 135 ohms (120 ohms typical) and a capacitance below 30pF/meter.

b) The 120Ω termination resistors must be rated at 0.2W minimum. Do not use winded resistors, which are inductive.

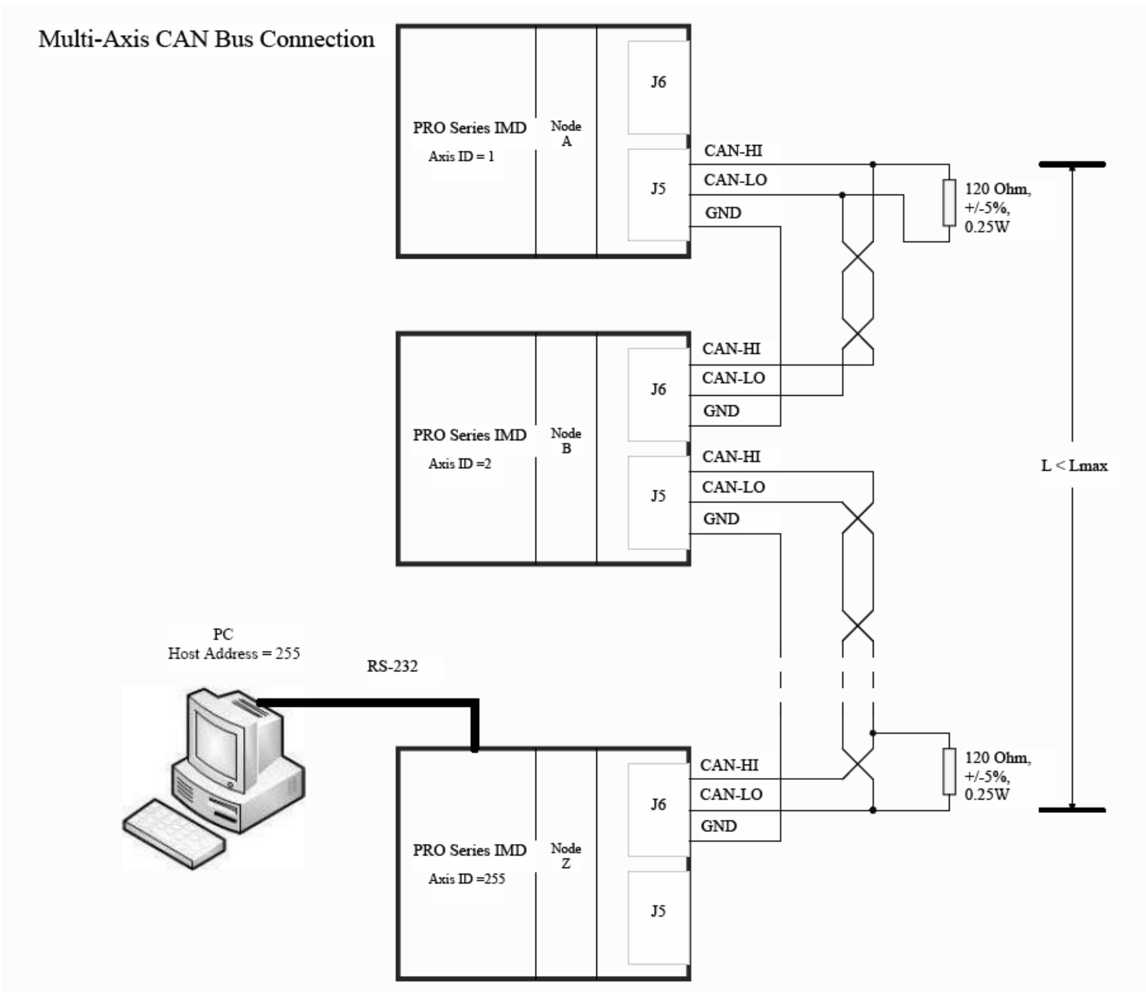


Figure 3.6. *Multiple-Axis CAN network*

Remarks:

1. *The axis IDs in* ***Figure 3.6****, are valid for MPLCAN mode. For CANopen mode, the highest axis ID a drive can have is 127.*
2. *Lmax is the bus length defined in paragraph* ***2.3.11.***

3.3.13. Disabling Autorun Mode

When a PRO Series IMD is set in MPLCAN operation mode, by default after power-on it enters automatically in *Autorun* mode. In this mode, if the drive has in its local EEPROM a valid MPL application (motion program), this is automatically executed as soon as the motor supply V_{MOT} is turned on.

To disable *Autorun* mode write value 0x0001 in first EEPROM location at address 0x4000 in software, this will invalidate the MPL program. On next power on, in absence of a valid MPL application, the drive enters the *non-Autorun/slave* mode.

3.4. Operation Mode and Axis ID Selection

3.4.1. Selection of the Operation Mode

On the PRO Series IMD, the selection of the operation mode CANopen or MPLCAN is done by setting the JU1 jumper:

- MPLCAN mode, JU1 = 1-2
- CANopen mode, JU1 = 2-3

3.4.2. Selection of the Axis ID

The axis ID can be selected by using either hardware or software settings. The hardware Axis ID selection is done by connecting a resistor between AXISID (J4 pin 2) and GND (J4 pin 7). The resistance is then measured by the PRO Series IMD at power-on and the axis ID set according to the following table:

Resistance (Ohm)	AxisID in MPLCAN mode	AxisID in CANopen mode
Open (no connection)	255	LSS non-configured state
294K	4	4
232K	8	8
88.7K	12	12
48.7K	13	13
34K	14	14
0 (short circuit)	15	15

Note: LSS "non-configured" state, is a state in which the drive does not have assigned an active Axis ID while connected to the CAN network. In this mode the axis ID for RS232 communication is 255. The axis ID can be configured via a LSS master using CiA-305 protocol, which can set and save a new unique value. While the drive has a non-configured axis ID, it cannot communicate with other drives in the network.

4. Step 2. Drive Setup

4.1. Installing PROconfig

PROconfig is a PC software platform for the setup of the ElectroCraft drives. PROconfig is part of the ElectroCraft Motion PRO Suite is available as part of a PRO Series Drive Evaluation Kit. PROconfig comes with an ***Update via Internet tool*** through which you can check if your software version is up-to-date, and when necessary download and install the latest updates.

PROconfig can be installed independently or together with the **MotionPRO Suite** platform for motion programming using MPL. You will need MotionPRO Suite only if you plan to use the advanced features presented in Section 5.3.

On request, PROconfig can be provided on a CD too. In this case, after installation, use the update via internet tool to check for the latest updates. Once you have started the installation package, follow its indications.

4.2. Getting Started with PROconfig

Using PROconfig you can quickly setup a drive for your application. The drive can be connected with your PC in one of the following ways:

1. Via an RS232 link, directly connected to the PC, or via an USB to RS232 adapter or using ElectroCraft Ethernet to RS232 adapter, function of your PC communication interfaces;
2. Via a CAN-bus link, directly connected to the PC through a PC-CAN interface, or using ElectroCraft Ethernet to CAN adapter
3. Via another drive from the same CAN-bus network, which is connected to the PC via one of the above options from point 1.

The output of PROconfig is a set of *setup data*, which can be downloaded into the drive EEPROM or saved on your PC for later use.

PROconfig includes a set of evaluation tools like the Data Logger, the Control Panel and the Command Interpreter which help you to quickly measure, check and analyze your drive commissioning.

PROconfig works with **setup** data. A **setup** contains all the information needed to configure and parameterize a ElectroCraft drive. This information is preserved in the drive EEPROM in the *setup table*. The setup table is copied at power-on into the RAM memory of the drive and is used during runtime. With PROconfig it is also possible to retrieve the complete setup information from a drive previously programmed.

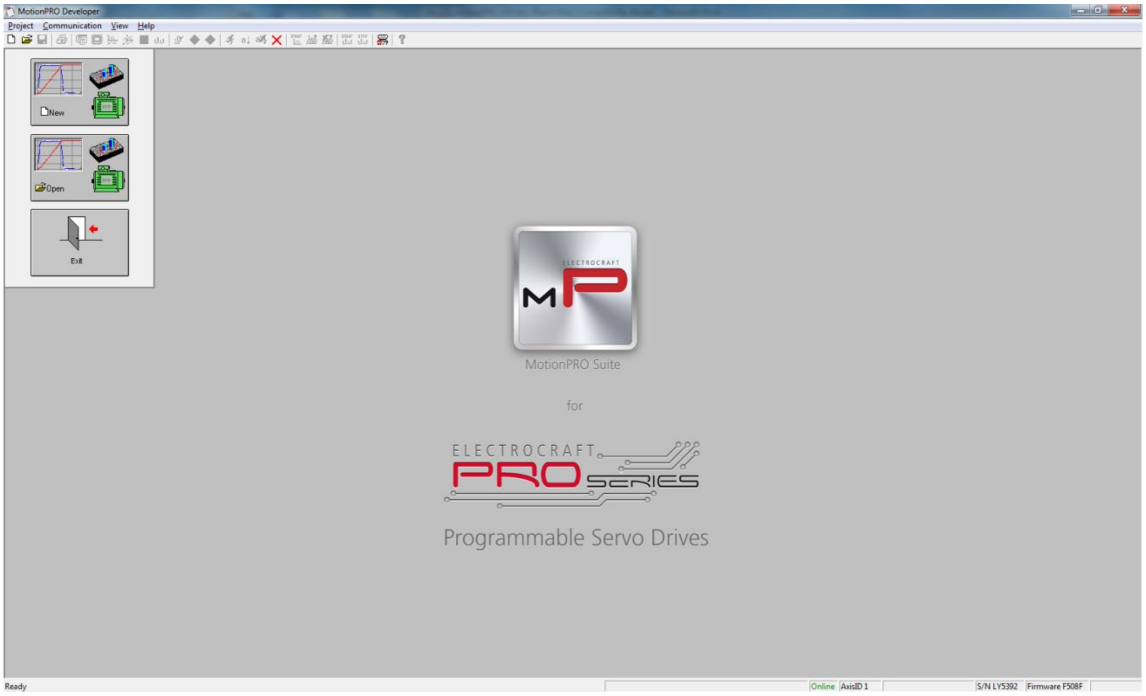
Note that with PROconfig you do only your drive/motor commissioning. For motion programming you have the following options:

- Use a **CANopen** master to control the PRO Series IMD as a standard CANopen drive
- Use **MotionPRO Developer** to create and download a **MPL** program into the drive/motor memory
- Use one of the **MPL_LIB** motion libraries to control the drives/motors from your host/master. If your host is a **PC**, MPL_LIB offers a collection of high level motion functions which can be called from applications written in C/C++, Visual Basic, Delphi Pascal or LabVIEW. If your host is a **PLC**, MPL_LIB offers a collection of function blocks for motion programming, which are **IEC61131-3 compatible** and can be integrated in your PLC program.
- **Implement** on your master the MPL commands you need to send to the drives/motors using one of the supported communication channels. The implementation must be done according with ElectroCraft communication protocols.

- **Combine** MPL programming at drive level with one of the other options (see Section 5.3)

4.2.1. Establish communication

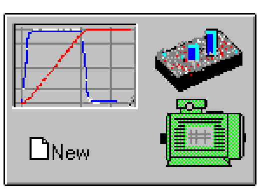
PROconfig starts with an empty window from where you can create a **New** setup, **Open** a previously created setup which was saved on your PC, or **Upload** the setup from the drive/motor.



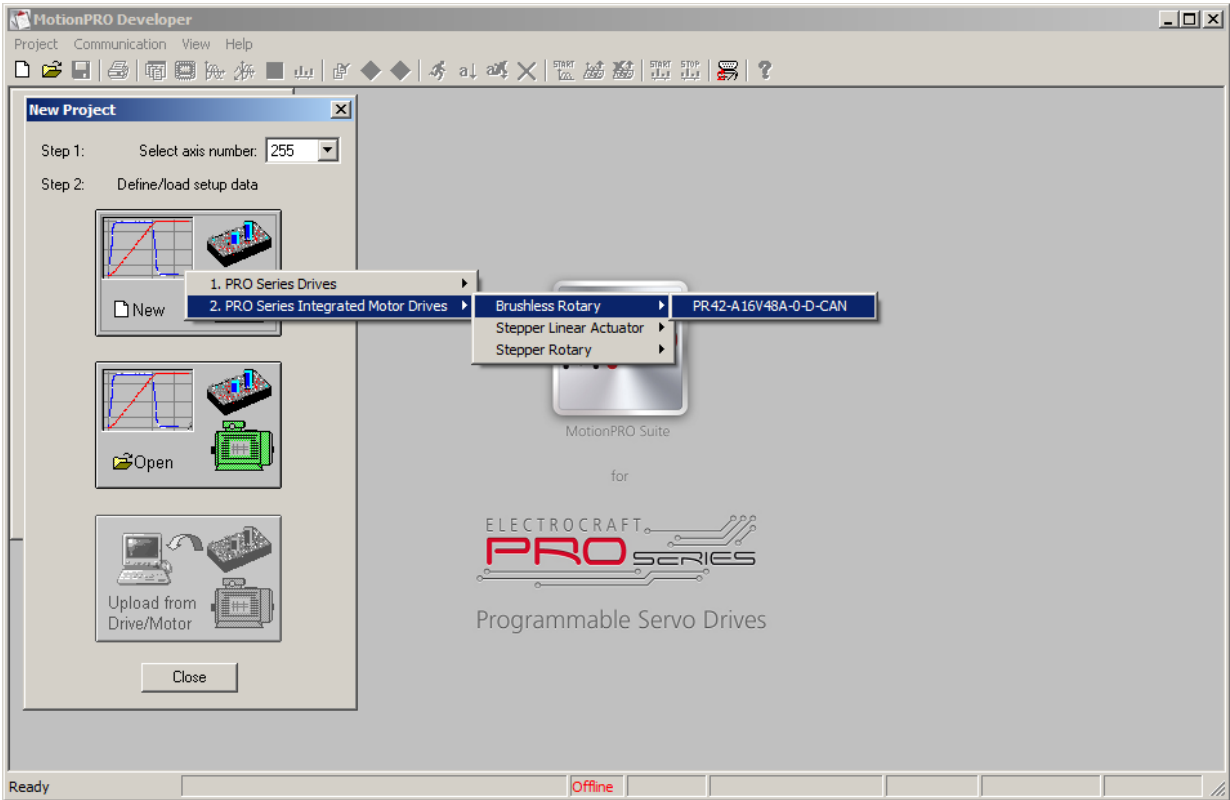
Before selecting one of the above options, you need to establish the communication with the PRO Series IMD you want to commission. Use menu command **Communication | Setup** to check/change your PC communication settings. Press the **Help** button of the dialogue opened. Here you can find detailed information about how to setup your IMD and connect. Power on the IMD, then close the Communication | Setup dialogue with OK. If the communication is established, PROconfig displays in the status bar (the bottom line) the text "**Online**" plus the axis ID of your IMD and its firmware version. Otherwise the text displayed is "**Offline**" and a communication error message tells you the error type. In this case, return to the Communication | Setup dialogue, press the Help button to troubleshoot.

Remark: *When first started, PROconfig tries to communicate via RS-232 and COM1 with an IMD having axis ID=255 (default communication settings). If the IMD has a different axis ID and you don't know it, select in the Communication | Setup dialogue and set "Axis ID of drive/motor connected to PC" to the option* ***Autodetected****. If this IMD is part of a CANbus network, use the menu command* ***Communication | Scan Network.***

4.2.2. Setup drive/motor

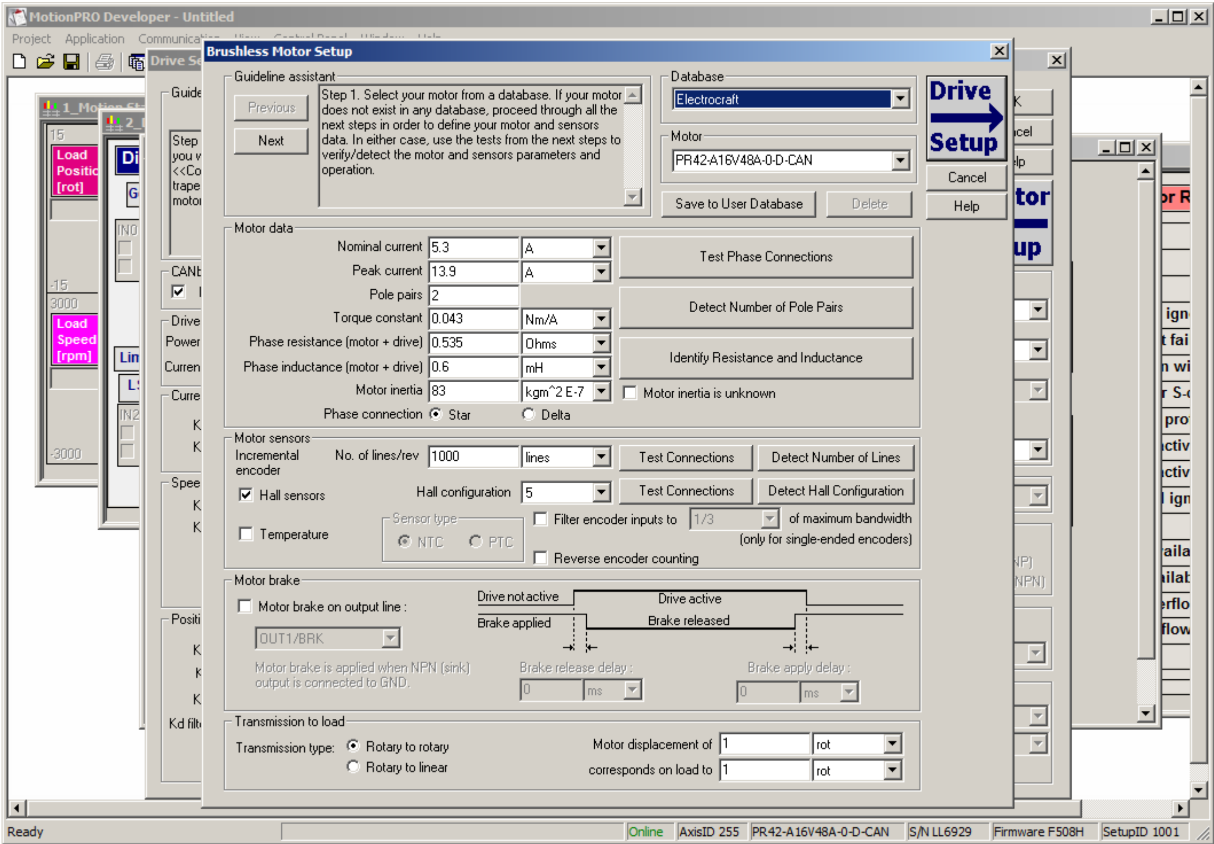
Press **New** button 

and select your PRO Series IMD type.



The selection continues with the motor technology (for example: brushless or stepper)

The selection opens two setup dialogues: for **Motor Setup** and for **Drive setup** through which you can configure and parameterize an ElectroCraft drive, plus several predefined control panels customized for the product selected.



In the **Motor setup** dialogue you can introduce the data of your motor and the associated sensors. Data introduction is accompanied by a series of tests having as goal to check the connections to the drive and/or to determine or validate a part of the motor and sensors parameters. In the **Drive setup** dialogue you can configure and parameterize the drive for your application. In each dialogue you will find a **Guideline Assistant**, which will guide you through the whole process of introducing and/or checking your data. Close the Drive setup dialogue with **OK** to keep all the changes regarding the motor and the drive setup.

4.2.3. Download setup data to drive/motor

Press the **Download to Drive/Motor** button 

to download your setup data in the drive/motor EEPROM memory in the *setup table*. From now on, at each power-on, the setup data is copied into the drive/motor RAM memory which is used during runtime. It is also possible to **Save** 

the setup data on your PC and use it in other applications.

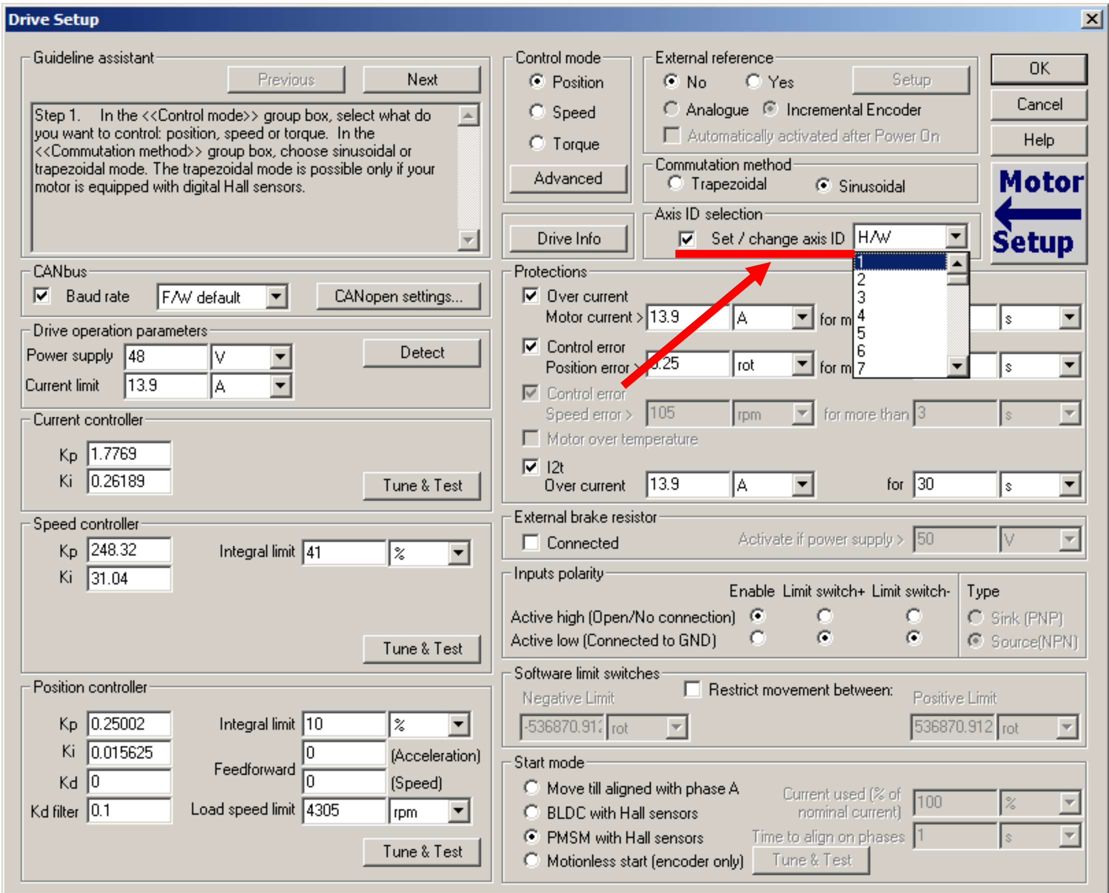
To summarize, you can define or change the setup data in the following ways:

- create a new setup data by going through the motor and drive dialogues
- use setup data previously saved in the PC
- upload setup data from a drive/motor EEPROM memory

4.2.4. Evaluate drive/motor behavior (optional)

You can use the **Data Logger** or the **Control Panel** evaluation tools to quickly measure and analyze your application behavior. In case of errors like protections triggered, use the Drive Status control panel to find the cause.

4.3. Changing the drive Axis ID



The axis ID of a PRO Series IMD can be set in 3 ways:

- Hardware (H/W) – in accordance with axis ID resistance value par. 3.4.2.
- Software (via Setup) – any value between 1 and 255, stored in the setup table. If the IMD is in CANopen mode, a Node ID value above 127 is automatically converted into 255 and the IMD is set with CAN communication in “non-configured” mode waiting for a CANopen master to configure it using CiA-305 protocol. A “non-configured” IMD answers only to CiA-305 commands. All other CANopen commands are ignored and transmission of all other CANopen messages (including boot-up) is disabled. In absence of a CANopen master, you can get an IMD out of “non-configured” mode, by setting another axis ID between 1 and 127, from above dialogue using a serial link between the IMD and the PC.
- Software (via CANopen master) – using CiA-305 protocol

The axis ID is initialized at power on, using the following algorithm:

a) If a valid setup table exists, and this setup table was created with the *Axis ID Selection* checkbox checked in the Drive Setup dialogue (see above) – with the value read from the setup table. This value can be an axis number 1 to 255 or can indicate that axis ID will be set according with the AxisID hex switch. If the drive is set in CANopen mode and the Axis ID is over 127 it is converted into 255 and the drive enters in CAN communication "LSS non-configured" mode.

b) If a valid the setup table exists, and this was created with the *Axis ID Selection* checkbox unchecked in the Drive Setup dialogue (see above) – with the last value set either from a valid setup table or by a CANopen master via CiA-305 protocol. This value can be an axis number 1 to 255 for MPLCAN, 1 to 127 for CANopen, or can indicate that axis ID will be set in accordance with the axis ID resistor value connected between AXISID (J4 pin 2) and GND (J4 pin 7).

c) If the setup table is invalid, with the last value set either from a valid setup table or by a CANopen master via CiA-305 protocol. This value can be an axis number 1 to 255 for MPLCAN, 1 to 127 for CANopen, or can indicate that axis ID will be set in accordance with the axis ID resistor value connected between AXISID (J4 pin 2) and GND (J4 pin 7).

d) If the setup table is invalid, there is no previous axis ID set from a valid setup table or by a CANopen master, the axis ID will be set in accordance with the axis ID resistor value connected between AXISID (J4 pin 2) and GND (J4 pin 7).

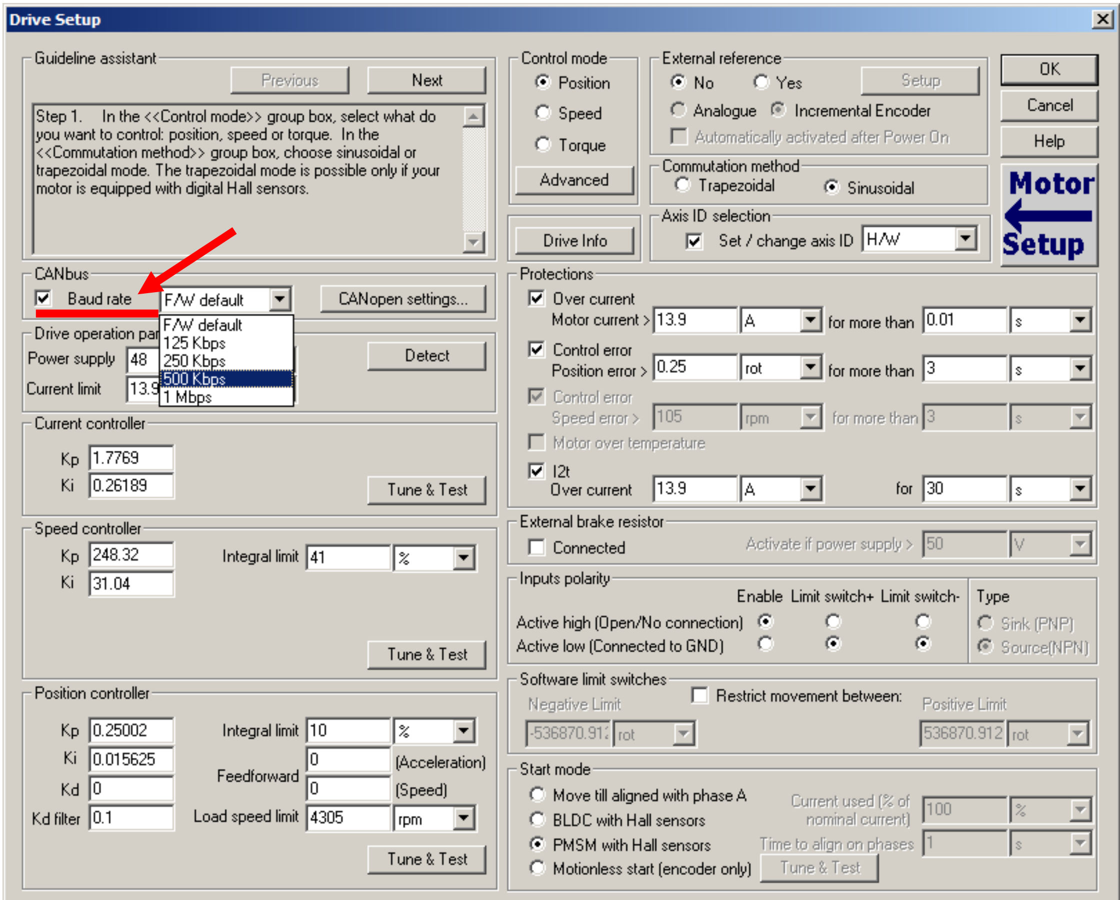
Remark: *If you don't know the axis ID set in an IMD, you can find it in the following way:*

a) Connect the IMD via a serial RS232 link to a PC where PROconfig or MotionPRO Developer are installed.

b) *With the IMD powered, open PROconfig or MotionPRO Developer and check the status bar. If communication with the IMD is established, the status bar displays* ***Online*** *in green and nearby the IMD's axis ID. If the status bar displays* ***Offline*** *in red, execute menu command "Communication|Setup…" and in the dialogue opened select at "Channel Type"* ***RS232*** *and at "axis ID of IMD/motor connected to PC" the option* ***Autodetected****. After closing the dialogue with OK, communication with the IMD must be established and the status bar must display the IMD's axis ID.*

c) *If the access to the IMD with the unknown Axis ID is difficult, but this IMD is connected via CANbus with other ElectroCraft IMDs having an easier access, connect your PC serially to one of the other IMDs. Use PROconfig or MotionPRO Developer menu command* ***Communication | Scan Network*** *to find the axis IDs of all the ElectroCraft IMDs present in the network.*

4.4. Setting CANbus rate



The PRO Series IMDs accept the following CAN rates: 125Kbps, 250 Kbps, 500kbps and 1Mbps. Using the Drive Setup dialogue you can choose the initial CAN rate after power on. This information is stored in the setup table The CAN rate is initialized using the following algorithm:

a) If a valid setup table exists, and this setup table was created with the *Set baud rate* checkbox <u>checked</u> in the Drive Setup dialogue (see above) – with the value read from the setup table. This value can be one of the above 4 values or the firmware default (F/W default) which is 500kbs.

b) If a valid setup table exists, and this setup table was created with the *Set baud rate* checkbox <u>unchecked</u> in the Drive Setup dialogue (see above) – with the last value set either from a valid setup table or by a CANopen master via CiA-305 protocol

c) If the setup table is invalid, with the last value set either from a valid setup table or by a CANopen master via CiA-305 protocol.

d) If the setup table is invalid, there is no previous CAN rate set from a valid setup table or by a CANopen master, with f/w default value which is 500kbs.

4.5. Creating an Image File with the Setup Data

Once you have validated your setup, you can create with the menu command **Setup | Create EEPROM Programmer File** a software file (with extension **.sw**) which contains all the setup data to write in the EEPROM of your IMD.

A software file is a text file that can be read with any text editor. It contains blocks of data separated by an empty row. Each block of data starts with the block start address, followed by data values to place in ascending order at consecutive addresses: first data – to write at start address, second data – to write at start address + 1, etc. All the data are hexadecimal 16- bit values (maximum 4 hexadecimal digits). Each row contains a single data value. When less than 4 hexadecimal digits are shown, the value must be right justified. For example 92 represent 0x0092.

The **.sw** file can be programmed into an IMD:

- from a CANopen master, using the communication objects for writing data into the IMD EEPROM.
- from a host PC or PLC, using the MPL_LIB functions for writing data into the IMD EEPROM.
- using the PRO EEPROM Programmer tool, which comes with PROconfig but may also be installed separately. The PRO EEPROM Programmer was specifically designed for repetitive fast and easy programming of **.sw** files into the ElectroCraft IMDs during production.

5. Step 3. Motion Programming

5.1. Using a CANopen Master (for PRO Series IMD CANopen execution)

The PRO Series IMD conforms to **CiA 301 v.4.2** application layer and communication profile, **CiA WD 305 v.2.2.13** layer settings services and protocols and **CiA DSP 402 v3.0** device profile for drives and motion control the now included in IEC 61800-7-1 Annex A, IEC 61800-7-201 and IEC 61800-7-301 standards. For details see ElectroCraft PRO Series ***CANopen Programming Manual (ElectroCraft* Document *Number* A11226*)*** .

5.1.1. CiA-301 Application Layer and Communication Profile Overview

The PRO Series IMD accepts the following basic services and types of communication objects of the CANopen communication profile CiA301 v4.2:

- **Service Data Object (SDO)**

Service Data Objects (SDOs) are used by CANopen master to access any object from the drive's Object Dictionary. Both expedited and segmented SDO transfers are supported. SDO transfers are confirmed services. The SDOs are typically used for drive configuration after power-on, for PDOs mapping and for infrequent low priority communication between the CANopen master and the drives.

- **Process Data Object (PDO)**

Process Data Objects (PDO) are used for high priority, real-time data transfers between CANopen master and the drives. The PDOs are unconfirmed services which are performed with no protocol overhead. Transmit PDOs are used to send data from the drive, and receive PDOs are used to receive on to the drive. The PRO Series IMD accepts 4 transmit PDOs and 4 receive PDOs. The contents of the PDOs can be set according with the application needs using the dynamic PDO-mapping. This operation can be done during the drive configuration phase using SDOs.

- **Synchronization Object (SYNC)**

The SYNC message provides the basic network clock, as the SYNC producer broadcasts the synchronization object periodically. The service is unconfirmed. The PRO Series IMD supports both SYNC consumer and producer.

- **Time Stamp Object (TIME)**

The Time Stamp Object is supported by the PRO Series IMD device.

- **Emergency Object (EMCY)**

Emergency objects are triggered by the occurrence of a drive internal error situation. An emergency object is transmitted only once per 'error event'. As long as no new errors occur, the drive will not transmit further emergency objects.

- **Network Management Objects (NMT)**

The Network Management is node oriented and follows a master-slave structure. NMT objects are used for executing NMT services. Through NMT services the drive can be initialized, started, monitored, reset or stopped. The PRO Series IMD is a NMT slave in a CANopen network.

- **Module Control Services** – through these unconfirmed services, the NMT master controls the state of the drive. The following services are implemented: Start Remote Node, Stop Remote Node, Enter Pre-Operational, Reset Node, Reset Communication
- **Error Control Services** – through these services the NMT master detects failures in a CAN-based network. Both error control services defined by DS301 v4.02 are supported by the PRO Series IMD: Node Guarding (including Life Guarding) and Heartbeat
- **Bootup Service** - through this service, the drive indicates that it has been properly initialized and is ready to receive commands from a master

5.1.2. CiA-305 Layer Setting Services (LSS) and Protocols Overview

When used in a CANopen network, the PRO Series IMD accept node-ID and CAN bus bit timing settings according with CiA 305 protocol. This allows a CANopen master supporting CiA WD 305 to configure each PRO Series IMD from the network with the desired node-ID and CAN bus bit timing. CiA-305 protocol allows connecting non-configured drives to a CANopen network and performing the drives configuration on-the-fly via the CANopen master.

5.1.3. CiA-402 and Manufacturer Specific Device Profile Overview

The PRO Series IMDsupports the following CiA 402 modes of operation:

- **Profile position and velocity modes**
- **Homing mode**
- **Interpolated position mode**

Additional to these modes, there are also several manufacturer specific modes defined:

- **External reference modes (position, speed or torque)**
- **Electronic gearing and camming position mode**

5.1.4. ElectroCAN Extension

In order to take full advantage of the powerful ElectroCraft Motion PROgramming Language (MPL) built into the PRO Series IMD, ElectroCraft has developed an extension to CANopen, called ElectroCAN through which MPL commands can be exchanged with the drives. Thanks to ElectroCAN you can inspect or reprogram any of the ElectroCraft drives from a CANopen network using PROconfig or MotionPRO Developer and an RS-232 link between your PC and any of the drives.

ElectroCAN uses only identifiers outside of the range used by the default by the CANopen predefined connection set (as defined by CiA 301). Thus, ElectroCAN protocol and CANopen protocol can co-exist and communicate simultaneously on the same physical CAN bus, without disturbing each other.

5.1.5. Checking Setup Data Consistency

During the configuration phase, a CANopen master can quickly verify using the checksum objects and a reference **.sw** file (see 4.5 and 5.2.4 for details) whether the non-volatile EEPROM memory of an PRO Series IMD contains the right information. If the checksum reported by the drive doesn't match with that computed from the **.sw** file, the CANopen master can download the entire **.sw** file into the IMD EEPROM using the communication objects for writing data into the IMD EEPROM.

5.2. Using the built-in Motion Controller and MPL

One of the key advantages of the ElectroCraft IMDs is their capability to execute complex motions without requiring an external motion controller. This is possible because ElectroCraft IMDs offer in a single compact package both a state of art digital drive and a powerful motion controller.

5.2.1. ElectroCraft Motion PROgramming Language Overview

Programming motion directly on a ElectroCraft IMD requires creating and downloading a MPL (ElectroCraft Motion PROgramming Language) program into the IMD memory. The MPL allows you to:

- Set various motion modes (profiles, PVT, PT, electronic gearing or camming[1], etc.)
- Change the motion modes and/or the motion parameters

[1] Optional for PRO-A04V36x CANopen execution

- Execute homing sequences[1]
- Control the program flow through:
    - Conditional jumps and calls of MPL functions
    - MPL interrupts generated on pre-defined or programmable conditions (protections triggered, transitions on limit switch or capture inputs, etc.)
    - Waits for programmed events to occur
- Handle digital I/O and analog input signals
- Execute arithmetic and logic operations
- Perform data transfers between axes
- Control motion of an axis from another one via motion commands sent between axes
- Send commands to a group of axes (multicast). This includes the possibility to start simultaneously motion sequences on all the axes from the group
- Synchronize all the axes from a network

In order to program a motion using MPL you need MotionPRO Suite software platform.

5.2.2. Installing MotionPRO Suite

MotionPRO Suite is an integrated development environment for the setup and motion programming of ElectroCraft programmable IMDs. It comes with an ***Update via Internet tool*** through which you can check if your software version is up-to-date, and when necessary download and install the latest updates.

MotionPRO Suite, including the fully functional version of PROconfig, is part of a PRO Series IMD Evaluation Kit. Please contact ElectroCraft or your local ElectroCraft sales representative for more information on obtaining MotionPRO Suite or an evaluation kit.

MotionPRO Suite is delivered on a CD. Once you have started the installation package, follow its indications. After installation, use the update via internet tool to check for the latest updates. Alternately, you can first install the demo version and then purchase a license.

5.2.3. Getting Started with MotionPRO Suite

Using MotionPRO Suite you can quickly do the setup and the motion programming of an ElectroCraft IMD according with your application needs. The drive can be connected with your PC in one of the following ways:

1. Via an RS232 link, directly connected to the PC, or via an USB to RS232 adapter or using ElectroCraft Ethernet to RS232 adapter, function of your PC communication interfaces;
2. Via a CAN-bus link, directly connected to the PC through a PC-CAN interface, or using ElectroCraft Ethernet to CAN adapter
3. Via another drive from the same CAN-bus network, which is connected to the PC via one of the above options from point 1.

The output of the MotionPRO Suite is a set of setup data and a motion program, which can be downloaded to the IMD EEPROM or saved on your PC for later use.

MotionPRO Suite includes a set of evaluation tools like the Data Logger, the Control Panel and the Command Interpreter which help you to quickly develop, test, measure and analyze your motion application.

---

[1] The customization of the homing routines is available only for PRO-A04V36x CAN execution

MotionPRO Suite works with **projects***.* A project contains one or several **Applications**.

Each application describes a motion system for one axis. It has 2 components: the **Setup** data and the **Motion** program and an associated axis number: an integer value between 1 and 255. An application may be used either to describe:

1. One axis in a multiple-axis system
2. An alternate configuration (set of parameters) for the same axis.

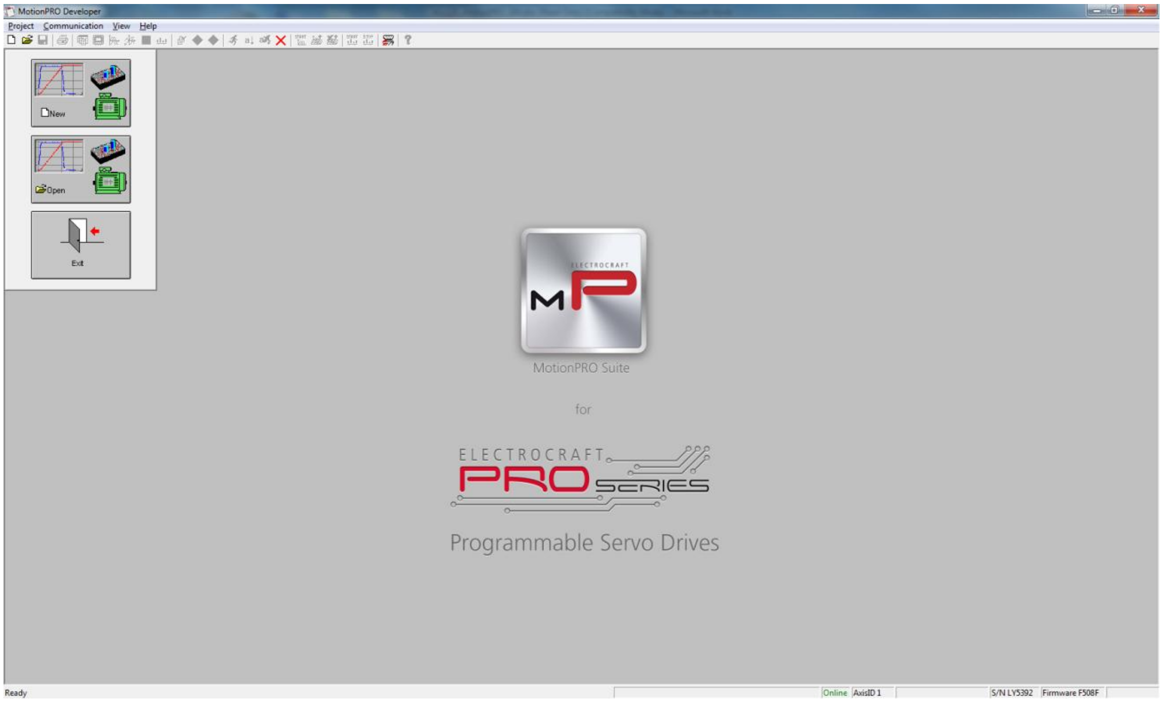
In the first case, each application has a different axis number corresponding to the axis ID of the IMD from the network. All data exchanges are done with the IMD having the same address as the selected application. In the second case, all the applications have the same axis number.

The setup component contains all the information needed to configure and parameterize an ElectroCraft IMD. This information is preserved in the drive/motor EEPROM in the *setup table*. The setup table is copied at power-on into the RAM memory of the drive/motor and is used during runtime.

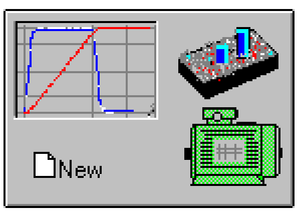
The motion component contains the motion sequences to do. These are described via a MPL (ElectroCraft Motion PROgramming Language) program, which is executed by the IMDs built-in motion controller.

5.2.3.1 Create a new project

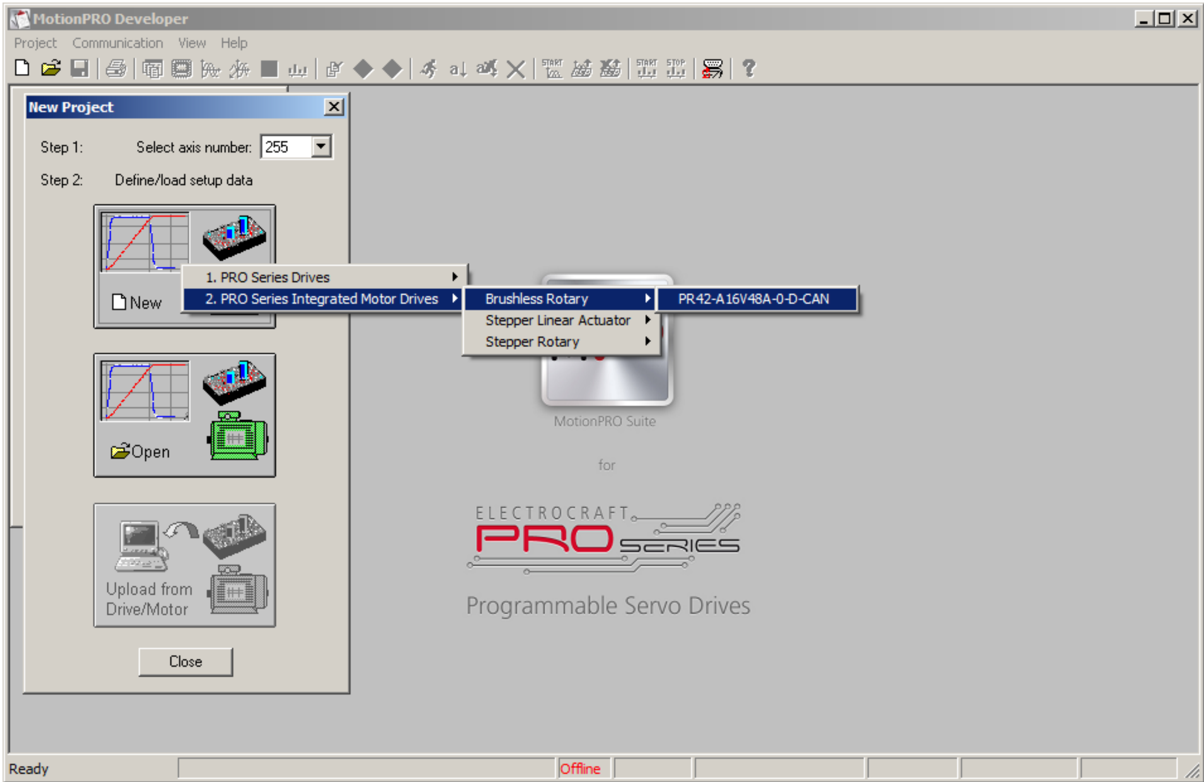
MotionPRO Developer starts with an empty window from where you can create a new project or open a previously created one.



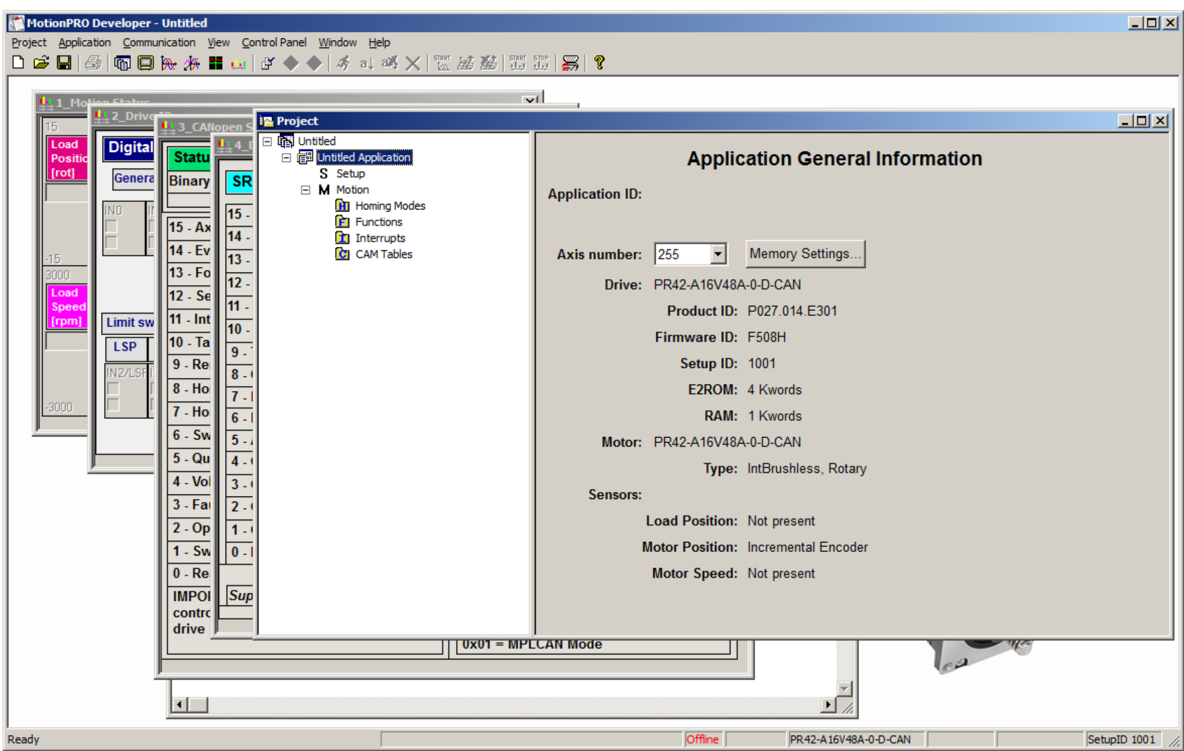
When you start a new project, MotionPRO Developer automatically creates a first application. Additional applications can be added later. You can duplicate an application or insert one defined in another project.

Press **New** button 

to open the “New Project” dialogue. Set the axis number for your first application equal with your IMD axis ID. The initial value proposed is 255 which is the default axis ID of the IMD. Press **New** button and select your IMD type.



Click on your selection. MotionPRO Developer opens the Project window where on the left side you can see the structure of a project. At beginning both the new project and its first application are named “Untitled”. The application has 2 components: **S** Setup and **M** Motion (program).



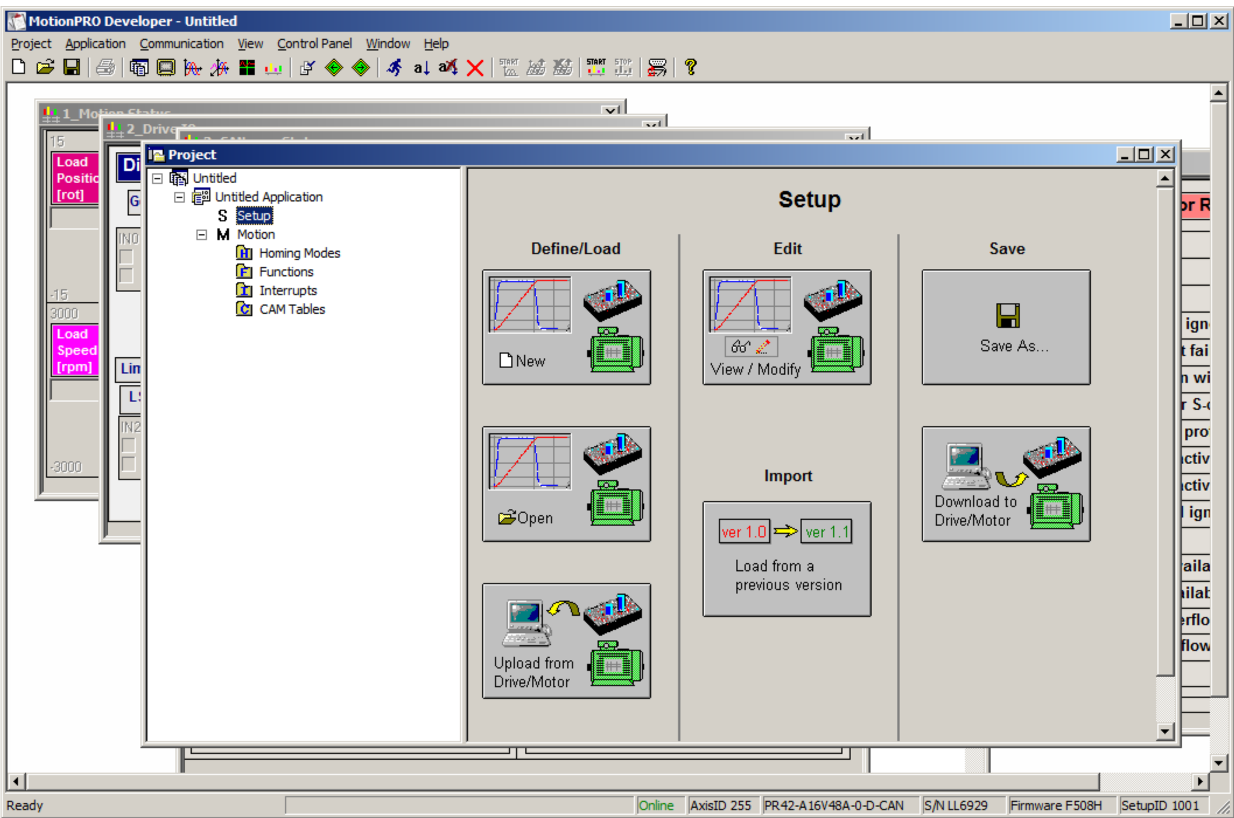
5.2.3.2 Step 2 Establish communication

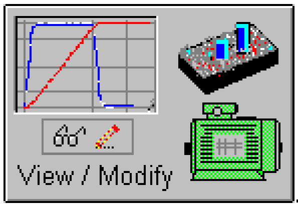
If you have an IMD connected with your PC, now it's time to check the communication. Use menu command **Communication | Setup** to check/change your PC communication settings. Press the **Help** button of the dialogue opened. Here you can find detailed information about how to setup your IMD and the connections. Power on the drive, then close the Communication | Setup dialogue with OK. If the communication is established, MotionPRO Developer displays in the status bar (the bottom line) the text "**Online**" plus the axis ID of your IMD and its firmware version. Otherwise the text displayed is "**Offline**" and a communication error message tells you the error type. In this case, return to the Communication | Setup dialogue, press the Help button and check troubleshoots.

Remark: *When first started, MotionPRO Developer tries to communicate via RS-232 and COM1 with a drive having axis ID=255 (default communication settings).). If the drive has a different axis ID and you don't know it, select in the Communication | Setup dialogue at "Axis ID of IMD connected to PC" the option* ***Autodetected.*** *If this drive is part of a CANbus network and the PC is serially connected with another drive, use the menu command* ***Communication | Scan Network***

5.2.3.3 Setup drive/motor

In the project window left side, select "**S** Setup", to access the setup data for your application.



Press **View/Modify** button 

. This opens 2 setup dialogues: for **Motor Setup** and for **Drive Setup** (same as PROconfig) through which you can configure and parameterize a ElectroCraft IMD. In the **Motor setup** dialogue you can introduce the data of your motor and the associated sensors. Data introduction is accompanied by a series of tests having as goal to check the connections to the drive and/or to determine or validate a part of the motor and sensors parameters. In the **Drive setup** dialogue you can configure and parameterize the drive for your application. In each dialogue you will find a **Guideline Assistant**, which will guide you through the whole process of introducing and/or checking your data.

Press the **Download to Drive/Motor** button 

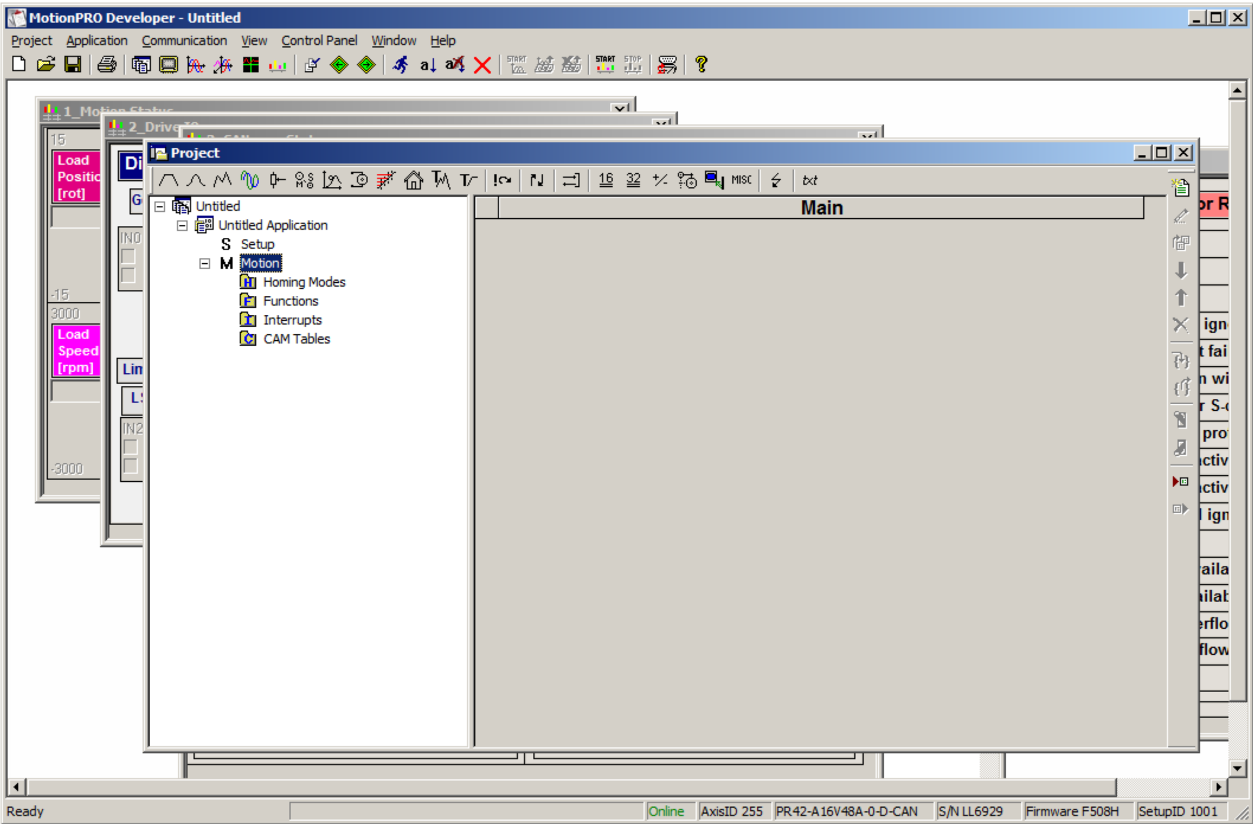
to download your setup data in the IMD EEPROM memory in the *setup table*. From now on, at each power-on, the setup data is copied into the drive/motor RAM memory which is used during runtime. It is also possible to save the setup data on your PC and use it in other applications. Note that you can upload the complete setup data from an IMD.

To summarize, you can define or change the setup data of an application in the following ways:

- create a new setup data by going through the motor and drive dialogues
- use setup data previously saved in the PC
- upload setup data from a IMD EEPROM memory

5.2.3.4 Program motion

In the project window left side, select "**M** Motion", for motion programming. This automatically activates the **Motion Wizard**.



The Motion Editor offers you the possibility to program all the motion sequences using high level dialogues which automatically generate the corresponding MPL instructions. Therefore with Motion Editor you can develop motion programs using almost all the MPL instructions without needing to learn them. A MPL program includes a main section, followed by the subroutines used: functions, interrupt service routines and homing procedures[1]. The MPL program may also include cam tables used for electronic camming applications[2].

When activated, Motion Editor adds a set of toolbar buttons in the project window just below the title. Each button opens a programming dialogue. When a programming dialogue is closed, the associated MPL instructions are automatically generated. Note that, the MPL instructions generated are not a simple text included in a file, but a motion object. Therefore with the Motion Editor you define your motion program as a collection of motion objects.

The major advantage of encapsulating programming instructions in motion objects is that you can very easily manipulate them. For example, you can:

- Save and reuse a complete motion program or parts of it in other applications
- Add, delete, move, copy, insert, enable or disable one or more motion objects
- Group several motion objects and work with bigger objects that perform more complex functions

As a starting point, push for example the leftmost Motion Editor button – Trapezoidal profiles, and set a position or speed profile. Then press the **Run** button. At this point the following operations are done automatically:

- A MPL program is created by inserting your motion objects into a predefined template

[1] The customization of the interrupt service routines and homing routines is available only for PRO-A04V36x CAN execution
[2] Optional for PRO-A04V36x CANopen execution

- The MPL program is compiled and downloaded to the IMD
- The MPL program execution is started

For learning how to send MPL commands from your host/master, using one of the communication channels and protocols supported by the drives use menu command **Application | Binary Code Viewer…** Using this tool, you can get the exact contents of the messages to send and of those expected to be received as answers.

5.2.3.5 Evaluate motion application performances

MotionPRO Suite includes a set of evaluation tools like the **Data Logger**, the **Control Panel** and the **Command Interpreter** which help you to quickly measure and analyze your motion application.

5.2.4. Creating an Image File with the Setup Data and the MPL Program

Once you have validated your application, you can create with the menu command **Application | Create PRO EEPROM Programmer File** a software file (with extension **.sw**) which contains all the data to write in the EEPROM of your drive. This includes both the setup data and the motion program. For details regarding the **.sw** file format and how it can be programmed into a drive, see paragraph 4.5

5.3. Combining CANopen /or other host with MPL

Due to its embedded motion controller, an PRO Series IMDoffers many programming solutions that may simplify a lot the task of a CANopen master. This paragraph overviews a set of advanced programming features which arise when combining MPL programming at drive level with CANopen master control. A detailed description of these advanced programming features is included in the ***CANopen Programming* (Document No. A11226)** manual. All features presented below require usage of MotionPRO Suite as MPL programming tool

Remark: *If you don't use the advanced features presented below you don't need MotionPRO Suite. In this case the PRO Series IMD is treated like a standard CANopen IMD, whose setup is done using PROconfig.*

5.3.1. Using MPL Functions to Split Motion between Master and IMDs

With ElectroCraft programmable IMDs you can really distribute the intelligence between a CANopen master and the IMDs in complex multi-axis applications. Instead of trying to command each step of an axis movement, you can program the IMDs using MPL to execute complex tasks and inform the master when these are done. Thus for each axis, the master task may be reduced at: calling MPL functions (with possibility to abort their execution) stored in the IMDs EEPROM and waiting for a message, which confirms the finalization of the MPL functions execution.

5.3.2. Executing MPL programs

The distributed control concept can go one step further. You may prepare and download into an IMD a complete MPL program including functions, homing procedures[1], etc. The MPL program execution can be started by simply writing a value in a dedicated object,

5.3.3. Loading Automatically Cam Tables Defined in MotionPRO Developer

The PRO Series IMD offers others motion modes like[1]: electronic gearing, electronic camming, external modes with analog or digital reference etc. When electronic camming is used, the cam tables can be loaded in the following ways:

[1] The customization of the interrupt service routines and homing routines is available only for PRO-A0xV36 CAN executions

a) The master downloads the cam points into the IMD active RAM memory after each power on;

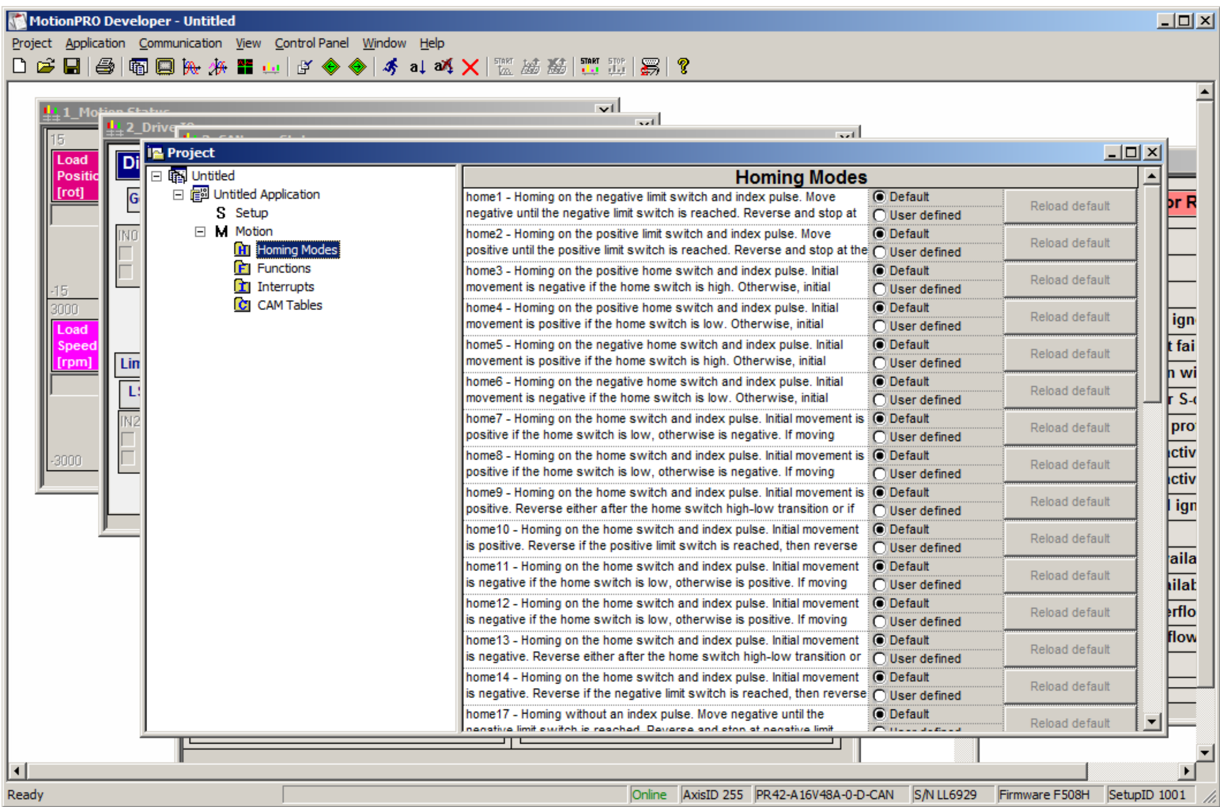
b) The cam points are stored in the IMD EEPROM and the master commands their copy into the active RAM memory

c) The cam points are stored in the IMD EEPROM and during the IMD initialization (transition to Ready to Switch ON status) are automatically copied from EEPROM to the active RAM

For the last 2 options the cam table(s) are defined in MotionPRO Suite and are included in the information stored in the EEPROM together with the setup data and the MPL programs/functions.
Remark: The cam tables are included in the **.sw** file generated with MotionPRO Developer. Therefore, the IMDs can check the cam presence in the IMD EEPROM using the same procedure as for testing of the setup data.

5.3.4. Customizing the Homing Procedures

The PRO Series IMD supports all homing modes defined in CiA402 device profile, plus 4 custom based on hard stop. If needed, any of these homing modes can be customized. In order to do this you need to select the Homing Modes from your MotionPRO Developer application and in the right side to set as “User defined” one of the Homing procedures. Following this operation the selected procedure will occur under Homing Modes in a subtree, with the name *HomeX* where X is the number of the selected homing.



If you click on the *HomeX* procedure, on the right side you’ll see the MPL function implementing it. The homing routine can be customized according to your application needs. It’s calling name and method remain unchanged.

[1] Optional for the PRO-A0xV36x CANopen execution

5.3.5. Customizing the IMD Reaction to Fault Conditions

Similarly to the homing modes, the default service routines for the MPL interrupts can be customized according to your application needs. However, as most of these routines handle the drive reaction to fault conditions, it is mandatory to keep the existent functionality while adding your application needs, in order to preserve the correct protection level of the drive. The procedure for modifying the MPL interrupts is similar with that for the homing modes.

5.4. Using Motion Libraries for PC-based Systems

A **MPL Library for PC** is a collection of high-level functions allowing you to control from a PC a network of ElectroCraft programmable IMDs. It is an ideal tool for quick implementation on PCs of motion control applications with ElectroCraft products.

With the MPL Motion Library functions you can: communicate with a IMD via any of its supported channels (RS-232, CAN-bus, etc.), send motion commands, get automatically or on request information about IMD status, check and modify its setup parameters, read inputs and set outputs, etc.

The MPL Motion Library can work under a **Windows** or **Linux** operating system. Implemented as a .dll/.so, it can be included in an application developed in **C/C++/C#, Visual Basic, Delphi Pascal** or **Labview**.

Using a MPL Motion Library for PC, you can focus on the main aspects of your application, while the motion programming part can be reduced to calling the appropriate functions and getting the confirmation when the task was done.

All ElectroCraft's MPL Motion Libraries for PCs are provided with PROconfig.

5.5. Using Motion Libraries for PLC-based Systems

A **MPL Motion Library for PLC** is a collection of high-level functions and function blocks allowing you to control from a PLC the ElectroCraft programmable IMDs. The motion control function blocks are developed in accordance with the **PLC IEC61131-3 standard** and represent an ideal tool for quick implementation on PLCs of motion control applications with ElectroCraft products.

With the MPL Motion Library functions you can: communicate with a IMD via any of its supported channels, send motion commands, get automatically or on request information about IMD status, check and modify its setup parameters, read inputs and set outputs, etc. Depending on the PLC type, the communication is done either directly with the CPU unit, or via a CANbus or RS-232 communication module.

Using a MPL Motion Library for PLC, you can focus on the main aspects of your PLC application, while the motion programming part can be reduced to calling the appropriate functions and monitoring the confirmations that the task was done.

All these blocks have been designed using the guidelines described in the PLC standards, so they can be used on any development platform that is **IEC 61136 compliant.**

All ElectroCraft's MPL Motion Libraries for PLC are provided with PROconfig.

6. Scaling factors

ElectroCraft IMDs work with parameters and variables represented in the IMD internal units (IU). These correspond to various signal types: position, speed, current, voltage, etc. Each type of signal has its own internal representation in IU and a specific scaling factor. This chapter presents the IMD internal units and their relation with the international standard units (SI).
In order to easily identify them, each internal unit has been named after its associated signal. For example the **position units** are the internal units for position, the **speed units** are the internal units for speed, etc.

6.1. Position units

6.1.1. Brushless motor with quadrature encoder on motor

The internal position units are encoder counts. The correspondence with the load **position in SI units**[1] is:

$$Load\_Position[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr} \times Motor\_Position[IU]$$

where:
No_encoder_lines – is the rotary encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

6.1.2. Step motor open-loop control. No feedback device

The internal position units are motor µsteps. The correspondence with the load **position in SI units** is:

$$Load\_Position[SI] = \frac{2 \times \pi}{No\_\mu steps \times No\_steps \times Tr} \times Motor\_Position[IU]$$

where:
No_steps – is the number of motor steps per revolution
No_µsteps – is the number of microsteps per step. You can read/change this value in the “Drive Setup” dialogue from PROconfig.
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

6.1.3. Step motor closed-loop control. Incremental encoder on motor

The internal position units are motor encoder counts. The correspondence with the load **position in SI units** [2] is:

$$Load\_Position[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr} \times Motor\_Position[IU]$$

where:
No_encoder_lines – is the motor encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

6.2. Speed units

The internal speed units are internal position units / (slow loop sampling period) i.e. the position variation over one slow loop sampling period

6.2.1. Brushless motor with quadrature encoder on motor

The internal speed units are encoder counts / (slow loop sampling period). The correspondence with the load **speed in SI units** is:

$$Load\_Speed[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr \times T} \times Motor\_Speed[IU]$$

where:
No_encoder_lines – is the rotary encoder number of lines per revolution

---

[1] SI units for position are: [rad] for a rotary movement, [m] for a linear movement
[2] SI units for position are [rad] for a rotary movement , [m] for a linear movement

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.2.2. Step motor open-loop control. No feedback device

The internal speed units are motor µsteps / (slow loop sampling period). The correspondence with the load **speed in SI units** is:

$$Load\_Speed[SI] = \frac{2 \times \pi}{No\_\mu steps \times No\_steps \times Tr \times T} \times Motor\_Speed[IU]$$

where:
No_steps – is the number of motor steps per revolution
No_µsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from PROconfig.
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.2.3. Step motor closed-loop control. Incremental encoder on motor

The internal speed units are motor encoder counts / (slow loop sampling period). The correspondence with the load **speed in SI units**[1] is:

$$Load\_Speed[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr \times T} \times Motor\_Speed[IU]$$

where:
No_encoder_lines – is the motor encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup".

6.3. Acceleration units

The internal acceleration units are internal position units / (slow loop sampling period)^2 i.e. the speed variation over one slow loop sampling period.

6.3.1. Brushless motor with quadrature encoder on motor

The internal acceleration units are encoder counts / (slow loop sampling period)^2. The correspondence with the load **acceleration in SI units** is

$$Load\_Acceleration[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr \times T^2} \times Motor\_Acceleration[IU]$$

where:
No_encoder_lines – is the rotary encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.3.2. Step motor open-loop control. No feedback device

The internal acceleration units are motor µsteps / (slow loop sampling period)^2. The correspondence with the load **acceleration in SI units** is:

$$Load\_Acceleration[SI] = \frac{2 \times \pi}{No\_\mu steps \times No\_steps \times Tr \times T^2} \times Motor\_Acceleration[IU]$$

where:

[1] SI units for speed are [rad/s] for a rotary movement , [m/s] for a linear movement

No_steps – is the number of motor steps per revolution
No_µsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from PROconfig.
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.3.3. Step motor closed-loop control. Incremental encoder on motor

The internal acceleration units are motor encoder counts / (slow loop sampling period)^2. The transmission is rotary-to-rotary. The correspondence with the load **acceleration in SI units** is:

$$Load\_Acceleration[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr \times T^2} \times Motor\_Acceleration[IU]$$

where:
No_encoder_lines – is the motor encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.4. Jerk units

The internal jerk units are internal position units / (slow loop sampling period)^3 i.e. the acceleration variation over one slow loop sampling period.

6.4.1. Brushless motor with quadrature encoder on motor

The internal jerk units are encoder counts / (slow loop sampling period)^3. The correspondence with the load **jerk in SI units**[1] is:

$$Load\_Jerk[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr \times T^3} \times Motor\_Jerk[IU]$$

where:
No_encoder_lines – is the rotary encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.4.2. Step motor open-loop control. No feedback device

The internal jerk units are motor µsteps / (slow loop sampling period)^3. The correspondence with the load **jerk in SI units**[2] is:

$$Load\_Jerk[SI] = \frac{2 \times \pi}{No\_\mu steps \times No\_steps \times Tr \times T^3} \times Motor\_Jerk[IU]$$

where:
No_steps – is the number of motor steps per revolution
No_µsteps – is the number of microsteps per step. You can read/change this value in the "Drive Setup" dialogue from PROconfig.
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the "Advanced" dialogue, which can be opened from the "Drive Setup"

6.4.3. Step motor closed-loop control. Incremental encoder on motor

The internal jerk units are motor encoder counts / (slow loop sampling period)^3. The correspondence with the load jerk in SI units is:

[1] SI units for jerk are [rad/s^3] for a rotary movement, [m/s^3] for a linear movement
[2] SI units for jerk are [rad/s^3] for a rotary movement, [m/s^3] for a linear movement

$$Load\_Jerk[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines \times Tr \times T^3} \times Motor\_Jerk[IU]$$

where:
No_encoder_lines – is the motor encoder number of lines per revolution
Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units
T – is the slow loop sampling period expressed in [s]. You can read this value in the “Advanced” dialogue, which can be opened from the “Drive Setup”.

6.5. Current units

The internal current units refer to the motor phase currents. The correspondence with the motor currents in [A] is:

$$Current[A] = \frac{2 \times Ipeak}{65520} \times Current[IU]$$

where Ipeak – is the IMD peak current expressed in [A]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”.

6.6. Voltage command units

The internal voltage command units refer to the voltages applied on the motor. The significance of the voltage commands as well as the scaling factors, depend on the motor type and control method used.

In case of **brushless motors** driven in **sinusoidal** mode, a field oriented vector control is performed. The voltage command is the amplitude of the sinusoidal phase voltages. In this case, the correspondence with the motor phase voltages in SI units i.e. [V] is:

$$Voltage\ command[V] = \frac{1.1 \times Vmot}{65534} \times Voltage\ command[IU]$$

where Vmot – is the IMD motor power supply voltage expressed in [V].

In case of **brushless** motors driven in **trapezoidal** mode, the voltage command is the voltage to apply between 2 of the motor phases, according with Hall signals values. In this case, the correspondence with the voltage applied in SI units i.e. [V] is:

$$Voltage\ command[V] = \frac{Vdc}{32767} \times Voltage\ command[IU]$$

6.7. Voltage measurement units

The internal voltage measurement units refer to the drive V_{MOT} supply voltage. The correspondence with the supply voltage in [V] is:

$$Voltage\_measured[V] = \frac{VdcMaxMeasurable}{65520} \times Voltage\_measured[IU]$$

where VdcMaxMeasurable – is the maximum measurable DC voltage expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”.

Remark: *the voltage measurement units occur in the scaling of the over voltage and under voltage protections and the supply voltage measurement*

6.8. Time units

The internal time units are expressed in slow loop sampling periods. The correspondence with the time in [s] is:

$$Time[s] = T \times Time[IU]$$

where T – is the slow loop sampling period expressed in [s]. You can read this value in the “Advanced” dialogue, which can be opened from the “Drive Setup”. For example, if T = 1ms, one second = 1000 IU.

6.9. Master position units

When the master position is sent via a communication channel, the master position units depend on the type of position sensor present on the master axis.

6.10. Master speed units

The master speed is computed in internal units (IU) as master position units / slow loop sampling period i.e. the master position variation over one position/speed loop sampling period.

6.11. Motor position units

6.11.1. Brushless motor with quadrature encoder on motor

The internal motor position units are encoder counts. The correspondence with the motor **position in SI units**[1] is:

$$Motor\_Position[SI] = \frac{2 \times \pi}{4 \times No\_encoder\_lines} \times Motor\_Position[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

6.11.2. Step motor open-loop control. No feedback device

The internal motor position units are motor µsteps. The correspondence with the motor **position in SI units**[1] is:

$$Motor\_Position[SI] = \frac{2 \times \pi}{No\_\mu steps \times No\_steps} \times Motor\_Position[IU]$$

where:

No_steps – is the number of motor steps per revolution

No_µsteps – is the number of microsteps per step. You can read/change this value in the “Drive Setup” dialogue from PROconfig.

6.11.3. Step motor closed-loop control. Incremental encoder on motor

[1] SI units for motor position are: [rad] for a rotary motor, [m] for a linear motor

The internal motor position units are motor encoder counts. The correspondence with the motor position in SI units is:

$$Motor\_Position[SI] = \frac{2\times\pi}{4\times No\_encoder\_lines}\times Motor\_Position[IU]$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

6.12. Motor speed units

6.12.1. Brushless motor with quadrature encoder on motor

The internal motor speed units are encoder counts / (slow loop sampling period). The correspondence with the motor speed in SI units is:

For rotary motors:

$$Motor\_Speed[SI] = \frac{2\times\pi}{4\times No\_encoder\_lines\times T}\times Motor\_Speed[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the “Advanced” dialogue, which can be opened from the “Drive Setup”

6.12.2. Step motor open-loop control. No feedback device

The internal motor speed units are motor µsteps / (slow loop sampling period). The correspondence with the motor **speed in SI units** is:

$$Motor\_Speed[SI] = \frac{2\times\pi}{No\_\mu steps\times No\_steps\times T}\times Motor\_Speed[IU]$$

where:

No_steps – is the number of motor steps per revolution

No_µsteps – is the number of microsteps per step. You can read/change this value in the “Drive Setup” dialogue from PROconfig.

T – is the slow loop sampling period expressed in [s]. You can read this value in the “Advanced” dialogue, which can be opened from the “Drive Setup”

6.12.3. Step motor closed-loop control. Incremental encoder on motor

The internal motor speed units are motor encoder counts / (slow loop sampling period). The correspondence with the load speed in SI units is:

$$Motor\_Speed[SI] = \frac{2\times\pi}{4\times No\_encoder\_lines\times T}\times Motor\_Speed[IU]$$

where:

No_encoder_lines – is the motor encoder number of lines per revolution

T – is the slow loop sampling period expressed in [s]. You can read this value in the “Advanced” dialogue, which can be opened from the “Drive Setup”.

7. Memory Map

PRO Series IMD has 2 types of memory available for user applications: 1K×16 SRAM and 4K×16 serial E^2ROM.

The SRAM memory is mapped in the address range: 9000h to 9FFFh. It can be used to download and run a MPL program, to save real-time data acquisitions and to keep the cam tables during run-time.

The E^2ROM is mapped in the address range: 4000h to 4FFFh. It is used to keep in a non-volatile memory the MPL programs, the cam tables and the drive setup information.

Remark: *MotionPRO Suite handles automatically the memory allocation for each motion application. The memory map can be accessed and modified from the main folder of each application*

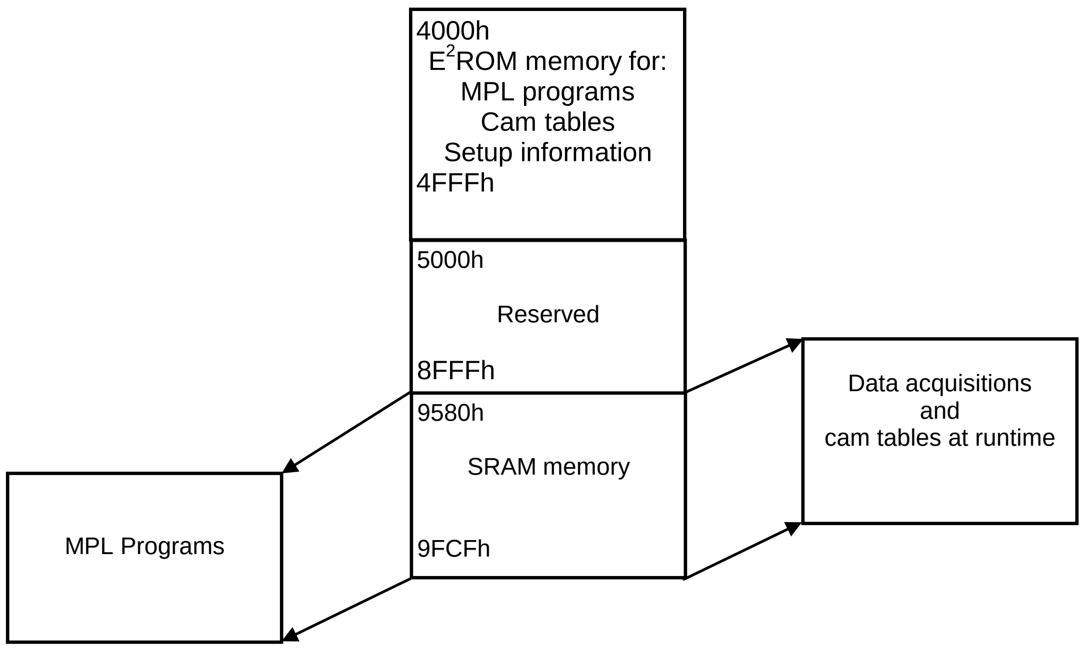


Figure 7.1. *PRO Series IMD Memory Map*

8. APPENDIX A: PRO Series IMD Mechanical Dimensions

8.1. Rotary BLDC Models

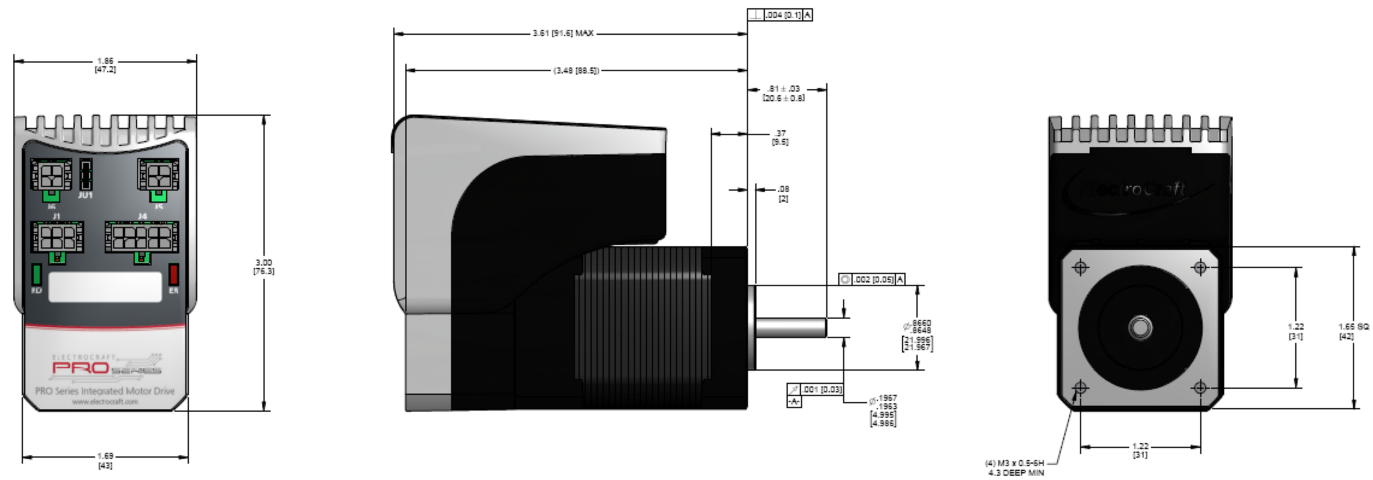


Figure 8.1: PR42-A16V48A mechanical dimensions.

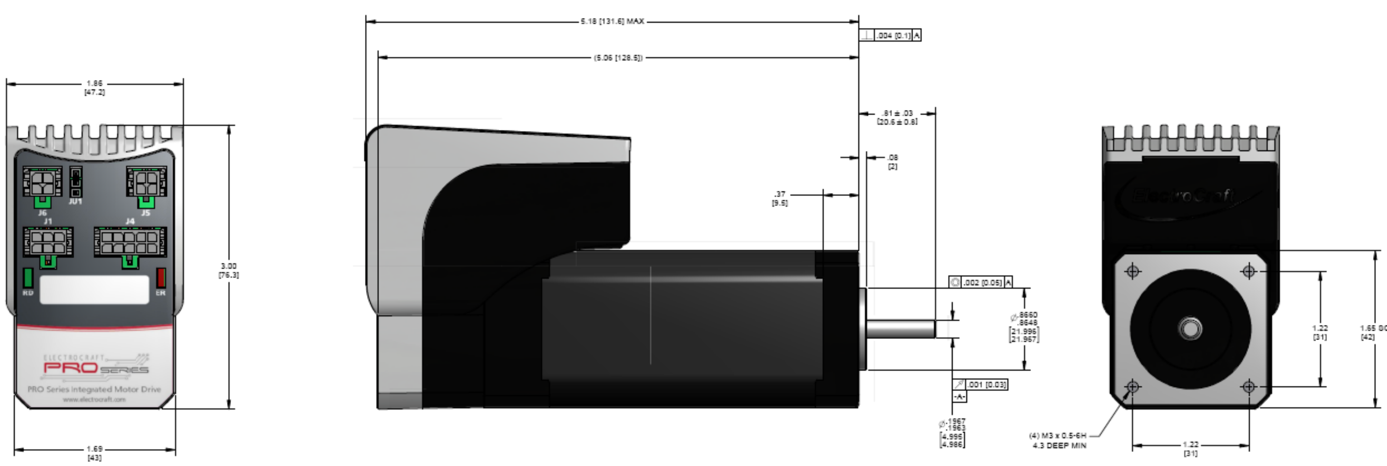


Figure 8.2: PR42-A32V48A mechanical dimensions.

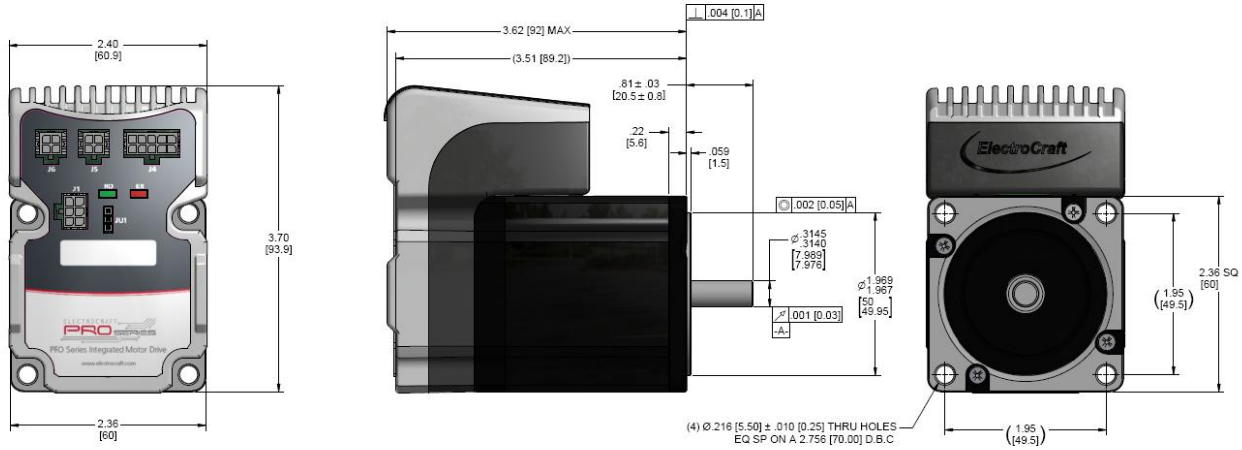


Figure 8.3: PR60-A25V48A mechanical dimensions.

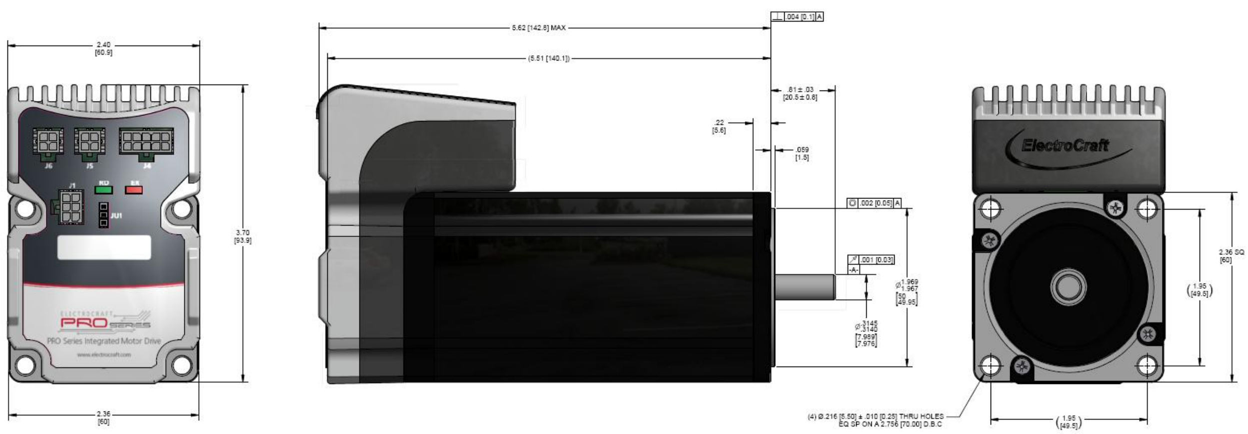


Figure 8.4: PR60-A52V48A mechanical dimensions.

8.2. Linear Stepper Models

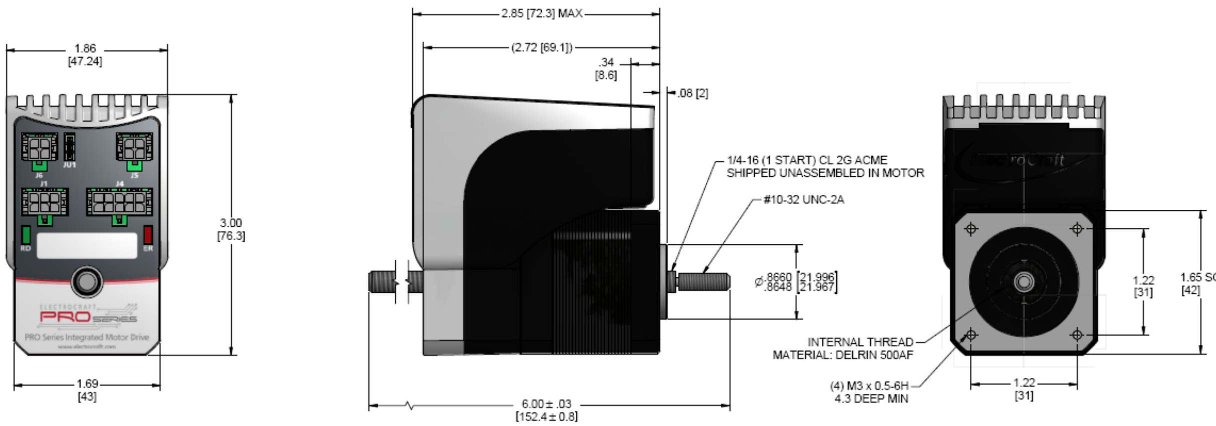


Figure 8.5: PS42-A44V48A-0AB4-D-CAN mechanical dimensions.

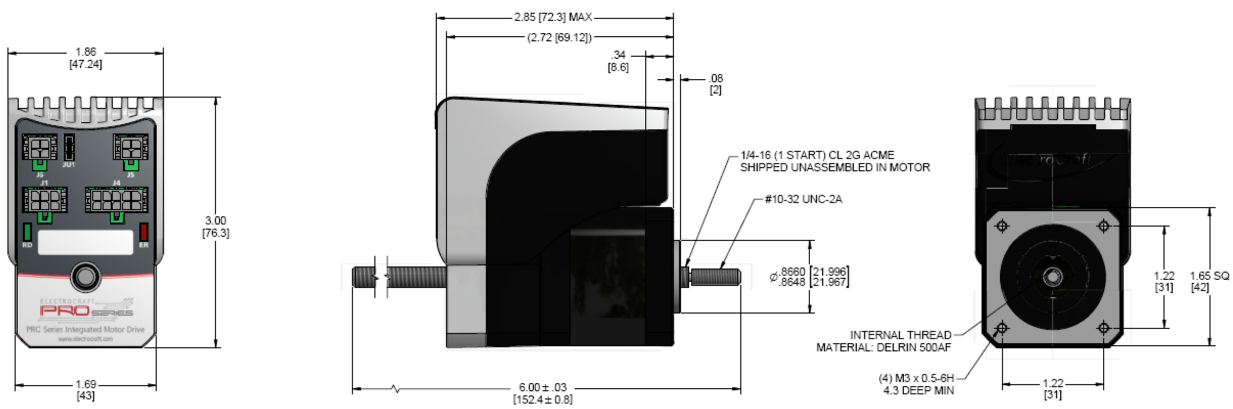


Figure 8.6: PS42-A44V48A-0AB4-X-CAN mechanical dimensions.

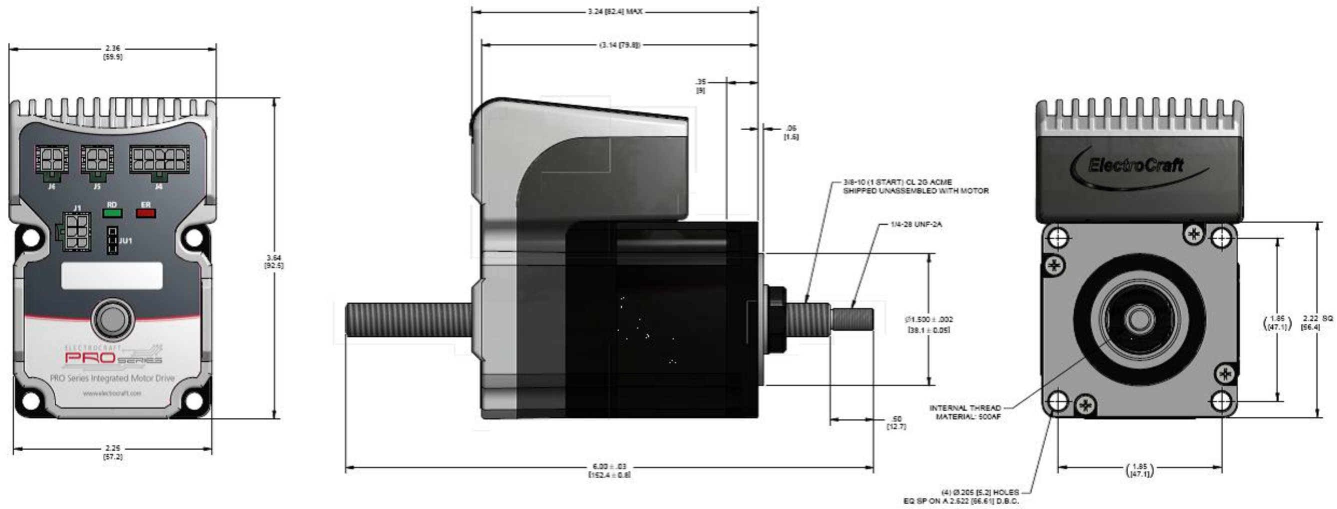


Figure 8.7: PS56-A106V48A-0SB7-D-CAN mechanical dimensions.

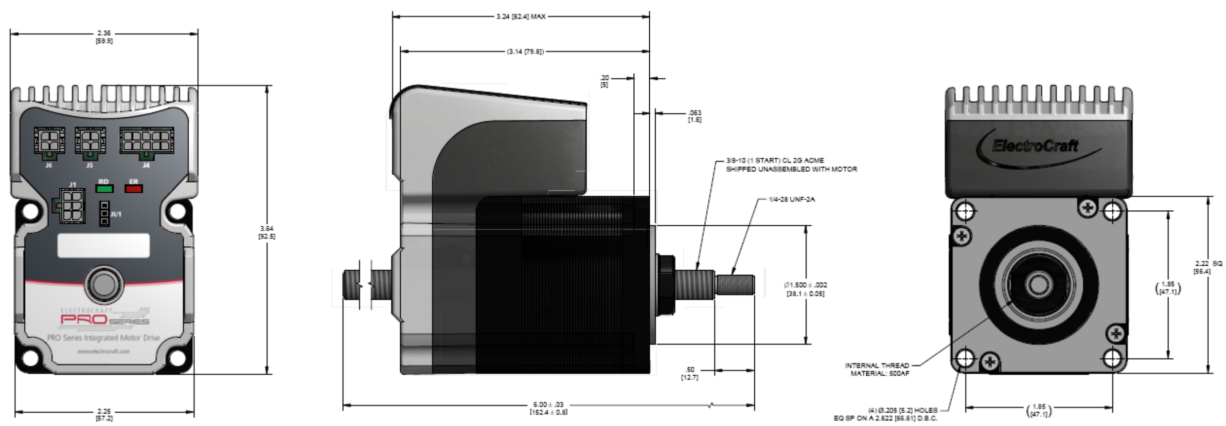


Figure 8.8: PS56-A106V48A-0SB7-X-CAN mechanical dimensions.

8.3. Rotary Stepper Models

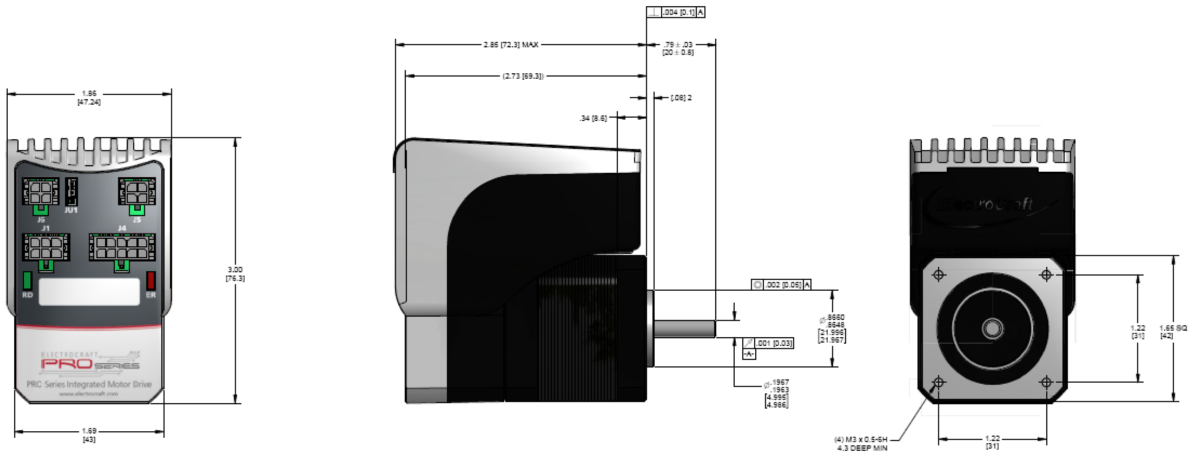


Figure 8.9: PT42-A44V48A-0-D-CAN mechanical dimensions.

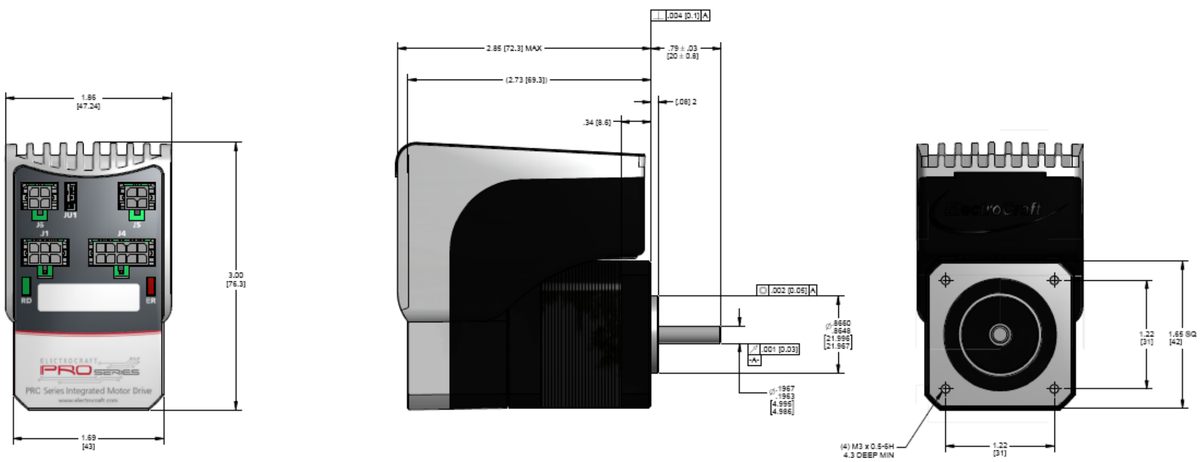


Figure 8.10: PT42-A44V48A-0-X-CAN mechanical dimensions.

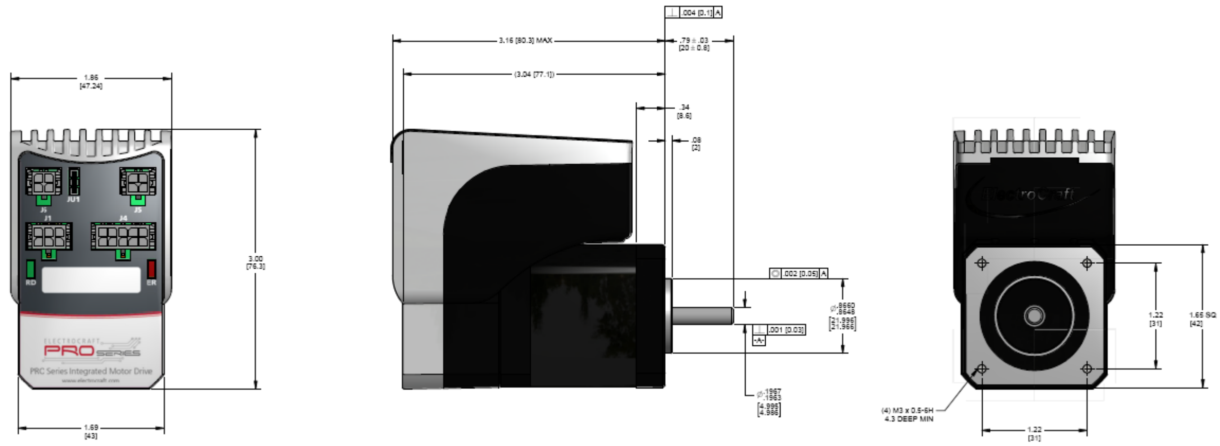


Figure 8.11: PT42-A55V48A-0-D-CAN mechanical dimensions.

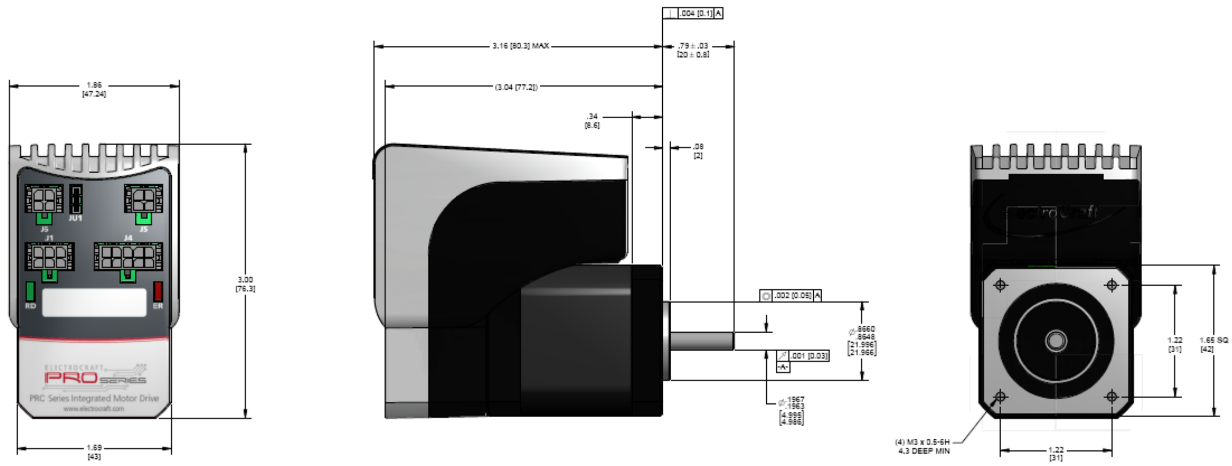


Figure 8.12: PT42-A55V48A-0-X-CAN mechanical dimensions.

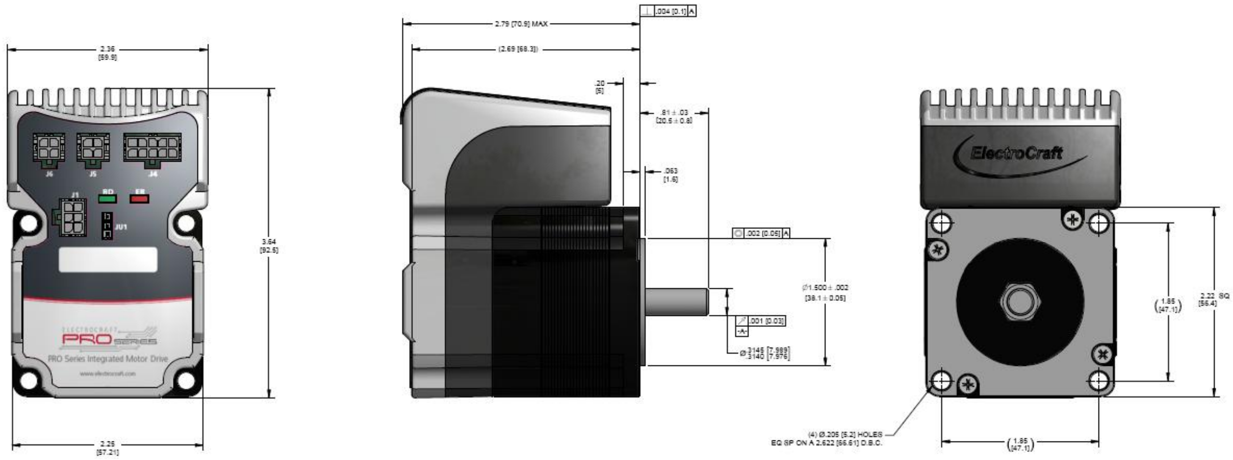


Figure 8.13: PT56-A64V48A-0-D-CAN mechanical dimensions.

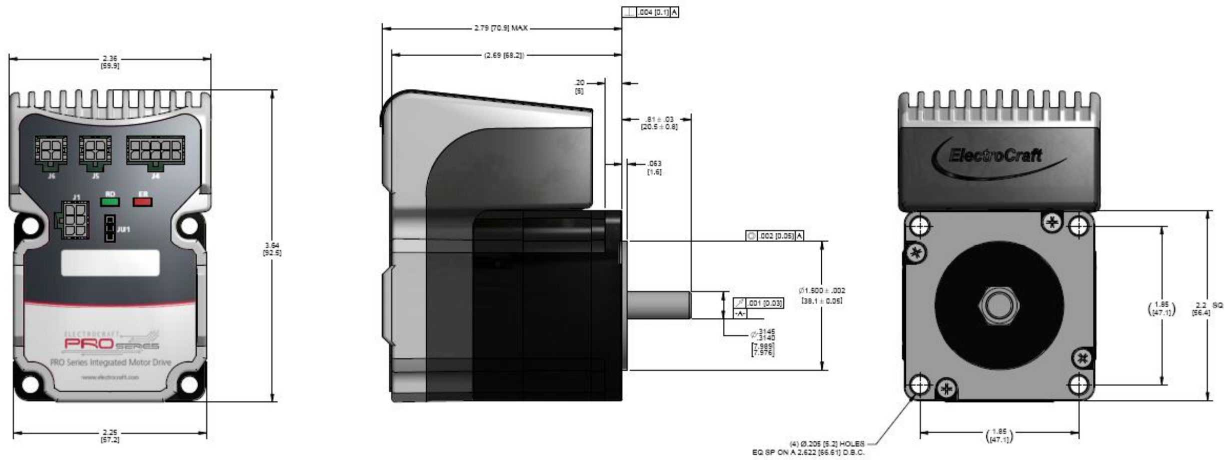


Figure 8.14: PT56-A64V48A-0-X-CAN mechanical dimensions.

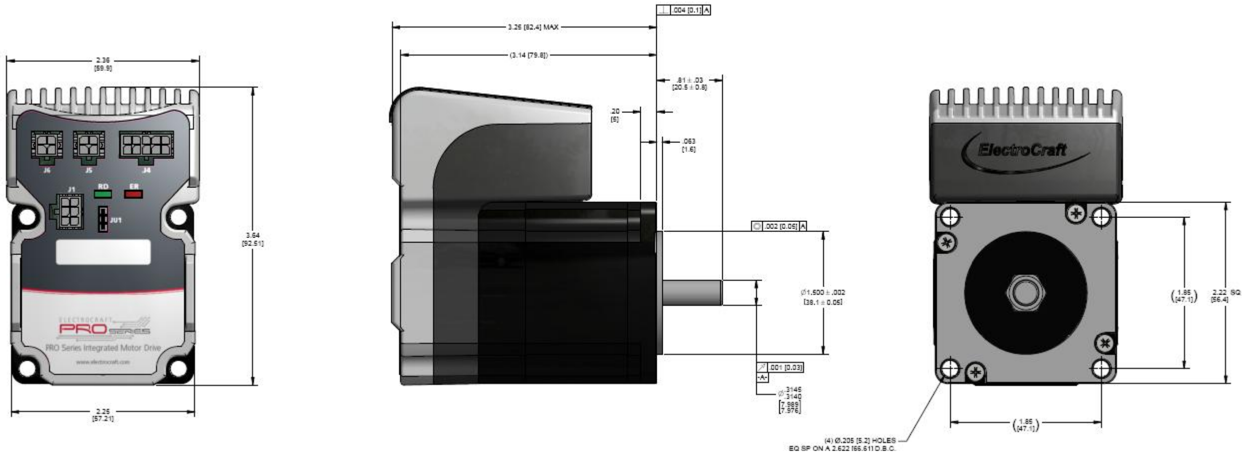


Figure 8.15: PT56-A106V48A-0-D-CAN mechanical dimensions.

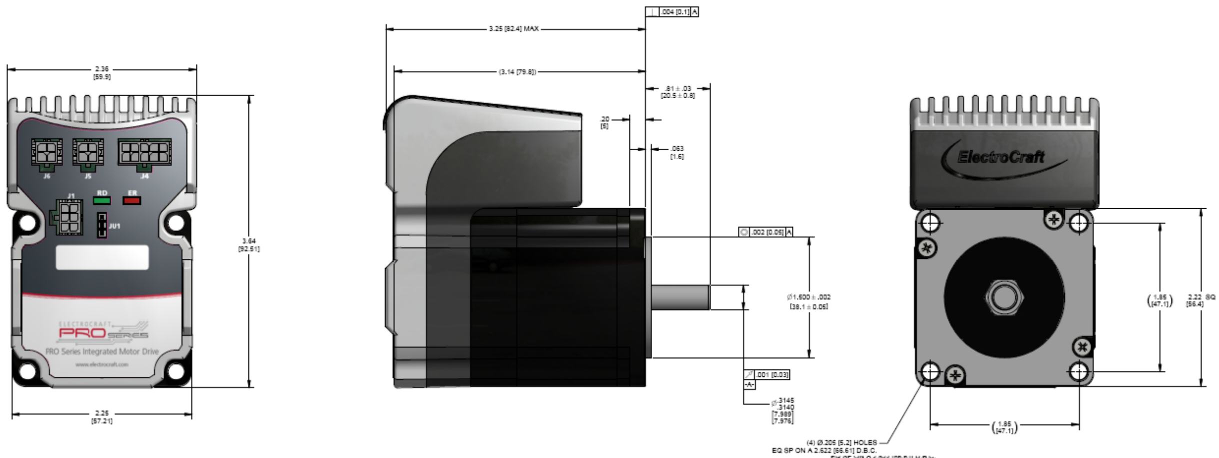


Figure 8.16: PT56-A106V48A-0-X-CAN mechanical dimensions.

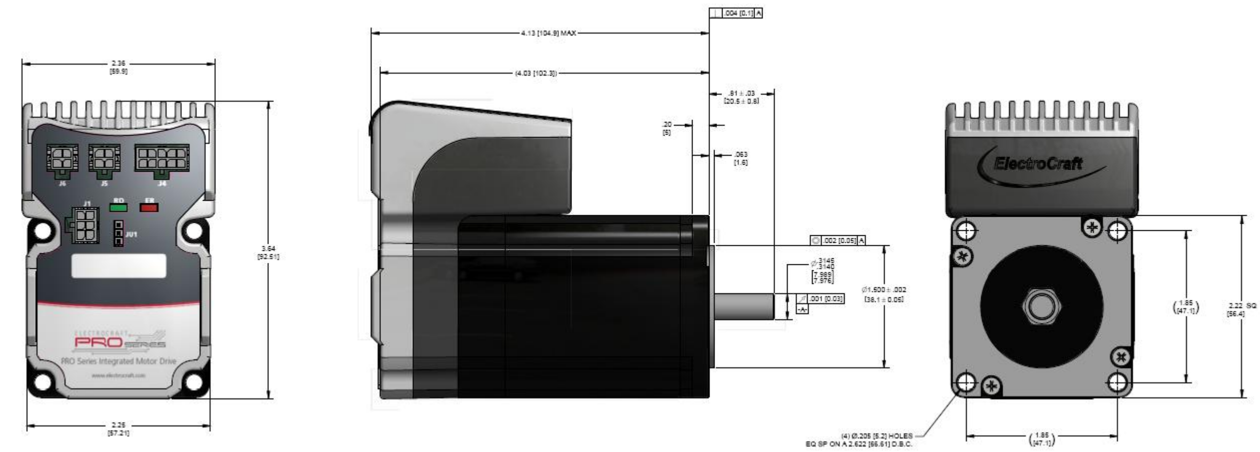


Figure 8.17: PT56-A170V48A-0-D-CAN mechanical dimensions.

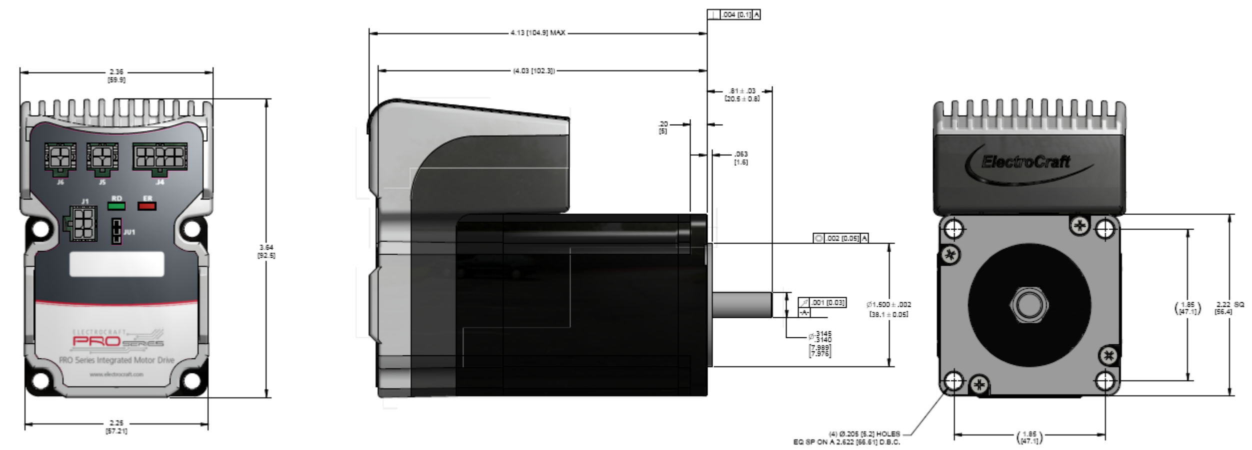
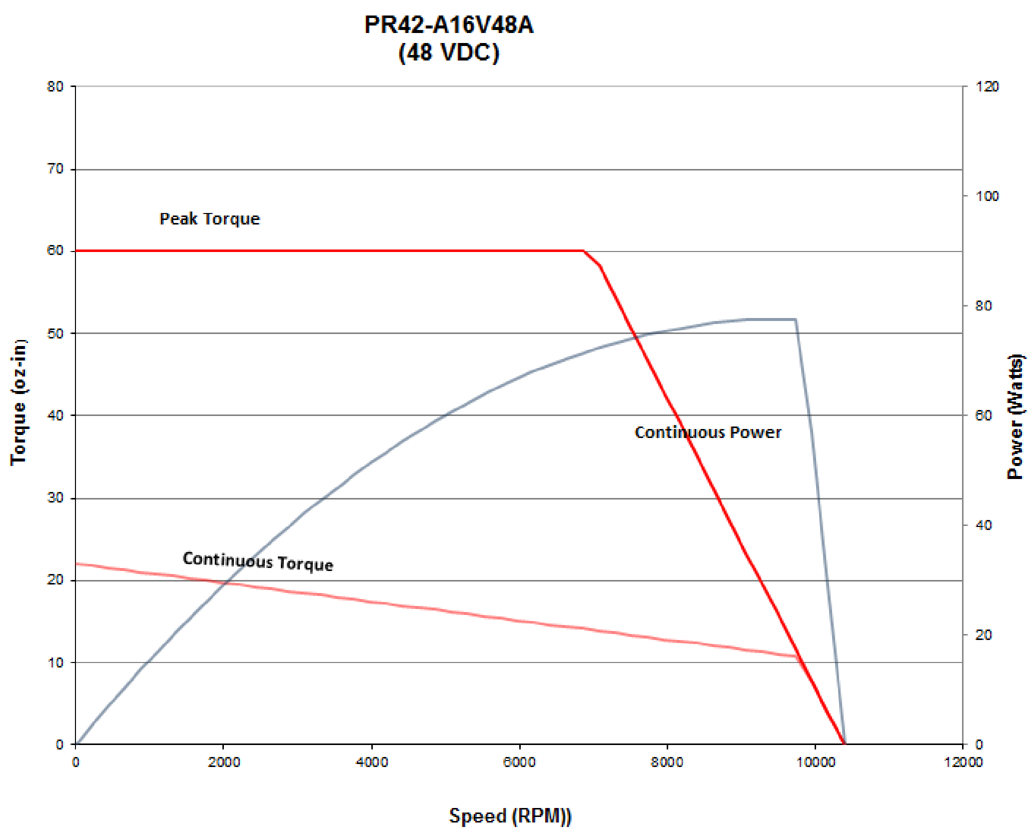
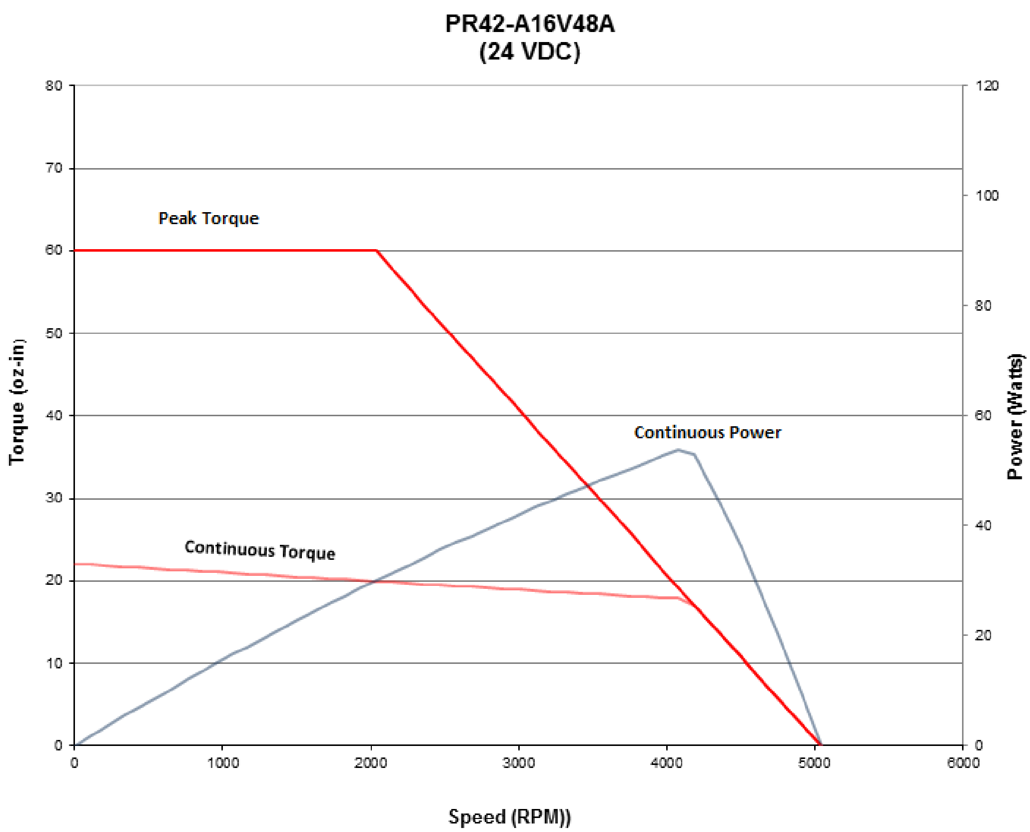


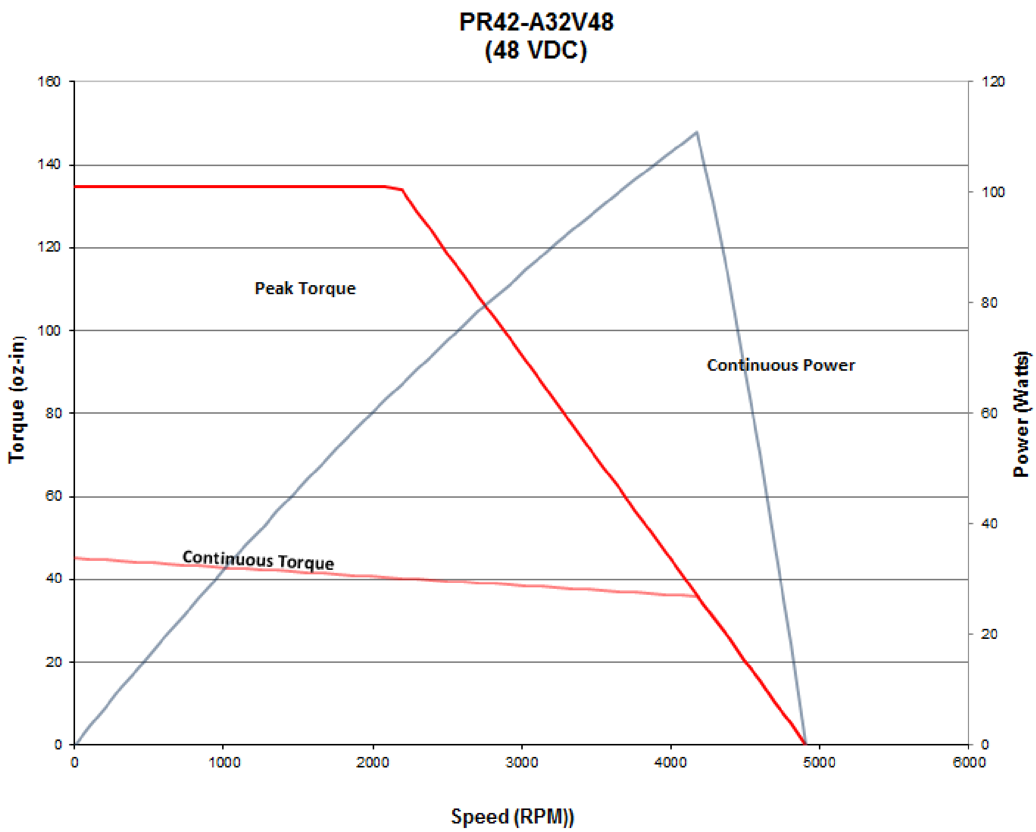
Figure 8.18: PT56-A170V48A-0-X-CAN mechanical dimensions.

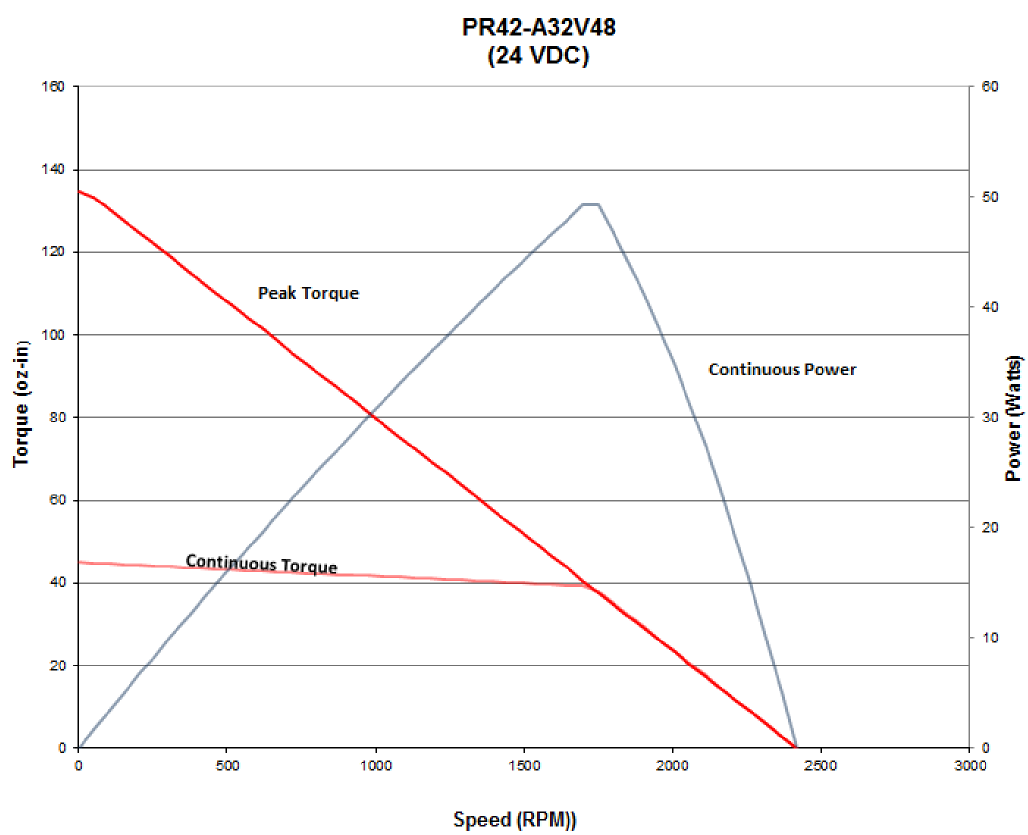
9. APPENDIX B: PRO Series IMD Torque-Speed Curves

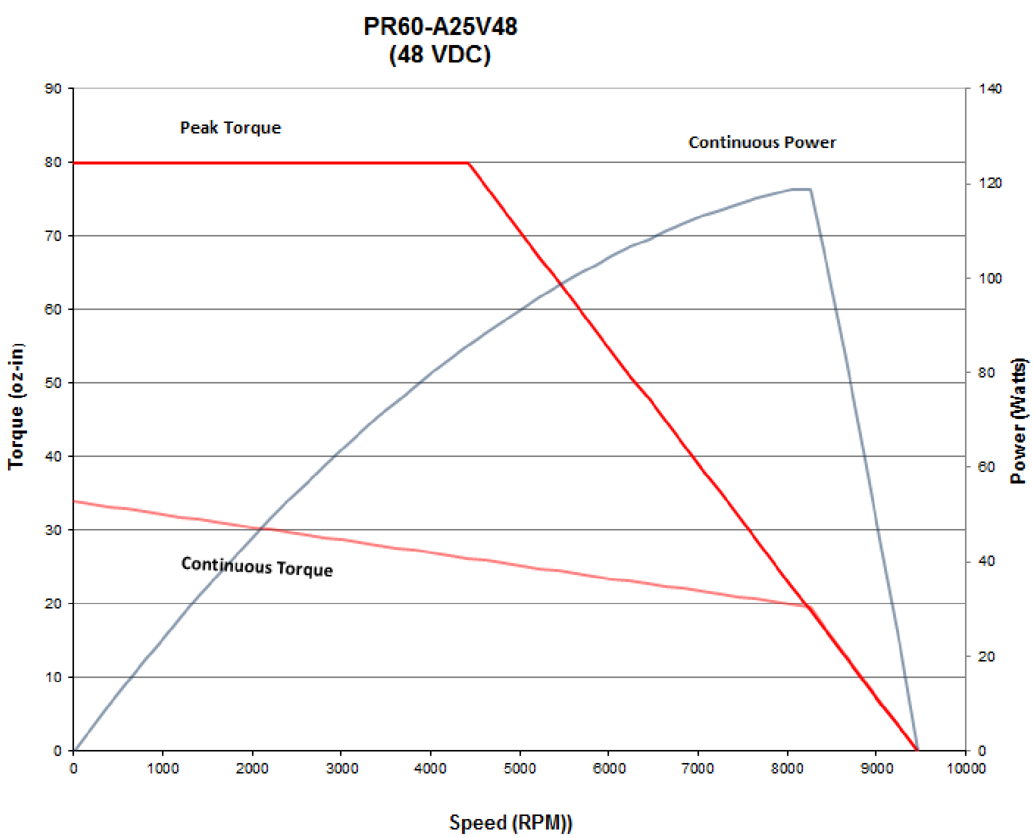
9.1. Rotary BLDC models

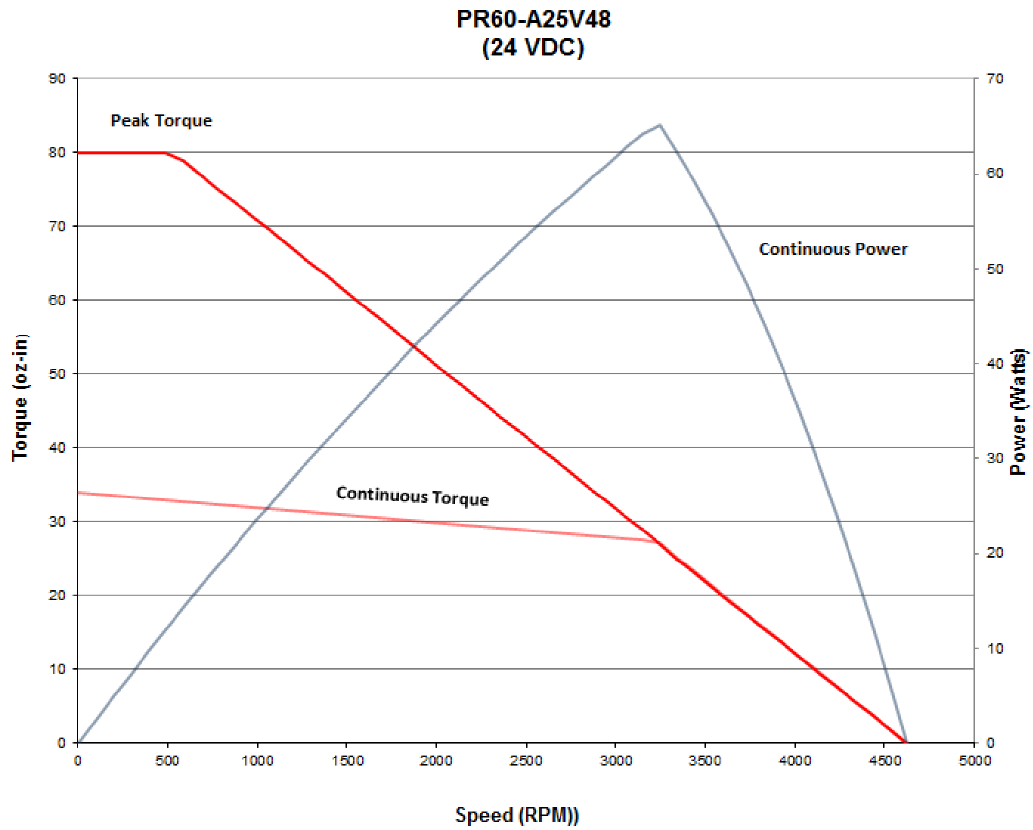


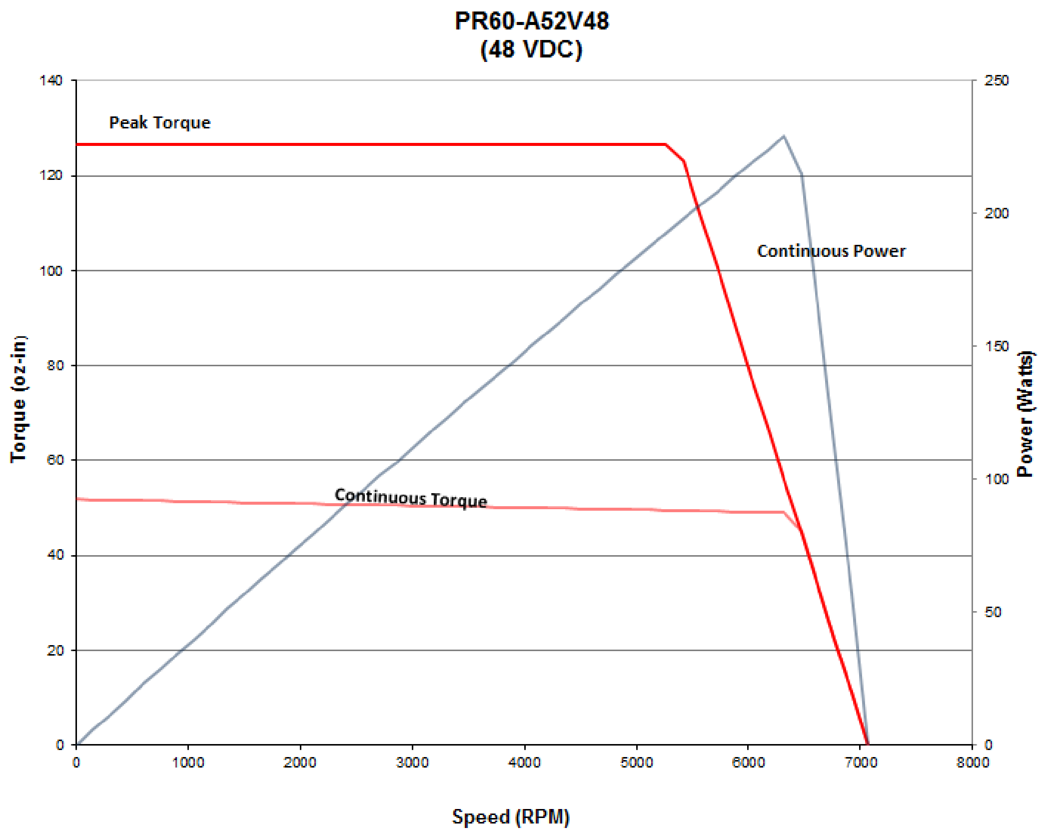

PR42-A16V48A
(24 VDC)
Peak Torque
Continuous Power
Continuous Torque
Torque (oz-in)
Power (Watts)
Speed (RPM))
0
10
20
30
40
50
60
70
80
0
20
40
60
80
100
120
0
1000
2000
3000
4000
5000
6000

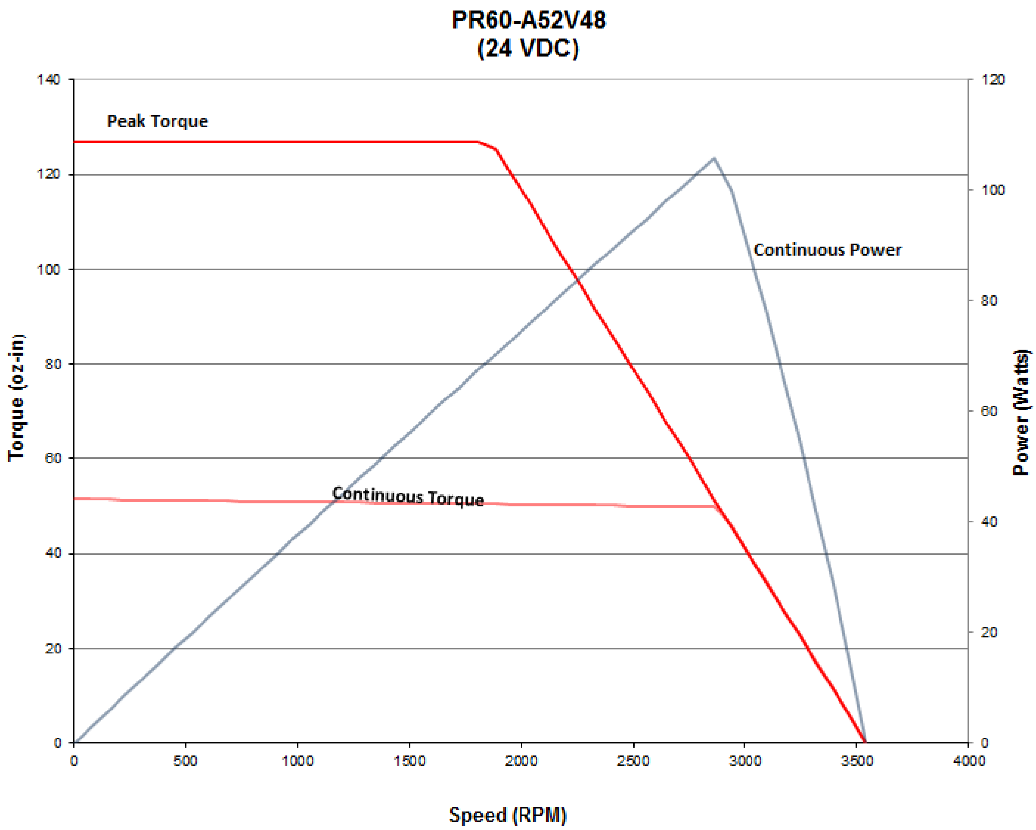

PR42-A32V48
(48 VDC)
Torque (oz-in)
Power (Watts)
Speed (RPM))
Peak Torque
Continuous Power
Continuous Torque
0
20
40
60
80
100
120
140
160
0
1000
2000
3000
4000
5000
6000
0
20
40
60
80
100
120


PR42-A32V48
(24 VDC)
Torque (oz-in)
Power (Watts)
Speed (RPM))
Peak Torque
Continuous Power
Continuous Torque
160
140
120
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
60
50
40
30
20
10

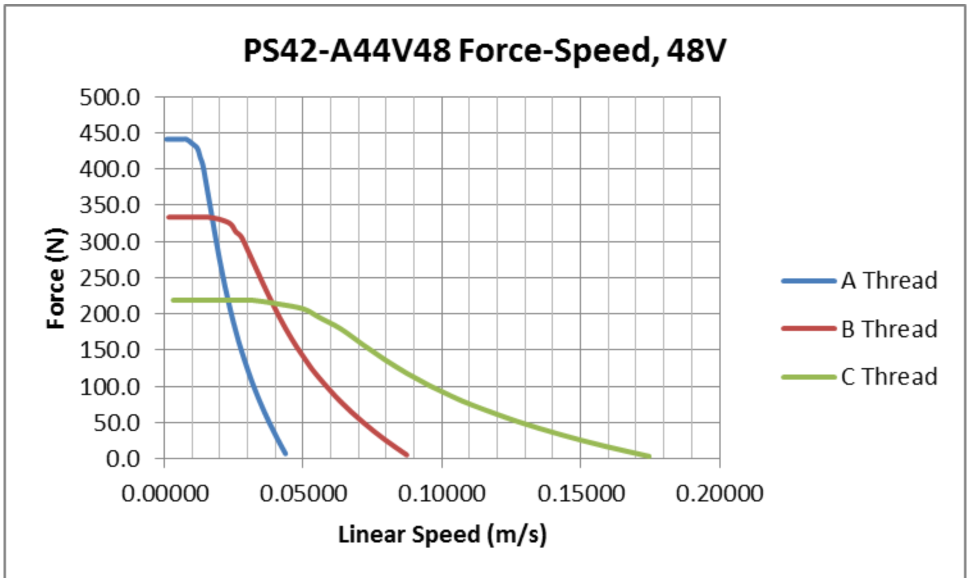

PR60-A25V48
(48 VDC)
Peak Torque
Continuous Power
Continuous Torque
Torque (oz-in)
Power (Watts)
Speed (RPM))
0
10
20
30
40
50
60
70
80
90
0
20
40
60
80
100
120
140
0
1000
2000
3000
4000
5000
6000
7000
8000
9000
10000

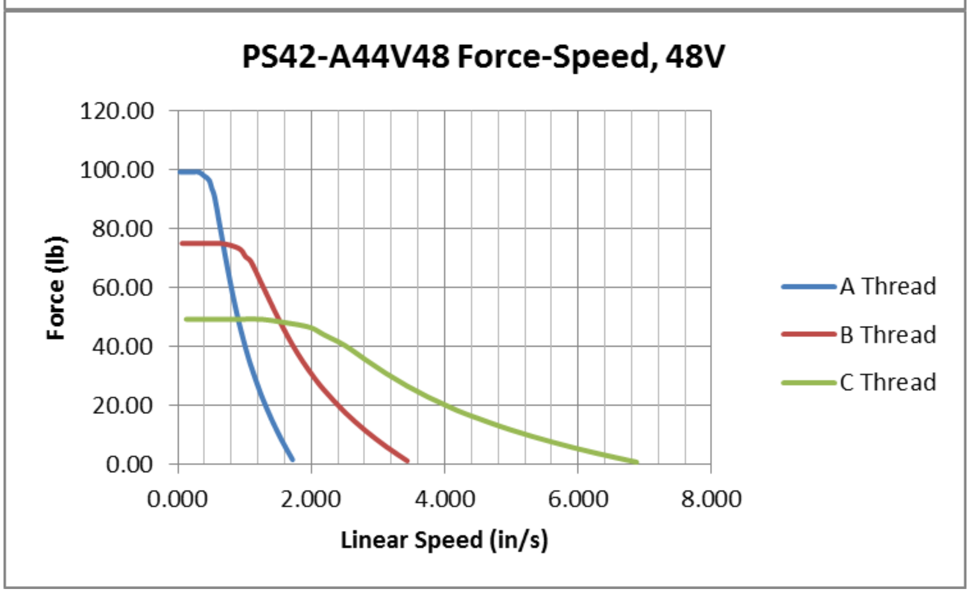

PR60-A25V48
(24 VDC)
Peak Torque
Continuous Power
Continuous Torque
Torque (oz-in)
Power (Watts)
Speed (RPM))
0
10
20
30
40
50
60
70
80
90
500
1000
1500
2000
2500
3000
3500
4000
4500
5000

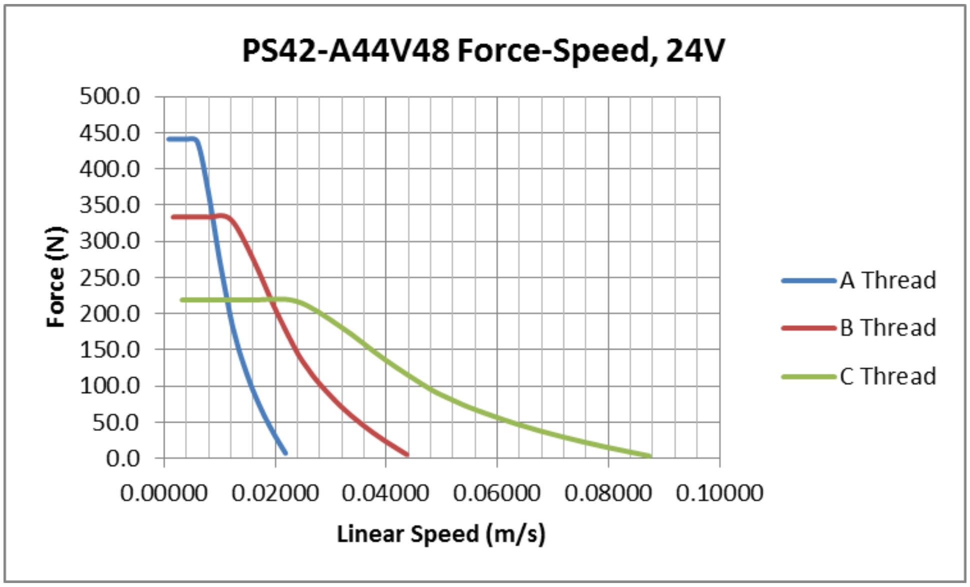

PR60-A52V48
(48 VDC)
Peak Torque
Continuous Power
Continuous Torque
Torque (oz-in)
Power (Watts)
Speed (RPM)
0
20
40
60
80
100
120
140
0
50
100
150
200
250
0
1000
2000
3000
4000
5000
6000
7000
8000

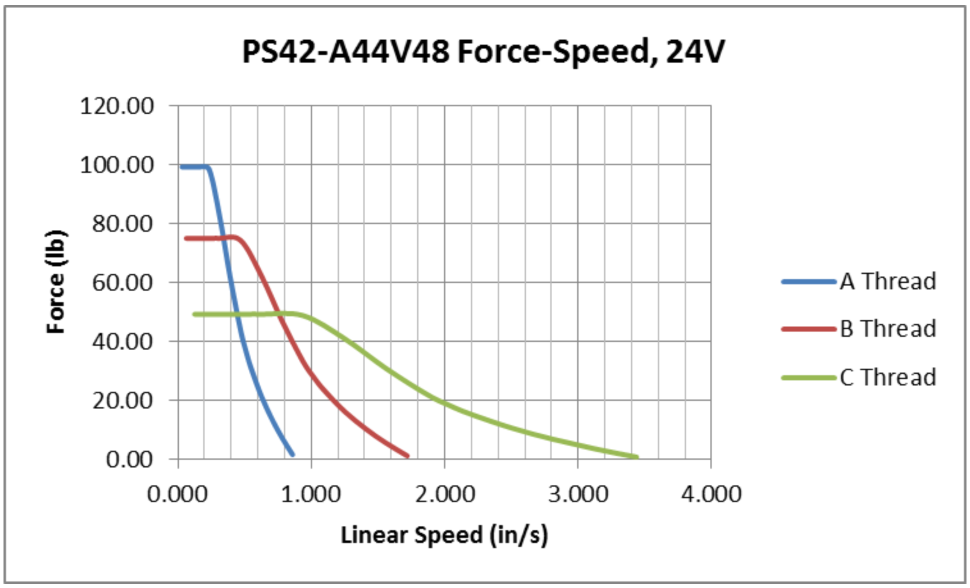

PR60-A52V48
(24 VDC)
Peak Torque
Continuous Power
Continuous Torque
Torque (oz-in)
Power (Watts)
Speed (RPM)
140
120
100
80
60
40
20
0
0
500
1000
1500
2000
2500
3000
3500
4000

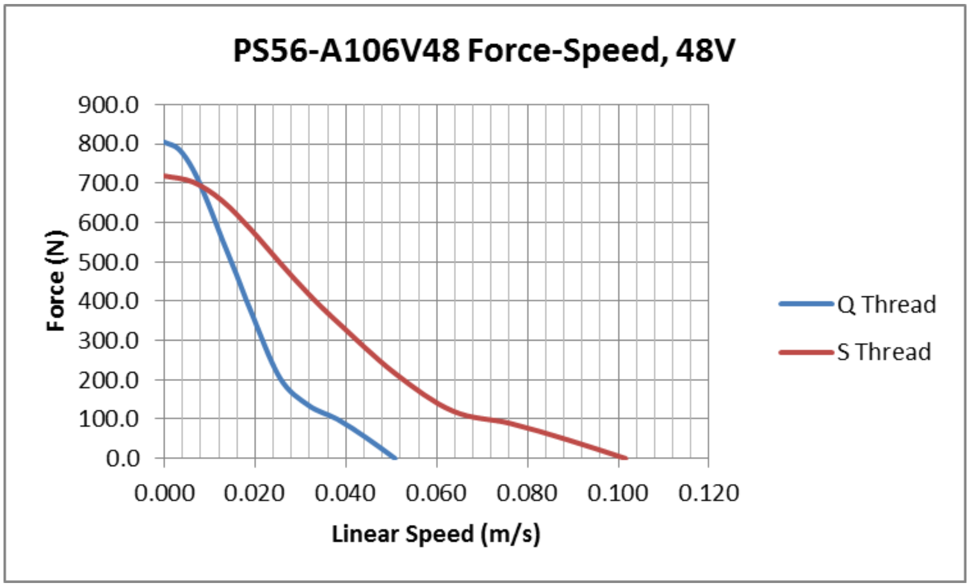
9.2. Linear Stepper Models

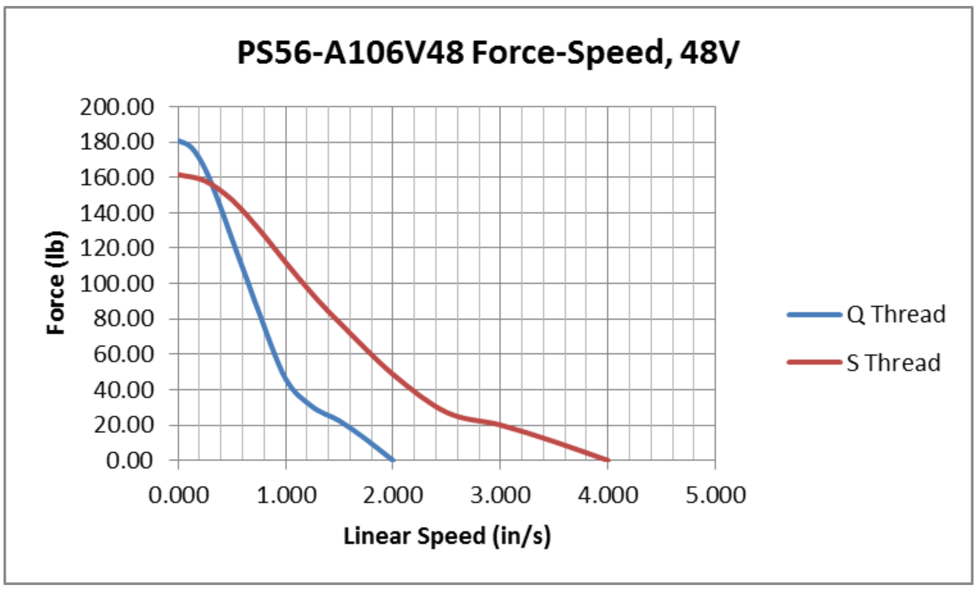


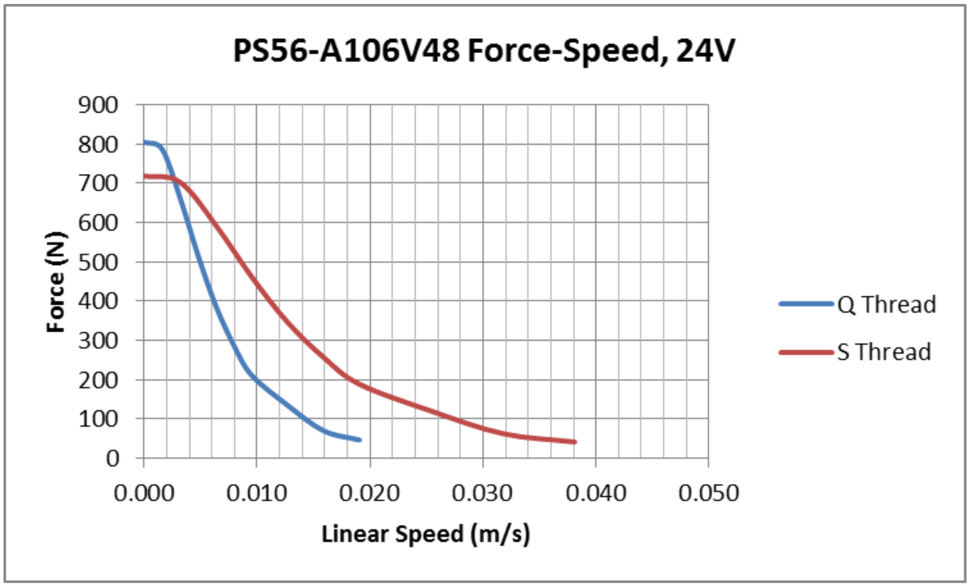


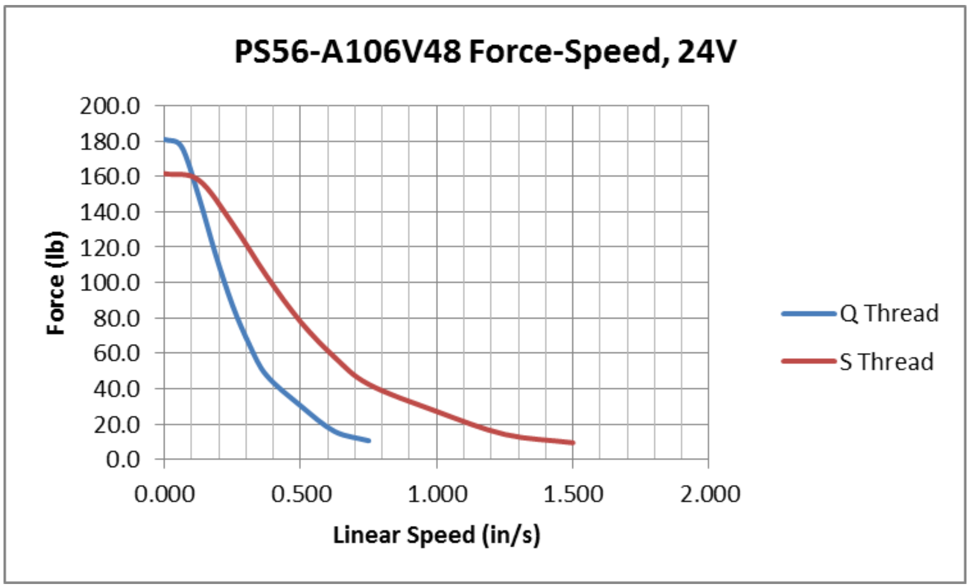

PS42-A44V48 Force-Speed, 24V
Force (N)
500.0
450.0
400.0
350.0
300.0
250.0
200.0
150.0
100.0
50.0
0.0
0.00000
0.02000
0.04000
0.06000
0.08000
0.10000
Linear Speed (m/s)
A Thread
B Thread
C Thread


PS42-A44V48 Force-Speed, 24V
Force (lb)
120.00
100.00
80.00
60.00
40.00
20.00
0.00
0.000
1.000
2.000
3.000
4.000
Linear Speed (in/s)
A Thread
B Thread
C Thread

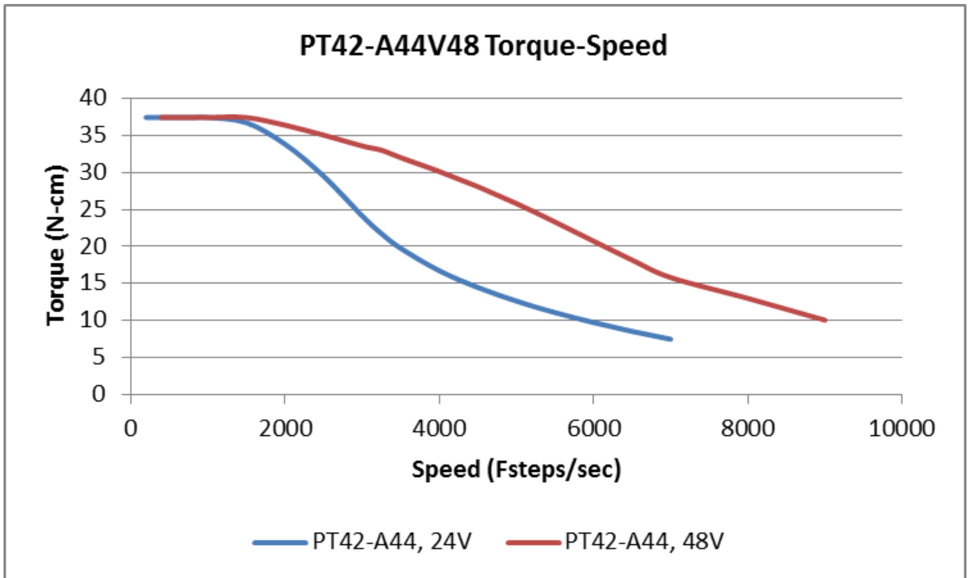

PS56-A106V48 Force-Speed, 48V
Force (N)
900.0
800.0
700.0
600.0
500.0
400.0
300.0
200.0
100.0
0.0
0.000
0.020
0.040
0.060
0.080
0.100
0.120
Linear Speed (m/s)
Q Thread
S Thread

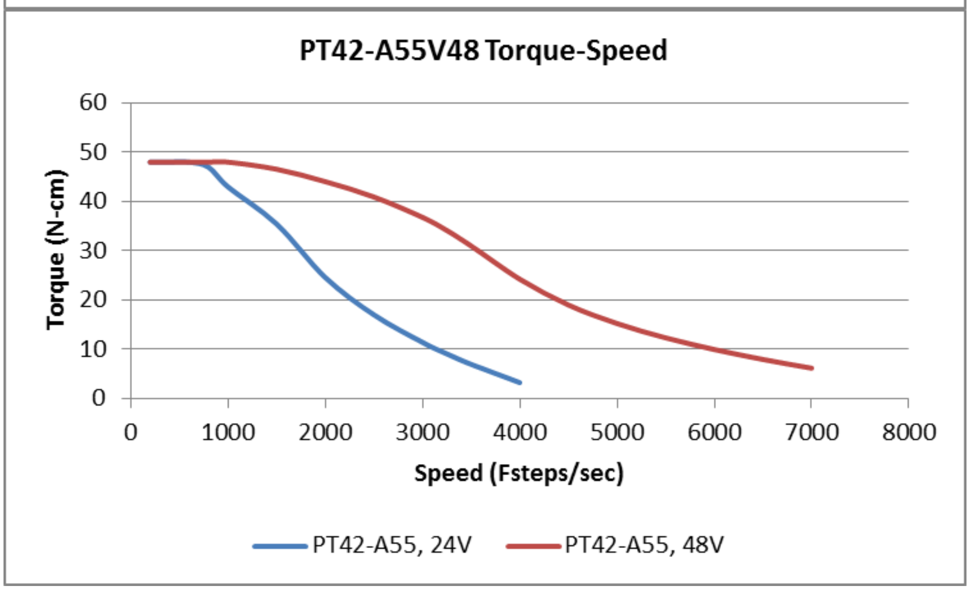

PS56-A106V48 Force-Speed, 48V
Force (lb)
200.00
180.00
160.00
140.00
120.00
100.00
80.00
60.00
40.00
20.00
0.00
0.000
1.000
2.000
3.000
4.000
5.000
Linear Speed (in/s)
Q Thread
S Thread

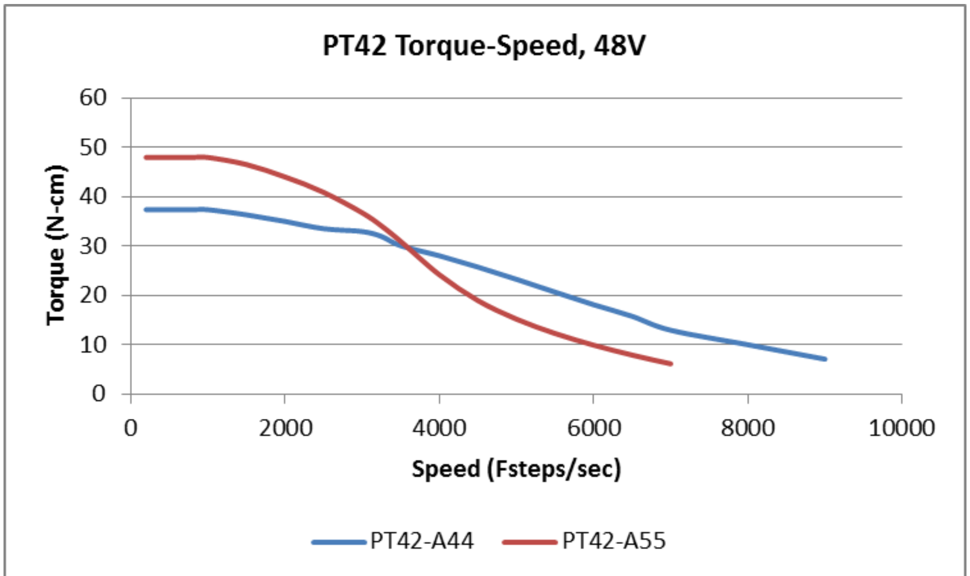

PS56-A106V48 Force-Speed, 24V
900
800
700
600
500
400
300
200
100
0
Force (N)
0.000
0.010
0.020
0.030
0.040
0.050
Linear Speed (m/s)
Q Thread
S Thread

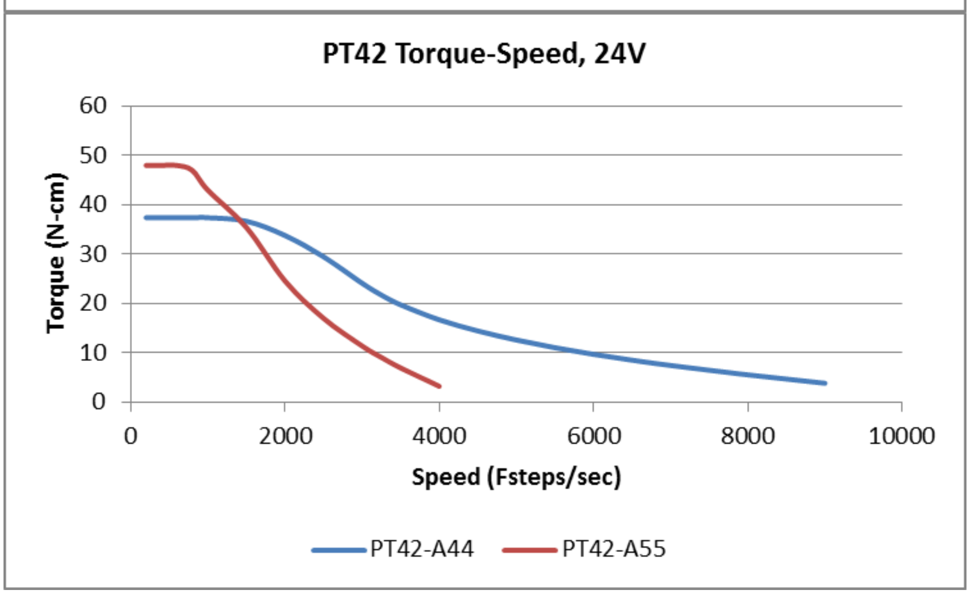

PS56-A106V48 Force-Speed, 24V
200.0
180.0
160.0
140.0
120.0
100.0
80.0
60.0
40.0
20.0
0.0
Force (lb)
0.000
0.500
1.000
1.500
2.000
Linear Speed (in/s)
Q Thread
S Thread

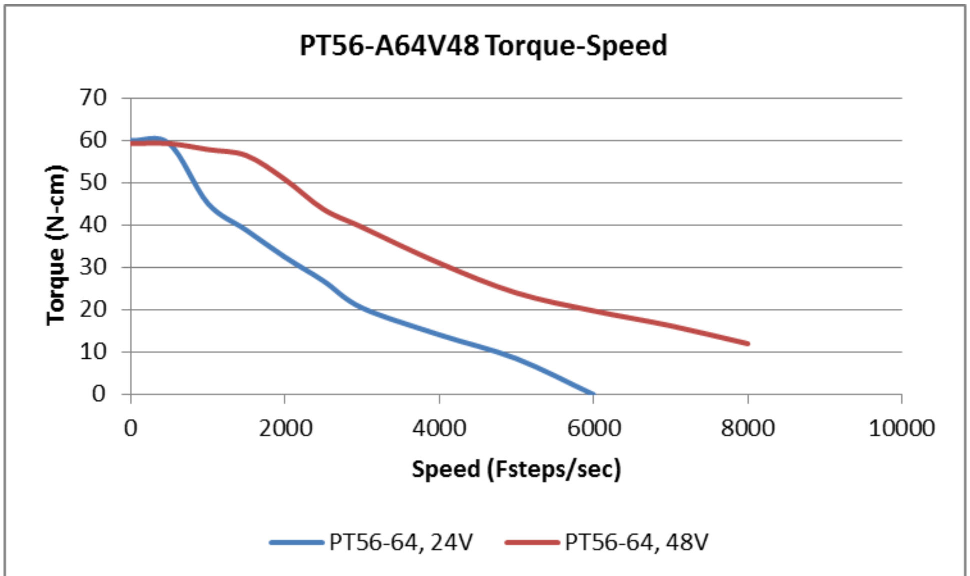
9.3. Rotary Stepper Models

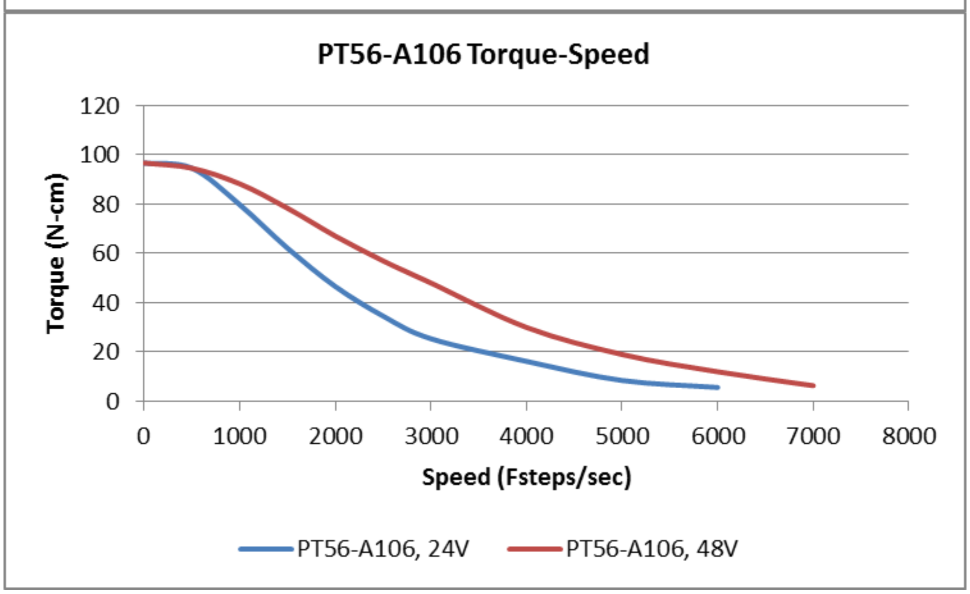


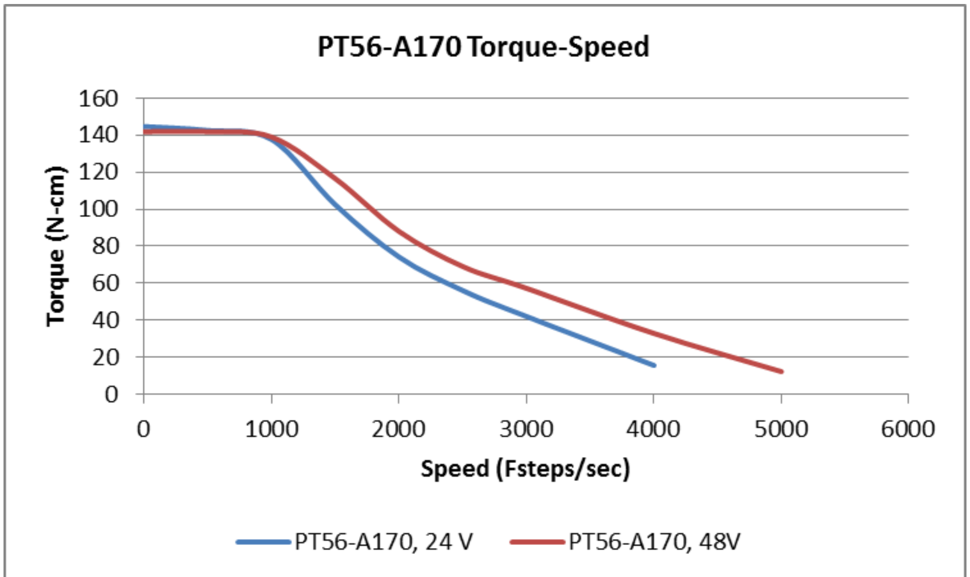


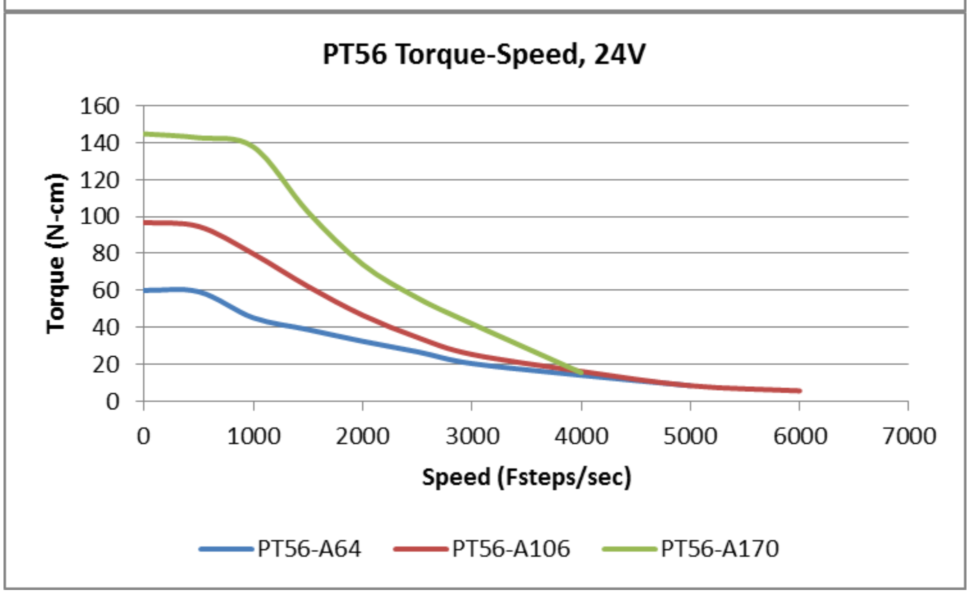

PT42 Torque-Speed, 48V
Torque (N-cm)
60
50
40
30
20
10
0
0
2000
4000
6000
8000
10000
Speed (Fsteps/sec)
PT42-A44
PT42-A55

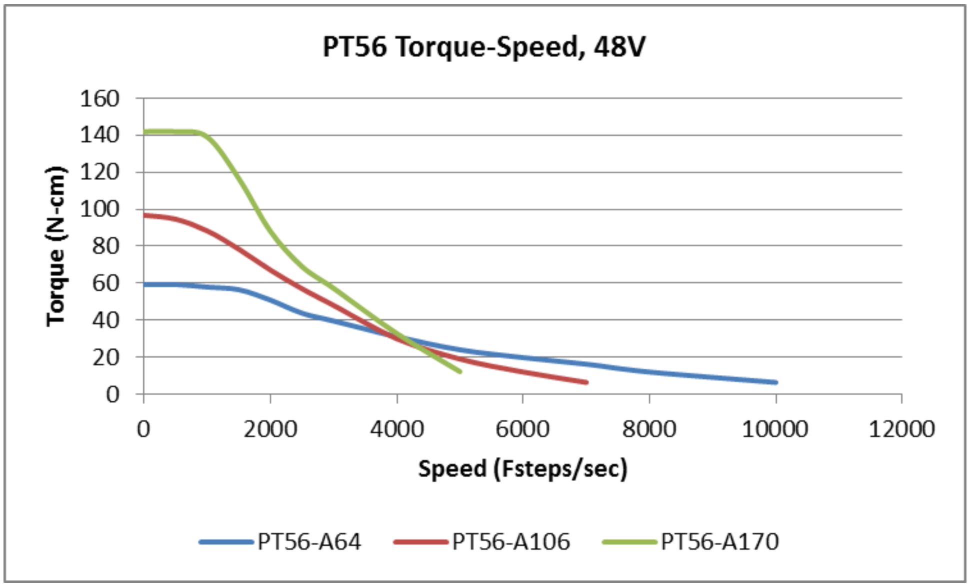

PT42 Torque-Speed, 24V
Torque (N-cm)
60
50
40
30
20
10
0
0
2000
4000
6000
8000
10000
Speed (Fsteps/sec)
PT42-A44
PT42-A55


PT56-A64V48 Torque-Speed
70
60
50
40
30
20
10
0
Torque (N-cm)
0
2000
4000
6000
8000
10000
Speed (Fsteps/sec)
PT56-64, 24V
PT56-64, 48V


PT56-A106 Torque-Speed
120
100
80
60
40
20
0
Torque (N-cm)
0
1000
2000
3000
4000
5000
6000
7000
8000
Speed (Fsteps/sec)
PT56-A106, 24V
PT56-A106, 48V


PT56-A170 Torque-Speed
Torque (N-cm)
160
140
120
100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
Speed (Fsteps/sec)
PT56-A170, 24 V
PT56-A170, 48V


PT56 Torque-Speed, 24V
Torque (N-cm)
160
140
120
100
80
60
40
20
0
0
1000
2000
3000
4000
5000
6000
7000
Speed (Fsteps/sec)
PT56-A64
PT56-A106
PT56-A170


PT56 Torque-Speed, 48V
160
140
120
100
80
60
40
20
0
Torque (N-cm)
0
2000
4000
6000
8000
10000
12000
Speed (Fsteps/sec)
PT56-A64
PT56-A106
PT56-A170

This page is empty