PRO-A10V80x-SA-CAN
PRO-A20V80x-SA-CAN

Programmable Servo Drive for Step, DC, Brushless DC and AC Motors

ELECTROCRAFT PRO SERIES

Programmable Servo Drive

ElectroCraft Document Number
A11297 Rev 2

Technical Reference

ELECTROCRAFT

PRO-A10V80x-SA-CAN
PRO-A20V80x-SA-CAN

Technical Reference

ElectroCraft Document Number
A11297 Revision 2

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

www.electrocraft.com

Read This First

About This Manual

This book is a technical reference manual for the **PRO-Ax0V80x-SA-CAN** family of intelligent servo drives, including the following products:
PRO-A10V80x-SA-CAN
PRO-A20V80x-SA-CAN

In order to operate the **PRO-Ax0V80** drives, you need to pass through 3 steps:

- **Step 1 Hardware installation**
- **Step 2 Drive setup** using ElectroCraft **PRO Config** software for drive commissioning
- **Step 3 Motion programming** using one of the options:
  - A **CANopen master**[1]
  - The drive's **built-in motion controller** executing an Electrocraft Motion Programming Language (**MPL**) program developed using Electrocraft's **MotionPRO Developer** 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 drives in MPL

This manual covers **Step 1** in detail. It describes the **PRO-Ax0V80x-SA-CAN** 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 drive internal units. For detailed information regarding the next steps, refer to the related documentation.

[1] when PRO-Ax0V80 is set in CANopen mode

Notational Conventions

This document uses the following conventions:

- **PRO-Ax0V80** – PRO-A10V80x-SA-CAN or PRO-A20V80x-SA-CAN
- **IU units** – Internal units of the drive
- **SI units** – International standard units (meter for length, seconds for time, etc.)
- **MPL** – Electrocraft's Motion Programming Language
- **MPLCAN** – Electrocraft protocol for exchanging MPL commands via CAN-bus

Related Documentation

Help Screens within the PRO Config software – describes how to use **PRO Config** to quickly setup any ElectroCraft PRO Series drive for your application using only 2 dialogues. The output of PRO Config is a set of setup data that can be downloaded into the drive EEPROM or saved on a PC file. At power-on, the drive is initialized with the setup data read from its EEPROM. With PRO Config it is also possible to retrieve the complete setup information from a drive previously programmed. PRO Config is part of the ElectroCraft Motion PRO Suite. Motion PRO Suite is available as part of a PRO Series Drive 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 drives 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 includes **PRO Config** for the drive/motor 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 Developer you can fully benefit from a key advantage of ElectroCraft drives – their capability to execute complex motions without requiring an external motion controller, thanks to their built-in motion controller.* Motion PRO Suite is available as part of a PRO Series Drive Evaluation Kit. Please contact ElectroCraft or your local ElectroCraft sales representative for more information on obtaining MotionPRO Suite or an evaluation kit.

PRO Series and 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 drives using MPL_LIB v2.0, 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 LabVIEW v2.0 Compatibility **(Document No. A11231)** – explains how to program in **LabVIEW** a motion application for the ElectroCraft programmable drives 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 drives using MPL_LIB_S7 motion control library. The MPL_LIB_S7 library is **IEC61131-3 compatible**.

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

PRO Series and X20 **(Document No. A11234)** – explains how 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: http://www.electrocraft.com/
Receive general information or assistance (see Note)	World Wide Web: http://www.electrocraft.com Email: drivesupport@electrocraft.com
Ask questions about product operation or report suspected problems (see Note)	Tel : +1 734.662-7771 Email: drivesupport@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

Contents

Read This First .... **III**

1. Safety information **13**

1.1. Warnings 13
1.2. Cautions 13

2. Product Overview **15**

2.1. Introduction 15
2.2. Key Features 17
2.3. Supported Motor-Sensor Configurations 19
2.4. PRO-Ax0V80x-SA-CAN Drive Dimensions 26
2.5. Identification Labels 26
2.6. Electrical Specifications 27
2.6.1. Operating Conditions 27
2.6.2. Storage Conditions 27
2.6.3. Mechanical Mounting 27
2.6.4. Environmental Characteristics 27
2.6.5. Logic Supply Input (+V_{LOG}) 28
2.6.6. Motor Supply Input (+V_{MOT}) 28
2.6.7. Enable/STO inputs 28
2.6.8. Motor Outputs (A/A+, B/A-, C/B+, BR/B-) 28
2.6.9. Digital Inputs (IN0, IN1, IN2/LSP, IN3/LSN) 29
2.6.10. Digital Outputs (OUT0, OUT1, OUT2/Error, OUT3/Ready) 31
2.6.11. Digital Hall Inputs (Hall1, Hall2, Hall3) 31
2.6.12. Encoder Inputs (A1+, A1-, B1+, B1-, Z1+, Z1-, A2+, A2-, B2+, B2-, Z2+, Z2-) ... 32
2.6.13. Sin-Cos Encoder Inputs (Sin+, Sin-, Cos+, Cos-)[1] 32
2.6.14. Analog 0…5V Inputs (REF, FDBK) 32
2.6.15. Analog ±10V Input (Ref) 33
2.6.16. RS-232 33
2.6.17. CAN-Bus 33
2.6.18. Supply Output (+5V) 34
2.7. EMC Compliance 34

3. Step 1. Hardware Installation **35**

3.1. Mechanical Mounting 35
3.2. Mating Connectors 35
3.3. Connectors and Connection Diagrams 36
3.3.1. Connector Layout 36
3.3.2. J1 Power supply input connector pinout 36
3.3.3. J2 Motor output 37
3.3.4. J3 Primary feedback connector pinout 37
3.3.5. J4 Secondary feedback connector pinout 37
3.3.6. J5 Digital, analog I/O and logic supply connector pinout 37
3.3.7. J6, J7 CAN-Bus connectors 38
3.3.8. J8 Enable/STO connector pinout (Labelled STO) 38
3.3.9. J9 RS232 connector pinout 38
3.3.10. AxisID DIP switch settings 39
3.3.11. 24V Digital I/O Connection 40
3.3.12. 5V Digital I/O Connection 42
3.3.13. Analog Inputs Connection 43
3.3.14. Motor connections 45
3.3.15. Feedback connections 48
3.3.16. Power Supply Connection 53
3.3.17. Serial RS-232 connection 56
3.3.18. CAN-bus connection 57
3.3.19. Disabling Autorun Mode 59
3.4. Selection of the Operation mode and Axis ID, DIP-Switch Settings 60
3.5. LED Indicator 60

4. Step 2. Drive Setup 61

4.1. Installing PRO Config 61
4.2. Getting Started with PRO Config 61
4.2.1. Establish communication 62
4.2.2. Setup drive/motor 62
4.2.3. Selecting NPN/PNP inputs type in Setup 65
4.2.4. Download setup data to drive/motor 65
4.2.5. Evaluate drive/motor behavior (optional) 66
4.3. Changing the drive Axis ID 66
4.4. Setting CANbus rate 67
4.5. Creating an Image File with the Setup Data 68

5. **Step 3. Motion Programming 69**
5.1. Using a CANopen Master (for PRO-Ax0V80 CANopen execution) 69
5.1.1. CiA-301 Application Layer and Communication Profile Overview 69
5.1.2. CiA-305 Layer Setting Services (LSS) and Protocols Overview 70
5.1.3. CiA-402 and Manufacturer Specific Device Profile Overview 70
5.1.4. ElectroCAN Extension 70
5.1.5. Checking Setup Data Consistency 70
5.2. Using the built-in Motion Controller and MPL 70
5.2.1. Electrocraft Motion Programming Language Overview 70
5.2.2. Installing MotionPRO Developer 71
5.2.3. Getting Started with MotionPRO Developer 72
5.2.4. Creating an Image File with the Setup Data and the MPL Program 77
5.3. Combining CANopen /or other host with MPL 77
5.3.1. Using MPL Functions to Split Motion between Master and Drives 77
5.3.2. Executing MPL programs 78
5.3.3. Loading Automatically Cam Tables Defined in MotionPRO Developer 78
5.3.4. Customizing the Homing Procedures 78
5.3.5. Customizing the Drive Reaction to Fault Conditions 78
5.4. Using Motion Libraries for PC-based Systems 79
5.5. Using Motion Libraries for PLC-based Systems 79

6. **Scaling factors 80**
6.1. Position units 80
6.1.1. Brushless / DC brushed motor with quadrature encoder on motor 80
6.1.2. Brushless motor with linear Hall signals 80
6.1.3. DC brushed motor with quadrature encoder on load and tacho on motor 80
6.1.4. Step motor open-loop control. No feedback device 81
6.1.5. Step motor open-loop control. Incremental encoder on load 81
6.1.6. Brushless motor with sine/cosine encoder on motor 81
6.2. Speed units 82
6.2.1. Brushless / DC brushed motor with quadrature encoder on motor 82
6.2.2. Brushless motor with linear Hall signals 82
6.2.3. DC brushed motor with quadrature encoder on load and tacho on motor 83
6.2.4. DC brushed motor with tacho on motor 83
6.2.5. Step motor open-loop control. No feedback device 83
6.2.6. Step motor closed-loop control. Incremental encoder on motor 84
6.2.7. Brushless motor with sine/cosine encoder on motor 84

6.3. Acceleration units 85
6.3.1. Brushless / DC brushed motor with quadrature encoder on motor 85
6.3.2. Brushless motor with linear Hall signals 85
6.3.3. DC brushed motor with quadrature encoder on load and tacho on motor 86
6.3.4. Step motor open-loop control. No feedback device 86
6.3.5. Step motor open-loop control. Incremental encoder on load 86
6.3.6. Step motor closed-loop control. Incremental encoder on motor 87
6.3.7. Brushless motor with sine/cosine encoder on motor 87
6.4. Jerk units 88
6.4.1. Brushless / DC brushed motor with quadrature encoder on motor 88
6.4.2. Brushless motor with linear Hall signals 88
6.4.3. DC brushed motor with quadrature encoder on load and tacho on motor 88
6.4.4. Step motor open-loop control. No feedback device 89
6.4.5. Step motor open-loop control. Incremental encoder on load 89
6.4.6. Step motor closed-loop control. Incremental encoder on motor 89
6.4.7. Brushless motor with sine/cosine encoder on motor 90
6.5. Current units 90
6.6. Voltage command units 90
6.7. Voltage measurement units 91
6.8. Time units 91
6.9. Master position units 91
6.10. Master speed units 91
6.11. Motor position units 91
6.11.1. Brushless / DC brushed motor with quadrature encoder on motor 91
6.11.2. Brushless motor with linear Hall signals 92
6.11.3. DC brushed motor with quadrature encoder on load and tacho on motor 92
6.11.4. Step motor open-loop control. No feedback device 92
6.11.5. Step motor open-loop control. Incremental encoder on load 92
6.11.6. Step motor closed-loop control. Incremental encoder on motor 92
6.11.7. Brushless motor with sine/cosine encoder on motor 93
6.12. Motor speed units 93
6.12.1. Brushless / DC brushed motor with quadrature encoder on motor 93
6.12.2. Brushless motor with linear Hall signals 93
6.12.3. DC brushed motor with quadrature encoder on load and tacho on motor 94
6.12.4. DC brushed motor with tacho on motor 94
6.12.5. Step motor open-loop control. No feedback device or incremental encoder on load 94

6.12.6. Step motor closed-loop control. Incremental encoder on motor 95
6.12.7. Brushless motor with sine/cosine encoder on motor 95

7. Memory Map 96

This page is empty.

1. Safety information

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

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

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

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 DRIVE, 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 CABLES INTO THE PRO-Ax0V80x-SA-CAN WHILE THE POWER SUPPLIES ARE ON!***

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

WARNING!	**DURING DRIVE 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 DRIVE 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-Ax0V80x-SA-CAN** is part of a family of fully digital intelligent servo drives, based on the latest DSP technology and they offer unprecedented drive performance combined with an embedded motion controller.

Suitable for control of brushless DC, brushless AC (vector control), DC brushed motors and step motors, the PRO-Ax0V80x-SA-CAN drives accept as position feedback incremental encoders (quadrature or sine/cosine) and linear Halls signals.

All drives perform position, speed or torque control and work in single, multi-axis or stand-alone configurations. Thanks to the embedded motion controller, the PRO-Ax0V80x-SA-CAN drives combine controller, drive and PLC functionality in a single compact unit and are capable to execute complex motions 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 or camming, 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 one 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 drive level you can really distribute the intelligence between the master and the drives 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 drives 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 at: calling MPL functions stored in the drive EEPROM and waiting for a message, which confirms the MPL functions execution completion.

All PRO-Ax0V80 drives are equipped with a serial RS232 and a CAN 2.0B interface and can be set (hardware, via a switch) to operate in 2 modes:

- ❑ **CANopen**
- ❑ **MPLCAN**

When **CANopen** mode is selected, the PRO-Ax0V80 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 standards. In this mode, the PRO-Ax0V80 may be controlled via a CANopen master. As a bonus, PRO-Ax0V80 offers a CANopen master the option to call motion sequences, written in MPL and stored in the drive EEPROM, using manufacturer specific objects (see for details par. 5.3).

When **MPLCAN** mode is selected, the PRO-Ax0V80 behaves as standard ElectroCraft intelligent drive and conforms to ElectroCraft protocol for exchanging MPL commands via CAN-bus. When MPLCAN protocol is used, it is not mandatory to have a master. Any PRO-Ax0V80 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 drives.

When higher level coordination is needed, apart from a CANopen master, the PRO-Ax0V80 drives can also be controlled via a PC or a PLC using one of the **MPL_LIB** motion libraries.

For PRO-Ax0V80 commissioning **PRO Config** or **MotionPRO Developer** PC applications may be used.

PRO Config is a subset of MotionPRO Developer, including only the drive setup part. The output of PRO Config is a set of setup data that can be downloaded into the drive EEPROM or saved on a PC file. At power-on, the drive is initialized with the setup data read from its EEPROM. With PRO Config it is also possible to retrieve the complete setup information from a drive previously programmed. PRO Config shall be used for drive setup in all cases where the motion commands are sent exclusively from a master. Hence neither the PRO-Ax0V80 MPL programming capability nor the drive camming mode are used.

MotionPRO Developer platform includes PRO Config for the drive 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 Developer you can fully benefit from a key advantage of ElectroCraft drives – their capability to execute complex motions 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-Ax0V80 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 drive 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 drive, wherever needed.

2.2. Key Features

- Fully digital servo drive suitable for the control of rotary or linear brushless, DC brush, and step motors with CANopen interface and built-in motion controller with high-level MPL motion language
- Very compact design
- Sinusoidal (FOC) or trapezoidal (Hall-based) control of brushless motors
- Open or closed-loop control of 2 and 3-phase steppers
- Enable/STO function. Two Enable/STO inputs are provided, which when left not connected will disable the motor outputs. This provides a dual redundant hardware protection that cannot be overdriven by the software or other hardware components.
- 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
  - Dual RJ45 CAN-Bus connectors up to 1Mbit/s
- Digital and analog I/Os:
  - 4 digital inputs: 5-36 V, programmable polarity: sourcing/NPN or sinking/PNP: 2 Limit switches and 2 general-purpose
  - 4 digital outputs: 5-36 V, OUT0 with 2A and others with 0.5 A, sinking/NPN open-collector (Ready, Error and 2 general-purpose)
  - 2 analog inputs: 12 bit, +/-10V and 0-5V: Reference and Feedback or general purpose
- Feedback devices (dual-loop support)

  1st feedback devices supported:
  - Incremental encoder interface (differential)
  - Analog sin/cos encoder interface (differential $1V_{PP}$)
  - Digital Hall sensor interface (single-ended and open collector)

  2nd feedback devices supported:
  - Incremental encoder interface (differential)
  - Pulse & direction interface (differential) for external (master) digital reference
- 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
  - 35 Homing modes
- 127 h/w addresses selectable by DIN switch
- MPLCAN and CANopen (CiA 301 v4.2, CiA 305 v.2.2.13 and CiA 402 v3.0) protocols selectable by DIN switch
- 16K × 16 internal SRAM memory for data acquisition
- 16K × 16 E^2ROM to store MPL motion programs, cam tables and other user data

- PWM switching frequency up to 125kHz
- Motor supply: 12-75V
- Logic supply: 9-36V.Separate supply is optional
- Output current: PRO-A10V80x-SA-CAN: 10A[1] continuous; 20A peak
  PRO-A20V80x-SA-CAN: 20A[2] continuous; 40A peak
- Operating ambient temperature: 0-40°C (over 40°C with derating)
- Protections:
    - Short-circuit between motor phases
    - Short-circuit from motor phases to ground
    - Over-voltage
    - Under-voltage
    - Over-current
    - Over-temperature
    - Communication error
    - Control error

[1] 10A cont. with DC, step and BLDC motors (trapezoidal), 10A amplitude (7.1A_{RMS}) for PMSM (sinusoidal)
[2] 20A cont. with DC, step and BLDC motors (trapezoidal), 20A amplitude (14.2A_{RMS}) for PMSM (sinusoidal)

2.3. Supported Motor-Sensor Configurations

PRO-Ax0V80x-SA-CAN supports the following configurations:

1. Position, speed or torque control of a **brushless AC rotary motor** with an **incremental quadrature encoder** on its shaft. The brushless motor is vector controlled like a permanent magnet synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

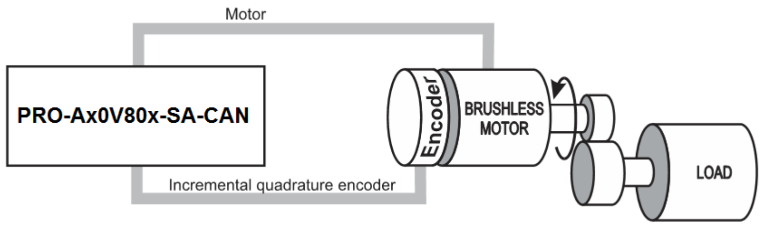


Figure 2.1. *Brushless AC rotary motor. Position / speed / torque control. Quadrature encoder on motor.*

2. Position, speed or torque control of a **brushless AC rotary motor** with an **incremental sine/cosine encoder** on its shaft. The brushless motor is vector controlled like a permanent magnet synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

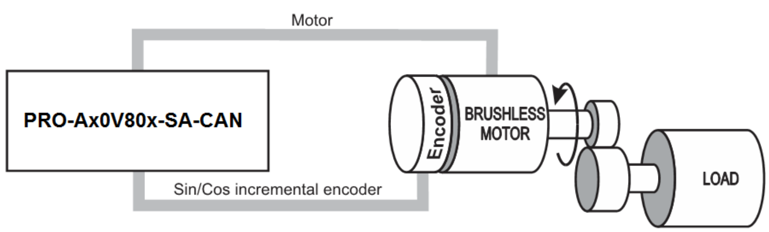


Figure 2.2. *Brushless AC rotary motor. Position/speed/torque control. Sine/cosine incremental encoder on motor*

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

3. Position, speed or torque control of a **brushless DC rotary motor** with **digital Hall sensors** and an **incremental quadrature encoder** on its shaft. The brushless motor is controlled using Hall sensors for commutation. It works with rectangular currents and **trapezoidal** BEMF voltages. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

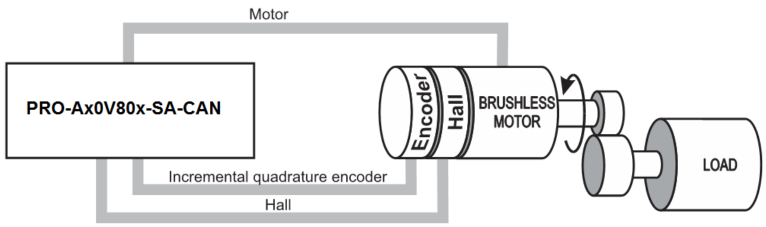


Figure 2.3. *Brushless DC rotary motor. Position / speed / torque control. Hall sensors and quadrature encoder on motor.*

4. Dual loop position and speed control of a **brushless DC rotary motor** with **digital Hall sensors,** an **incremental quadrature or Sin/Cos encoder** on one feedback and another incremental encoder on the second feedback. The speed loop is controlled with the encoder on the motor and the position loop is controlled with the encoder on the load.
Remark: the Sin/Cos encoder is present only on the Feedback #1 interface.

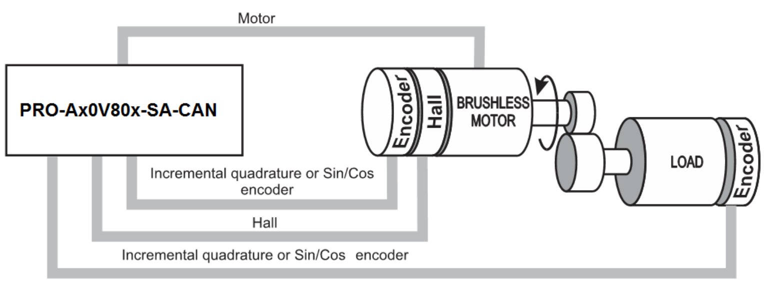


Figure 2.4. *Brushless DC rotary motor. Position and speed control. Hall sensors and encoders on motor and load.*

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

5. Position, speed or torque control of a **brushless AC linear motor** with an **incremental sine/cosine encoder**. The brushless motor is vector controlled like a permanent magnet synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

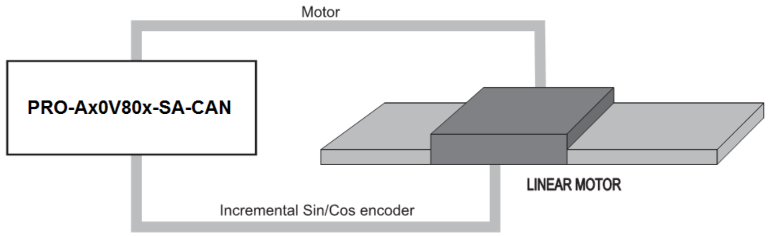


Figure 2.5. *Brushless AC linear motor. Position/speed/torque control. Sine/cosine incremental encoder on motor*

6. Position, speed or torque control of a **brushless AC linear motor** with an **incremental quadrature linear encoder** on the track. The brushless motor is vector controlled like a permanent magnet synchronous motor. It works with **sinusoidal** voltages and currents. Scaling factors take into account the transmission ratio between motor and load (linear or rotary). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

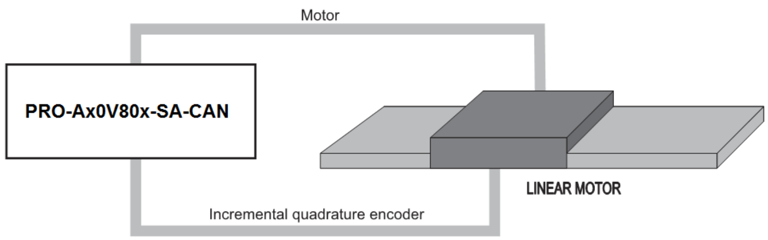


Figure 2.6. *Brushless AC rotary motor with incremental quadrature linear encoder. Position / speed / torque control.*

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

7. Dual loop position and speed control of a **brushless DC linear motor** with **digital Hall sensors,** an **incremental quadrature or Sin/Cos encoder** on one feedback and another incremental encoder on the second feedback. The speed loop is controlled with the encoder on the motor and the position loop is controlled with the encoder on the load.
   Remark: the Sin/Cos encoder is present only on the Feedback #1 interface.

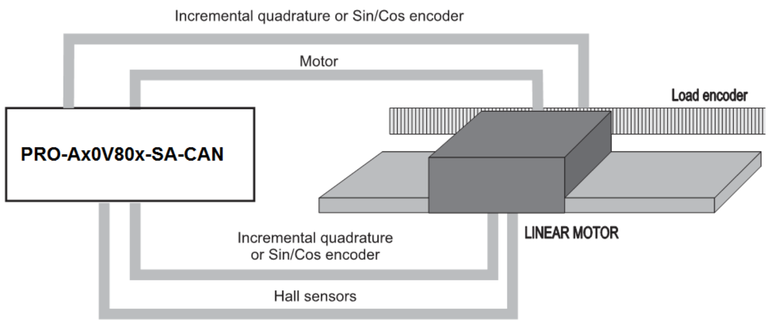


Figure 2.7. *Brushless DC rotary motor. Position and speed control.*
Hall sensors and encoders on motor and load.

8. Position, speed or torque control of a **DC brushed rotary motor** with an **incremental quadrature encoder** on its shaft. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

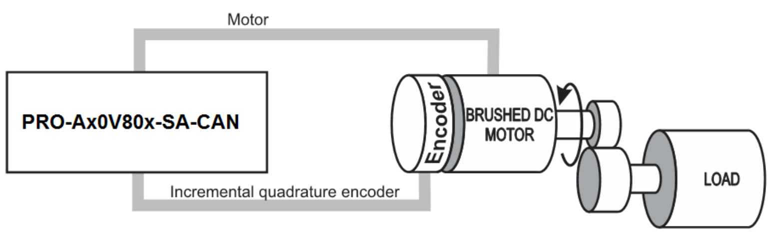


Figure 2.8. *DC brushed rotary motor. Position / speed / torque control.*
Quadrature encoder on motor.

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

9. Dual loop, position and speed control of a **brushed DC rotary motor** with an **incremental quadrature** on the motor and an incremental encoder on the load. The speed loop is controlled with the encoder on the motor and the position loop is controlled with the encoder on the load.

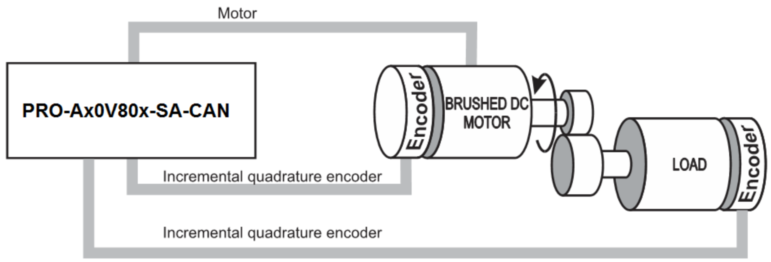


Figure 2.9. *DC brushed rotary motor. Position and speed control. Quadrature encoders on motor and load.*

10. Load position control using an **incremental quadrature encoder** on load, combined with speed control of a **DC brushed rotary motor** having a **tachometer** on its shaft. The motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

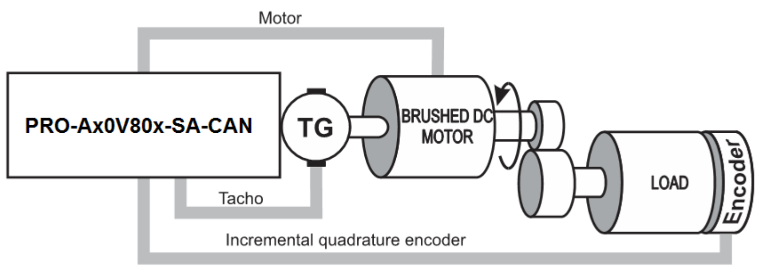


Figure 2.10. *DC brushed rotary motor. Position / speed / torque control. Quadrature encoder on load plus tachometer on motor.*

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

11. Speed or torque control of a **DC brushed rotary motor** with a **tachometer** on its shaft. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for speed and acceleration) expressed in SI units (or derivatives) refer to the load[1], while the same commands, expressed in IU units, refer to the motor.

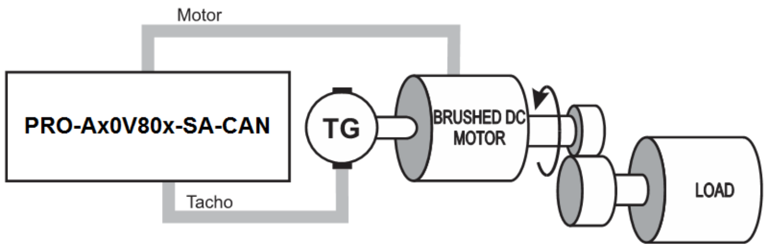


Figure 2.11. *DC brushed rotary motor. Speed/torque control.*
Tachometer on motor.

12. Open-loop control of a 2 or 3-phase **step motor** in position or speed. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

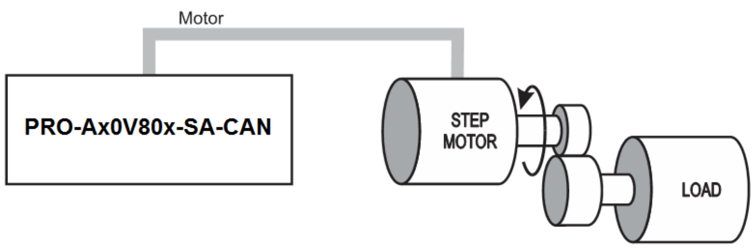


Figure 2.12. *No position or speed feedback.*
Open-loop control: motor position or speed.

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

13. Closed-loop control of **load position using an encoder on load**, combined with open-loop control of **a 2 or 3-phase step motor** in speed, with speed reference provided by the position controller. The motion commands in both SI and IU units refer to the load.

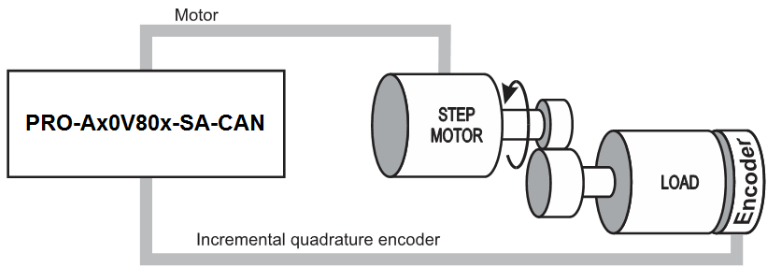


Figure 2.13. *Encoder on load.*
Closed-loop control: load position, open-loop control: motor speed.

14. Closed-loop control of **a 2-phase step motor** in position, speed or torque. Scaling factors take into account the transmission ratio between motor and load (rotary or linear). Therefore, the motion commands (for position, speed and acceleration) expressed in SI units (or derivatives) always refer to the load[1].

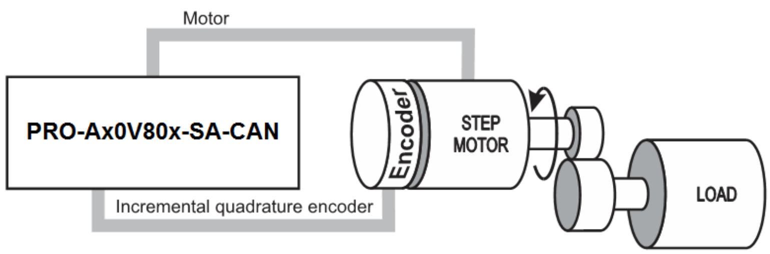


Figure 2.14. *Encoder on motor shaft.*
Closed-loop control: motor position, speed or torque.

[1] Motion commands can be referred to the motor by setting in PRO Config a rotary to rotary transmission with ratio 1:1

2.4. PRO-Ax0V80x-SA-CAN Drive Dimensions

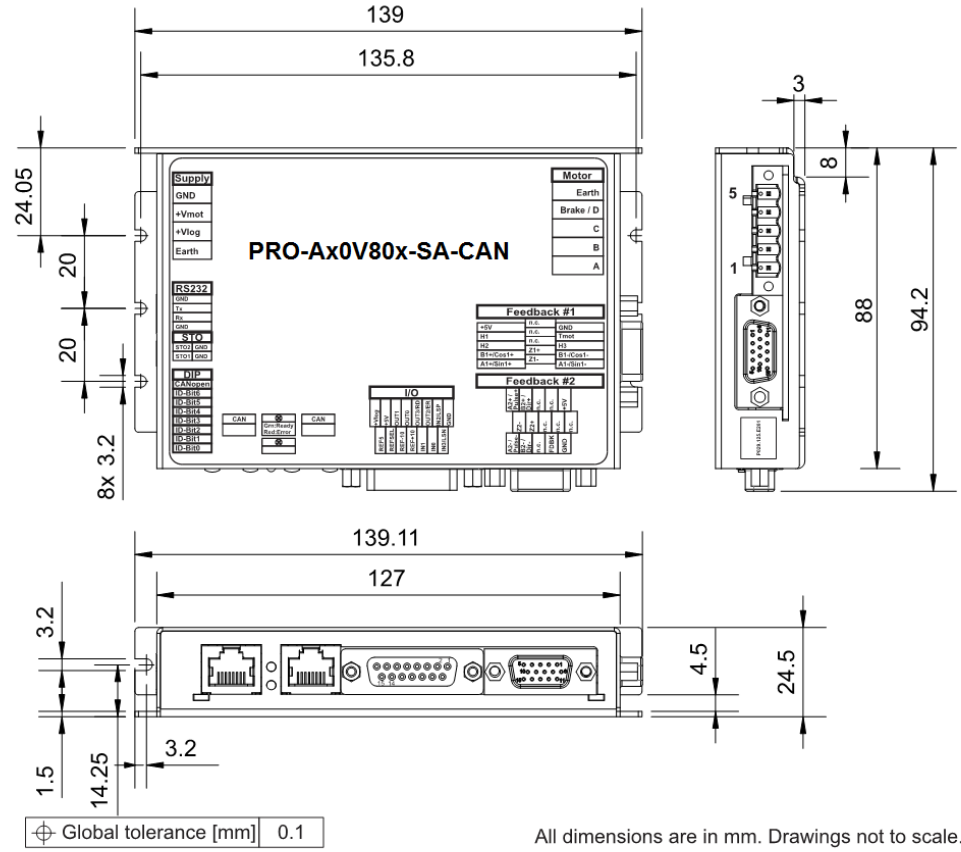


Figure 2.15. *PRO-Ax0V80x-SA-CAN drive dimensions*

2.5. Identification Labels

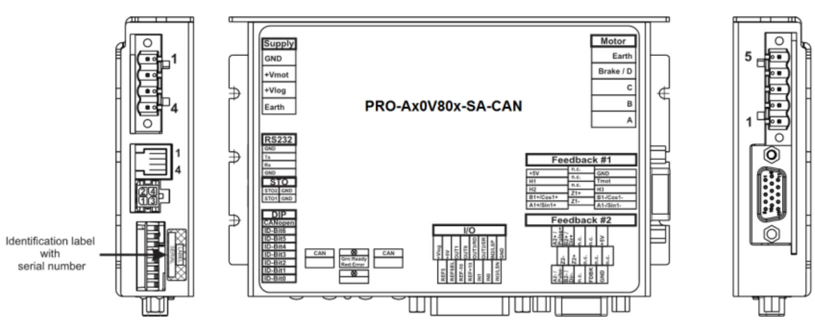


Figure 2.16. *PRO-Ax0V80x-SA-CAN Identification Labels*

2.6. Electrical Specifications

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

T_{amb} = 0…40°C, V_{LOG} = 24 V_{DC}; V_{MOT} = 75V_{DC}; Supplies start-up / shutdown sequence: -*any*-
Load current (sinusoidal amplitude / continuous BLDC, DC, stepper) = 20A PRO-A20V80; 10A PRO-A10V80

2.6.1. Operating Conditions

		Min.	Typ.	Max.	Units
Ambient temperature		0		+40	ºC
Ambient humidity	Non-condensing	0		90	%Rh
Altitude / pressure[1]	Altitude (referenced to sea level)	-0.1	0 ÷ 2.5	2	Km
	Ambient Pressure	0 [2]	0.75 ÷ 1	10.0	atm

2.6.2. Storage Conditions

		Min.	Typ.	Max.	Units
Ambient temperature		-40		+85	ºC
Ambient humidity	Non-condensing	0		100	%Rh
Ambient Pressure		0		10.0	atm

2.6.3. Mechanical Mounting

Airflow		natural convection, closed box

2.6.4. Environmental Characteristics

		Min.	Typ.	Max.	Units
Size (Length x Width x Height)	Without mating connector	139 x 94.2 x 24.5			mm
		~5.47 x 3.7 x 0.97			inch
Weight	Without mating connectors	240			g
Power dissipation	Idle (no load)		3.6		W
	Operating		11		
Efficiency			98		%
Cleaning agents	Dry cleaning is recommended	Only Water- or Alcohol- based			
Protection degree	According to IEC60529, UL508	IP30			-

[1] PRO-Ax0V80 can be operated in vacuum (no altitude restriction), but at altitudes over 2,500m, current and power rating are reduced due to thermal dissipation efficiency.

2.6.5. 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		8		40	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms) †		-1		+45	V
Supply current	No Load on Digital Outputs	+V_{LOG} = 9V		300		mA
		+V_{LOG} = 12V		250		
		+V_{LOG} = 24V		150		
		+V_{LOG} = 40V		100		

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

			Min.	Typ.	Max.	Units
Supply voltage	Nominal values		12	80	90	V_{DC}
	Absolute maximum values, drive operating but outside guaranteed parameters		11		94	V_{DC}
	Absolute maximum values, surge (duration ≤ 10ms) †		-1		95	V
Supply current	Idle			1	5	mA
	Operating	PRO-A10V80	-20	±10	+20	A
		PRO-A20V80	-40	±20	+40	
	Absolute maximum value, short-circuit condition (duration ≤ 10ms) †	PRO-A10V80			22.5	A
		PRO-A20V80			45	

2.6.7. Enable/STO inputs

		Min.	Typ.	Max.	Units
Mode compliance		PNP			
Default state	Input floating (wiring disconnected)	Logic Low			
Input voltage	Logic "LOW"	-10		2.2	V
	Logic "HIGH"	6.3		36	
	Absolute maximum, continuous	-10		+39	
Input current	Logic "LOW", pulled to GND		0		mA
	Logic "HIGH", pulled to +Vlog			0.4	
Pulse duration	Ignored low-high-low			tbd	ms
	Ignored high-low-high			tbd	
	Accepted pulse	tbd			
ESD protection	Human body model	±2			KV

2.6.8. Motor Outputs (A/A+, B/A-, C/B+, BR/B-)

	Min.	Typ.	Max.	Units

Nominal output current, continuous	for DC brushed, steppers and BLDC motors with Hall-based trapezoidal control	PRO-A10V80			10	A
		PRO-A20V80			20	
	for PMSM motors with FOC sinusoidal control (sinusoidal amplitude value)	PRO-A10V80			10	
		PRO-A20V80			20	
	for PMSM motors with FOC sinusoidal control (sinusoidal effective value)	PRO-A10V80			7.1	
		PRO-A20V80			14.2	
Motor output current, peak	maximum 10s		-20		+20	A
	maximum 3.6s		-40		+40	
Short-circuit protection threshold	measurement range	PRO-A10V80			±22.5	A
		PRO-A20V80			±45	
Short-circuit protection delay			5	10		μS
On-state voltage drop	for nominal output current; including typical mating connector contact resistance			±0.3	±0.5	V
Off-state leakage current				±0.5	±1	mA
Motor inductance (phase-to-phase)	Recommended value, for ripple ±5% of measurement range; $+V_{MOT}$ = 80 V	F_{PWM} = 20 kHz	330			μH
		F_{PWM} = 40 kHz	150			
		F_{PWM} = 60 kHz	120			
		F_{PWM} = 80 kHz	80			
		F_{PWM} = 100 kHz	60			
	Absolute minimum value, limited by short-circuit protection; $+V_{MOT}$ = 80 V	F_{PWM} = 20 kHz	120			μH
		F_{PWM} = 40 kHz	40			
		F_{PWM} = 60 kHz	30			
		F_{PWM} = 80 kHz	15			
		F_{PWM} = 100 kHz	8			
Motor electrical time-constant (L/R)	Recommended value, for ±5% current measurement error due to ripple	F_{PWM} = 20 kHz	250			µs
		F_{PWM} = 40 kHz	125			
		F_{PWM} = 60 kHz	100			
		F_{PWM} = 80 kHz	63			
		F_{PWM} = 100 kHz	50			
Current measurement accuracy	FS = Full Scale			±5	±8	%FS

2.6.9. Digital Inputs (IN0, IN1, IN2/LSP, IN3/LSN)[1]

		Min.	Typ.	Max.	Units
Mode compliance		PNP			
Default state	Input floating (wiring disconnected)	Logic LOW			
Input voltage	Logic "LOW"	-10	0	2.2	V
	Logic "HIGH"	6.3		36	
	Floating voltage (not connected)		0		

[1] The digital inputs are software selectable as PNP or NPN

	Absolute maximum, continuous	-10		+39	
	Absolute maximum, surge (duration ≤ 1S) †	-20		+40	
Input current	Logic "LOW"; Pulled to GND		0		mA
	Logic "HIGH"		1.3	2	

		Min.	Typ.	Max.	Units
Mode compliance		NPN			
Default state	Input floating (wiring disconnected)	Logic HIGH			
Input voltage	Logic "LOW"	-10		2.2	V
	Logic "HIGH"	6.3		36	
	Floating voltage (not connected)		3		
	Absolute maximum, continuous	-10		+36	
	Absolute maximum, surge (duration ≤ 1S) †	-20		+40	
Input current	Logic "LOW"; Pulled to GND	-1.6	0.6	1	mA
	Logic "HIGH"; Pulled to +24V	0	0	0.3	

Input frequency		0		150	KHz
Minimum pulse width		3.3			µS
ESD protection	Human body model	±2			KV

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

Parameter	Condition		Min.	Typ.	Max.	Units
Mode compliance	All outputs (OUT0, OUT1, OUT2/Error, OUT3/Ready)		NPN 24V			
Default state	Not supplied (+V_{LOG} floating or to GND)		High-Z (floating)			
	Immediately after power-up	OUT0, OUT1	Logic "HIGH"			
		OUT2/Error, OUT3/Ready	Logic "LOW"			
	Normal operation	OUT0, OUT1, OUT2/Error	Logic "HIGH"			
		OUT3/Ready	Logic "LOW"			
Output voltage	Logic "LOW"; output current = 0.5A				0.8	V
	Logic "HIGH"; output current = 0, no load	OUT2/Error, OUT3/ Ready	2.9	3	3.3	
		OUT0, OUT1	4	4.5	5	
	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	OUT0			2	A
		OUT1, OUT2, OUT3			0,5	
	Logic "LOW", sink current, pulse ≤ 5 sec	OUT0			4	A
		OUT1, OUT2, OUT3			1	
	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			2			µS
ESD protection	Human body model		±15			KV

2.6.11. Digital Hall Inputs (Hall1, Hall2, Hall3)

Parameter	Condition	Min.	Typ.	Max.	Units
Mode compliance		TTL / CMOS / Open-collector			
Default state	Input floating (wiring disconnected)	Logic HIGH			
Input voltage	Logic "LOW"		0	0.8	V
	Logic "HIGH"	1.8			
	Floating voltage (not connected)		4.5		
	Absolute maximum, surge (duration ≤ 1S) †	-10		+15	
Input current	Logic "LOW"; Pull to GND		3	5	mA
	Logic "HIGH"; Internal 1KΩ pull-up to +5	0	0	0	

Minimum pulse width		2			µS
ESD protection	Human body model	±5			KV

2.6.12. Encoder Inputs (A1+, A1-, B1+, B1-, Z1+, Z1-, A2+, A2-, B2+, B2-, Z2+, Z2-)[1]

		Min.	Typ.	Max.	Units
Differential mode compliance		TIA/EIA-422-A			
Input voltage, differential mode	Hysteresis	±0.06	±0.1	±0.2	V
	Differential mode	-14		+14	
	Common-mode range (A+ to GND, etc.)	-11		+14	
Input impedance, differential	A1+, A2+, B1+, B2+, Z1+, Z2+		2.2		KΩ
	A1-, A2-, B1-, B2-, Z1-, Z2-		1.6		
	Differential mode	0		10	MHz
	Differential mode	50			nS
ESD protection	Human body model	±1			KV

2.6.13. Sin-Cos Encoder Inputs (Sin+, Sin-, Cos+, Cos-)[1]

		Min.	Typ.	Max.	Units
Input voltage, differential	Sin+ to Sin-, Cos+ to Cos-		1	1.25	V_{PP}
Input voltage, any pin to GND	Operational range	-1	2.5	4	V
	Absolute maximum values, continuous	-7		+7	
	Absolute maximum, surge (duration ≤ 1S) †	-11		+14	
Input impedance	Differential, Sin+ to Sin-, Cos+ to Cos-	4.2	4.7		KΩ
	Common-mode, to GND		2.2		KΩ
Resolution with interpolation	Software selectable, for one sine/cosine period	2		10	bits
Frequency	Sin-Cos interpolation	0		450	KHz
	Quadrature, no interpolation	0		10	MHz
ESD protection	Human body model	±2			KV

2.6.14. Analog 0…5V Inputs (REF, FDBK)

		Min.	Typ.	Max.	Units
Input voltage	Operational range	0		4.95	V
	Absolute maximum values, continuous	-12		+18	

[1] All differential input pins have internal 120Ω termination resistors connected across

	Absolute maximum, surge (duration ≤ 1S) †			±36	
Input impedance	To GND		8		KΩ
Resolution			12		bits
Integral linearity				±2	bits
Offset error			±2	±10	bits
Gain error			±1%	±3%	% FS[1]
Bandwidth (-3dB)	Software selectable	0		1	KHz
ESD protection	Human body model	±2			KV

2.6.15. Analog ±10V Input (Ref)

		Min.	Typ.	Max.	Units
Differential voltage range			±10		V
Common-mode voltage range	Referenced to GND	-12	0..10	+50	V
Input impedance	Differential		40		KΩ
Common-mode impedance	Referenced to GND		20		KΩ
Resolution			12		bits
Integral linearity				0.036	%FS[1]
Offset error	Common-mode voltage = 0…10 V		±0.2	±0.5	%FS[1]
Gain error	Common-mode voltage = 0…10 V		±10	±12	%FS[1]
Bandwidth (-3dB)	Depending on software settings		1.5		kHz

2.6.16. 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

2.6.17. CAN-Bus

		Min.	Typ.	Max.	Units
Compliance		ISO11898, CiA-301v4.2, CiA 305 v2.2.13, 402v3.0			
Bit rate	Software selectable	125		1000	125
Bus length	1Mbps			25	m
	500Kbps			100	
	≤ 250Kbps			250	
Resistor	Between CAN-Hi, CAN-Lo	none on-board			
Node addressing	Hardware: by DIP switch	1 ÷ 127 & LSS non-configured (CANopen); 1-127 & 255 (MPLCAN)			
	Software	1 ÷ 127 (CANopen); 1- 255 (MPLCAN)			
Voltage, CAN-Hi or CAN-Lo to GND		-26		26	V
ESD protection	Human body model	±15			KV

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

2.6.18. Supply Output (+5V)

		Min.	Typ.	Max.	Units
+5V output voltage	Current sourced = 250mA	4.8	5	5.2	V
+5V output current		600	650		mA
Short-circuit protection		NOT protected			
Over-voltage protection		NOT protected			
ESD protection	Human body model	±2			KV

† Stresses beyond values listed under "absolute maximum ratings" may cause permanent damage to the device. Exposure to absolute-maximum-rated conditions for extended periods may affect device reliability.

2.7. EMC Compliance

This drive has been tested and found complaint with IEC61800-3 2004 for use in the second environment Category C3. This type of drive is not intended to be used on a low-voltage public network which supplies domestic premises. Radio frequency interference is expected if used on such a network.

The following installation guidelines must be followed for EMC compliance.

- Quality shielded multiconductor cable must be used for all connections to the drive. Connection details are specified in Section 3.3 Connectors and Connection Diagrams.
- For radiated emissions, install series inductors on motor signals A, B and C near drive. For a typical installation 15uH toroidal type inductors rated at 20A Multicomp model MCAP1130140114K-150MU or equivalent inductors are recommended. However, since the radiated emissions are strongly influenced by the system architecture, the installer should confirm the system performance.
- For surge immunity, install a capacitor between +Vlog and GND close to the drive. A 50V ceramic, X7S or better dielectric 100uF capacitor is required, TDK model CKG57NX7S1H226M500JH or equivalent is recommended.
- To avoid the risk of cross talk to signal cables, power interface cables should be segregated from signal cables. Specific recommendations are made in Section 3.3 Connectors and Connection Diagrams.

3. Step 1. Hardware Installation

3.1. Mechanical Mounting

The PRO-Ax0V80x-SA-CAN drive is intended to be mounted horizontally on a metallic support using the provided mounting holes and the recommended mating connectors, as specified in chapter 3.2.

The metallic support must act as a cooling heat sink.

3.2. Mating Connectors

Connector	Description	Manufacturer	Part Number
J1	Supply input, 4x5.08 female counter part for cable	Camden	CTBA9208/4FL
J2	Motor power, 5x5.08 female counter part for cable	Camden	CTBA9208/5FL
J3,J4	Feedback #1 +Hall & feedback #2		generic 26-pin High Density D-Sub male
J9	RS232		generic RJ10-4/4 phone plug
J5	I/O ; Analog		generic 15-pin D-Sub male, DB15
J8	MICROFIT RECEPTACLE HOUSING, 2x2 WAY	MOLEX	43045-0400
J8	CRIMP PIN, MICROFIT, 5A	MOLEX	43030-0007
J6,J7	Standard 8P8C modular jack (RJ-45) male	-	-

3.3. Connectors and Connection Diagrams

3.3.1. Connector Layout

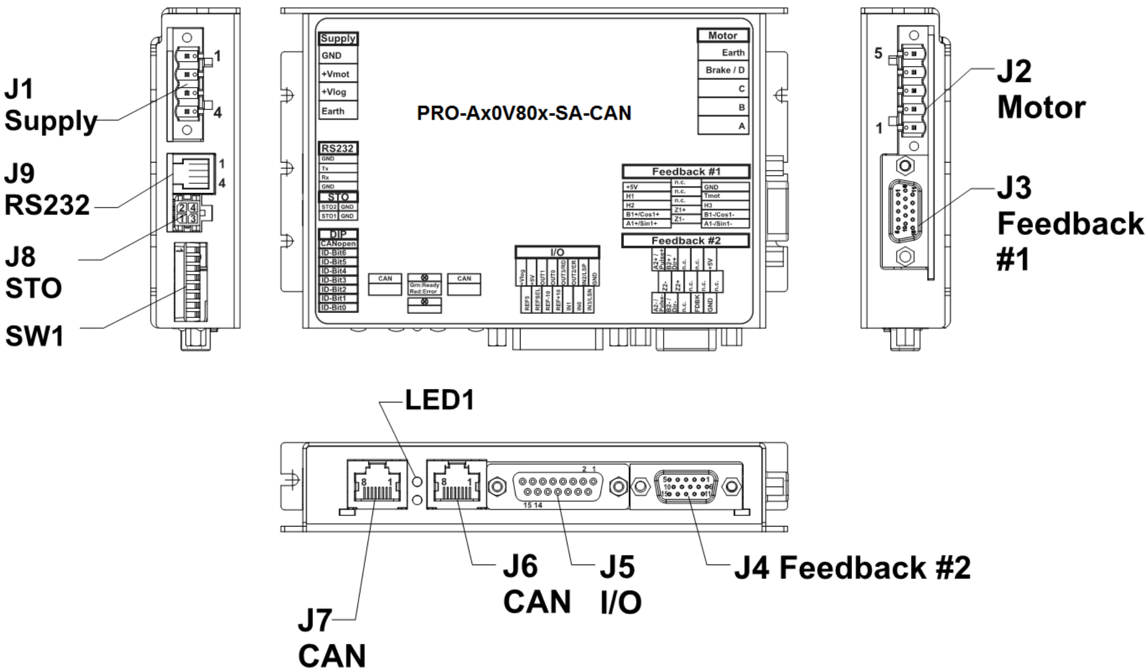


Figure 3.1. *PRO-Ax0V80x-SA-CAN drive connectors*

3.3.2. J1 Power supply input connector pinout

Connector description				
	Pin	**Name**	**Type**	**Description**
J1	**1**	**GND**	-	Negative return (ground) of the power supply
	2	**+V_{MOT}**	**I**	Positive terminal of the motor supply: to 75V_{DC} .
	3	**+V_{LOG}**	**I**	Positive terminal of the logic supply input: 9 to 36V_{DC}
	4	**Earth**	-	Earth connection

3.3.3. J2 Motor output

Connector description				
	Pin	Name	Type	Description
J2	1	A/A+	O	**Phase A** for 3-ph motors, A+ for 2-ph steppers, Motor+ for DC brush motors
	2	B/A-	O	**Phase B** for 3-ph motors, A- for 2-ph steppers, Motor- for DC brush motors
	3	C/B+	O	**Phase C** for 3-ph motors, B+ for 2-ph steppers
	4	BR/B-	O	**Brake resistor / Phase B-** for step motors
	5	Earth	-	Earth connection

3.3.4. J3 Primary feedback connector pinout

Connector description				
	Pin	Name	Type	Description
J3	1	+5V$_{OUT}$	O	5V output supply for I/O usage
	2	Hall 1	I	Digital input Hall 1 sensor
	3	Hall 2	I	Digital input Hall 2 sensor
	4	B1+/Cos+	I	Incr. encoder1 B+ diff. input, or analog encoder Cos+ diff. input
	5	A1+/Sin+	I	Incr. encoder1 A+ diff. input, or analog encoder Sin+ diff. input
	6	n.c.		
	7	n.c.		
	8	n.c.		
	9	Z1+	I	Incr. encoder1 Z+ diff. input
	10	Z1-	I	Incr. encoder1 Z- diff. input
	11	GND	-	Return ground for sensors supply
	12	Temp Mot	I	Analog input, 12-bit, 0-5V. Used to read an analog temperature value
	13	Hall 3	I	Digital input Hall 3 sensor
	14	B1-/Cos-	I	Incr. encoder1 B- diff. input, or analog encoder Cos- diff. input
	15	A1- /Sin-	I	Incr. encoder1 A- diff. input, or analog encoder Sin- diff. input

3.3.5. J4 Secondary feedback connector pinout

Connector description				
	Pin	Name	Type	Description
J4	1	+5V$_{OUT}$	O	5V output supply for I/O usage
	2	n.c.		
	3	n.c.		
	4	B2+/Dir+	I	Incr. encoder2 B+ diff. input, or Dir+
	5	A2+/Pulse+	I	Incr. encoder2 A+ diff. input, or analog encoder Sin+ diff. input
	6	n.c.		
	7	n.c.		
	8	n.c.		
	9	Z2+/PWM+	I	Incr. encoder2 Z+ diff. input
	10	Z2-/PWM-	I	Incr. encoder2 Z- diff. input
	11	GND	-	Return ground for sensors supply
	12	FDBK	I	Analog input, 12-bit, 0-5V. Used to read an analog position or speed feedback (as tacho), or used as general purpose analog input
	13	n.c.		
	14	B2-/Dir-	I	Incr. encoder2 B- diff. input, or Dir-
	15	A2- /Pulse-	I	Incr. encoder2 A- diff. input, or Pulse-

3.3.6. J5 Digital, analog I/O and logic supply connector pinout

Connector description				
	Pin	**Name**	**Type**	**Description**
J5	**1**	**GND**	-	Return ground for I/O pins
	2	**IN2/LSP**	**I**	5-36V digital PNP/NPN input. Positive limit switch input
	3	**OUT2/Error**	**O**	5-36V 0.5A, drive Error output, active low, NPN open-collector/TTL pull-up. Also drives the red LED
	4	**OUT3/Ready**	**O**	5-36V 0.5A, drive Ready output, active low, NPN open-collector/TTL pull-up. Also drives the green LED.
	5	**OUT0**	**O**	5-36V 0.5A, general-purpose digital output, NPN open-collector/TTL pull-up
	6	**OUT1**	**O**	5-36V 0.5A, general-purpose digital output, NPN open-collector/TTL pull-up
	7	**$+5V_{OUT}$**	**O**	5V output supply for I/O usage
	8	**$+V_{LOG}$**	**I**	Positive terminal of the logic supply input: 9 to $36V_{DC}$
	9	**IN3/LSN**	**I**	5-36V digital PNP/NPN input. Negative limit switch input
	10	**IN0**	**I**	5-36V general-purpose digital PNP/NPN input
	11	**IN1**	**I**	5-36V general-purpose digital PNP/NPN input
	12	**REF+10**	**I**	Analog input, 12-bit, +/-10V input. Used to read an analog position, speed or torque reference.
	13	**REF-10**	**I**	Connect REF-10 to GND when REF5 is used.
	14	**REFSEL**	**I**	Analog selection, floating for +/-10V input, GND connected when REF5+ is used.
	15	**REF5**	**I**	Analog input, 12-bit, 0-5V input. Used to read an analog position, speed or torque reference.

3.3.7. J6, J7 CAN-Bus connectors

Connector description				
	Pin	**Name**	**Type**	**Description**
J6,J7	**1**	**Can-Hi**	**I/O**	CAN-Bus positive line (dominant high)
	2	**Can-Lo**	**I/O**	CAN-Bus negative line (dominant low)
	3	**GND**	-	Return ground for CAN-Bus
	4, 5	**-**	-	Reserved. Do not use.
	6..8	**n.c.**	-	Not connected

3.3.8. J8 Enable/STO connector pinout (Labelled STO)

Connector description				
	Pin	**Name**	**Type**	**Description**
J8	**1**	**STO1**	**I**	Enable/STO input1. Set STO1 and STO2 HIGH to enable motor outputs.
	2	**STO2**	**I**	Enable/STO input2. Set STO1 and STO2 HIGH to enable motor outputs.
	3	**GND**	-	Return ground for Enable/STO pins
	4	**GND**	-	Return ground for Enable/STO pins

Remark: *Enable/STO pins set HIGH usually means they should be connected to +Vlog.*

3.3.9. J9 RS232 connector pinout

Connector description				
	Pin	**Name**	**Type**	**Description**
J7	**1**	**GND**	-	Return ground for RS-232 pins
	2	**232TX**	O	RS-232 Data Transmission
	3	**232RX**	I	RS-232 Data Reception
	4	**GND**	-	Return ground for RS-232 pins

3.3.10. AxisID DIP switch settings

Connector description				
	Pin	**Name**	**Type**	**Description**
SW1	**1**	**CANopen**	-	ON (down): CANopen communication protocol OFF(up): MPLCAN communication protocol
	2	**ID-Bit6**	-	Hardware AxisID selection switches
	3	**ID-Bit5**	-	
	4	**ID-Bit4**	-	
	5	**ID-Bit3**	-	
	6	**ID-Bit2**	-	
	7	**ID-Bit1**	-	
	8	**ID-Bit0**	-	

3.3.11. 24V Digital I/O Connection

3.3.11.1 PNP inputs

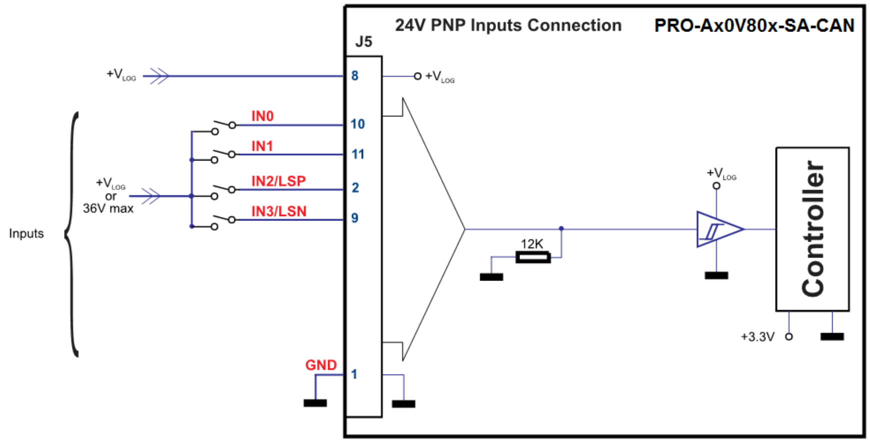


Figure 3.2. *24V Digital PNP Inputs connection*

Remarks:

1. *The inputs are selectable as PNP/ NPN by software.*
2. *The inputs are compatible with PNP type outputs (input must receive a positive voltage value (5-36V) to change its default state)*

3.3.11.2 NPN inputs

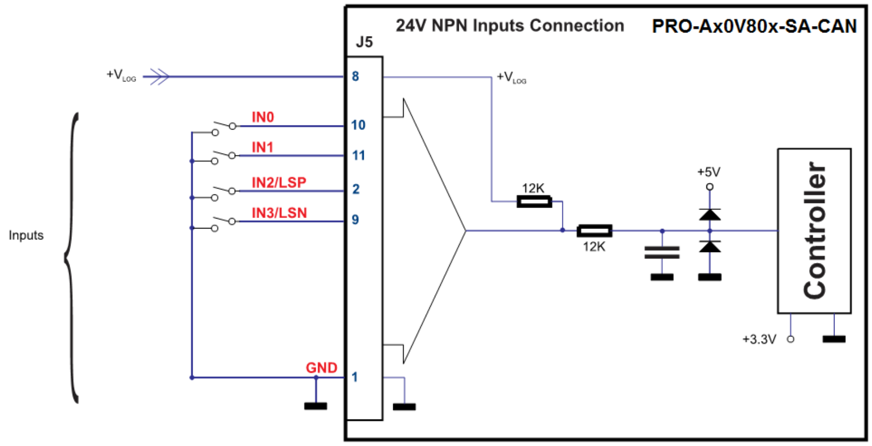


Figure 3.3. *24V Digital NPN Inputs connection*

Remarks:

1. *The inputs are selectable as PNP/ NPN by software.*
2. *The inputs are compatible with NPN type outputs (input must be pulled to GND to change its default state)*

3.3.11.3 NPN outputs

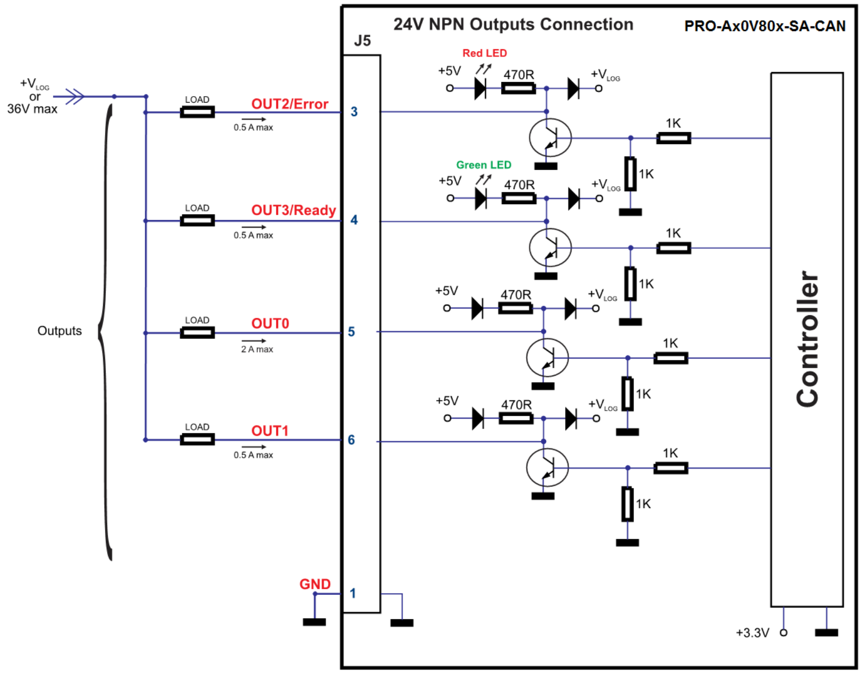


Figure 3.4. *24V Digital NPN Outputs connection*

Remark:

The outputs are compatible with NPN type inputs (load is tied to common +V_{LOG}, output pulls to GND when active and is floating when inactive)

3.3.12. 5V Digital I/O Connection

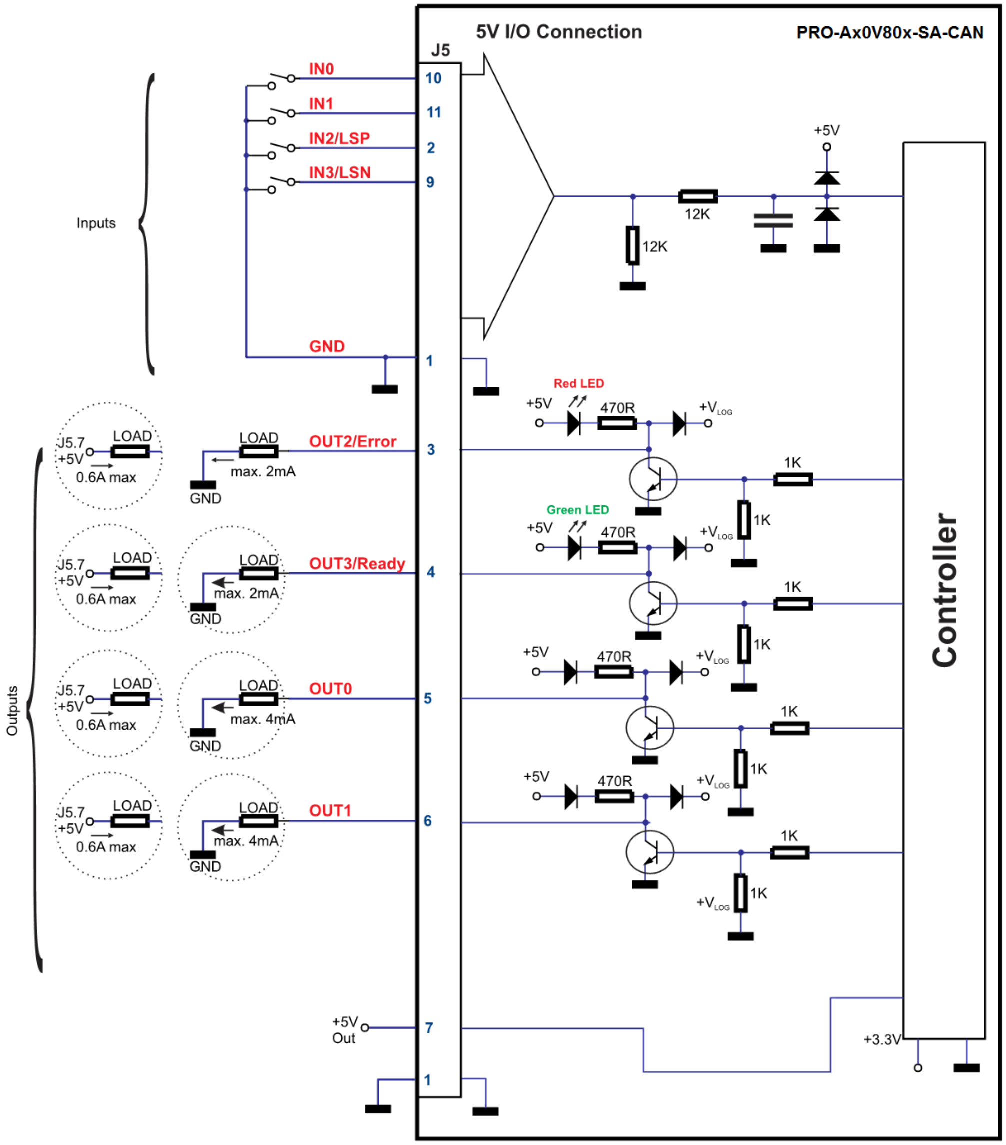


Figure 3.5. *5V Digital I/O connection*

Remarks:

1. *The inputs are selectable as PNP/ NPN by software. For this connection they are selected as PNP.*
2. *The inputs are compatible with TTL(5V), LVTTL(3.3V), CMOS (3.3V-24V) outputs*
3. *The outputs are compatible with TTL (5V) and CMOS (5V) inputs*
4. *The output loads can be individually and independently connected to +5V or to GND.*

3.3.13. Analog Inputs Connection

3.3.13.1 0-5V Input Range

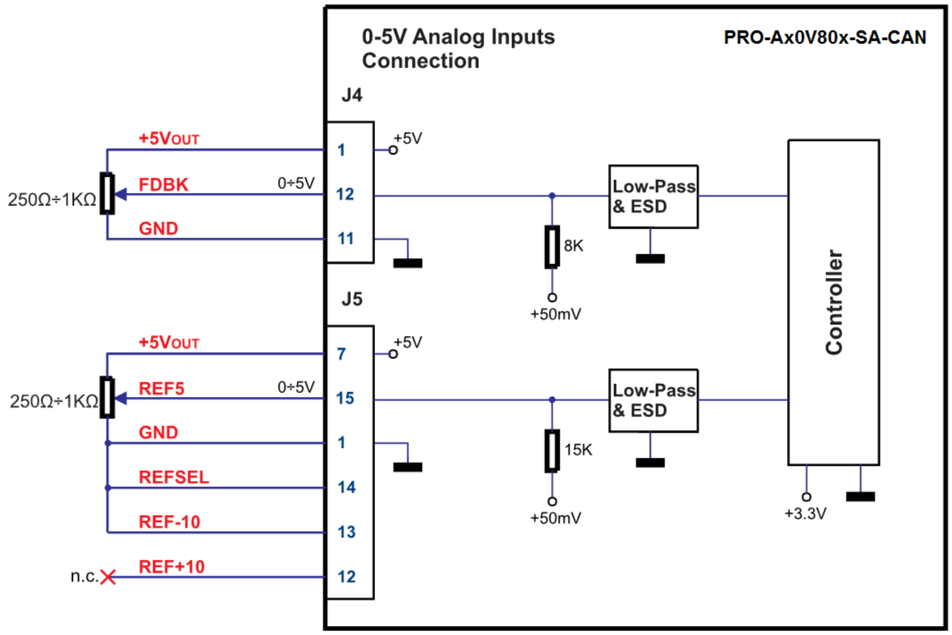


Figure 3.6. 0-5V *Analog inputs connection*

3.3.13.2 +/-10V Input Range

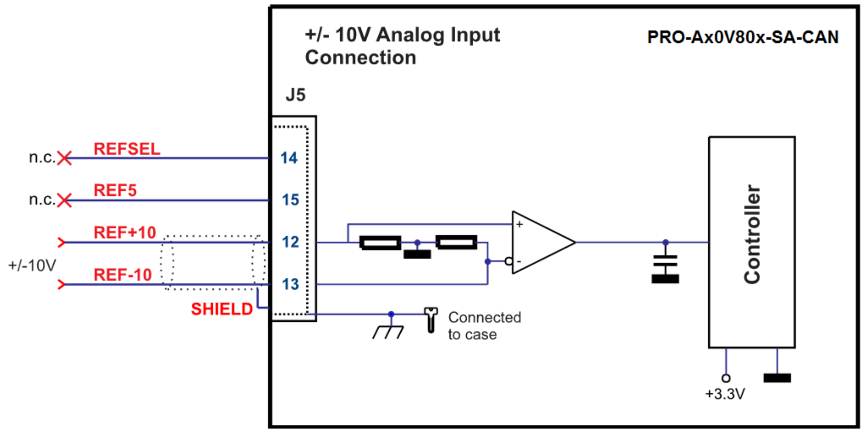


Figure 3.7. +/-10V *Analog inputs connection*

3.3.13.3 Recommendations for Analog Signals Wiring

a) If the analog signal source is single-ended, use a 2-wire shielded cable as follows: 1st wire connects the live signal to the drive positive input (+); 2nd wire connects the signal ground to the drive negative input(-).

b) If the analog signal source is differential and the signal source ground is isolated from the drive GND, use a 3-wire shielded cable as follows: 1st wire connects the signal plus to the drive positive input (+); 2nd wire connects the signal minus to the drive negative input (-) and 3rd wire connects the source ground to the drive GND

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 signal plus to the drive positive input (+); 2nd wire connects the signal minus to the drive negative input (-)

d) For all of the above cases, connect the cable shield to the drive GND and leave the other shield end unconnected to the signal source. To further increase the noise protection, use a double shielded cable with inner shield connected to drive GND and outer shield connected to the motor chassis (earth).

3.3.14. Motor connections

Connector J2 provides electrical power to the motor phase wires. Series inductors may be required in the motor phases to balance the motor inductance and cable capacitance. Type and value are dependent on the motor and cable used.

3.3.14.1 Brushless Motor connection

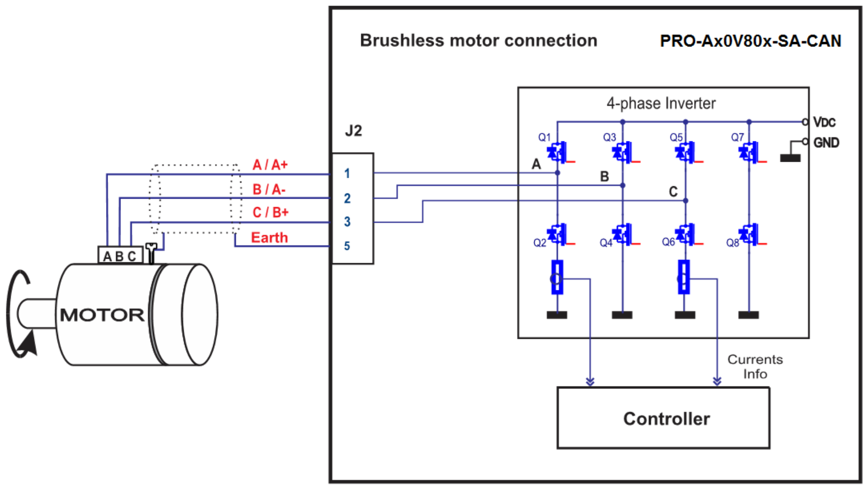


Figure 3.8. *Brushless motor connection*

Remark: *To hardware enable the motor outputs, set Enable/STO1 and Enable/STO2 pins HIGH. This usually means they should be connected to +Vlog.*

3.3.14.2 2-phase Step Motor connection

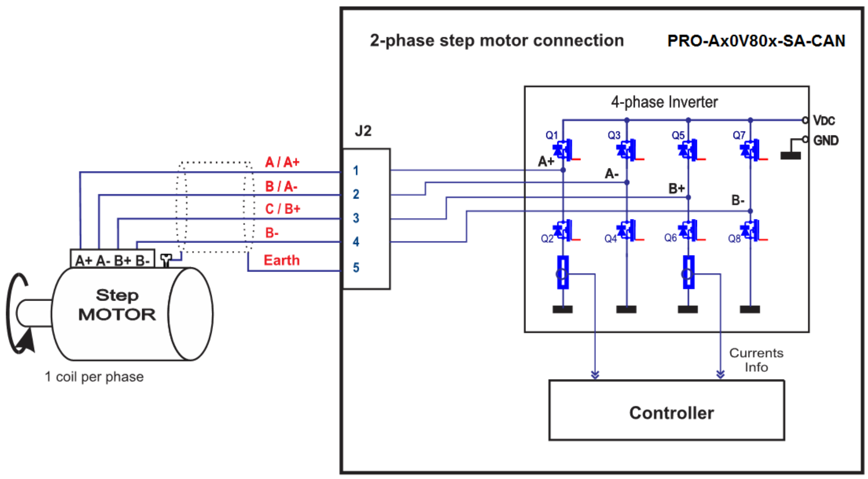


Figure 3.9. *2-phase step motor connection, one coil per phase*

Remark: *To hardware enable the motor outputs, set Enable/STO1 and Enable/STO2 pins HIGH. This usually means they should be connected to +Vlog.*

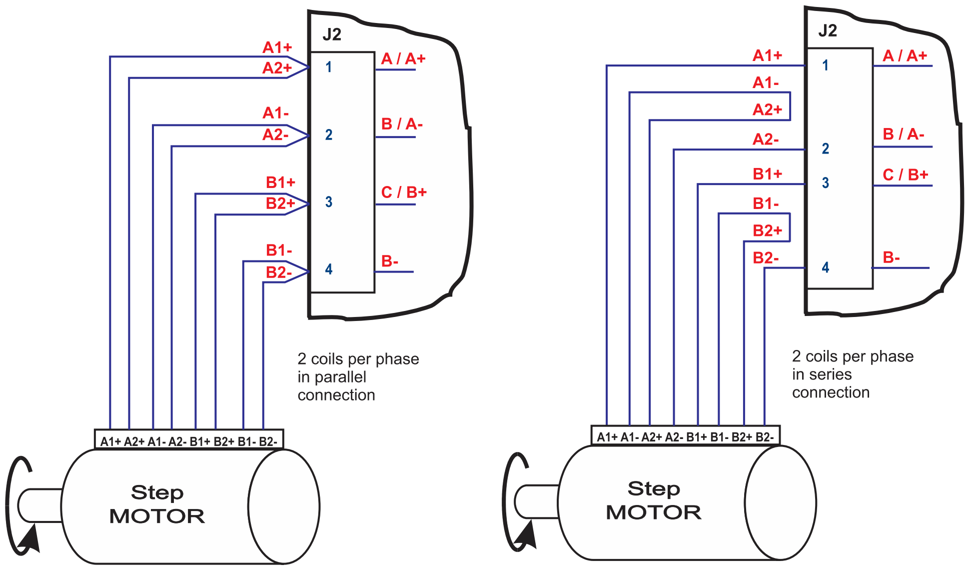


Figure 3.10. *2-phase step motor connection, two coils per phase*

3.3.14.3 3-Phase Step Motor connection

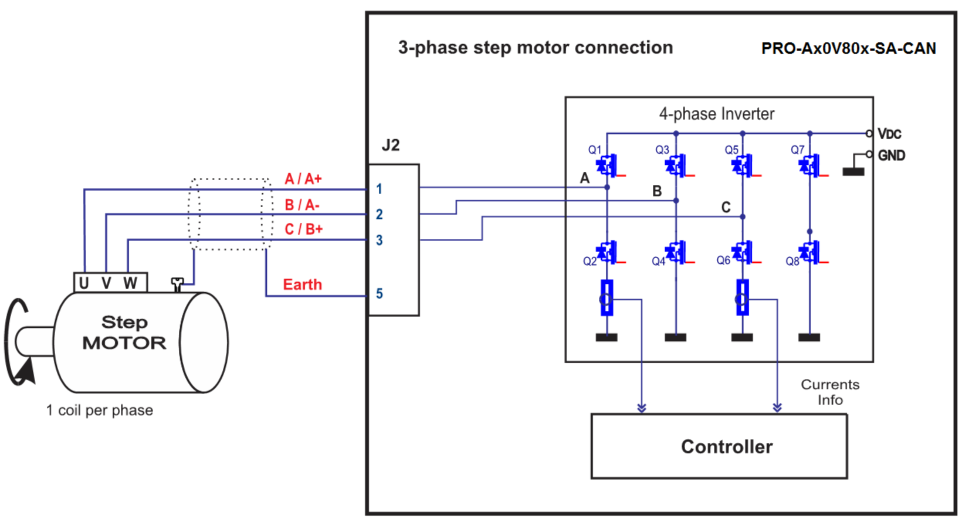


Figure 3.11. *3-phase step motor connection*

Remark: *To hardware enable the motor outputs, set Enable/STO1 and Enable/STO2 pins HIGH. This usually means they should be connected to +Vlog.*

3.3.14.4 DC Brushed Motor connection

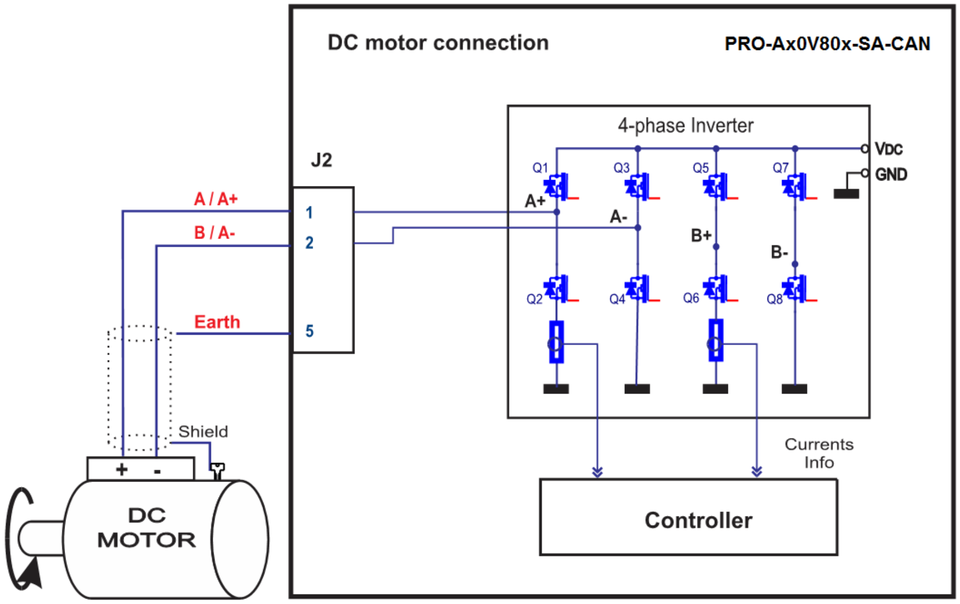


Figure 3.12. *DC Motor connection*

Remark: *To hardware enable the motor outputs, set Enable/STO1 and Enable/STO2 pins HIGH. This usually means they should be connected to +Vlog.*

3.3.14.5 Recommendations for motor wiring

a) Avoid running the motor wires in parallel with other wires for a distance longer than 2 meters. If this situation cannot be avoided, use a shielded cable for the motor wires. Connect the cable shield to the PRO-Ax0V80 GND pin. Leave the other end disconnected.

b) The parasitic capacitance between the motor wires must not bypass 10nF. If very long cables (tens of meters) are used, this condition may not be met. In this case, add series inductors between the PRO-Ax0V80 outputs and the cable. The inductors must be magnetically shielded (toroidal, for example), and must be rated for the motor surge current. Typically the necessary values are around 100 μH.

c) A good shielding can be obtained if the motor wires are running inside a metallic cable guide.

3.3.15. Feedback connections

3.3.15.1 Differential Incremental Encoder1 Connection

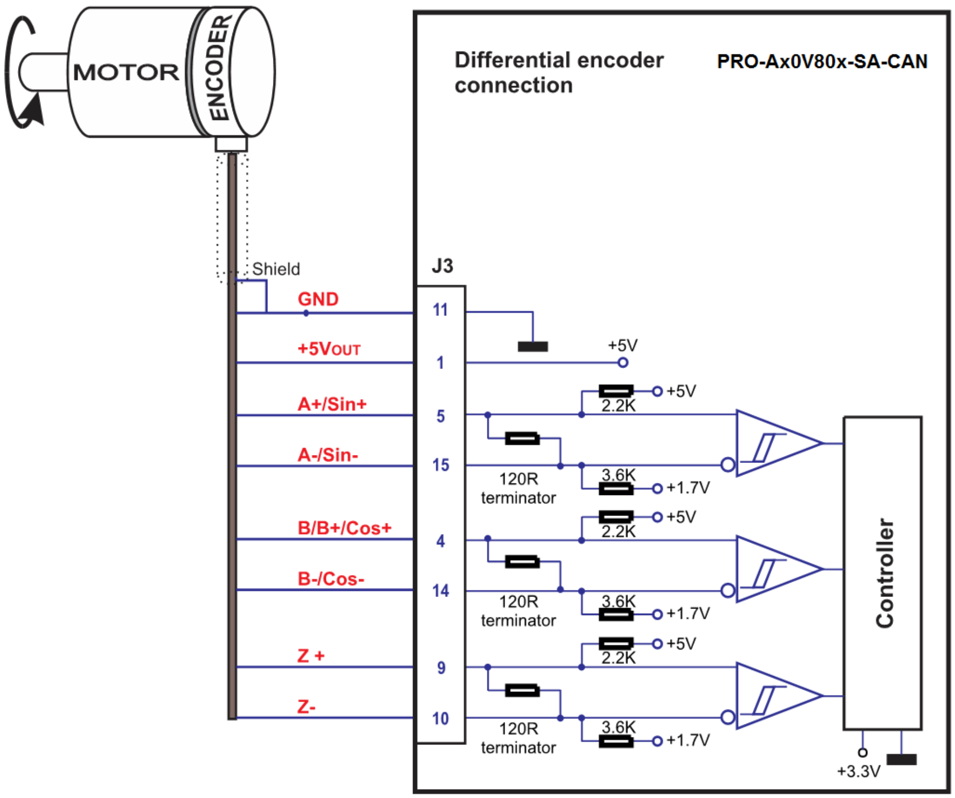


Figure 3.13. *Differential incremental encoder connection*

Remark: *120Ω termination resistors are present in the drive.*

3.3.15.2 Digital Hall Connection

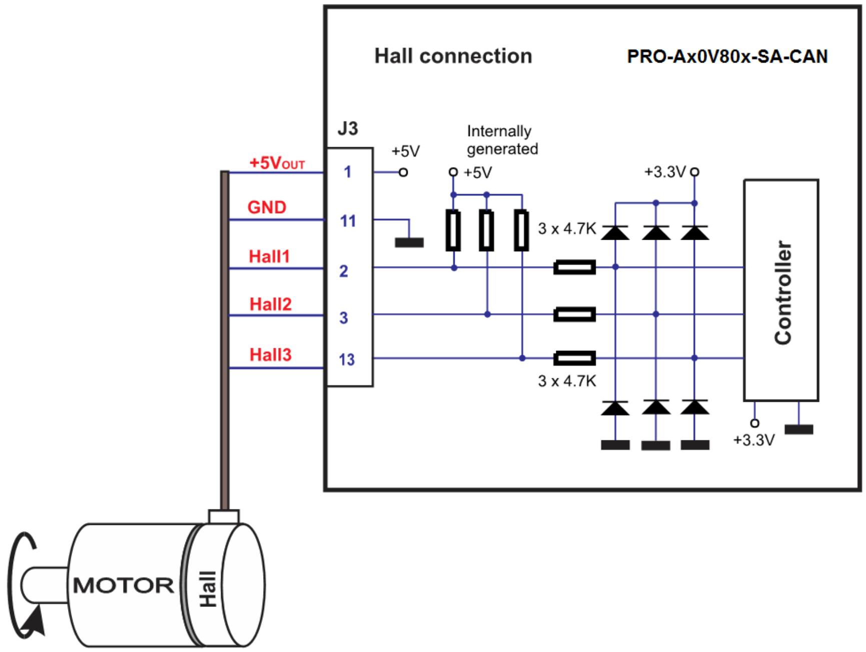


Figure 3.14. *Digital Hall connection*

3.3.15.3 Pulse and direction connection

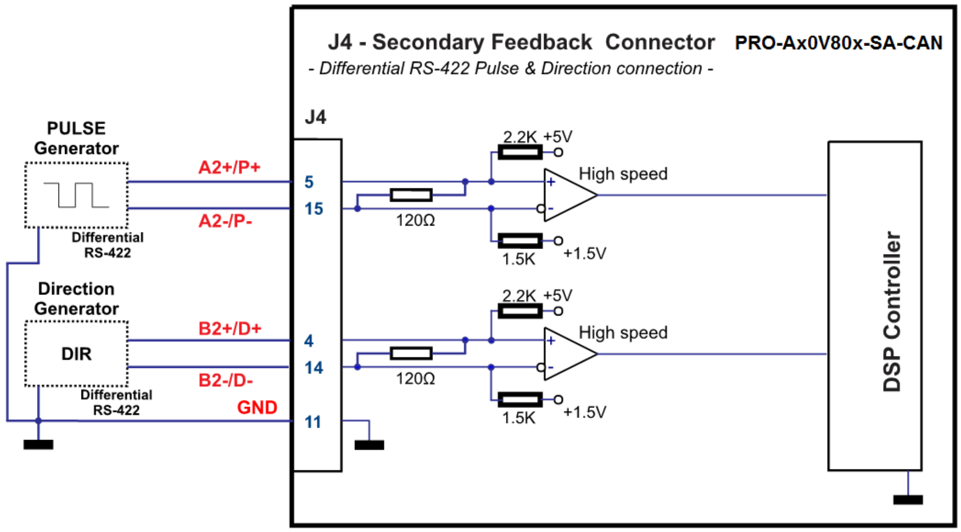


Figure 3.15. **J4 – Differential (RS-422) Pulse & Direction connection**

Remark: *Termination resistors (120Ω) are present in the drive.*

3.3.15.4 Differential Incremental Encoder2 Connection

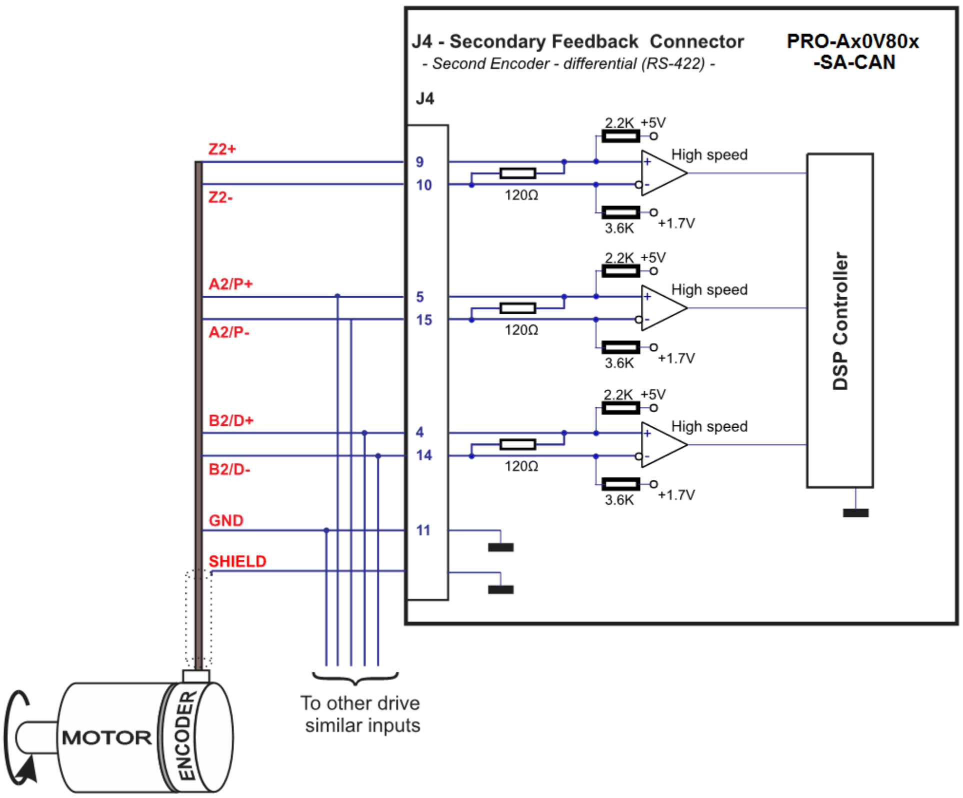


Figure 3.16. **J9 – Second encoder – differential (RS-422) connection**

Remark: *Termination resistors (120Ω) are present in the drive.*

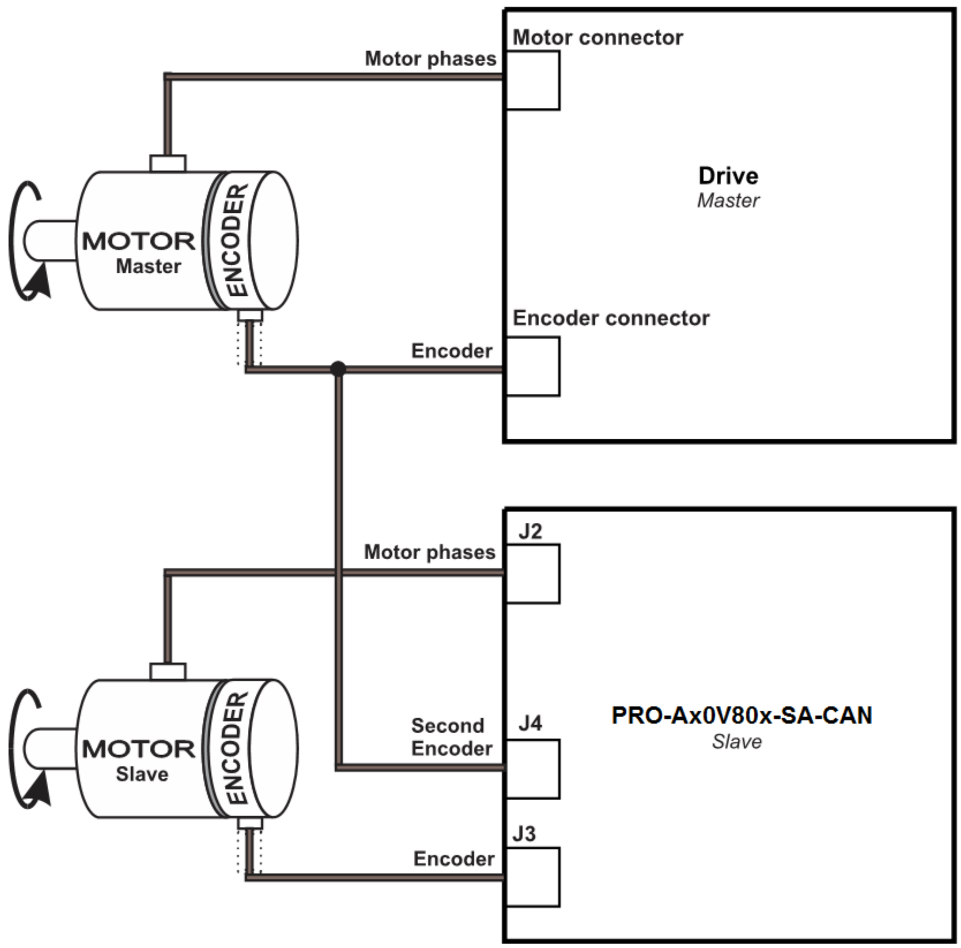


Figure 3.17. **J4 – Master – Slave connection using second encoder input**

3.3.15.5 Sine-Cosine Analog Encoder Connection

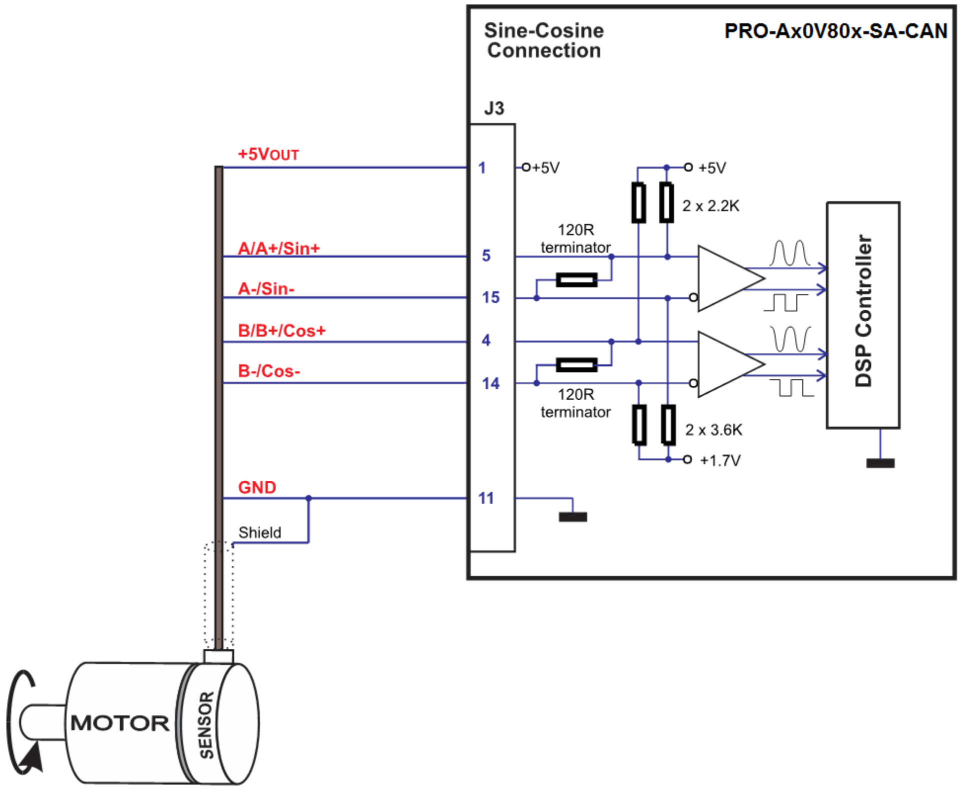


Figure 3.18. *Sine-Cosine analog encoder connection*

Remark: *Termination resistors (120Ω) are present in the drive.*

3.3.15.6 Recommendations for wiring

a) Always connect both positive and negative signals when the position sensor is differential and provides them. Use one twisted pair for each differential group of signals as follows: A+/Sin+ with A-/Sin-, B+/Cos+ with B-/Cos-, Z+ with Z-. Use another twisted pair for the 5V supply and GND.

b) Always use shielded cables to avoid capacitive-coupled noise when using single-ended encoders or Hall sensors with cable lengths over 1 meter. Connect the cable shield to the GND, at only one end. This point could be either the PRO-Ax0V80 (using the GND pin) or the encoder / motor. Do not connect the shield at both ends.

c) If the PRO-Ax0V80 5V supply output is used by another device (like for example an encoder) and the connection cable is longer than 5 meters, add a decoupling capacitor near the supplied device, between the +5V and GND lines. The capacitor value can be 1...10 µF, rated at 6.3V.

3.3.16. Power Supply Connection

3.3.16.1 Supply Connection

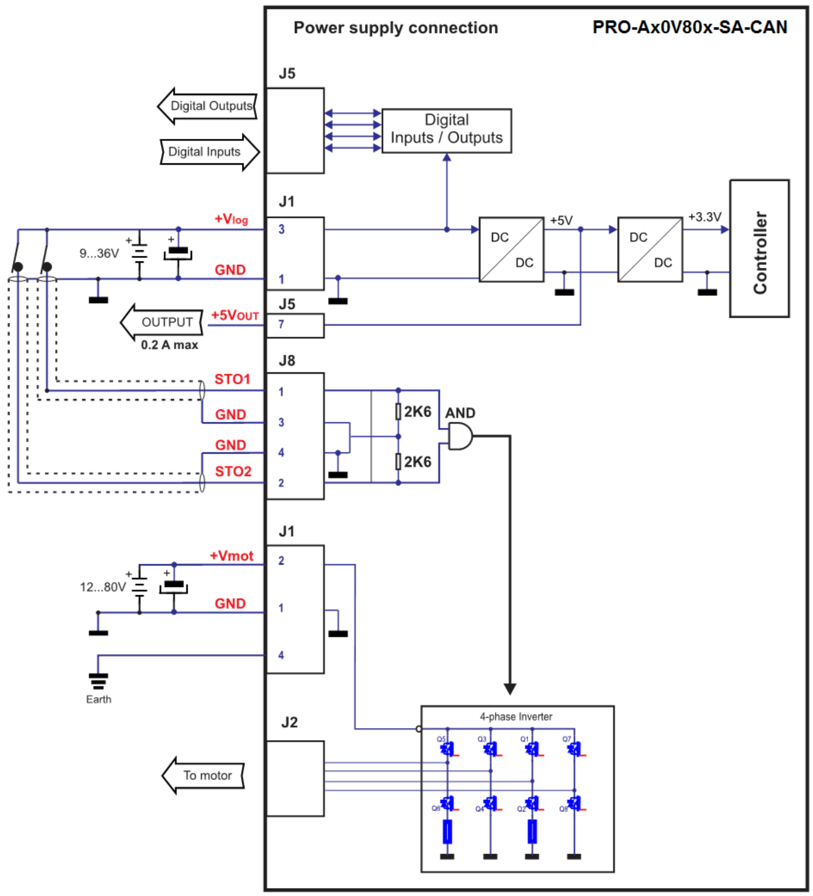


Figure 3.19. *Supply connection*

3.3.16.2 Recommendations for Supply Wiring

The PRO-Ax0V80x-SA-CAN always requires two supply voltages: V_{log} and V_{mot}.
A 100uF capacitor (rated at an appropriate voltage) is recommended to be connected from V_{log} to GND.
Use short, thick wires between the PRO-Ax0V80 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-Ax0V80.
It is recommended to connect the negative motor supply return (GND) to the Earth protection near the power supply terminals.

3.3.16.3 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 84V, the drive over-voltage protection is triggered and the drive power stage is disabled. In order to avoid this situation you have 2 options:

Option 1. Add a capacitor on the motor supply big enough to absorb the overall energy flowing back to the supply. 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} = 84V 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:

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

where:

J_M – total rotor inertia [kgm²]

J_L – total load inertia as seen at motor shaft after transmission [kgm²]

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/s²]

$h_{initial}$ – initial system altitude [m]

h_{final} – final system altitude [m]

I_M – motor current during deceleration [A_{RMS}/phase]

R_{Ph} – motor phase resistance [Ω]

t_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].

Option 2. Connect a brake resistor R_{BR} between phase BR/B- (J2/ pin 4) and ground (J1/ pin 1), and activate the software option of dynamic braking (see below).

This option is not available when the drive is used with a step motor.

The dynamic braking option can be found in the Drive Setup dialogue within MotionPRO Developer / PRO Config. The braking will occur when DC bus voltage increases over U_{BRAKE}. This parameter (U_{BRAKE}) should be adjusted depending on the nominal motor supply. Optimally (from a braking point of view), U_{BRAKE} should be a few volts above the maximum nominal supply voltage. This setting will activate the brake resistor earlier, before reaching dangerous voltages – when the over-voltage protection will stop the drive. Of course, U_{BRAKE} must always be less than U_{MAX} – the over-voltage protection threshold.

Remark: *This option can be combined with an external capacitor whose value is not enough to absorb the entire regenerative energy E_M but can help reducing the brake resistor size.*

Brake resistor selection

The brake resistor value must be chosen to respect the following conditions:

1. to limit the maximum current below the drive peak current I_{PEAK} = 38.3A

$$R_{BR} > \frac{U_{MAX}}{I_{PEAK}}$$

2. to sustain the required *braking power*:

$$P_{BR} = \frac{E_M - \frac{1}{2}C(U_{MAX}^2 - U_{brake}^2)}{t_d}$$

where C is the capacitance on the motor supply (external), i.e:

$$R_{BR} < \frac{U_{BRAKE}^2}{2 \times P_{BR}}$$

3. to limit the average current below the drive nominal current I_{NOM}=10A

$$R_{BR} > \frac{P_{BR} \times t_d}{t_{CYCLE} \times I_{NOM}^2}$$

where t_{CYCLE} is the time interval between 2 brakes in case of repetitive moves.

4. to be rated for an average power $P_{AV} = \frac{P_{BR} \times t_d}{t_{CYCLE}}$ and a peak power $P_{PEAK} = \frac{U_{MAX}^2}{R_{BR}}$

Remarks:

1. *If* $\frac{U_{MAX}}{I_{PEAK}} > \frac{U_{BRAKE}^2}{2 \times P_{BR}}$ *the braking power P_{BR} must be reduced by increasing either t_d – the time to decelerate or C – the external capacitor on the motor supply*

2. *If* $\frac{P_{BR} \times t_d}{t_{CYCLE} \times I_{NOM}^2} > \frac{U_{BRAKE}^2}{2 \times P_{BR}}$ *either the braking power must be reduced (see Remark 1) or t_{CYCLE} – the time interval between braking cycles must be increased*

	WARNING!	***THE BRAKE RESISTOR MAY HAVE HOT SURFACES DURING OPERATION.***

3.3.17. Serial RS-232 connection

3.3.17.1 Serial RS-232 connection

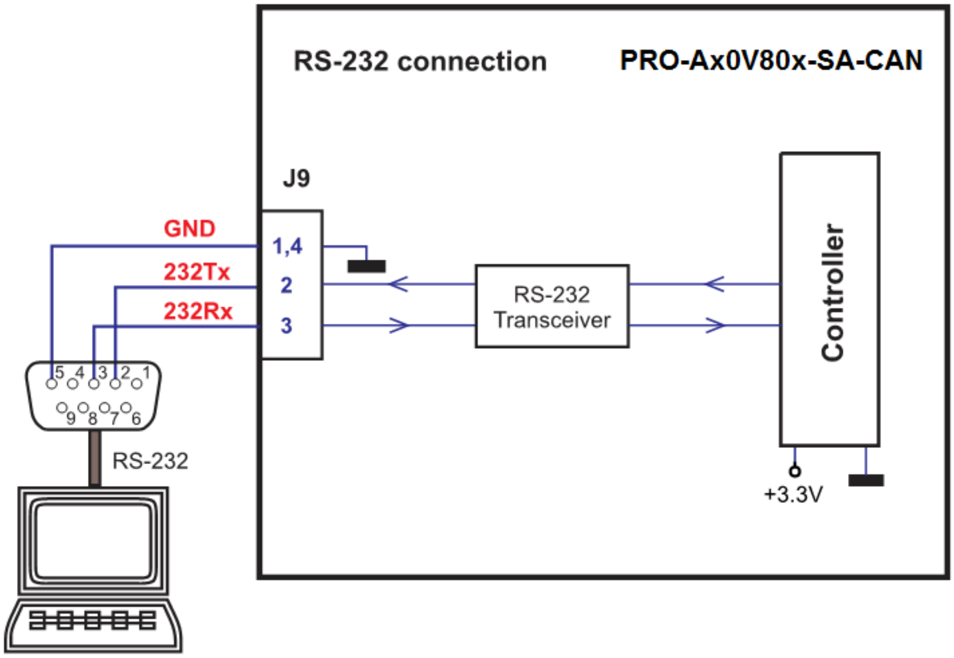


Figure 3.20. *Serial RS-232 connection*

3.3.17.2 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 1 or 4 of J9) must be included inside the shield, like the 232Rx and 232Tx signals

b) Always power-off all the PRO-Ax0V80x-SA-CAN supplies before inserting/removing the RS-232 serial connector

c) Do not rely on an earthed PC to provide the PRO-Ax0V80 GND connection! The drive must be earthed through a separate circuit. Most communication problems are caused by the lack of such connection

CAUTION!	***DO NOT CONNECT/DISCONNECT THE RS-232 CABLE WHILE THE DRIVE IS POWERED ON. THIS OPERATION CAN DAMAGE THE DRIVE***

3.3.18. CAN-bus connection

3.3.18.1 CAN connection

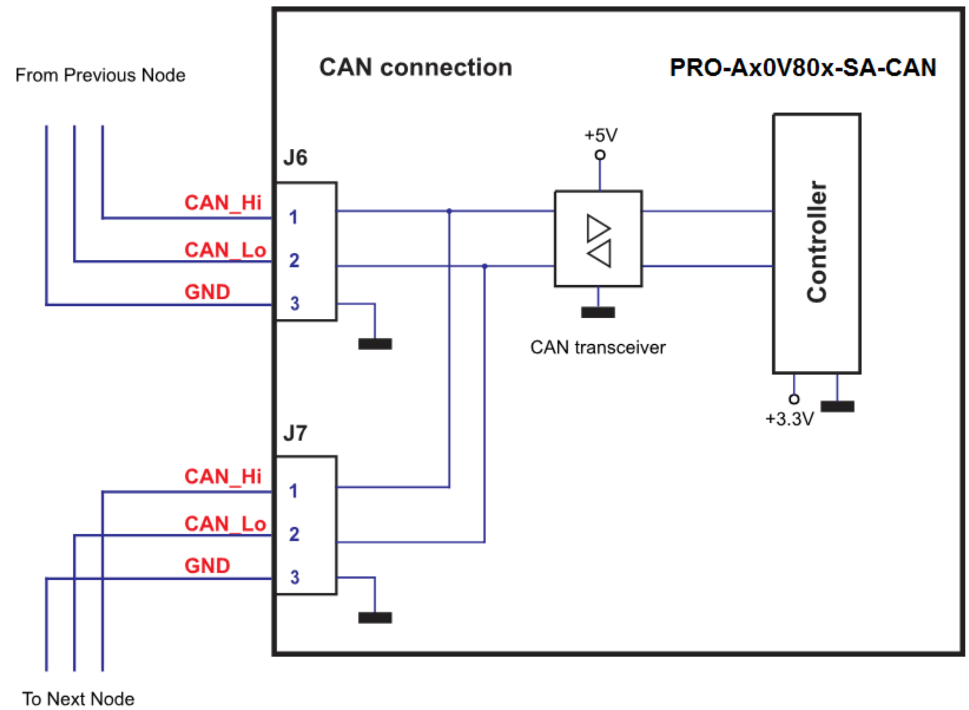


Figure 3.21. *CAN connection*

Remarks:

1. *The CAN network requires a 120-Ohm terminator. This is not included in the drive. shows how to connect it on your network*
2. *CAN signals are not insulated from other PRO-Ax0V80 circuits.*

3.3.18.2 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.

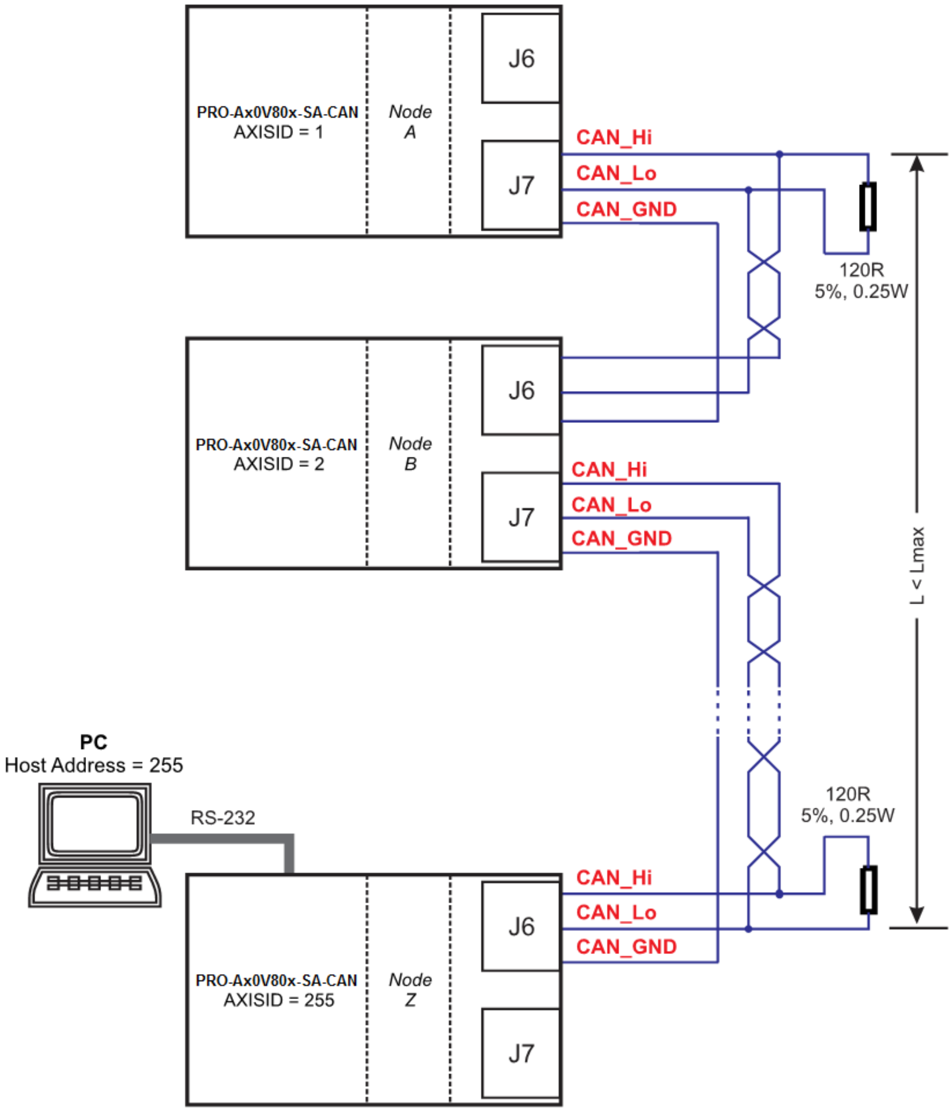


Figure 3.22. *Multiple-Axis CAN network*

Remarks:

1. *The axis IDs in* ***Figure 3.22****, 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* ***2.6.17.***

3.3.19. Disabling Autorun Mode

When an PRO-Ax0V80x-SA-CAN 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.

In order to disable *Autorun* mode, there are 2 methods:

a) Software - by writing value 0x0001 in first EEPROM location at address 0x4000

b) Hardware – by temporary connecting all digital Hall inputs to GND, during the power-on for about 1 second, until the green LED is turned on, as shown in ***Figure 3.23***. This option is particularly useful when it is not possible to communicate with the drive.

After the drive is set in *non-Autorun/slave* mode using 2^{nd} method, the 1^{st} method may be used to invalidate the MPL application from the EEPROM. On next power on, in absence of a valid MPL application, the drive enters in the *non-Autorun/slave* mode independently of the digital Hall inputs status.

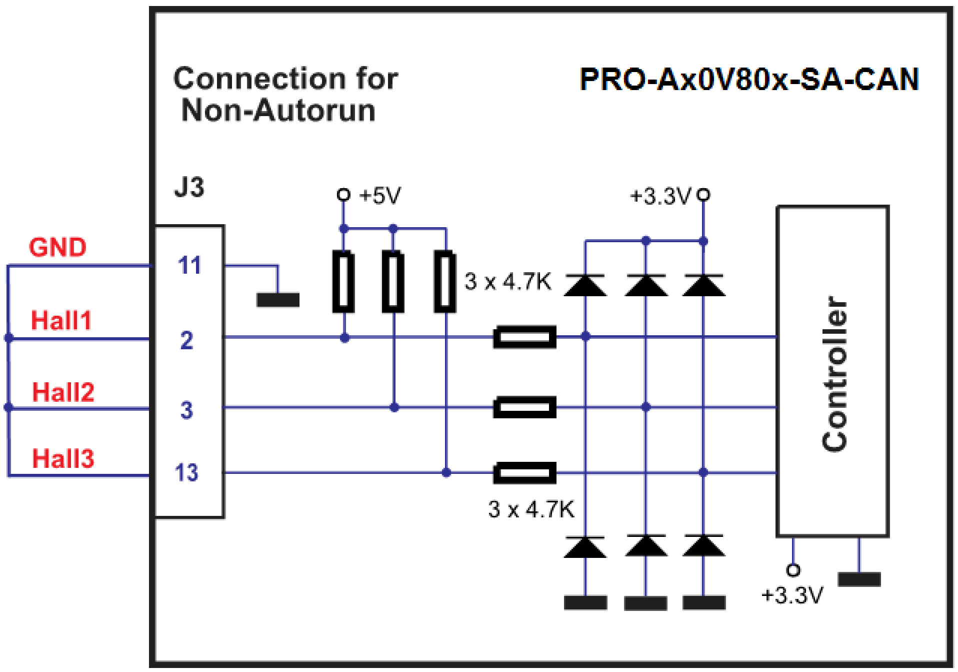


Figure 3.23. *Temporary connection during power-on to disable* ***Autorun*** *mode*

3.

3.4. Selection of the Operation mode and Axis ID, DIP-Switch Settings

The Operation mode is selected by the DIP switch pin1:

ON= CANopen mode / OFF= MPLCAN mode

The drive AxisID value is set after power on by:

- Software, setting via PRO Config a specific AxisID value in the range 1-255.
- Hardware, by setting h/w in PRO Config and selecting a value between 1-127 from the switch SW1

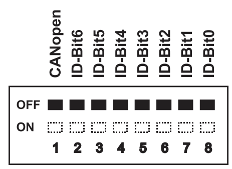


Figure 3.24. **SW1 – DIP Switch**

  - **Position 1**: On = CANopen mode; Off = MPLCAN mode

- **Positions 2 … 8**: ID-Bitx.
  - Axis ID switches - The drive axis/address number is set when H/W is selected in Drive Setup under AxisID field or when the Setup is invalid.
  - The axis ID is an 8 bit unsigned number. Its first 7 bits are controlled by the ID-bit0 to ID-bit6. Bit7 of this variable is always 0. In total, 127 axis ID HW values can result from the DIP switch combinations.

Remark: All switches are sampled at power-up, and the drive is configured accordingly

3.5. LED Indicator

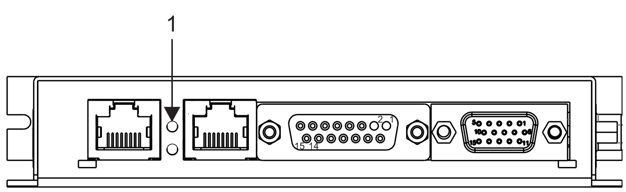


Table 3.1 **– LED Indicators**

LED no.	LED name	LED color	Function
1	Drive Ready/ Error	green	Lit after power-on when the drive initialization ends. Turned off when an error occurs.
		red	Turned on when the drive detects an error condition or when OUT2/Error is set to +Vlog with OUT(2)=0 MPL instruction.

4. Step 2. Drive Setup

4.1. Installing PRO Config

PRO Config is a PC software platform for the setup of the ElectroCraft drives. It can be downloaded ***free of charge*** from ElectroCraft web page. PRO Config 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.

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

On request, PRO Config 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 PRO Config

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

PRO Config 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.

PRO Config 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 PRO Config it is also possible to retrieve the complete setup information from a drive previously programmed.

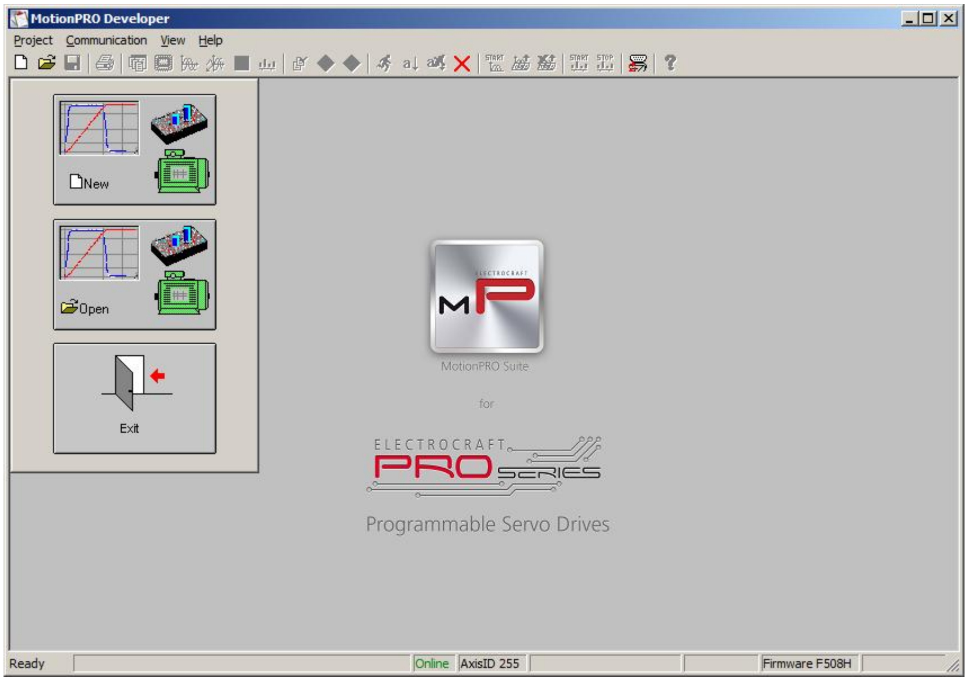
Note that with PRO Config you do only your drive/motor commissioning. For motion programming you have the following options:

- Use a **CANopen** master to control the PRO-Ax0V80 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

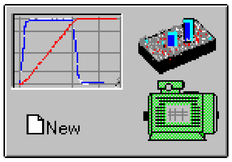
MotionPRO Developer / PRO Config starts with an empty window from where you can create a **New** setup, **Open** a previously created setup which was saved on your PC, or **Upload** the setup from the drive/motor.



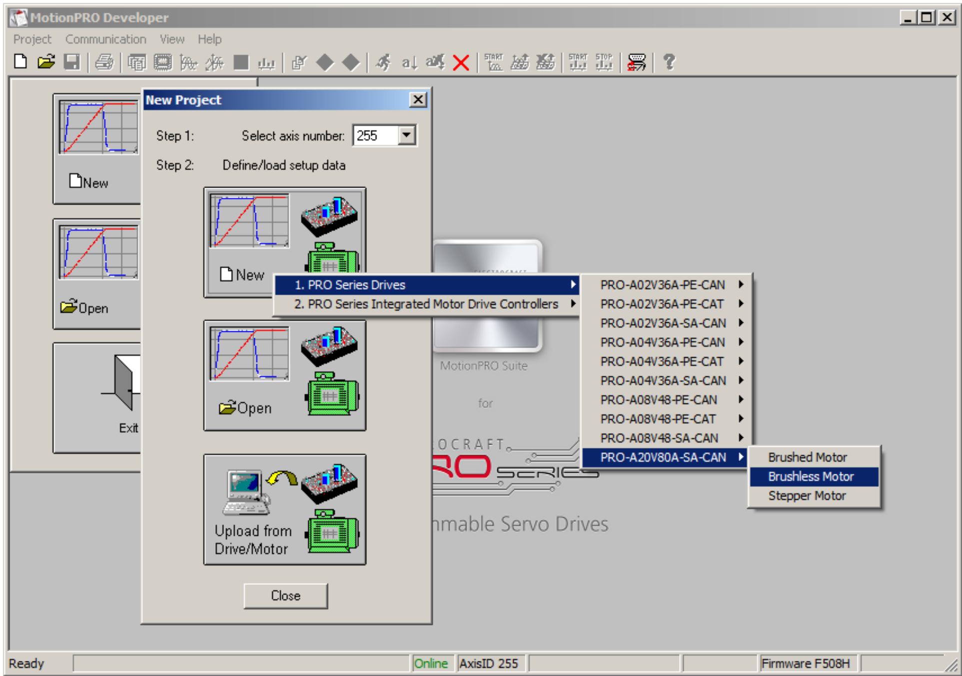
Before selecting one of the above options, you need to establish the communication with the drive you want to commission. Use menu command **Communication | Setup** to check/change your PC communication settings. Press the **Help** button of the dialogue opened. Here you can find detailed information about how to setup your drive and do the connections. Power on the drive, then close the Communication | Setup dialogue with OK. If the communication is established, PRO Config displays in the status bar (the bottom line) the text "**Online**" plus the axis ID of your drive/motor 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, PRO Config 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 drive/motor connected to PC" the option* ***Autodetected****. If this drive is part of a CANbus network, use the menu command* ***Communication | Scan Network***

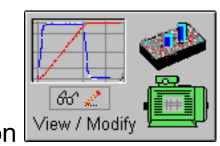
4.2.2. Setup drive/motor

Press **New** button 

and select your drive type.

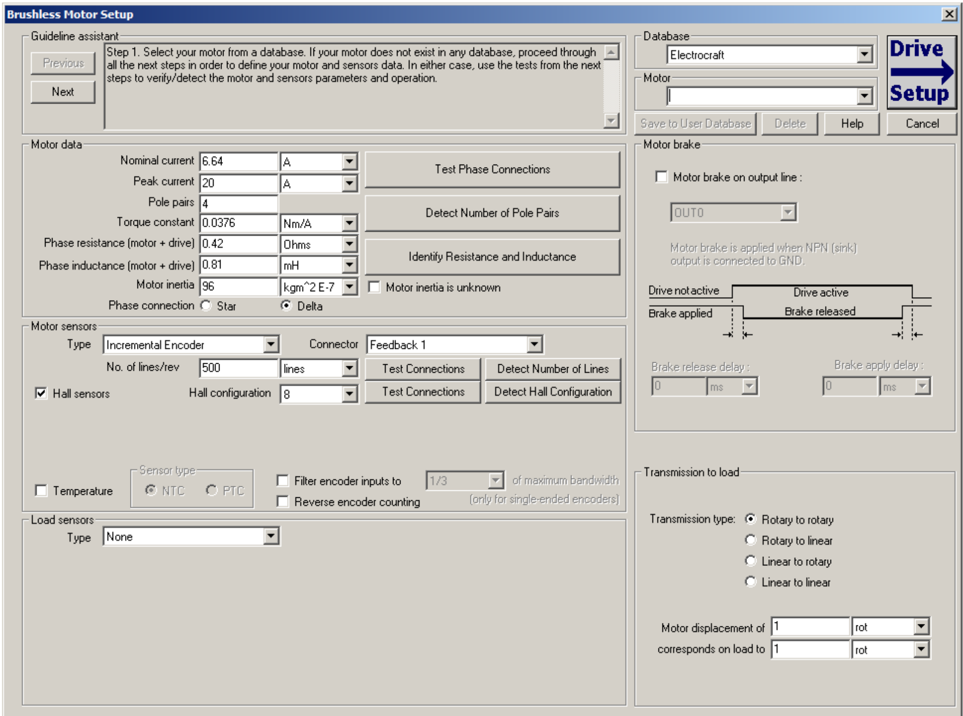


The selection continues with the motor technology (for example: brushless, brushed or stepper).

Next, select ‘Setup’ in the Project window, then press the Edit: View/Modify button 

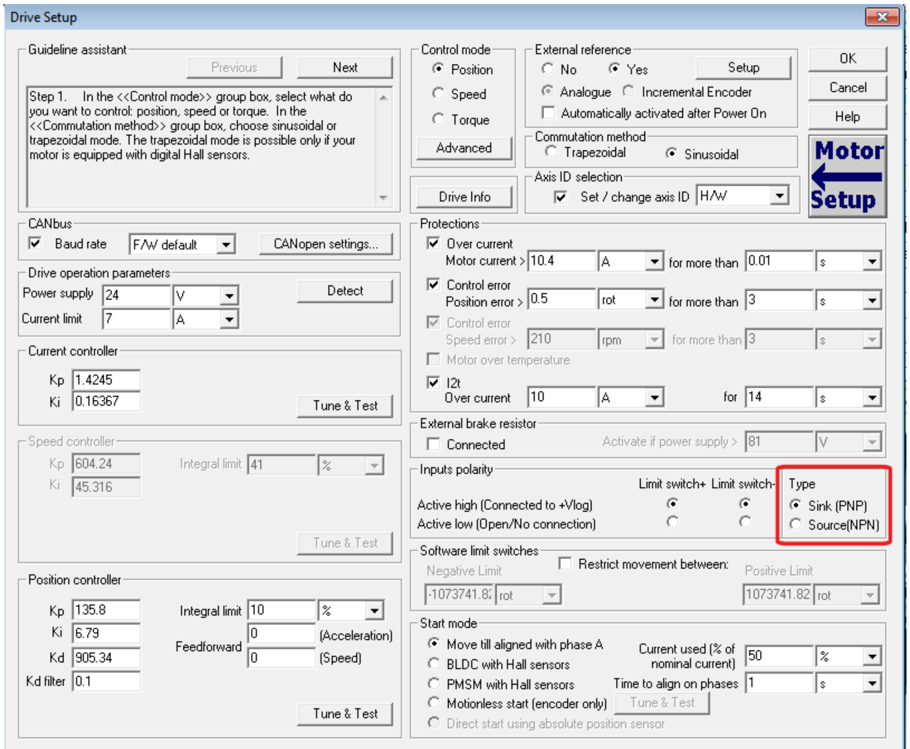
.

The selection opens 2 setup dialogues: for **Motor Setup** and for **Drive setup** through which you can configure and parameterize a 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 feedback 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.

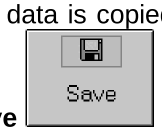
4.2.3. Selecting NPN/PNP inputs type in Setup



In Drive Setup, choose the inputs type PNP or NPN.

4.2.4. 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.

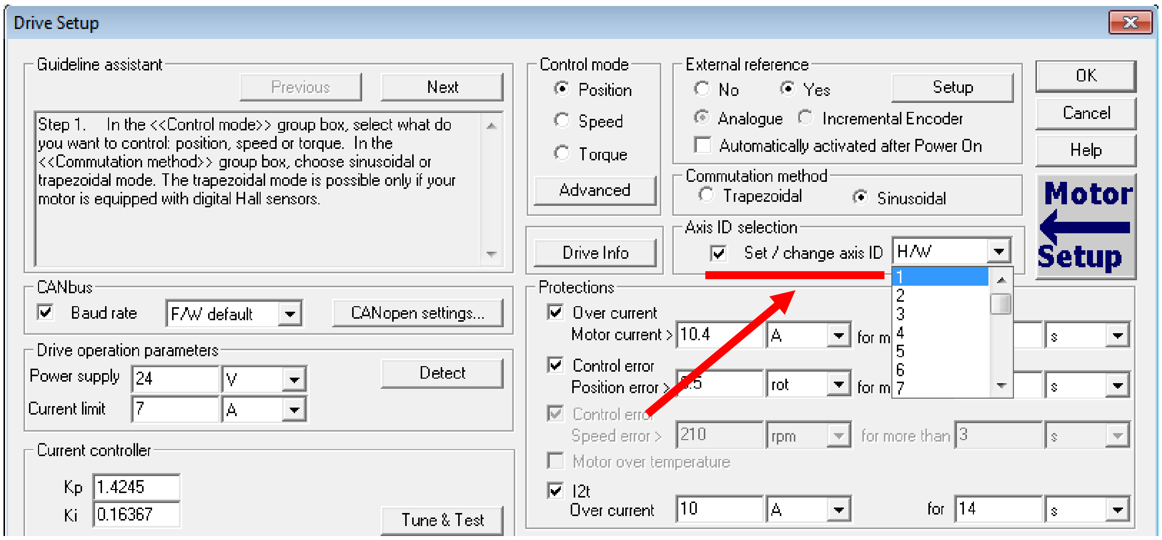
To summarize, you can define or change the setup data in the following ways:

- 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

4.2.5. 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 an PRO-Ax0V80 drive can be set in 3 ways:

- Hardware (H/W) – according with AxisID DIP switch SW1 par. **3.4**.
- Software (via Setup)– any value between 1 and 255, stored in the setup table. If the drive is in CANopen mode, a Node ID value above 127 is automatically converted into 255 and the drive is set with CAN communication in "non-configured" mode waiting for a CANopen master to configure it using CiA-305 protocol. A "non-configured" drive 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 out a drive from "non-configured" mode, by setting another axis ID between 1 and 127, from above dialogue using a serial link between the drive 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 DIP 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 according with the AxisID DIP switch

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 according with the AxisID DIP switch

d) If the setup table is invalid, there is no previous axis ID set from a valid setup table or by a CANopen master, according with the AxisID DIP switch

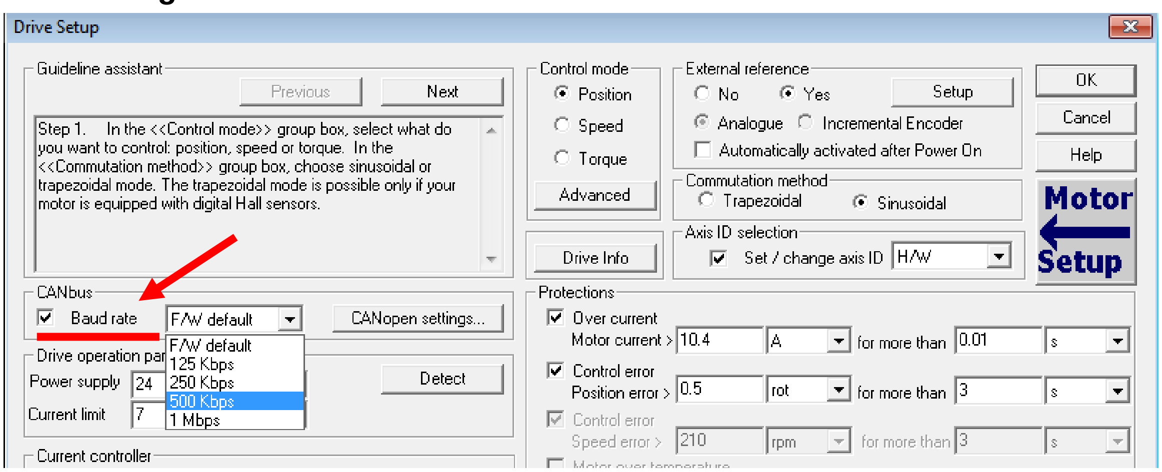
Remark: *If you don't know the axis ID set in a drive, you can find it in the following way:*

a) Connect the drive via a serial RS232 link to a PC where PRO Config or MotionPRO Developer are installed

b) *With the drive powered, open PRO Config or MotionPRO Developer and check the status bar. If communication with the drive is established, the status bar displays **Online** in green and nearby the drive'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 drive/motor connected to PC" the option **Autodetected**. After closing the dialogue with OK, communication with the drive shall be established and the status bar shall display the drive's Axis ID*

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

4.4. Setting CANbus rate



The PRO-Ax0V80 drives 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 checked 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 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

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 drive.

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 represents 0x0092.

The **.sw** file can be programmed into a drive:

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

5. Step 3. Motion Programming

5.1. Using a CANopen Master (for PRO-Ax0V80 CANopen execution)

The PRO-Ax0V80 drive 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.

5.1.1. CiA-301 Application Layer and Communication Profile Overview

The PRO-Ax0V80 drive 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-Ax0V80 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-Ax0V80 supports both SYNC consumer and producer.

- **Time Stamp Object (TIME)**

The Time Stamp Object is supported by the PRO-Ax0V80 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-Ax0V80 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-Ax0V80: 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-Ax0V80 drives 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-Ax0V80 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-Ax0V80 supports 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-Ax0V80, 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 PRO Config 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 a PRO-Ax0V80 drive 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 drive EEPROM using the communication objects for writing data into the drive EEPROM.

5.2. Using the built-in Motion Controller and MPL

One of the key advantages of the ElectroCraft drives is their capability to execute complex motions without requiring an external motion controller. This is possible because ElectroCraft drives 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 an ElectroCraft drive requires creating and downloading a MPL (Motion Programming Language) program into the drive 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
- Execute homing sequences[2]
- 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 Developer software platform.

5.2.2. Installing MotionPRO Developer

MotionPRO Developer is an integrated development environment for the setup and motion programming of ElectroCraft intelligent drives. 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 Developer 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.

[1] Optional for PRO-Ax0V80 CANopen execution
[2] The customization of the homing routines is available only for PRO-Ax0V80 CAN execution

5.2.3. Getting Started with MotionPRO Developer

Using MotionPRO Developer you can quickly do the setup and the motion programming of an ElectroCraft a drive 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 Developer is a set of setup data and a motion program, which can be downloaded to the drive/motor EEPROM or saved on your PC for later use.

MotionPRO Developer 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.

MotionPRO Developer 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.

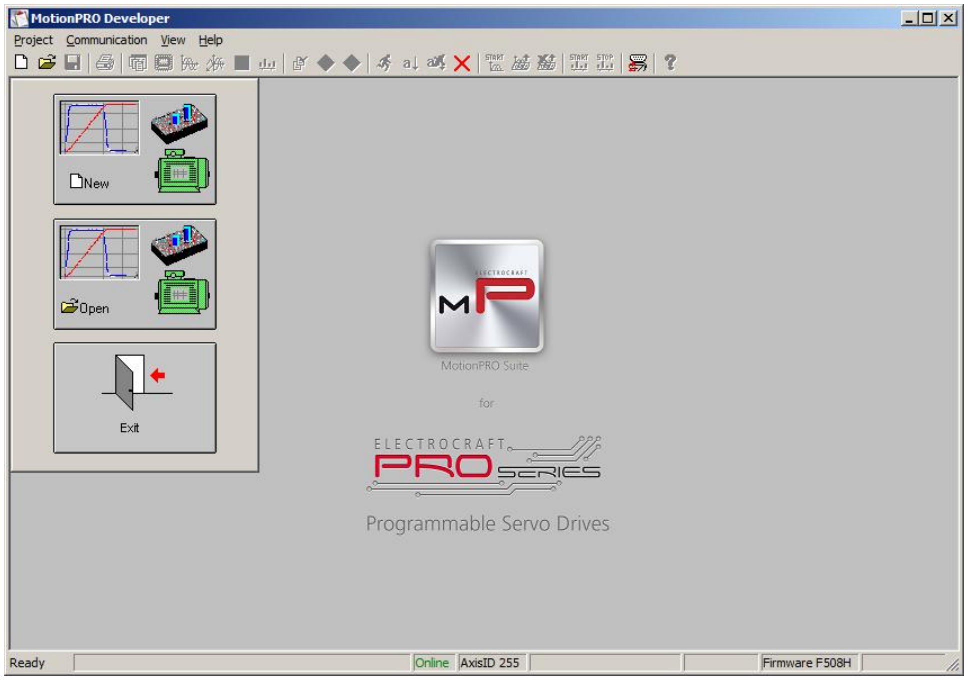
In the first case, each application has a different axis number corresponding to the axis ID of the drives/motors from the network. All data exchanges are done with the drive/motor having the same address as the selected application. In the second case, all the applications have the same axis number.

The setup component contains all the information needed to configure and parameterize an ElectroCraft drive. 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.

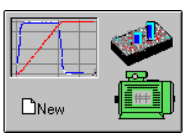
The motion component contains the motion sequences to do. These are described via a MPL (Motion Programming Language) program, which is executed by the drives/motors built-in motion controller.

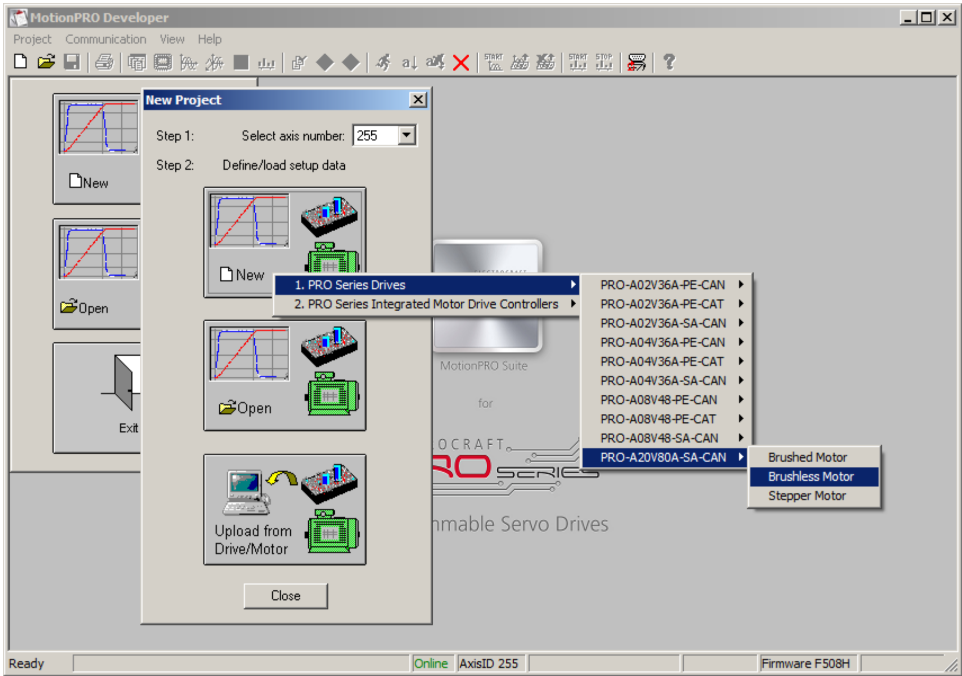
5.2.3.1 Create a new project

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

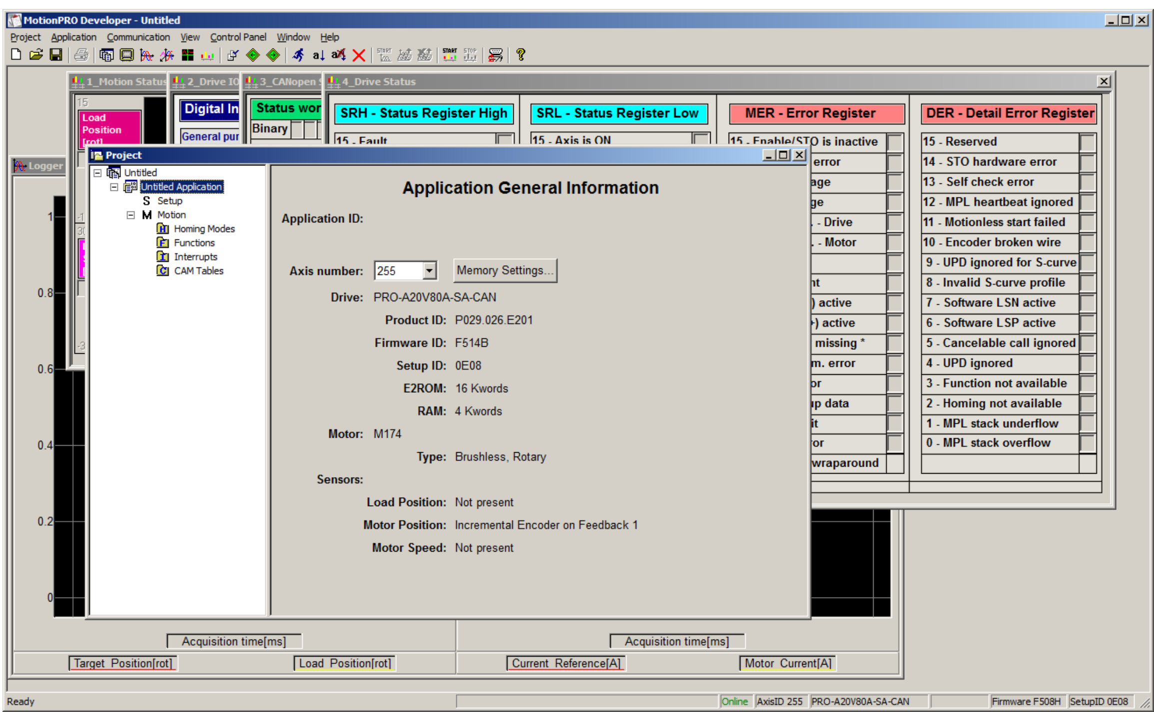


When you start a new project, MotionPRO Developer automatically creates a first application. Additional applications can be added later. You can duplicate an application or insert one defined in another project.

Press **New** button  to open the “New Project” dialogue. Set the axis number for your first application equal with your drive/motor axis ID. The initial value proposed is 255 which is the default axis ID of the drives. Press **New** button and select your drive type. Depending on the product chosen, the selection may continue with the motor technology (for example: brushless or brushed).



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).



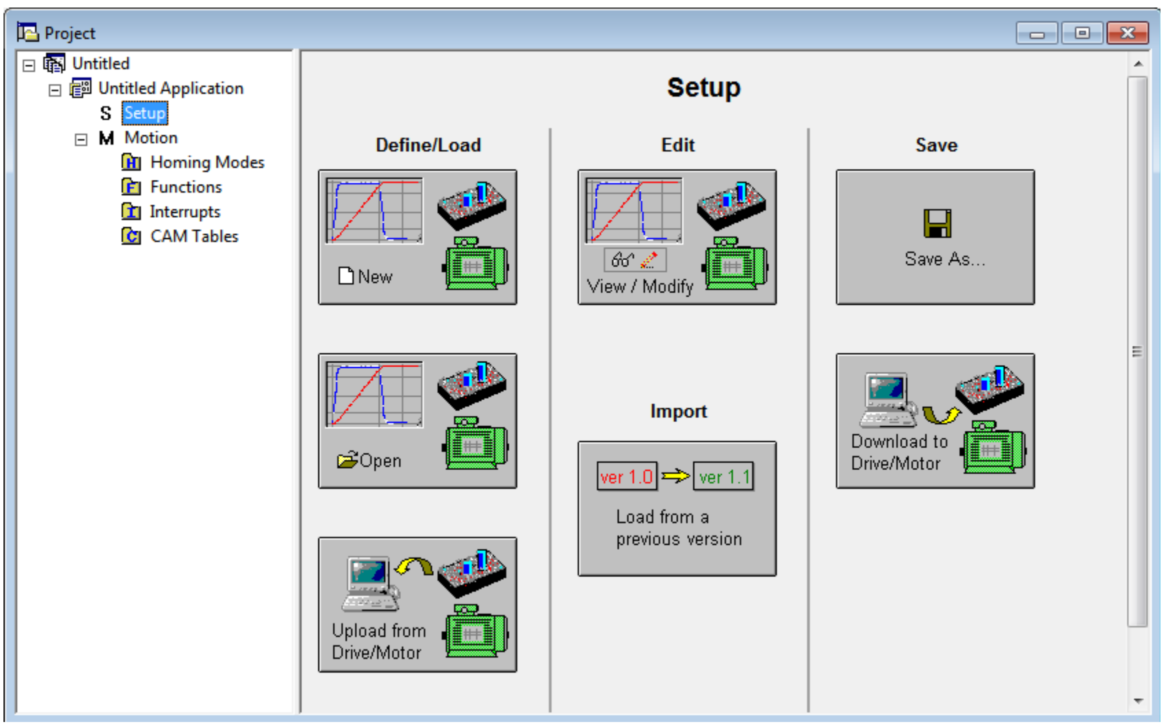
5.2.3.2 Step 2 Establish communication

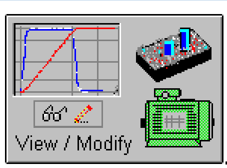
If you have a drive/motor connected with your PC, now it is time to check the communication. Use menu command **Communication | Setup** to check/change your PC communication settings. Press the **Help** button of the dialogue opened. Here you can find detailed information about how to setup your drive/motor 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 drive/motor 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 drive/motor 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 like on PRO Config) through which you can configure and parameterize an

ElectroCraft drive. 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 

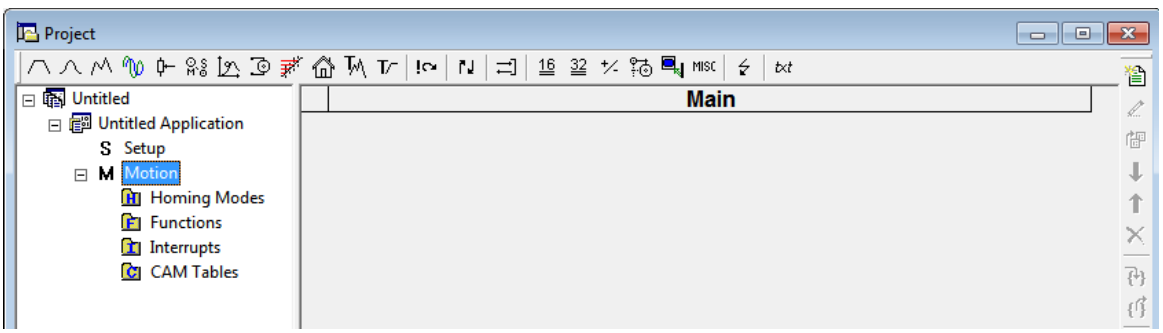
to download your setup data in the drive/motor EEPROM memory in the *setup table*. From now on, at each power-on, the setup data is copied into the drive/motor RAM memory which is used during runtime. It is also possible to save the setup data on your PC and use it in other applications. Note that you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

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 drive/motor 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 Wizard offers you the possibility to program all the motion sequences using high level graphical dialogues which automatically generate the corresponding MPL instructions. Therefore with Motion Wizard 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 Wizard 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 Motion Wizard you define your motion program as a collection of motion objects.

[1] The customization of the interrupt service routines and homing routines is available only for PRO-Ax0V80 CAN execution
[2] Optional for PRO-Ax0V80 CANopen execution

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 Wizard 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
- The MPL program is compiled and downloaded to the drive/motor
- 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 Developer 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 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, a PRO-Ax0V80 offers 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*** manual. All features presented below require usage of MotionPRO Developer as MPL programming tool

Remark: *If you don't use the advanced features presented below you don't need MotionPRO Developer. In this case the PRO-Ax0V80 is treated like a standard CANopen drive, whose setup is done using PRO Config.*

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

With ElectroCraft intelligent drives you can really distribute the intelligence between a CANopen master and the drives in complex multi-axis applications. Instead of trying to command each step of an axis movement, you can program the drives 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 drives 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 a drive 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-Ax0V80 offers others motion modes like[2]: 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:

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

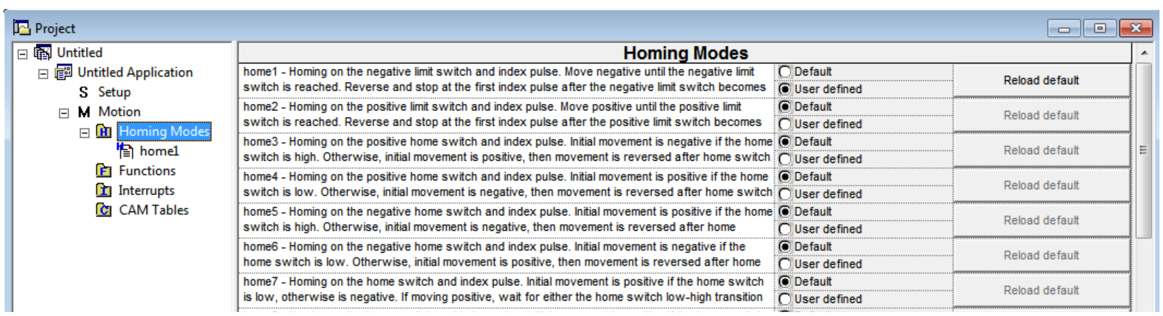
b) The cam points are stored in the drive EEPROM and the master commands their copy into the active RAM memory

c) The cam points are stored in the drive EEPROM and during the drive 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 Developer 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 drives can check the cam presence in the drive EEPROM using the same procedure as for testing of the setup data.

5.3.4. Customizing the Homing Procedures

The PRO-Ax0V80 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. Its calling name and method remain unchanged.

5.3.5. Customizing the Drive Reaction to Fault Conditions

---

[1] The customization of the interrupt service routines and homing routines is available only for PRO-Ax0V80 CAN executions
[2] Optional for the PRO-Ax0V80 CANopen execution

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 intelligent drives. 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 drive / motor via any of its supported channels (RS-232, CAN-bus, etc.), send motion commands, get automatically or on request information about drive / motor 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 PRO Config.

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 intelligent drives. 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 drive/motor via any of its supported channels, send motion commands, get automatically or on request information about drive/motor 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 PRO Config.

6. Scaling factors

ElectroCraft drives work with parameters and variables represented in the drive 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 drive 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 / DC brushed motor with quadrature encoder on motor

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

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

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. Brushless motor with linear Hall signals

The internal position units are counts. The motor is rotary. The resolution i.e. number of counts per revolution is programmable as a power of 2 between 512 and 8192. By default it is set at 2048 counts per turn. The correspondence with the load position in SI units is:

For rotary motors: $$\text{Load\_Position}[SI] = \frac{2 \times \pi}{\text{resolution} \times \text{Tr}} \times \text{Motor\_Position}[IU]$$

For linear motors: $$\text{Load\_Position}[SI] = \frac{\text{Pole\_Pitch}}{\text{Tr}} \times \text{Motor\_Position}[IU]$$

where:

resolution – is the motor position resolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

Pole_Pitch – is the magnetic pole pitch NN (distance expressed in [m])

6.1.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal position units are encoder counts. The motor is rotary and the transmission is rotary-to-rotary. The correspondence with the load position in SI units is:

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

$$\text{Load\_Position[rad]} = \frac{2 \times \pi}{4 \times \text{No\_encoder\_lines}} \times \text{Load\_Position[IU]}$$

where:

No_encoder_lines – is the encoder number of lines per revolution

6.1.4. 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:

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

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 PRO Config.

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

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** [1] is:

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

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

6.1.5. Step motor open-loop control. Incremental encoder on load

The internal position units are load encoder counts. The transmission is rotary-to-rotary. The correspondence with the load position in SI units is:

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

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

6.1.6. Brushless motor with sine/cosine encoder on motor

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

For rotary motors:

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

$$Load\_Position[SI] = \frac{2 \times \pi}{4 \times Enc\_periods \times Interpolaton \times Tr} \times Motor\_Position[IU]$$

For linear motors:

$$Load\_Position[SI] = \frac{Encoder\_accuracy}{Interpolaton \times Tr} \times Motor\_Position[IU]$$

where:

Enc_periods – is the rotary encoder number of sine/cosine periods or lines per revolution

Interpolation – is the interpolation level inside an encoder period. It is a number power of 2 between 1 and 256. 1 means no interpolation

Encoder_accuracy – is the linear encoder accuracy in [m] for one sine/cosine period

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 / DC brushed 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

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. Brushless motor with linear Hall signals

The internal speed units are counts / (slow loop sampling period). The motor is rotary. The position resolution i.e. number of counts per revolution is programmable as a power of 2 between 512 and 8192. By default it is set at 2048 counts per turn. The correspondence with the load speed in SI units is:

For rotary motors: $$Load\_Speed[SI] = \frac{2 \times \pi}{resolution \times Tr \times T} \times Motor\_Speed[IU]$$

For linear motors: $$Load\_Speed[SI] = \frac{Pole\_Pitch}{resolution \times Tr \times T} \times Motor\_Speed[IU]$$

where:

resolution – is the motor position resolution

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"

Pole_Pitch – is the magnetic pole pitch NN (distance expressed in [m])

6.2.3. DC brushed motor with quadrature encoder on load and tacho on motor

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

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

where:

No_encoder_lines – is the 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"

6.2.4. DC brushed motor with tacho on motor

When only a tachometer is mounted on the motor shaft, the internal speed units are A/D converter bits. The correspondence with the load **speed in SI units**[1] is:

$$\text{Load\_Speed}[\text{SI}] = \frac{\text{Analogue\_Input\_Range}}{4096 \times \text{Tacho\_gain} \times \text{Tr}} \times \text{Motor\_Speed}[\text{IU}]$$

where:

Analog_Input_Range – is the range of the drive analog input for feedback, expressed in [V]. You can read this value in the "Drive Info" dialogue, which can be opened from the "Drive Setup"

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

6.2.5. 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:

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

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 PRO Config.

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"

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

Step motor open-loop control. Incremental encoder on load

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

$$\text{Load\_Speed}[\text{rad/s}] = \frac{2 \times \pi}{4 \times \text{No\_encoder\_lines} \times T} \times \text{Load\_Speed}[\text{IU}]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Tr – transmission ratio between the motor displacement in [rad] and load displacement in [rad] or [m]

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”.

6.2.6. 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:

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

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”.

6.2.7. Brushless motor with sine/cosine encoder on motor

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

For rotary motors:

$$\text{Load\_Speed}[\text{SI}] = \frac{2 \times \pi}{4 \times \text{Enc\_periods} \times \text{Interpolaton} \times \text{Tr} \times T} \times \text{Motor\_Speed}[\text{IU}]$$

For linear motors:

$$\text{Load\_Speed}[\text{SI}] = \frac{\text{Encoder\_accuracy}}{\text{Interpolaton} \times \text{Tr} \times T} \times \text{Motor\_Speed}[\text{IU}]$$

where:

Enc_periods – is the rotary encoder number of sine/cosine periods or lines per revolution

Encoder_accuracy – is the linear encoder accuracy in [m] for one sine/cosine period

Interpolation – is the interpolation level inside an encoder period. It is a number power of 2 between 1 and 256. 1 means no interpolation

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

6.3. 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 / DC brushed 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:

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

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. Brushless motor with linear Hall signals

The internal acceleration units are counts / (slow loop sampling period)^2. The motor is rotary. The position resolution i.e. number of counts per revolution is programmable as a power of 2 between 512 and 8192. By default it is set at 2048 counts per turn. The correspondence with the load **acceleration in SI units**[1] is:

For rotary motors:

$$\text{Load\_Acceleration[SI]} = \frac{2 \times \pi}{\text{resolution} \times \text{Tr} \times \text{T}^2} \times \text{Motor\_Acceleration[IU]}$$

For linear motors:

$$\text{Load\_Acceleration[SI]} = \frac{\text{Pole\_Pitch}}{\text{resolution} \times \text{Tr} \times \text{T}^2} \times \text{Motor\_Acceleration[IU]}$$

where:

resolution – is the motor position resolution

Tr – transmission ratio between the motor displacement in SI units and load displacement in SI units

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

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”

Pole_Pitch – is the magnetic pole pitch NN (distance expressed in [m])

6.3.3. DC brushed motor with quadrature encoder on load and tacho on motor

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

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

where:

No_encoder_lines – is the 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”

6.3.4. 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:

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 PRO Config.

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”

6.3.5. Step motor open-loop control. Incremental encoder on load

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

For rotary-to-rotary transmission:

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

For rotary-to-linear transmission:

$$Load\_Acceleration[m/s^2] = \frac{Encoder\_accuracy}{T^2} \times Load\_Acceleration[IU]$$

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

Encoder_accuracy – is the linear encoder accuracy i.e. distance in [m] between 2 pulses

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”.

6.3.6. 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:

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

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”

6.3.7. Brushless motor with sine/cosine encoder on motor

The internal acceleration units are interpolated encoder counts / (slow loop sampling period)^2. The correspondence with the load **acceleration in SI units**[1] is:

For rotary motors:

$$\text{Load\_Acceleration}[SI] = \frac{2 \times \pi}{4 \times \text{Enc\_periods} \times \text{Interpolation} \times \text{Tr} \times T^2} \times \text{Motor\_Acceleration}[IU]$$

For linear motors:

$$\text{Load\_Acceleration}[SI] = \frac{\text{Encoder\_accuracy}}{\text{Interpolation} \times \text{Tr} \times T^2} \times \text{Motor\_Acceleration}[IU]$$

where:

Enc_periods – is the rotary encoder number of sine/cosine periods or lines per revolution

Encoder_accuracy – is the linear encoder accuracy in [m] for one sine/cosine period

Interpolation – is the interpolation level inside an encoder period. It is a number power of 2 between 1 and 256. 1 means no interpolation

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”

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

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 / DC brushed 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:

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

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. Brushless motor with linear Hall signals

The internal jerk units are counts / (slow loop sampling period)^3. The motor is rotary. The position resolution i.e. number of counts per revolution is programmable as a power of 2 between 512 and 8192. By default it is set at 2048 counts per turn. The correspondence with the load acceleration in SI units is:

For rotary motors: $\text{Load\_Jerk[SI]} = \frac{2 \times \pi}{\text{resolution} \times \text{Tr} \times \text{T}^3} \times \text{Motor\_Jerk[IU]}$

For linear motors: $\text{Load\_Jerk[SI]} = \frac{\text{Pole\_Pitch}}{\text{resolution} \times \text{Tr} \times \text{T}^3} \times \text{Motor\_Jerk[IU]}$

where:

resolution – is the motor position resolution

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”

Pole_Pitch – is the magnetic pole pitch NN (distance expressed in [m])

6.4.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal jerk units are encoder counts / (slow loop sampling period)^3. The motor is rotary and the transmission is rotary-to-rotary. 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

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

where:

No_encoder_lines – is the 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”

6.4.4. 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**[1] 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 PRO Config.

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”

6.4.5. Step motor open-loop control. Incremental encoder on load

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

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

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”.

6.4.6. 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:

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

where:

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

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".

6.4.7. Brushless motor with sine/cosine encoder on motor

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

For rotary motors: $$Load\_Jerk[SI] = \frac{2 \times \pi}{4 \times Enc\_periods \times Interpolaton \times Tr \times T^3} \times Motor\_Jerk[IU]$$

For linear motors: $$Load\_Jerk[SI] = \frac{Encoder\_accuracy}{Interpolaton \times Tr \times T^3} \times Motor\_Jerk[IU]$$

where:

Enc_periods – is the rotary encoder number of sine/cosine periods or lines per revolution
Encoder_accuracy – is the linear encoder accuracy in [m] for one sine/cosine period
Interpolation – is the interpolation level inside an encoder period. Its a number power of 2 between 1 an 256. 1 means no interpolation
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 drive 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:

$$Voltagecommand[V] = \frac{1.1 \times Vdc}{65534} \times Voltagecommand[IU]$$

where Vdc – is the drive 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:

$$\text{Voltagecommand}[V] = \frac{\text{Vdc}}{32767} \times \text{Voltagecommand}[IU]$$

This correspondence is also available for **DC brushed** motors which have the voltage command internal units as the brushless motors driven in trapezoidal mode.

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:

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

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:

$$\text{Time}[s] = T \times \text{Time}[IU]$$

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 / DC brushed 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:

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

where:

No_encoder_lines – is the rotary encoder number of lines per revolution

6.11.2. Brushless motor with linear Hall signals

The internal motor position units are counts. The motor is rotary. The resolution i.e. number of counts per revolution is programmable as a power of 2 between 512 and 8192. By default it is set at 2048 counts per turn. The correspondence with the motor position in SI units is:

For rotary motors: $$\text{Motor\_Position[SI]} = \frac{2 \times \pi}{\text{resolution}} \times \text{Motor\_Position[IU]}$$

For linear motors: $$\text{Motor\_Position[SI]} = \frac{\text{Pole\_Pitch}}{\text{resolution}} \times \text{Motor\_Position[IU]}$$

where:

resolution – is the motor position resolution

Pole_Pitch – is the magnetic pole pitch NN (distance expressed in [m])

6.11.3. DC brushed motor with quadrature encoder on load and tacho on motor

The motor position is not computed.

6.11.4. 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:

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

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 PRO Config.

6.11.5. Step motor open-loop control. Incremental encoder on load

In open-loop control configurations with incremental encoder on load, the motor position is not computed.

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

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

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

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

where:

No_encoder_lines – is the motor encoder number of lines per revolution

6.11.7. Brushless motor with sine/cosine encoder on motor

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

For rotary motors:

$$Motor\_Position[SI] = \frac{2 \times \pi}{4 \times Enc\_periods \times Interpolaton} \times Motor\_Position[IU]$$

For linear motors:

$$Motor\_Position[SI] = \frac{Encoder\_accuracy}{Interpolaton} \times Motor\_Position[IU]$$

where:

Enc_periods – is the rotary encoder number of sine/cosine periods or lines per revolution

Interpolation – is the interpolation level inside an encoder period. It is a number power of 2 between 1 and 256. 1 means no interpolation

Encoder_accuracy – is the linear encoder accuracy in [m] for one sine/cosine period

6.12.Motor speed units

6.12.1. Brushless / DC brushed 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. Brushless motor with linear Hall signals

The internal motor speed units are counts / (slow loop sampling period). The motor is rotary. The position resolution i.e. number of counts per revolution is programmable as a power of 2 between 512 and 8192. By default it is set at 2048 counts per turn. The correspondence with the motor speed in SI units is:

For rotary motors: $$Motor\_Speed[SI] = \frac{2 \times \pi}{resolution \times T} \times Motor\_Speed[IU]$$

For linear motors: $Motor\_Speed[SI] = \frac{Pole\_Pitch}{resolution \times T} \times Motor\_Speed[IU]$

where:

resolution – is the motor position resolution

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”

Pole_Pitch – is the magnetic pole pitch NN (distance expressed in [m])

6.12.3. DC brushed motor with quadrature encoder on load and tacho on motor

The internal motor speed units are A/D converter bits. The correspondence with the motor **speed in SI units**[1] is:

$$Motor\_Speed[SI] = \frac{Analogue\_Input\_Range}{4096 \times Tacho\_gain} \times Motor\_Speed[IU]$$

where:

Analog_Input_Range – is the range of the drive analog input for feedback, expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

6.12.4. DC brushed motor with tacho on motor

The internal motor speed units are A/D converter bits. The correspondence with the motor speed in SI units is:

$$Motor\_Speed[SI] = \frac{Analogue\_Input\_Range}{4096 \times Tacho\_gain} \times Motor\_Speed[IU]$$

where:

Analog_Input_Range – is the range of the drive analog input for feedback, expressed in [V]. You can read this value in the “Drive Info” dialogue, which can be opened from the “Drive Setup”

Tacho_gain – is the tachometer gain expressed in [V/rad/s]

6.12.5. Step motor open-loop control. No feedback device or incremental encoder on load

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

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

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

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”

6.12.6. 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”.

6.12.7. Brushless motor with sine/cosine encoder on motor

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

For rotary motors:

$$Motor\_Speed[SI] = \frac{2 \times \pi}{4 \times Enc\_periods \times Interpolaton \times T} \times Motor\_Speed[IU]$$

For linear motors:

$$Motor\_Speed[SI] = \frac{Encoder\_accuracy}{Interpolaton \times T} \times Motor\_Speed[IU]$$

where:

Enc_periods – is the rotary encoder number of sine/cosine periods or lines per revolution

Encoder_accuracy – is the linear encoder accuracy in [m] for one sine/cosine period

Interpolation – is the interpolation level inside an encoder period. It is a number power of 2 between 1 and 256. 1 means no interpolation

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”

7. Memory Map

PRO-Ax0V80x-SA-CAN has 2 types of memory available for user applications: 16K×16 SRAM and up to 16K×16 serial E^2ROM.
The SRAM memory is mapped in the address range: C000h to FFFFh. 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 7FFFh. It is used to keep in a non-volatile memory the MPL programs, the cam tables and the drive setup information.
Remark: *MotionPRO Developer handles automatically the memory allocation for each motion application. The memory map can be accessed and modified from the main folder of each application*

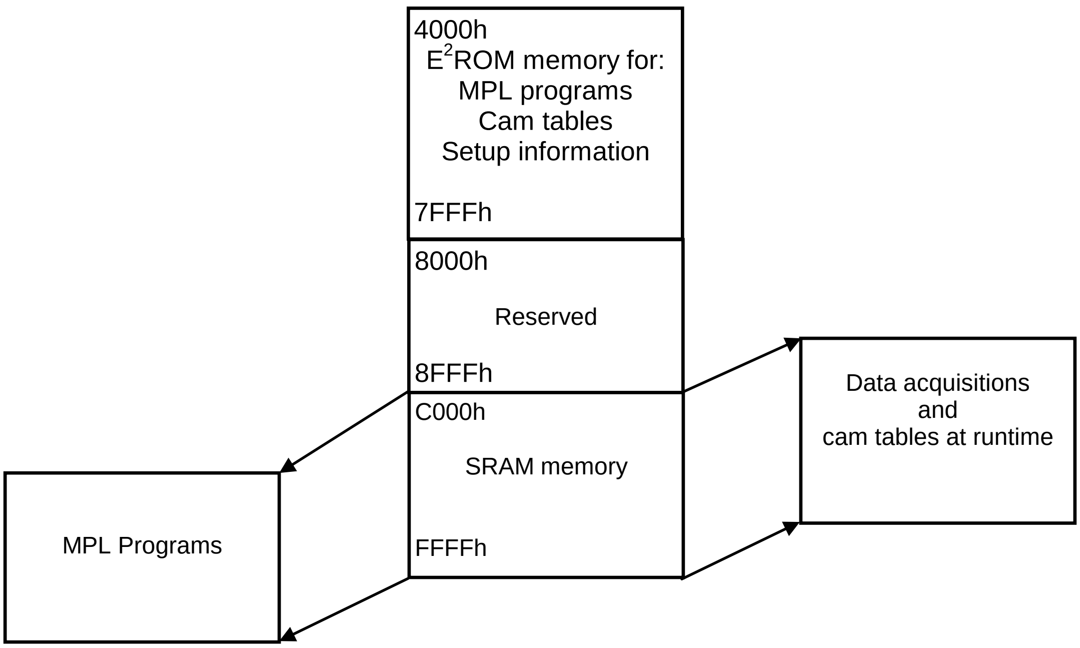


Figure 7.1. *PRO-Ax0V80x-SA-CAN Memory Map*

This page is empty

ElectroCraft®
powering innovation™