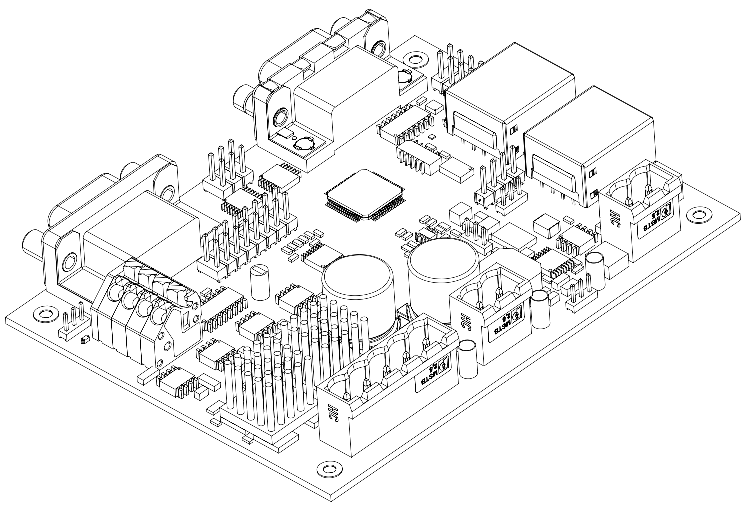


Juno™
MC7x112 Developer Kit

User Manual

Revision 1.4/ March 2026

Performance Motion Devices, Inc.
80 Central Street, Boxborough, MA 01719
www.pmdcorp.com

NOTICE

Warranty

Performance Motion Devices, Inc. warrants that its products shall substantially comply with the specifications applicable at the time of sale, provided that this warranty does not extend to any use of any Performance Motion Devices, Inc. product in an Unauthorized Application (as defined below). Except as specifically provided in this paragraph, each Performance Motion Devices, Inc. product is provided “as is” and without warranty of any type, including without limitation implied warranties of merchantability and fitness for any particular purpose.

Performance Motion Devices, Inc. reserves the right to modify its products, and to discontinue any product or service, without notice and advises customers to obtain the latest version of relevant information (including without limitation product specifications) before placing orders to verify the performance capabilities of the products being purchased. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgment, including those pertaining to warranty, patent infringement and limitation of liability.

Unauthorized Applications

Performance Motion Devices, Inc. products are not designed, approved or warranted for use in any application where failure of the Performance Motion Devices, Inc. product could result in death, personal injury or significant property or environmental damage (each, an “Unauthorized Application”). By way of example and not limitation, a life support system, an aircraft control system and a motor vehicle control system would all be considered “Unauthorized Applications” and use of a Performance Motion Devices, Inc. product in such a system would not be warranted or approved by Performance Motion Devices, Inc.

By using any Performance Motion Devices, Inc. product in connection with an Unauthorized Application, the customer agrees to defend, indemnify and hold harmless Performance Motion Devices, Inc., its officers, directors, employees and agents, from and against any and all claims, losses, liabilities, damages, costs and expenses, including without limitation reasonable attorneys’ fees, (collectively, “Damages”) arising out of or relating to such use, including without limitation any Damages arising out of the failure of the Performance Motion Devices, Inc. product to conform to specifications.

In order to minimize risks associated with the customer’s applications, adequate design and operating safeguards must be provided by the customer to minimize inherent procedural hazards.

Disclaimer

Performance Motion Devices, Inc. assumes no liability for applications assistance or customer product design. Performance Motion Devices, Inc. does not warrant or represent that any license, either express or implied, is granted under any patent right, copyright, mask work right, or other intellectual property right of Performance Motion Devices, Inc. covering or relating to any combination, machine, or process in which such products or services might be or are used. Performance Motion Devices, Inc.’s publication of information regarding any third party’s products or services does not constitute Performance Motion Devices, Inc.’s approval, warranty or endorsement thereof.

Patents

Performance Motion Devices, Inc. may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Patents and/or pending patent applications of Performance Motion Devices, Inc. are listed at https://www.pmdcorp.com/company/patents.

Related Documents

MC7x112 Torque Control IC User Guide

Complete description of the MC71112, MC71112N, MC73112 and MC73112N Juno torque control ICs including electrical characteristics, pin descriptions, and theory of operations.

Juno Velocity & Torque Control IC Programming Reference

Description of all Juno family IC host commands, with coding syntax and examples, listed alphabetically for quick reference.

Table of Contents

List of Figures

This page intentionally left blank.

1. Installation

1

In This Chapter

- Introduction
- DK Boards
- Installed MC7x112 IC Version
- Developer Kit Accessories
- Guide to this Manual
- Software Installation
- Recommended Hardware

1.1 Introduction

The PMD MC7x112 Developer Kit is an integrated board/software package that serves as an electrical and software design tool for building systems that use MC7x112 torque control ICs.

There are four different MC7x112 developer kit part numbers depending on the supported motor type and IC package (56-pin VQFN or 64pin TQFP). These four part numbers utilize two different boards; the DK73112N board which supports 56-pin package ICs and the DK78113 board which supports 64-pin package ICs:

Developer Kit P/N	Installed IC	Motor Supported	Board
DK71112N	MC71112N (56-pin VQFN)	DC Brush	DK73112N
DK73112N	MC73112N (56-pin VQFN)	Brushless DC	DK73112N
DK71112	MC71112 (64-pin TQFP)	DC Brush	DK78113
DK73112	MC73112 (64-pin TQFP)	Brushless DC	DK78113

Note that throughout this manual the term MC7x112 may be used to mean all members of the Juno Torque Control IC family including the MC71112N, MC73112N, MC71112, and MC73112. The term MC7x112 Developer Kit or MC7x112 DK may be used to mean the DK71112N, DK73112N, DK71112, and DK73112 developer kits.

1.2 DK Boards

1.2.1 DK73112N Board

The DK73112N printed circuit board contains interface and amplifier circuitry to allow various features of the MC71112N and MC73112N ICs to be accessed. Here is a summary of the features provided by the DK73112N board:

- Supports 56-pin VQFN package MC7x112 torque control ICs
- High performance on-board amplifier with current feedback
- Supports DC Brush and Brushless DC motors
- 3-pin serial UART host communications interface
- Easy to use terminal screw style connectors
- Single DC-voltage supply (12-48V)
- Quadrature encoder signal input with Index capture

- Overtemperature, overcurrent, over and undervoltage sense
- Compact (2.4" x 1.9") standalone form factor (6.1 cm x 4.8 cm)

1.2.2 DK78113 Board

The DK78113 printed circuit board contains interface and amplifier circuitry to allow various features of the MC71112 and MC73112 ICs to be accessed. Here is a summary of the features provided by the DK78113 board:

- Supports 64-pin TQFP Package MC7x112 torque control ICs
- High performance on-board amplifier with current feedback
- Supports DC Brush and Brushless DC motors
- DB9 RS232 host communications interface
- Single DC-voltage supply (12-56V)
- Quadrature encoder signal input with Index capture
- Overtemperature, overcurrent, over and undervoltage sense
- Supports on-board as well as off-board amplifier
- On-board temperature sensor
- 4.1" x 3.2" standalone form factor (10.4 cm x 8.1 cm)

1.2.3 DK73112N Versus DK78113 Board

The functions provided by the 56-pin package MC7x112N torque control ICs are identical to the functions provided by the 64-pin MC7x112 torque control ICs. The associated developer kit boards however, as indicated in the lists above, have some differences in the functions they provide.

In particular the DK78113 board supports an RS232 interface with a DB9 connector and the ability to support a remote amplifier. Support for an RS232 interface (rather than the UART interface provided by the DK73112N board) may provide more reliable communications particularly at higher levels of motor command current output. Support for a remote amplifier provides the possibility of prototyping MC7x112 ICs with user-designed higher power or alternate architecture amplifiers. In addition the DK78113 board provides a slightly higher HV input voltage of 12-56V versus 12-48V, and the DK78113 has an on-board temperature sensor whereas the DK73112N does not.

1.3 Installed MC7x112 IC Version

MC7x112 ICs, in addition to the base part number which is listed in the table in Section 1.1, “Introduction” are also identified with a version number. Version numbers represent revisions of the base part number, generally for the purpose of implementing functional corrections. This version information appears as digits or characters after the base part number.

To check the version of the installed MC7x112 family IC in your developer kit board view the Project window from Pro-Motion. The full part number will be listed after “Model” when Pro-Motion is connected to the developer kit board you are using. When you later order MC7x112 ICs for development or when the board goes into production it is recommended to order the same version. For more information on Pro-Motion see Chapter 3, *Going Further with Pro-Motion*.

You can order specific motion IC part numbers including the version directly from PMD. If you order PMD motion ICs from outlets such as Digikey or Mouser generally only one version is available and therefore you should be aware that the version number may be different. The version should be indicated in the description of the motion IC on those websites.

1.4 Developer Kit Accessories

1.4.1 DK73112N Board

In addition to the DK73112N board the DK71112N and DK73112N developer kits come with the following accessories:

PMD Component P/N	Description
Cable-USB-3P	UART serial cable. This cable connects the DK73112N's 3-pin serial UART port to a PC USB port.

1.4.2 DK78113 Board

In addition to the DK78113 board the DK71112 and DK73112 developer kits come with the following accessories:

PMD Component P/N	Description
Cable-USB-DB9	RS232 serial cable. This cable connects the DK78113's DB9 serial port to a PC USB port.
MC-HW-05	DB-15 breakout connector. This item provides terminal screw connections for the 15-pin axis connector.
CONN-0122-11	Power & shunt connector. This two-pin plug provides terminal screw connections for the power & shunt resistor connectors. Two connectors are included.
CONN-0121-11	Motor connector. This five-pin plug provides terminal screw connections for the motor connector.

1.5 Guide to this Manual

This manual is designed to help get your motion hardware setup connected and running quickly. In addition, this manual shows how to continue with development of your application by further exercising the connected motor hardware, and optimizing the motion system control parameters.

Here is a summary of the content in the remaining chapters of this manual:

Chapter 2, *Quick Start Guide*, provides instructions on connecting and verifying proper functioning of your motor with the MC7x112 DK.

Chapter 3, *Going Further with Pro-Motion*, describes the most frequently used features of Pro-Motion. You will find the content in this chapter helpful if you are new to Pro-Motion and PMD's MC7x112 ICs.

Chapter 4, *Operations Guide*, provides information on how various common and useful operations can be achieved using the MC7x112 developer kit boards and Pro-Motion.

Chapter 5, *DK73112N Board Reference*, is a complete functional reference for the DK73112N board including description of major functional elements and complete connector reference.

Chapter 6, *DK78113 Board Reference*, is a complete functional reference for the DK78113 board including description of major functional elements and complete connector reference.

Chapter 7, *Software Development*, provides an overview of how to write software for controlling PMD products generally, and MC7x112 ICs specifically.

Appendix A, *DK73112N Board Schematics*, provides complete schematics for the DK73112N board.

Appendix B, *DK78113 Board Schematics*, provides complete schematics for the DK78113 board.

1.6 Software Installation

The software distribution for the MC7x112 Developer Kit is downloaded from the PMD website at the URL: https://www.pmdcorp.com/resources/software.

All software applications are designed to work with Microsoft Windows.

To install the software:

1. Go to the Software Downloads section of PMD's website located at https://www.pmdcorp.com/resources/software and select download for "Developer Kit Software".
2. After selecting download you will be prompted to register your DK, providing the serial # for the DK and other information about you and your motion application.
3. After selecting submit the next screen will provide a link to the software download. The software download is a zip file containing various installation programs. Select this link and downloading will begin.
4. Once the download is complete extract the zip file. There is a *ReadMe.txt* file that may contain additional useful information. When ready execute the **Pro-Motion** install. Pro-Motion is a Windows application that will be used to communicate with and exercise your developer kit.
5. You can also extract one or both of the following SDKs (Software Development Kits) used with MC7x112 ICs.
   - **C-Motion Magellan SDK** – an SDK for creating C-language user applications for PMD products that utilize a Magellan/Juno formatted protocol
   - **C-Motion PRP SDK** – An SDK for creating PC-based applications using .NET (C#, VB) programming languages, and for creating C-language user applications for PMD products that utilize a Magellan/Juno or PRP (PMD Resource Access Protocol) formatted protocol

The next few sections give more information on each of these items.

1.6.1 Pro-Motion

Pro-Motion is a sophisticated, easy-to-use exerciser program which allows all MC7x112 IC parameters to be set and/or viewed, and allows all features to be exercised. Pro-Motion features include:

- Motion oscilloscope graphically displays axis parameters in real-time
- Axis Wizard to automate axis setup and configuration
- Project window for accessing motion resources and connections
- Ability to save and load settings
- Distance, time, and electrical units conversion
- Frequency sweep and bode plot analysis tools

For more information on Pro-Motion see Chapter 3, *Going Further with Pro-Motion*.

1.6.2 C-Motion

MC7x1112 torque control ICs can be used with direct digital or analog signals to command the desired torque value. Alternatively, the MC7x1112's host serial port may be used to command torque values. This section and the next section are generally only useful where serial host torque commands are delivered from user-developed code running on a microcontroller. For more information on options for providing the torque command refer to the *MC7x112 Torque Control IC User Guide*

C-Motion provides a convenient set of C language callable routines for controlling MC7x112 ICs. C-Motion includes the following features:

- Axis virtualization
- Ability to communicate to multiple PMD ICs or modules
- Ability to communicate via serial, CANbus, Ethernet, SPI (Serial Peripheral Interface), or 8/16 bit parallel bus
- Provided as source code, allowing easy compilation & porting onto various run-time environments including a PC, microprocessor, embedded board, or C-Motion Engine

There are three different versions of C-Motion; C-Motion Magellan, C-Motion PRP, and C-Motion PRP II. C-Motion Magellan is used with PMD products that utilize a direct Magellan or Juno formatted protocol. C-Motion PRP is used in systems that support both Magellan/Juno protocol devices and PRP (PMD Resource Access Protocol) protocol devices. C-Motion PRP is also used in motion applications that will use the .NET (C#, VB) programming languages. C-Motion PRP II is used with ION/CME N-Series Digital Drives.

For more information on C-Motion see Chapter 7, *Software Development*.

1.6.3 .NET Language Support

A complete set of methods and properties is provided for developing applications in Visual Basic and C# using a dynamically loaded library (DLL) containing PMD library software. The DLL may also be used from any language capable of calling C language DLL procedures, such as Labview, but no special software support is provided.

Includes the following features:

- Axis virtualization
- Ability to communicate to multiple PMD ICs or modules
- Ability to communicate via serial, CAN, Ethernet, SPI, or 8/16-bit parallel bus
- Provided as a single DLL and Visual Basic .NET source code for easy porting onto various PC environments

1.7 Recommended Hardware

To install either the DK73112N or DK78113 board the following hardware is recommended.

- PC with Intel (or compatible) processor, 1 Gbyte of available disk space, and 256 MB of available RAM. The supported PC operating system is Windows.
- DC Brush or Brushless DC motor according to the developer kit purchased. Encoder feedback is not necessary for DC Brush motors, and is optional with Brushless DC motors.
- Cables as required to connect to the motor and associated motion hardware.
- DC power supply with a voltage from 12-48 volts (DK73112N board) or 12-56 volts (DK78113 board). Only a single voltage supply is required. The board logic and other circuitry is powered from this input voltage using an on-board DC to DC converter.

This page intentionally left blank.

2. Quick Start Guide

2

In This Chapter

- Step #1 — Making Motion Hardware Connections
- Step #2 — Applying Power
- Step #3 — First-Time System Verification
- Next Steps

Here are the steps you will execute to set up either the DK73112N or DK78113 board so that it can control your motor or actuator successfully. If you haven't already installed the software you should do this now. See Section 1.6, "Software Installation" for instructions on installing the software.

Step 1 The first step is to connect your system's motion peripherals, motor, power supply, and PC to the DK board. See Section 2.1, "Step #1 — Making Motion Hardware Connections" for details.

Step 2 Next you will provide the board with power. See Section 2.2, "Step #2 — Applying Power" for more information.

Step 3 The final step to is to perform a functional test of the finished system. See Section 2.3, "Step #3 — First-Time System Verification" for a description of this procedure.

Once these steps have been accomplished setup is complete and the board and connected motion hardware are ready for operation.

2.1 Step #1 — Making Motion Hardware Connections

The next few sections detail how to make the needed connections between the DK board you are using and your motion hardware, PC, and power supply. Since there are two different MC7x112 DK boards the instructions for making motion hardware connections are split into two sections. The instructions for the DK73112N board begin below, and the instructions for the DK78113 board begin at Section 2.1.2, "DK78113 Board Connections."

2.1.1 DK73112N Board Connections

Figure 2-1 shows the location of the principle components of the DK73112N board.

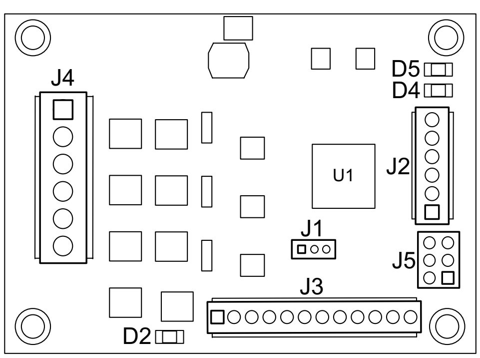


Figure 2-1: DK73112N Board Components Location

The following table identifies these components.

Label	Description
J4	HV Power & Motor Connector
J3	Feedback Connector
J2	I/O Signals Connector
J1	Serial Connector
J5	SPI Connector
D2, D4, D5	Amplifier Enable, Fault, and power LED indicators (respectively)

Note that not all of these connectors are used in this quick start guide sequence.

2.1.1.1 Encoder & Hall Connections

The following table summarizes encoder and Hall signal connections to the DK73112N board. All connections are made through the Feedback Connector (J3) which is a terminal screw style connector.

Encoders are optional with Brushless DC motors. If used, they enable encoder-based (sinusoidal) commutation which generally provides smoother motion than Hall-based commutation. Encoders are not used with DC Brush motors.

Quadrature encoders can use a differential wiring scheme where each signal (QuadA, QuadB, and Index) uses a positive (+) and negative (–) connection, or they can use a single-ended scheme with a single connection per signal. If available, use of differential connections is highly recommended. If single-ended encoders are used they should be connected to the positive (+) differential signal, and the negative (–) inputs may be left unconnected. In addition to QuadA, QuadB, and Index signals encoders also typically require 5V and GND connections.

Hall signals are used with Brushless DC motors only. Although they are not required, they should be used if available.

The following table shows the Feedback Connector (J3) connections.

Pin #	Signal Name	Description
	J3 - Feedback Connector	
1	+5V	+5V output
2	GND	Ground
3	QuadA+	Quadrature A+ encoder input
4	QuadA–	Quadrature A– encoder input
5	QuadB+	Quadrature B+ encoder input
6	QuadB–	Quadrature B– encoder input
7	Index+	Index+ input
8	Index–	Index– input
9	HallA	HallA signal input
10	HallB	HallB signal input
11	HallC	HallC signal input
12	GND	Ground

2.1.1.2 HV Power & Motor Coil Connections

The HV voltage is the voltage at which the motor will be driven and must be in the range of +12V to +48V. Use the HV and GND connections below to provide HV power to the board. This HV connection is also the power connection from which the board logic power is derived using an on-board DC-DC converter. The A, B, and C connections shown below are used to connect the DK73112N to the motor windings. For Brushless DC motors all three motor outputs are used, for DC Brush motors only Motor A and Motor B are used.

All of these connections are made via the HV Power & Motor Connector (J4).

Pin #	Signal Name	Description
	J4 - HV Power & Motor Connector	
1	HV	Positive motor voltage power
2	GND	Ground
3	Motor A	Motor coil A
4	Motor B	Motor coil B
5	Motor C	Motor coil C
6	Shunt	Shunt output

Note that although there isn't a separate Shield connection on the J4 Connector, the GND signal may be used for a shield connection to the motor. Motor shield connections are highly recommended to reduce EMI or other types of electrical noise.

Note that pin 6, Shunt,is not used in this quick start sequence.

2.1.1.3 Communication Connections

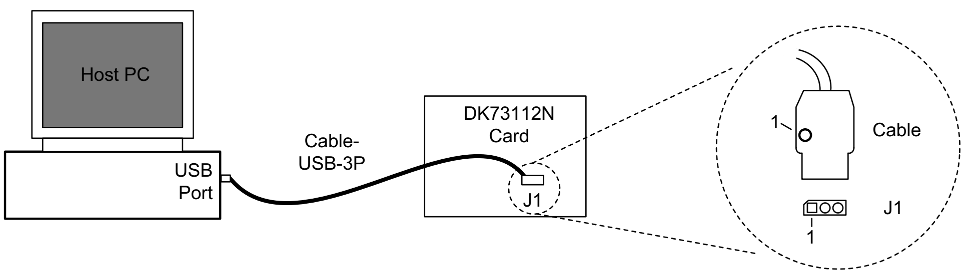


Figure 2-2: PC to DK73112N Host Serial Connection

Communication to the board is via a serial UART port.

A 3-pin cable for this purpose is included with the DK73112N. This cable (PMD p/n Cable-USB-3P) should be connected to the DK73112N board's J1 Serial Connector, while the opposite end of the cable should be connected to one of the PC's USB ports. When installing the cable at the DK board make sure the pin 1 orientation is correct. Refer to Figure 2-2 to determine the pin 1 location on the DK73112N board. Pin 1 on Cable-USB-3P is locatable via a small dot at the pin 1 location.

2.1.1.4 Enable Connection

The DK73112N requires an active Enable signal to operate. To accomplish this the I/O Signals Connector (J2), which is a terminal screw style connector, is used. Connect terminal #1 of J2 (Enable) to terminal #4 of J2 (GND) using a short wire.

For reference the following table provides the pinouts of the J2 connector:

Pin #	Signal Name	Description
	J2 - I/O Signals Connector	
1	Enable	Active low Enable digital input signal
2	FaultOut	Active low digital FaultOut output signal
3	Brake	Brake input signal
4	GND	Ground
5	AnalogCmd–	AnalogCmd– torque command input
6	AnalogCmd+	AnalogCmd+ torque command input

Note that the 2, 3, 5, and 6 pin connections above are not used in this quick start sequence.

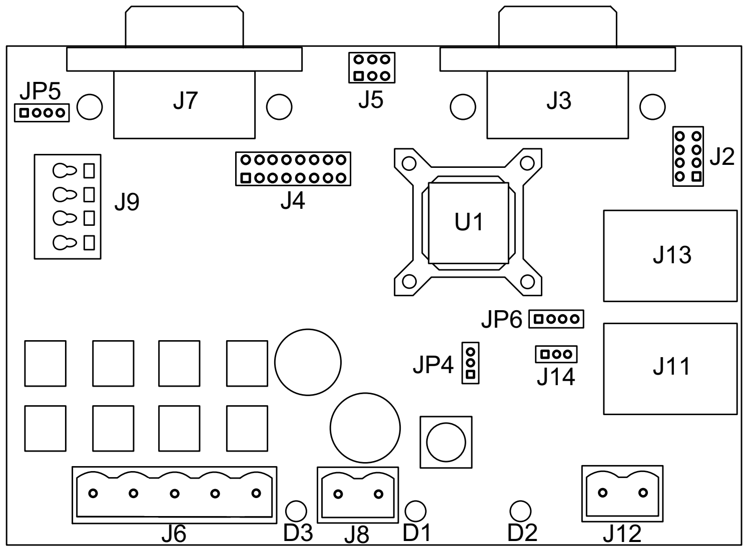
This completes step #1 of the quick start sequence for the DK73112N. To continue to step #2 go to Section 2.2, "Step #2 — Applying Power."

2.1.2 DK78113 Board Connections

The next few sections detail how to make the needed connections between the DK78113 board and your motion hardware, PC, and power supply.

Figure 2-3 shows the location of the principle components of the DK78113 board.

Figure 2-3: DK78113 Board Components Location



The following table identifies these components:

Label	Description
J8	HV Power Connector
J6	Motor Drive Connector
J7	Feedback Connector
J9	I/O Signals Connector
J4	Remote Amplifier Connector
J5	Analog Command Connector
J12	Shunt Connector
J3	RS232 Connector
J2	Host SPI, Pulse & Direction Connector
J14	Serial Connector
D1, D2, D3	Power, fault, and amplifier enabled LED indicators (respectively)
JP4, JP5	Remote amplifier enable jumpers

Note that not all of these connectors or components are used in this quick start sequence.

2.1.2.1 Encoder & Hall Connections

The following table summarizes encoder and Hall signal connections to the DK78113 board. All connections are made through the Feedback Connector (J7), which is a high density female DB-15.

Although you can create your own DB-15 cable to connect these signal wires, most users will use the DB-15 terminal screw breakout board included with the developer kit for connection of these signals.

Encoders are optional with Brushless DC motors. If used, they enable encoder-based (sinusoidal) commutation which generally provides smoother motion than Hall-based commutation. Encoders are not used with DC Brush motors.

Quadrature encoders can use a differential wiring scheme where each signal (QuadA, QuadB, and Index) uses a positive (+) and negative (–) connection, or they can use a single-ended scheme with a single connection per signal. If available use of differential connections is highly recommended. If single-ended encoders are used they should be connected to the positive (+) differential signal, and the negative (–) signals may be left unconnected. In addition to QuadA, QuadB, and Index signals encoders also typically require 5V and GND connections.

Hall signals are used with Brushless DC motors only. Although they are not required, they should be used if available.

Pin #	Signal Name	Description
	J7 - Axis Feedback Connector	
1	QuadA+	A+ quadrature input
2	QuadA–	A– quadrature input
3	QuadB+	B+ quadrature input
4	QuadB–	B– quadrature input
5	GND	Ground
6	Index+	Index+ quadrature input
7	Index–	Index– quadrature input
8	HallA	HallA signal input
9	HallB	HallB signal input
10	HallC	HallC signal input
11-13	NC	No Connect
14	+5V	+5V output
15	NC	No Connect

2.1.2.2 HV Power & Motor Coil Connections

The following table summarizes the connections from the DK78113 to your power supply made via the J8 HV Power Connector. The HV voltage is the voltage at which the motor is driven and must be in the range of 12V - 56V. This HV connection is also the power connection from which the board logic power is derived using an on-board DC-DC converter.

The HV Power Connector is a Phoenix Contact 2-circuit terminal block connector.

Pin #	Signal Name	Description
	J8 - HV Power Connector	
1	HV	Positive motor voltage power
2	GND	Motor voltage power ground

The following table summarizes the motor drive connections from the DK78113 to the motor made via the J6 Motor Drive Connector. There are three motor drive connections and a shield connection. For Brushless DC motors all three motor outputs are used, and for DC Brush motors only MotorA and MotorB are used.

The J6 Motor Drive Connector is a Phoenix Contact 5-circuit terminal block connector.

Pin #	Signal Name	Description
	J6 - Motor Drive Connector	
1	MotorA	Motor coil A
2	MotorB	Motor coil B
3	MotorC	Motor coil C
5	Case/shield	Connection to motor case/shield. A shield connection is strongly recommended for most motor setups

> Shield connections to the motor are strongly recommended. Not connecting the shield signal may result in increased EMI (electromagnetic interference), reduced immunity to ESD (electro static discharge), or electrical noise resulting in motor operation failure.

2.1.2.3 Communication Connections

An RS232 serial cable is included with the DK78113. This serial cable (PMD p/n Cable-USB-DB9) should be connected to the DK78113 board's J3 RS232 Connector, while the opposite end of the cable should be connected to one of the PC's USB ports.

2.1.2.4 Enable Connection

MC7x112 ICs require an active Enable signal to operate. To accomplish this the I/O Signals Connector (J9) is used. Connect terminal #2 of J9 (Enable) to terminal #4 of J9 (GND) using a short wire. J9 provides push-type connections, so no other hardware is needed to make this connection.

For reference the following table provides the pinouts of the J9 connector:

Pin #	Signal Name	Description
	J9 - I/O Signals Connector	
1	Brake	Active low Brake signal input to MC78113
2	Enable	Enable input. Must be tied low (GND) to enable the MC78113 for full operation.
3	FaultOut	Active high FaultOut signal output from MC78113.
4	GND	Digital ground

Note that the 1 and 3 pin connections above are not used in this quick start sequence.

2.2 Step #2 — Applying Power

Once you have made your motion hardware, power, and communication connections, hardware installation is complete and the board is ready for operation. When power is applied, the DK board's green power LED should light. This LED is locatable using Figure 2-1 or Figure 2-3 depending on which DK board you are using. If the LED does not light, recheck connections.

After power up no motor output will be applied. Therefore the motors should remain stationary. If the motors move or jump, power down the board and check the motor and encoder connections. If anomalous behavior is still observed, call PMD or your PMD representative for assistance.

2.3 Step #3 — First-Time System Verification

Here is a summary of the steps that will be used during first time system verification.

1. Initiate Pro-Motion and establish serial communication between the PC and the board using the host communication link.
2. Run Pro-Motion's Axis wizard to initialize parameters such as encoder direction and current loop parameters.
3. Manually enter torque commands to demonstrate that your motor is operating correctly.

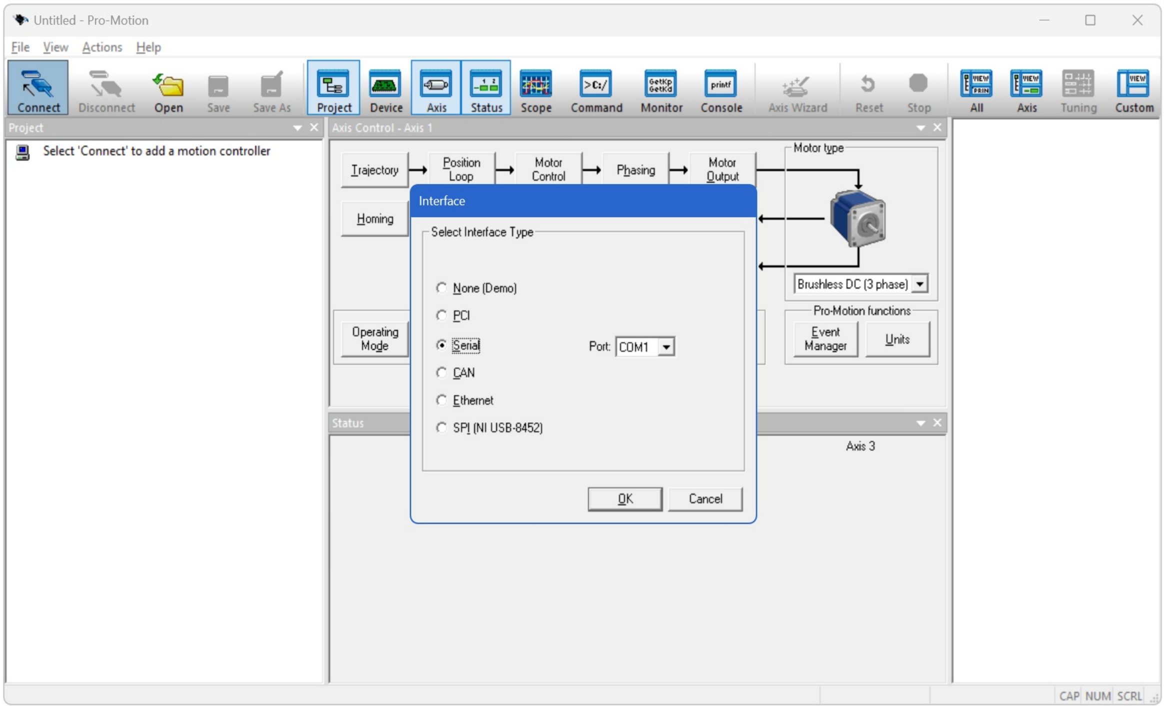
2.3.1 Establishing Serial Communications

To establish programming communications:

1. Make sure the DK board is powered and connected to the PC.
2. Launch the Pro-Motion application.

   When Pro-Motion is launched you will be prompted with an Interface selection window. A typical screen view when first launching Pro-Motion appears below.

   The purpose of the Interface dialog box is to indicate to Pro-Motion how your DK board is connected to the PC.



3. Click Serial and view the available COM ports listed in the Port field. If you know which COM port your USB serial cable is connected to, select it.

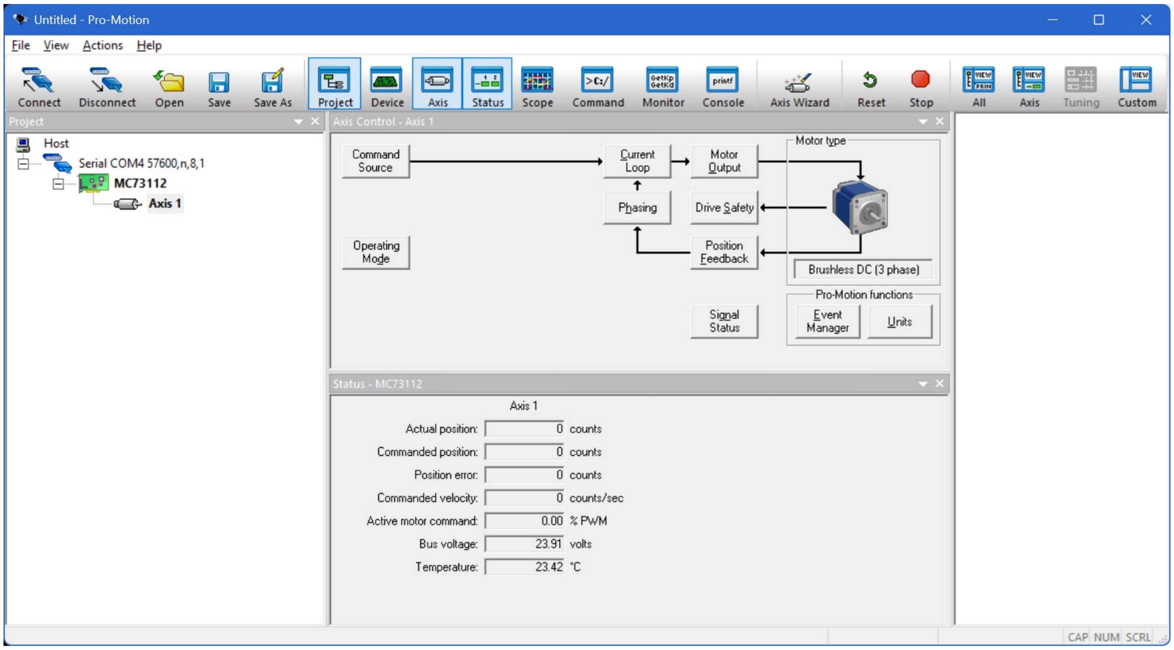
   If you are not sure which of the listed COM ports is the correct one, you may use the following procedure:

   - First, unplug the serial cable from your USB port and exit the Interface dialog box. Now re-enter the Interface dialog box by clicking "Connect" which is an icon at the far left of the top icon bar. Select Serial, and view the COM port list. Record the list of COM ports.
   - Next plug the serial cable back into the USB port and once again exit and re-enter the Interface dialog box. Select Serial and now when you view the COM port list you should see a new COM port listed. This is the COM port that is connected via the serial cable provided with the DK.
   - Select this COM port and hit the OK button.

   The Serial Port dialog box displays with default communication values of 57,600 baud, no parity, 1 stop bit, and point to point protocol.

4. Click OK without changing any of these settings.

If serial communication is correctly established, a set of object graphics loads into the Project window to the left, as shown in the following figure.



If serial communications are not correctly established, a message appears indicating that an error has occurred. If this is the case, recheck your connections and repeat from step 1.

2.3.2 Running the Axis Wizard

The next step is to initialize the axis, thereby verifying correct amplifier operation, encoder feedback connections (if an encoder is used), and other motion functions. All of this can be conveniently accomplished using Pro-Motion's Axis Wizard function.

To operate the Axis Wizard:

1 Select axis 1 in the Project window to the left of the screen.

2 Click the Axis Wizard toolbar button.

The Axis Wizard initialization window appears.

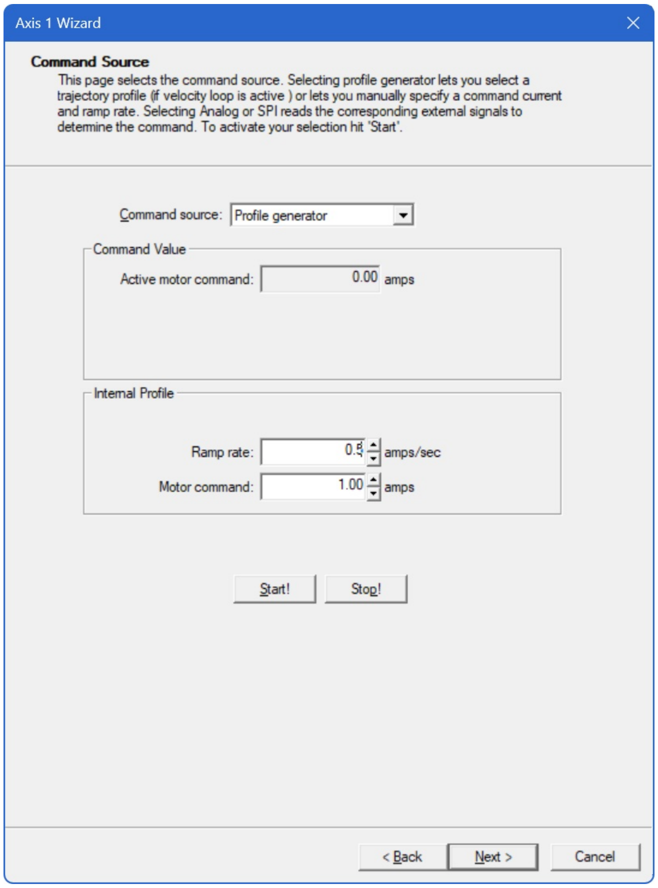
3 Click Next and follow the Axis Wizard instructions for each page of the axis initialization process.

Each screen contains instructions at the top that should guide you on how to provide the information requested, or how to use the functions provided by the screen to verify specific characteristics of your system. If you are still unsure of how to specify certain settings you can accept the default values by hitting "Next". Default settings, if available, have been chosen to work with a wide variety of motors.

4 The "Command Source" screen, which is one of the last setup screens in the wizard, provides the opportunity to command a motor torque.

2.3.3 Operating the Motor

The Command Source screen appears as follows.



To operate the Command Source screen:

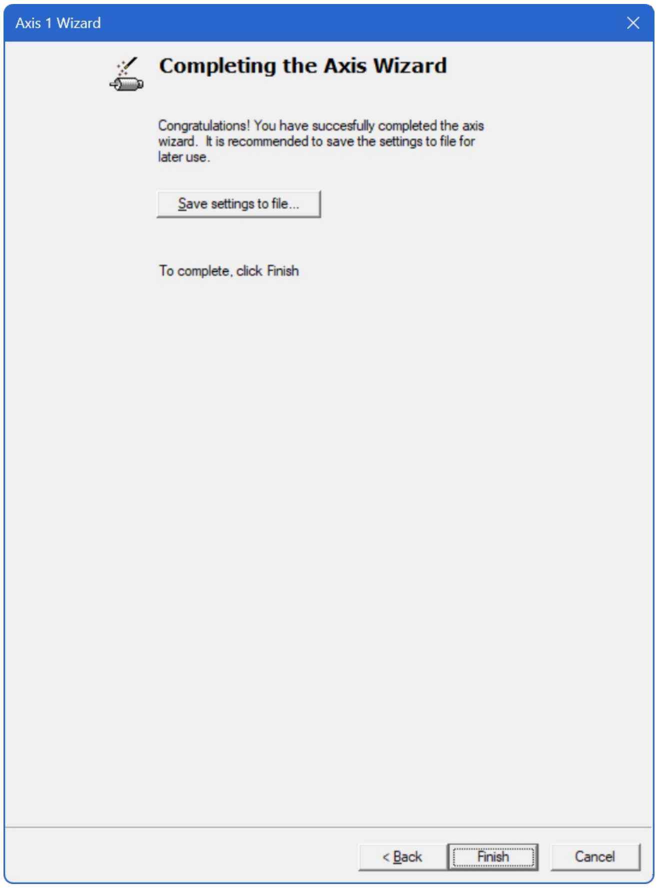
1. Set the source to profile generator.
2. Specify a Ramp rate and a Motor command (motor command here means torque command). Begin by specifying a ramp rate of perhaps 0.5 amps/sec, and a motor command of perhaps 1.0 amps. Hit "Start!" to activate the parameters you have entered. If the motor does not move increase the motor command until the motor starts spinning. You may enter positive or negative motor commands, which will result in position and negative direction motion.

   Beware that the axis velocity or position that may result from commanding the motor at a particular torque is not explicitly controlled. This generally means the motor will accelerate rather than stay at a particular velocity, bounded only by the supply voltage and the back-EMF at that supply voltage.

   If the motor doesn't move after increasing the current command or if the motion seems unstable exit the Axis Wizard by hitting "Cancel" at the bottom, exit Pro-Motion by clicking the windows "X" button at the upper right, turn off the power supply, and recheck connections.

Congratulations! First-time system verification is now complete and you have demonstrated that the motor and DK board are connected and operating correctly.

The last screen in the axis wizard allows you to save the various control parameters you have specified while in the Axis Wizard. This is convenient because it allows you to quickly load your control settings without the need to rerun the Axis Wizard.



3 To save the settings established using the Axis Wizard select "Save settings to file…" and specify a destination directory and file name. The Axis Wizard will create the file with your parameters loaded into it. For more information on saving and restoring configuration settings see Section 3.9, "Project Configuration Save & Restore."

4 When you have specified a file name and saved your settings select Finish at the bottom of the screen.

You will now be returned to the main Pro-Motion screen.

2.4 Next Steps

After first time verification is complete you may continue, in the same Pro-Motion session, to exercise the motor or change control parameters as desired. Alternatively, whether immediately or after additional interactions with Pro-Motion, you may wish to exit Pro-Motion for restart at a later time.

If you have changed control parameters and would like to save these parameters use the menu function File/Save Project As to save the current configuration to a file.

When restarting Pro-Motion it is important that you restore the configuration that you have saved. Do this by selecting File/Open Project, selecting the file previously stored.

> Operation of the DK7x112 without proper safety settings established during Axis Wizard operation may harm the DK7x112 board or the application hardware. To avoid this be sure to restore saved control settings using the File/Open Project menu item before operating the MC7x112 DK board.

Below are summaries of chapters in this manual that you will find helpful in developing your application:

Chapter 3, *Going Further with Pro-Motion,* shows you the most frequently used features of Pro-Motion. You will find the content in this chapter helpful if you are new to Pro-Motion and PMD's Magellan Motion Control ICs.

Chapter 4, *Operations Guide,* provides information on how various common operations can be achieved using Pro-Motion and the MC7x112 developer kit. Of particular interest for many users will be how to exercise the motor using either the analog signal torque command source or the digital SPI bus command source.

Chapter 5, *DK73112N Board Reference,* is a complete functional reference for the DK73112N board including description of major functional elements and complete connector reference.

Chapter 6, *DK78113 Board Reference,* is a complete functional reference for the DK78113 board including description of major functional elements and complete connector reference.

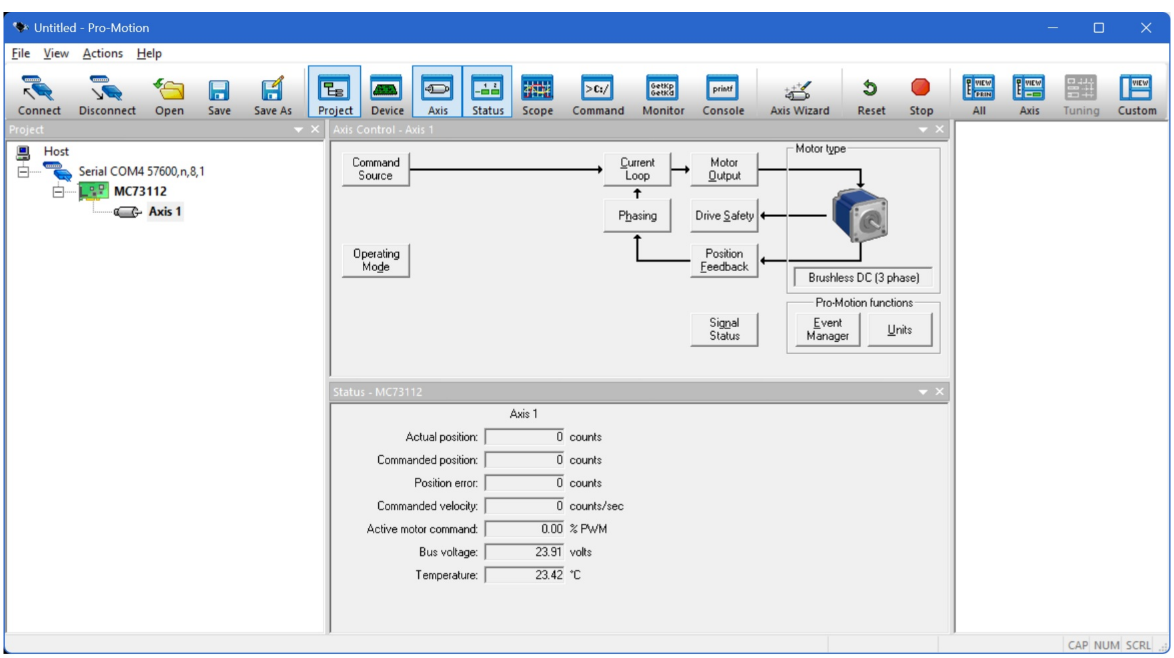
3. Going Further with Pro-Motion

In This Chapter

- Pro-Motion Screen Layout
- Project Window
- Device Control Window
- Axis Control Window
- Status Window
- Monitor Window
- Command Window
- Scope Window
- Project Configuration Save & Restore
- Configuration Export to C-Motion
- Pro-Motion Application Notes
- Troubleshooting Suggestions

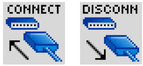
In this chapter we provide more information on Pro-Motion to help familiarize you with its most commonly used features.

3.1 Pro-Motion Screen Layout



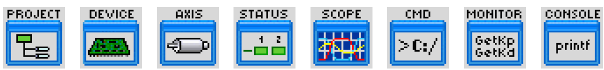
The screen capture above shows an example of Pro-Motion as it appears after first launching and connecting to a single-axis PMD controller. For information on connecting to a PMD controller after launching Pro-Motion see Section 2.3.1, "Establishing Serial Communications."

Pro-Motion follows the general form of Windows-based applications with a menu at the very top of the application window. In the case of Pro-Motion the available menu functions are *File, View, Actions*, and *Help*. In addition there is a tool bar with icons allowing the user to access the most commonly used Pro-Motion functions with a single click. Here is a description of these clickable icons in the order that they appear from left to right and grouped by function:



Connect, Disconnect
These toolbar icons let you connect or disconnect to PMD controllers in your motion hardware setup.

Open, Save, Save As
These toolbar icons let you call up and save your setup configuration. Section 3.9, "Project Configuration Save & Restore" describes these "project file' mechanisms in more detail.



Project, Device Control, Axis Control, Status, Scope, Command, Monitor, and CME Console
Each member in this group of icons represents a Pro-Motion window that, when clicked, opens if not yet being displayed or closes if being displayed. We will discuss the functions provided by many of these windows later on.

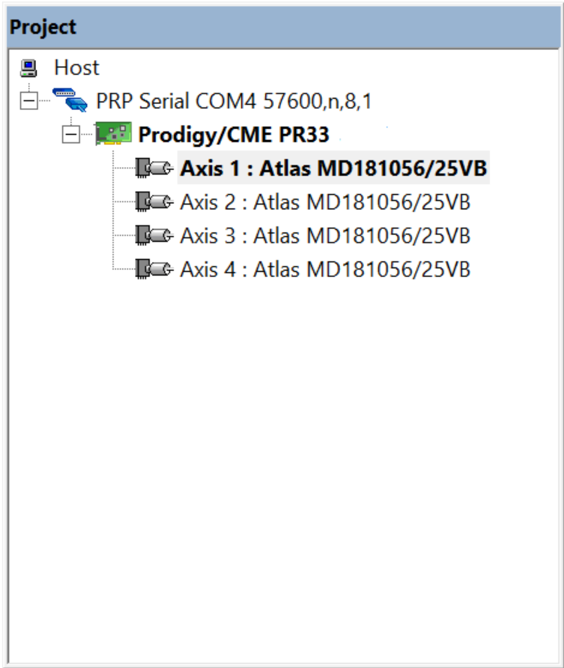
Axis Wizard
This icon starts the Axis Wizard. The Axis Wizard is the recommended way to establish and verify connections between the PMD controller and each axis of the motion hardware setup. See Section 2.3.2, "Running the Axis Wizard" for more on the Axis Wizard.

Reset, Stop
The Reset button will immediately reset the currently selected device in the Project window. The Stop button will immediately stop the motion of the axis selected in the Project window.

All, Axis, Tuning, Custom
This group of icons controls how the Pro-Motion windows are arranged. They provide convenient pre-programmed arrangements of windows designed to make specific tasks easier. The default arrangement is Axis.

Windows can also be arranged manually and saved as a custom window arrangement. To store the arrangement of windows in your Pro-Motion session use the *View/Save Custom View* menu function at the very top of the Pro-Motion screen. Thereafter, selecting the Custom icon will present the windows in this saved custom view scheme.

3.2 Project Window



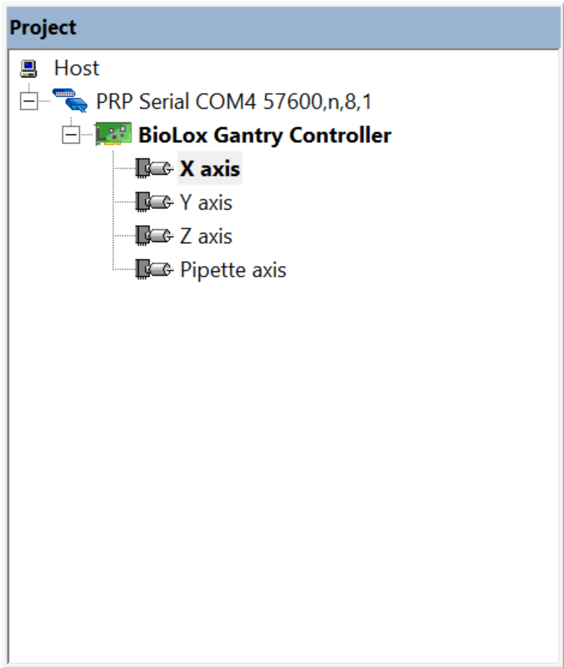
The Project window shows all PMD controllers presently connected to Pro-Motion. It presents these connections using a Windows Tree View scheme. The above screen capture shows an example Project window for a four axis controller.

In addition to displaying an icon and part number for each connected PMD controller the Project window displays a ‘motor’ icon for each axis within that controller. So a Prodigy board with four active axes would have four such motor icons displayed. Single axis devices such as a DK58113 board or N-Series ION drive would have two such icons displayed, one for the primary and one for the auxiliary encoder input. For Magellan axes that use an Atlas amplifier a special version of the motor icon is displayed and the part number of the attached Atlas is displayed next to the motor icon.

In addition to displaying the addressable PMD devices in the present Pro-Motion session, the Project window is used to select which axes are actively being processed by Pro-Motion windows. For example in the screen capture above, axis #1 is highlighted, which means axis-specific windows such as the Axis Control window will program the settings for axis #1. To change the currently selected axis simply click on the desired axis icon in the Project window.

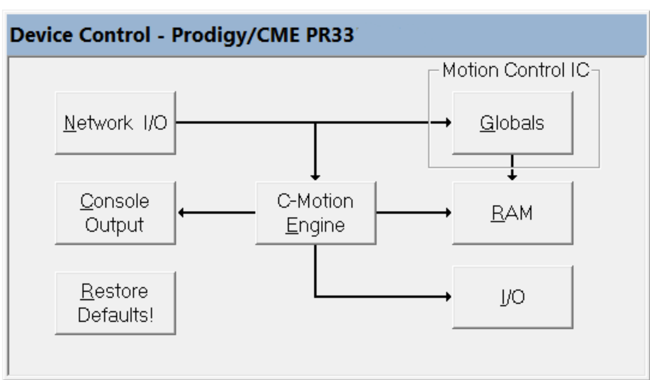
3.2.1 Device & Axis Label Customization

A very useful feature of the Project window is that the device and axis labels generated automatically by Pro-Motion can be customized to suit the application. This is shown in the screen capture below, where labels from the previous example Project window have been changed.



To change the device or axis labels slowly click twice on the existing text, and then type in your desired new entry.

3.3 Device Control Window



The Device Control window allows you to view and update ‘device level’ parameters. Devices are PMD controllers such as motion IC developer kit boards, Prodigy boards, and ION Drives. Different products have different device-level functions available and the Device Control window displays the available functions for the selected device as clickable boxes. To select a particular device from the Project window click any axis controlled by that device.

The list below briefly describes the clickable function boxes you may see in the Device Control window:

Network I/O – The Network I/O module of the Device Control window allows you to view and set various parameters for each of the communication ports supported by the connected-to device.

Motion Control IC Globals – Clicking this box lets you set the Magellan or Juno IC’s cycle time and lets you view various characteristics of the motion IC such as the IC family name, supported motor type(s), number of supported axes, and the version #.

NVRAM – Some Motion Control ICs support non-volatile memory which can contain initialization command sequences. For those products this function allows you to view, erase, and download script file content or the current Pro-Motion configuration to the Magellan IC's NVRAM.

RAM – Allows you to view the size of the PMD controller's RAM used for motion trace and User Defined Profile Mode storage. You can also change the usable size of the RAM with this function.

Analog Input – Allows you to view current reading(s) from the PMD controller's general purpose analog input function.

C-Motion Engine – This function allows you to view, erase, and download a *.bin* user code memory image to the C-Motion Engine. In addition this function allows you to manually reset, start, or stop code execution, and set whether user code begins execution automatically upon power-up or only after manual start.

Console – This function lets you specify the console communication channel and parameters used in connection with user code running on the C-Motion Engine.

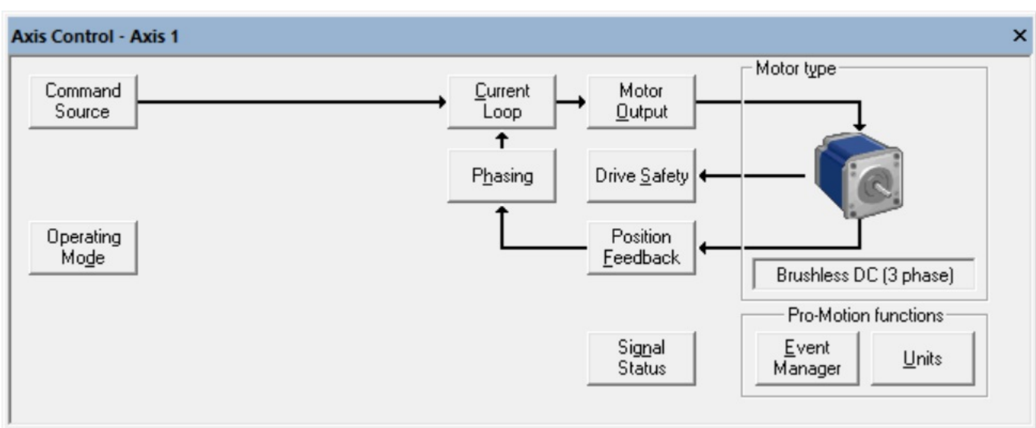
Restore Defaults – This function reverts the PMD controller's parameters to their default conditions. Any changes that may occur as a result of this operation occur in the device's NVRAM. Only after a reset or power cycle will NVRAM values become the active settings used by the PMD controller. Particular care should be taken when using this function if communication parameters have previously been altered because restoring them to their default values may mean that Pro-Motion no longer uses the correct settings to communicate with the PMD controller. For a detailed list of NVRAM-stored defaults that may be affected by this operation refer to the **DeviceSetDefault** command description in the user manual for the product you are using.

Digital I/O – This module allows device-level digital I/O registers to be read and set by the user.

The list of clickable function boxes above represents a superset of the boxes that may appear is the Device Control window for a particular PMD controller. The actual boxes displayed is based on the type of PMD controller that is selected. For example, motion IC developer kit boards such as DK78113 or DK73112N will display fewer selectable boxes while products such as ION/CME Digital Drives and Prodigy/CME boards will display more.

3.4 Axis Control Window

The Axis Control window allows you to view and update motion control parameters for the axis selected in the Project window. Each selectable box (technically these are Windows buttons) in the Axis Control window results in a dialog box being opened letting you access a sub-set of the motion control functions provided. Below is shown an example Axis Control window for MC7x112 ICs.



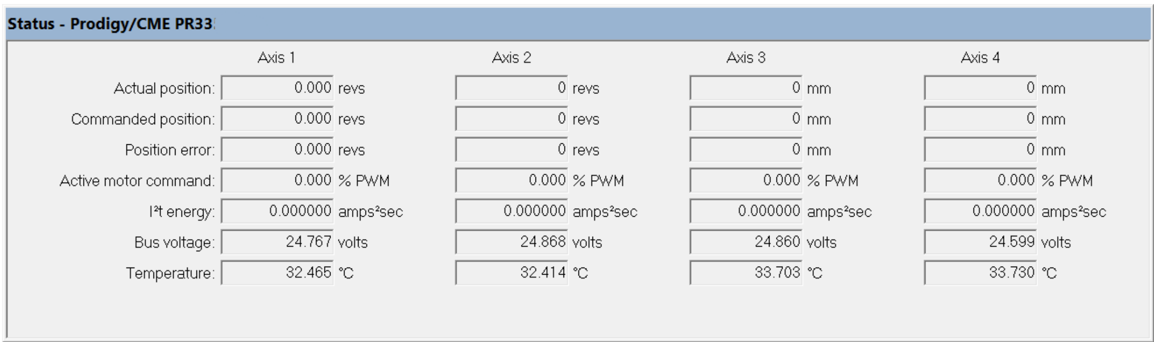
The Axis Control window presents these selectable boxes such that the overall control flow for the motor type selected is evident. For example the boxes and control flow arrows displayed for a Brushless DC motor are different than for a DC Brush motor reflecting the fact that Brushless motors are controlled differently.

Some of the boxes at the bottom and left hand side of the Axis Control window are not connected via the control flow arrows. These provide access to motion control settings such as for limit switches, breakpoints, axis I/O, signal status, event management, units, homing, and Atlas NVRAM (for axes which use an Atlas amplifier).

The box labeled ‘Operating Mode’ provides control of whether major axis control modules are active. These modules are trajectory, position loop, current loop, and motor output. While normally enabled, there may be circumstances, for example if a motion error occurs, where some modules will get disabled for safety reasons and may need to be manually re-enabled.

3.5 Status Window

The screen capture below shows an example of a Status window display, in this case for a four axis controller. The Status window displays all axes for the device selected in the Project window. For example if one of the axes in a four-axis Prodigy/CME Machine-Controller is selected all four axes of that controller will display in the Status window



For each axis of the selected PMD controller the value of several parameters are continuously displayed in the Status window. Note that some PMD products, particularly those without an amplifier, will not display all of these parameters. Here are brief descriptions of the displayed parameters in the image above:

Actual position – This is the actual position of the motor as measured by an encoder or by Hall sensors if Halls are programmed to provide position feedback.

Commanded position – This is the instantaneous commanded position from the trajectory generator.

Position Error – This parameter shows the difference between the commanded and the actual position. For servo-controlled motors (DC Brush and Brushless DC) this value measures how accurately the position loop is maintaining the commanded position. For step motors, if an encoder is present, this represents the amount that the actual motor lags the commanded position. A positive value means the commanded position is greater than the actual position, and vice versa for a negative value.

Active motor command – This parameter shows the torque (current) command being sent to the current loop/ amplifier. If there is no current loop present (or active), rather than units of amps this parameter will have units of % PWM (Pulse Width Modulation).

Bus voltage – For PMD products which include an amplifier function this parameter indicates the current measured value of the DC supply voltage.

Bus current return – For PMD products which include an amplifier function this parameter continually indicates the current flow from HV to ground in the amplifier’s switching bridge. Note that this current value does not include current used to power internal logic of the PMD controller.

Temperature – For PMD products which include an amplifier function this parameter indicates the instantaneous temperature of the amplifier bridge.

The quantities displayed in the Status window are displayed using the user-selected units. For example in the screen capture above axes 1 and 2 use position units of revolutions, and axes 3 and 4 use units of millimeters. To change Pro-Motion display units for a particular axis select the 'Units' box in the Axis Control window.

Note that the specific parameters shown in the status window are product specific. Therefore depending on the PMD product you are using the list of display parameters may be different than those in the image above.

3.5.1 Status Polling

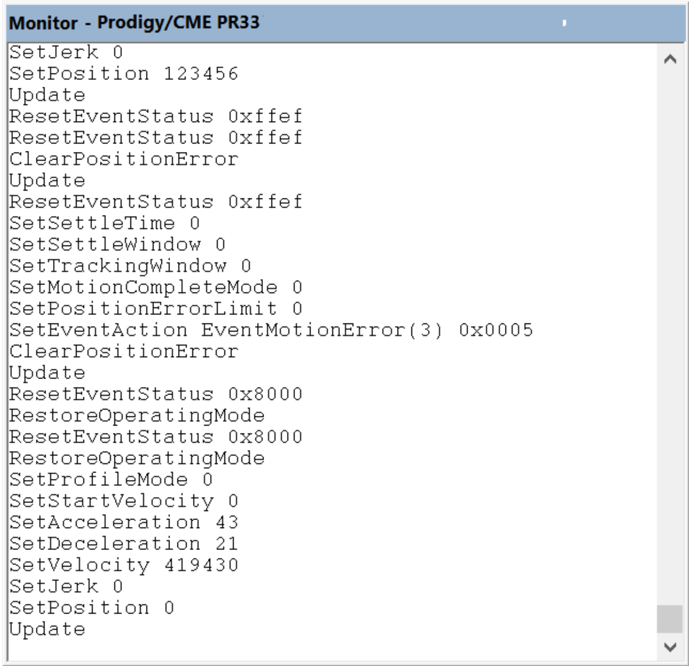
By default Pro-Motion continually polls the device selected in the Project window so that the Status window shows up-to-date values for all axes of that device. In addition, regardless of what device is selected in the Project window, Pro-Motion regularly polls all connected devices to report on safety events such as motion error, overcurrent, etc…

By default this background status polling is enabled, however there are situations where it may become disabled. One such situation is when certain types of communication errors occur between Pro-Motion and the connected PMD controller. To avoid continually generating communication error warnings automatic polling may be disabled by Pro-Motion. If this happens Pro-Motion will display this change in a dialog box that pops up after the communication error.

Automatic polling may also be manually disabled by the user. This can be accomplished via the *View/Status Polling* menu function. The main reason for manually disabling background polling is when using the Monitor window. Background polling results in a large number of messages being displayed in the Monitor window, thereby making it difficult to focus on the transactions of interest. Disabling automatic polling solves this problem. Note that after status polling is disabled it should eventually be re-enabled to restore safety monitoring.

Disabling status polling may result in erroneous or unsafe operation of the motion controller being unreported by Pro-Motion. It is therefore always recommended that Pro-Motion status polling be enabled during regular operation of the motion controller.

3.6 Monitor Window



Pro-Motion supports the ability to view communications between Pro-Motion and a connected PMD controller. These communications are in the form of hexadecimal-coded packets encoding short command-oriented transactions sent by the PC and responded to by the PMD controller. Packets can transmit messages which have meanings such as “Set the Profile Destination Position to 123,456” or “What is the current encoder position?” Transmitted command packets are displayed in the monitor as ASCII mnemonics. For example these two command functions are displayed in the monitor with the mnemonics **SetPosition** and **GetActualPosition**.

While many users will not need to concern themselves with the format of command mnemonics, software developers in particular may find the Monitor window traffic useful to learn how Pro-Motion commands the PMD controller to achieve certain functions.

Here is some additional information about the Monitor window that you may find helpful:

- To open the Monitor window click on the corresponding icon on the top menu bar.
- Displayed commands have zero, one, two, or three arguments. For details on command and argument formats refer to the *Juno Velocity & Torque Control IC Programming Reference.*
- Arguments that are prefaced with a “0x” (for example 0x1234) are being displayed in hexadecimal. Arguments without an 0x (for example 1234) are being displayed in decimal.
- To control which command packets are displayed right-click from within the Monitor window and select one or both of the ‘filter’ settings. *Filter Gets* will avoid displaying traffic due to regular background queries from Pro-Motion to the device. Note that an alternative to disabling all Get commands may be to disable automatic status polling. See Section 3.5.1, “Status Polling” for information on this. *Filter Reads* will avoid displaying traffic containing scope trace data.
- You can clear the content of the Monitor window using right-click and *Clear*.
- You can save the content of the Monitor window to a text file using right-click and *Save As...*

3.7 Command Window

Pro-Motion supports a command line interface known as the Command window that allows the user to directly type in and send commands to the attached PMD controller.

The Command window has a DOS command line style interface with a ">" command prompt. All Magellan and Juno Motion Control IC commands are accepted, and as described in Section 3.6, "Monitor Window" are expected to be in ASCII mnemonic format.

The *Juno Velocity & Torque Control IC Programming Reference* provides detailed syntax and argument definitions for each enterable command. Go to the alphabetized Instruction Reference section of this manual to lookup specific commands. The command syntax is indicated at the top with argument definitions provided just below. Note however that the *Axis* argument, which is shown as the first argument, should not be entered when typed into the Command window in the command line. Doing so will result in an error indicating there are too many arguments. As indicated above the addressed axis is specified using a separate #Axis command.

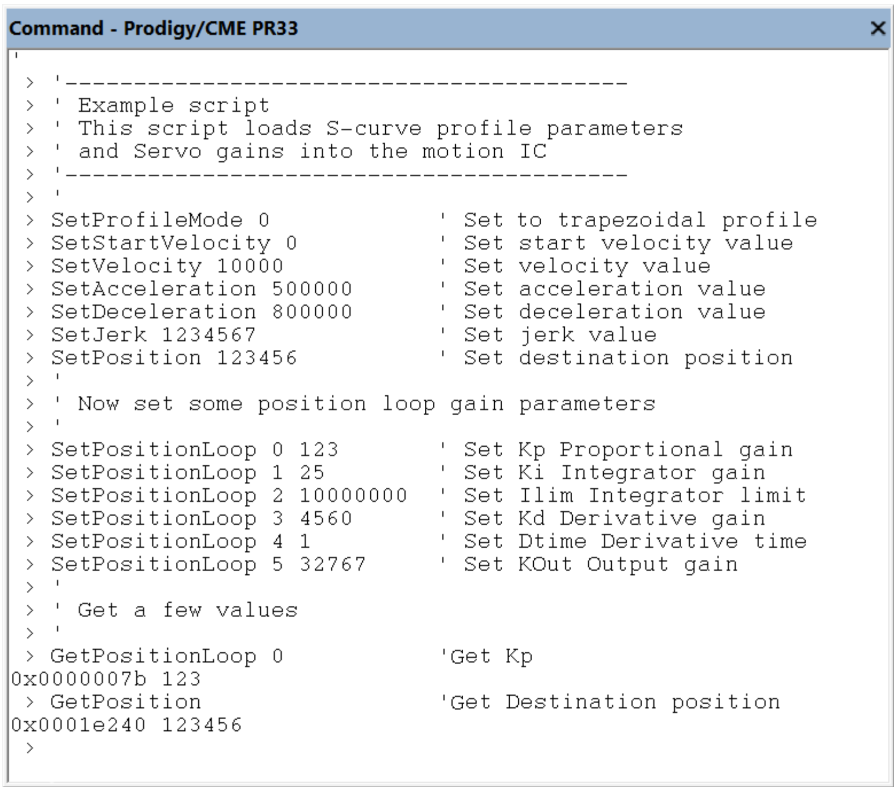
Here is some additional information on the Command window that you may find helpful:

- To open the Command window click on the corresponding icon on the top menu bar.
- Both the command mnemonic and associated argument values are entered on a single line followed by the Enter key. If the expected number of arguments are not entered an error message will display.
- Command mnemonics are not case sensitive.
- Entered arguments are separated from the command mnemonic and from each other (if the command accepts more than one argument) by spaces.
- All arguments are numerical. Arguments can be provided in decimal format (for example 1234) or in hexadecimal format by prefacing with "0x" (for example 0x1234).
- A facility for viewing the list of available commands is activated by hitting the Tab key. As characters are typed in, whenever the Tab key is pressed the command mnemonics that match the characters typed in are displayed in a "Select a Command" window that pops up. If Tab is pressed at the prompt a list of all available commands is displayed. To select a command from the list highlight the command and press Enter. The selected command appears at the command prompt (>) and the 'Select a Command' window closes.
- Upon entering the Command window typed-in commands are sent to the axis presently selected by the Project window. This can be changed while inside the Command window by entering a "#" followed without a space by "Axis n" (n being the axis number to change to) at the command prompt. For example after entering the command line "#Axis 3" all subsequent typed-in commands will apply

to axis 3 of the PMD controller. The addressed axis can be changed by further #Axis commands, or by exiting the Command window, at which point the active Pro-Motion axis returns to the axis selected in the Project window.

- If there is an error in processing a typed-in command a message will display indicating the nature of the error. An example of this in the form of a “Error, syntax error in command string” message can be seen in the Command window screen capture above. Commands that request information will return the requested value on the next command line. Commands that do not request information will display the command prompt at the next line.

3.7.1 Command Window Scripts



```
Command - Prodigy/CME PR33
'
> '----------------------------------------
> ' Example script
> ' This script loads S-curve profile parameters
> ' and Servo gains into the motion IC
> '----------------------------------------
> '
> SetProfileMode 0            ' Set to trapezoidal profile
> SetStartVelocity 0          ' Set start velocity value
> SetVelocity 10000           ' Set velocity value
> SetAcceleration 500000      ' Set acceleration value
> SetDeceleration 800000      ' Set deceleration value
> SetJerk 1234567             ' Set jerk value
> SetPosition 123456          ' Set destination position
> '
> ' Now set some position loop gain parameters
> '
> SetPositionLoop 0 123       ' Set Kp Proportional gain
> SetPositionLoop 1 25        ' Set Ki Integrator gain
> SetPositionLoop 2 10000000  ' Set Ilim Integrator limit
> SetPositionLoop 3 4560      ' Set Kd Derivative gain
> SetPositionLoop 4 1         ' Set Dtime Derivative time
> SetPositionLoop 5 32767     ' Set KOut Output gain
> '
> ' Get a few values
> '
> GetPositionLoop 0           'Get Kp
0x0000007b 123
> GetPosition                 'Get Destination position
0x0001e240 123456
>
```

In addition to the features described above the Command window supports a very useful feature which is the ability to execute a series of commands stored in a script file. Command window scripts do not support conditional statements or branching, instead they act more as macros executing a fixed sequence of pre-programmed commands. As such they are generally used just for sending parameter initialization sequences.

Command window script files are text files containing only ASCII characters. Within Windows the Notepad program is a common way to edit text files, but any editor that supports a text file format can be used.

From a Command window session, to execute a script the “<” character is entered followed without a space by the script file name, for example *<c:\scripts\TestScript.txt*. Alternatively entering just the “<” character following by the Enter key will call up a Windows Open dialog box. An alternate way to access this same dialog box is right-clicking from the Command window and selecting *Specify Script File*.

Here is information on the formatting of script text files:

File type – Script files are ASCII text files

Comments – Comments are content embedded in the script file that are not executed, but rather are used by the script developer to clarify its content. Comments may appear anywhere within a line and result in all subsequent characters to the end of the line being ignored during script processing. The character recognized as beginning a comment is the single quote '.

Blank lines, leading spaces and tabs – Blank lines are ignored as are leading spaces or tabs.

#Axis commands – Scripts may contain #Axis commands anywhere in the script. This is the mechanism by which a script can send commands to different axes on a device that supports multiple axes.

Commands and Newlines – Only one command and its associated arguments (if any are needed) are recognized per line. Recognized characters marking the end of the line are \R (ASCII 13) and \N (ASCII 10).

Argument delimiters – Arguments must be separated from adjacent commands or arguments by either a space or a tab. Commas are not an allowed delimiter.

Here is some additional information on Command window scripts that you may find helpful:

- If during script processing an error is encountered script processing will halt.
- Nesting of scripts is allowed, meaning scripts are allowed to contain command lines that execute other scripts. If using this feature caution should be exercised to not have a script call itself.
- The primary intended purpose of Command window scripts is to load Magellan Motion Control IC parameter settings so that they don't have to be typed in manually. While not specifically prohibited, script commands that result in axis motion being initiated or modified is not recommended and should instead be done via the Pro-Motion Axis Control window, or via user-written application code.

3.7.2 Motion IC Settings Export to Script

Pro-Motion provides the ability to export the IC controller settings to a Command window script.

The configuration information that is saved to a script file consists of IC settings such as loop gains, safety settings, signal sense, etc. Multi-axis products save settings for each axis. For example a four-axis Prodigy/CME Machine-Controller would have configuration information generated in the script for all four axes.

Here is some additional information that you may find helpful:

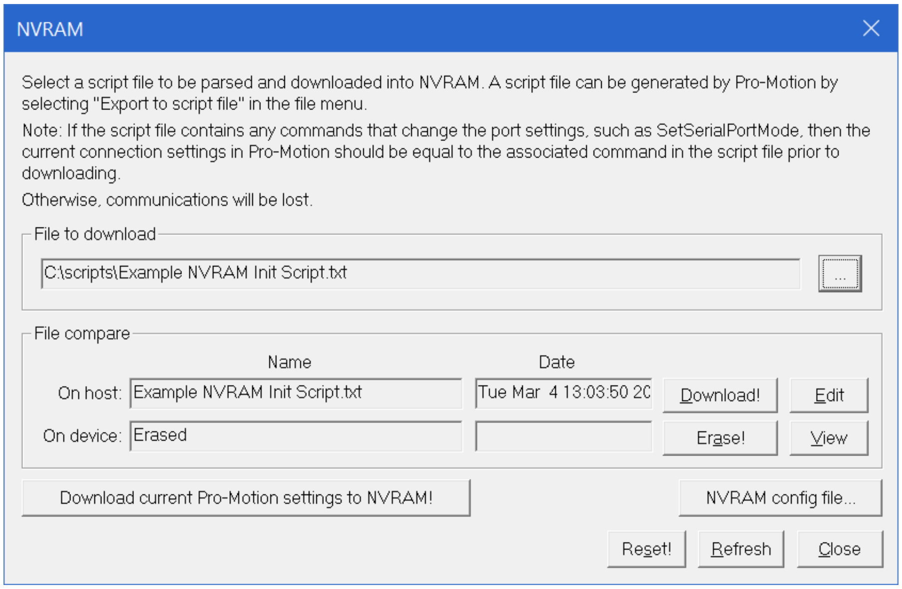
- To save the configuration to a Command window script file use the menu function *File/Export Motion IC Settings as Script*. A default file name is provided which can be changed. The specified script file will be created and displayed in Windows Notepad for convenience.
- Saved configuration script files are compatible with the Command window script facility and can be loaded and executed from the Command window. In addition the content can be edited or copied and incorporated into other script files as desired.
- Configuration scripts are ASCII-formatted files and therefore cannot be sent directly to the PMD controller using a standalone terminal or PC-based terminal emulator such as Procomm.exe. Scripts are parsed by the Command window and converted into hexadecimal-coded packets, which are then sent to the PMD controller.
- For script file exports there are generally more parameters saved than are actually used in a given application. If desired, the user can therefore review the saved configuration data and remove unused setting entries.

3.7.3 Downloading Scripts to the Motion IC NVRAM

MC58113 and MC7x11x-series ICs provide the ability to store configuration information such as communication settings, gain parameters, drive-related safety parameters, and other parameters into the internal NVRAM (non-volatile memory) of the motion IC. This initialization configuration content is specified using Pro-Motion Command window scripts.

Once loaded into the IC's NVRAM, the commands in the specified script are automatically executed and any parameter settings contained in the script become the active settings used by the IC at power-up or reset. The main advantage of storing initialization sequences in NVRAM is that the stored settings are available immediately after powerup of the IC, rather than having to be downloaded by a host PC or microcontroller.

Programming or re-programming of the motion IC's NVRAM is accomplished by selecting the NVRAM box in Pro-Motion's Device Control window. The screen capture below shows the dialog box that appears.



Specify the script file in the “File to download” field. If you click the small box to the right a Windows Open dialog box will appear. When ready click Download! to store the script file content into the IC’s NVRAM. Similar to scripts that are executed in the Command window and sent to the IC via a communication port, the script content is parsed by Pro-Motion, converted into hexadecimal-coded packets, and then sent to the motion IC for storage.

If the NVRAM already contains stored content it will be overwritten when a new script is downloaded. To erase the IC’s NVRAM memory select the Erase! Button. This will restore the motion IC to use its default parameter settings after powerup.

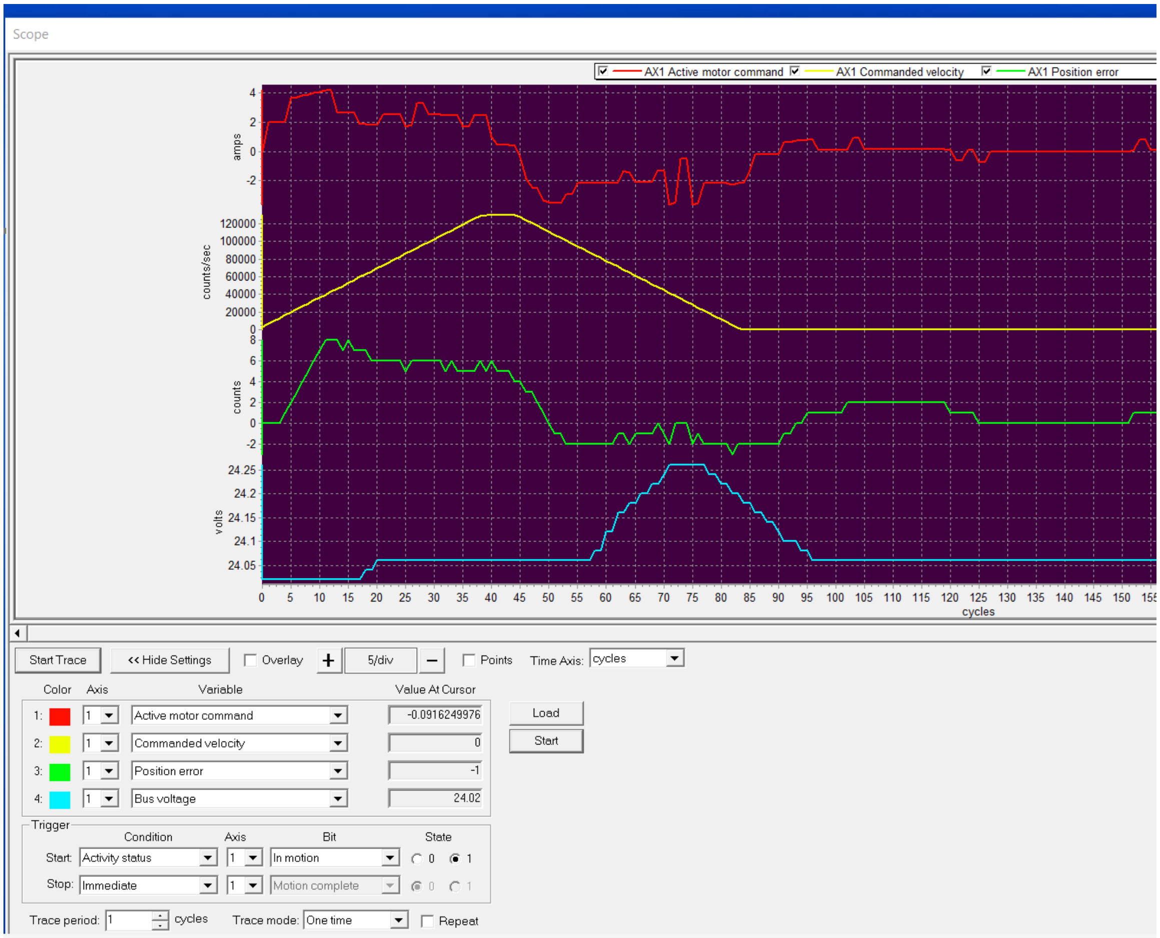
3.8 Scope Window

Pro-Motion’s Scope window ties directly to the motion control feature called trace, which provides the ability to capture and store in hardware memory up to four motion registers simultaneously, at up to 20 ksamples/second (depending on the PMD controller).

Once a trace operation is specified the motion IC captures the motion values at a programmed time interval and stores these values in its local RAM memory. Pro-Motion then sends commands to retrieve the data from the IC and display this data graphically. In addition to being displayed, traced data can be captured to a file for import to spreadsheets or other graphing and analysis software. This is accomplished using the *File/Export Trace* menu function.

The list of available traceable variables is product dependent, but may include parameters such as the motor output command, position error, commanded position, commanded velocity, actual position, and others.

To access the Pro-Motion scope function click the icon at the top bar labeled *Scope*. An example screen capture of the Scope window is shown below.



Here are some of the key settable fields of the Scope window:

Trace Variables – The core of the trace and scope function is the list of motion registers that will be traced and graphed. These are shown as Variables 1 through 4 with graph colors red, yellow, green, and blue respectively. For each trace variable click the down arrow which in turn displays a list of trace categories such as Commanded Trajectory, Feedback, Position Loop, etc... Selecting one of these categories then shows the specific available traceable motion variables. Note that for multi-axis devices variables can be traced from separate axes.

Time Axis – The top portion of the Scope window graphs captured data. Up to four variables can be graphed at the same time. The horizontal scale is time, with selectable units via the Time Axis field of cycles, milliseconds, and seconds.

Trigger Controls – Similar to a regular oscilloscope the conditions by which trace data collection can start or end is settable. Use the down arrow key to see the list of available trigger registers, and the associated bit or signal for each register.

Trace Period & Trace Mode – The data capture trace period is expressed in cycles, which are 51.2 µSec per cycle. So specifying a period of 1 cycle captures data at ~20 kHz. Trace mode can be set to *One-time* or *Rolling Buffer*. One-time capture fills the trace buffer only once beginning when the trigger start condition is satisfied. One time capture is recommended unless a particular reason for selecting rolling buffer exists.

The most common use of rolling buffer mode is to display trace data leading up to the moment an event occurs. For example with rolling buffer mode selected, if *Immediate* is set as the trigger start condition and the *Event Status* register selected with *Commutation Error* as the programmed bit and the state set to 1, after a commutation error occurs the captured and displayed data will show the trace data prior to the occurrence of the commutation error.

> When using rolling buffer mode caution should be exercised to avoid overflowing the buffer, which can result in the graphed data not matching the actual traced data. This can occur if the rate of data being put into the trace buffer by the Magellan IC is faster then the rate at which Pro-Motion can request it.

Trace Length – The total number of trace samples can be specified via this setting. Setting this value may help you better-manage how your data displays or how it exports to a file. The programmed value represents the number of data set captures. For example if this value is set to 500 and one variable is traced a total of 500 data values will be captured, if two variables are traced a total of 1,000 data values will be captured etc...

3.8.1 Trace Buffer Display Optimization

The speed of the scope display update is related to the size of the buffer that fills with data in the Magellan IC, and the speed with which the Windows PC can retrieve this data from the Magellan IC. For higher speed interfaces, if available in the controller you are using, such as CAN or Ethernet optimizing these settings is usually not needed. However for serial interfaces, improving the communication speed may be useful. To set the baud rate to a new value see Section 4.2, "Changing the Active Serial Settings."

Another way to speed up serial communications is to reduce the latency between message packet sends. This is done via the following sequence; First, open the Windows Device Manager by typing "Device Manager" in the Windows search box. Next, find Ports (COM & LPT) from the list and click on it, then right-click the USB Serial Port (COMn) of the serial port you are using and then select Properties. Click on the Port Settings tab and then select the Advanced button. Change the field for Latency Timer (msec) to a value of 1. Click OK on both Windows and close the Device Manager.

3.9 Project Configuration Save & Restore

Pro-Motion project files allow you to store a motion control configuration to a named file and later recall this saved configuration. You can save a project at any time while working with Pro-Motion by specifying *File/Save Project* or *File/Save Project* As from the Pro-Motion session menu.

Project files save the control parameters and communication settings for all axes of all PMD controllers connected to Pro-Motion and displayed in the Project window. The files that Pro-Motion uses to save and restore project configuration information have an extension of *.PMD*. These project files are not user-readable. They encode the project configuration information in a format that is written and read by Pro-Motion, but is not intended to be read or edited by the user.

The specific configuration information that is saved via the *File/Save Project* function is all of the PMD controller's configuration, control, and communication settings including all Motion Control IC settings. The main exceptions are content that is stored in NVRAM such as Magellan initialization commands, device SetDefaults settings, and *.bin* C-Motion Engine user code programs. Most of the Pro-Motion windows also have their content saved via the project save mechanism. This includes the Trajectory and Units dialog box settings of the Axis Control window, and the Scope window settings. Finally, the operating mode of the PMD controller is saved - in other words whether axes are enabled, the position loops are active, amplifier output is enabled, etc...

> .PMD project files are not user-readable. They encode the project configuration information in a format that can be read (and written) by Pro-Motion, but is not intended to be read or edited by the user.

3.9.1 Project Configuration Restore

To call up a saved project select *File/Open Project*. You will be asked to specify a project file. Once you specify a file the content will be parsed by Pro-Motion and sent to the appropriate PMD controller(s).

Once saved configuration settings are loaded Pro-Motion will attempt to return each axis of the system to the operating mode it had when the project was saved. Along those lines, if appropriate, dialog boxes will appear asking whether the motor is ready to be energized. In the case of a Brushless DC motor energizing means the commutation is initialized and depending on the PMD control product being used and the motor control mode setting the current loop and the position loop may be enabled. For DC Brush motors, similarly, the position loop and current loop may be enabled depending on the control systems in the project file. For step motors the drive current or holding current will be applied.

With communication connections and controller settings restored, after the project open operation the system should be restored to the same condition it had when the project was saved. At this point further development work on the project can occur and subsequent configuration changes stored, if desired, via the *File/Save Project* function. Alternatively a new project file can be created by saving the configuration via *File/Save Project As*.

> Selecting File/Open Project to restore a saved configuration should be done with the connected PMD controller(s) in a reset condition or when in a stable non-moving state. Systems that have axes that may move when de-energized (such as vertical axes subject to the force of gravity) should only be restored from a reset condition. Failure to observe these guidelines may result in damage to the controlled mechanics.

3.10 Configuration Export to C-Motion

Pro-Motion provides the ability to export the Pro-Motion motion control setup as a C-Motion source code file, suitable for input to PMD's user application code building system. To learn more about how this export operation can be used to help with user application code building see the *readme.txt* file in the SDK you will be using for code development. For more information on C-Motion SDKs see Section 7.1.4, "C-Motion SDKs."

Here is some additional information that you may find helpful:

- To save the configuration to a C-Motion file use the menu function *File/Export Motion IC Settings as C-Motion*. A default file name is provided which can be changed. The specified C-Motion source code file will be created and displayed in Windows Notepad for convenience.
- For C-Motion exports there are generally more parameters saved than are actually used in a given application. If desired, the user can therefore review the saved configuration data and remove unused setting entries.

3.11 Pro-Motion Application Notes

In the next section we provide a an example of a Pro-Motion Application Note illustrating how Pro-Motion can help characterize and optimize the operation of your motion control system.

3.11.1 Application Note – Monitoring and Enhancing DC Bus Voltage Stability

An important aspect of designing a robust motion control system is ensuring stability of the power supply voltage. This can be challenging because motors, as they accelerate and decelerate, may demand rapid increases in current from the power supply, or may actually send current back to the power supply.

In this application note we will walk through how to characterize the stability of the voltage supply under various motor operating conditions, and then suggest techniques for improving voltage stability.

There are three HV supply voltage conditions to consider for a given combination of power supply and motor operation:

1. Motor and load trajectories that result in stable operating voltage of the DC supply.
2. Motor and load trajectories that result in undesirable voltage drops because the DC supply's current output capacity is exceeded.
3. Motor and load trajectories that result in undesirable voltage increases because the DC supply cannot absorb any, or enough, current from the controller.

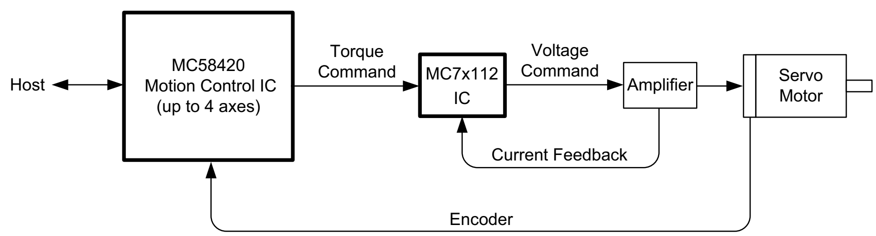
3.11.1.1 Stable DC Supply Voltage Operation

In this section we will use the trace capability of PMD controllers and Pro-Motion to show what stable operation of the DC supply looks like while the motor is moving. We will generate torque commands from a host controller that is executing a simple point to point trapezoidal profile move.

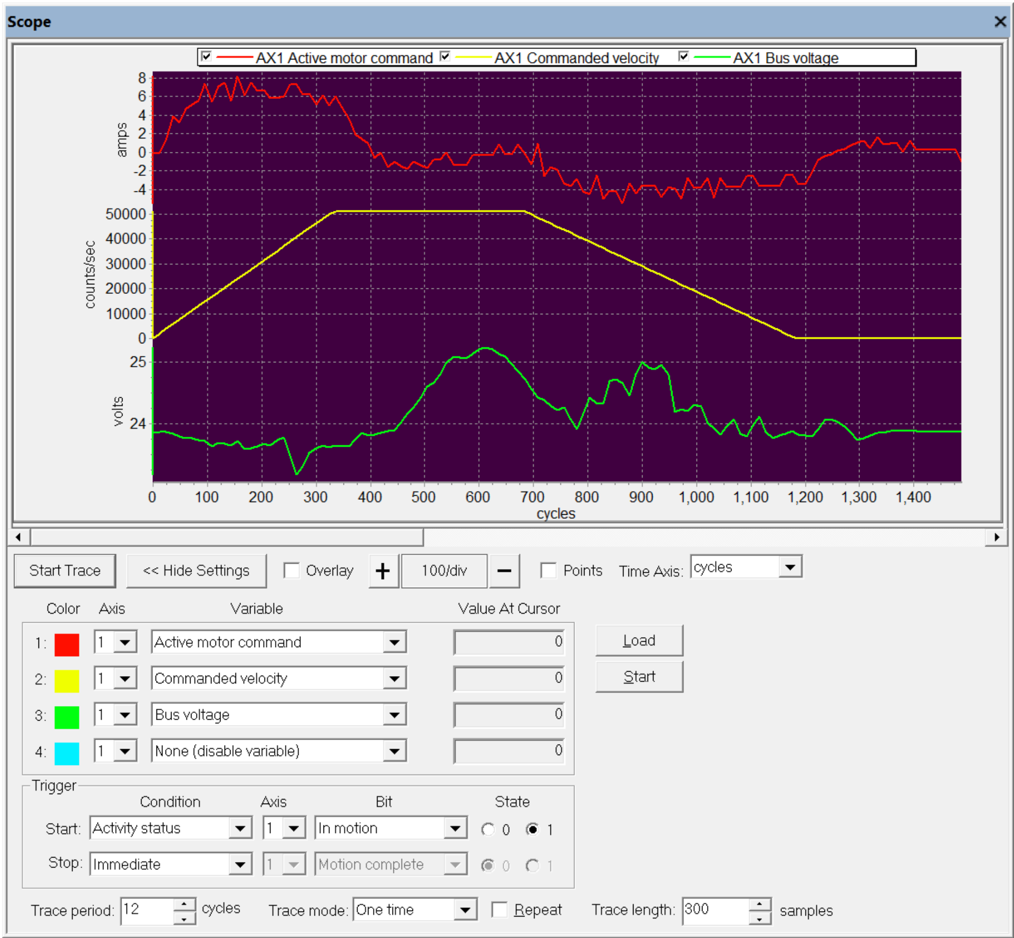
The traces displayed in this application note were made using a PMD MC58113 IC. This IC integrates profile generation, a position loop, a current loop, and a shunt control function. Use of this IC makes generation of the synchronized traces shown below easier. Although the MC7x112 torque control IC does not provide position control, if used with a profile generator and position loop it will behave similarly.

The diagram below shows such a setup. A DK58420, which contains PMD's MC58420 Motion Control IC performs profile generation and position loop servo control. It outputs a +/– 10V analog motor command which is input to the MC7x112 developer kit as the torque command.

Figure 3-1: PMD MC58420 Motion Control IC Driving MC7x112 Torque Control IC



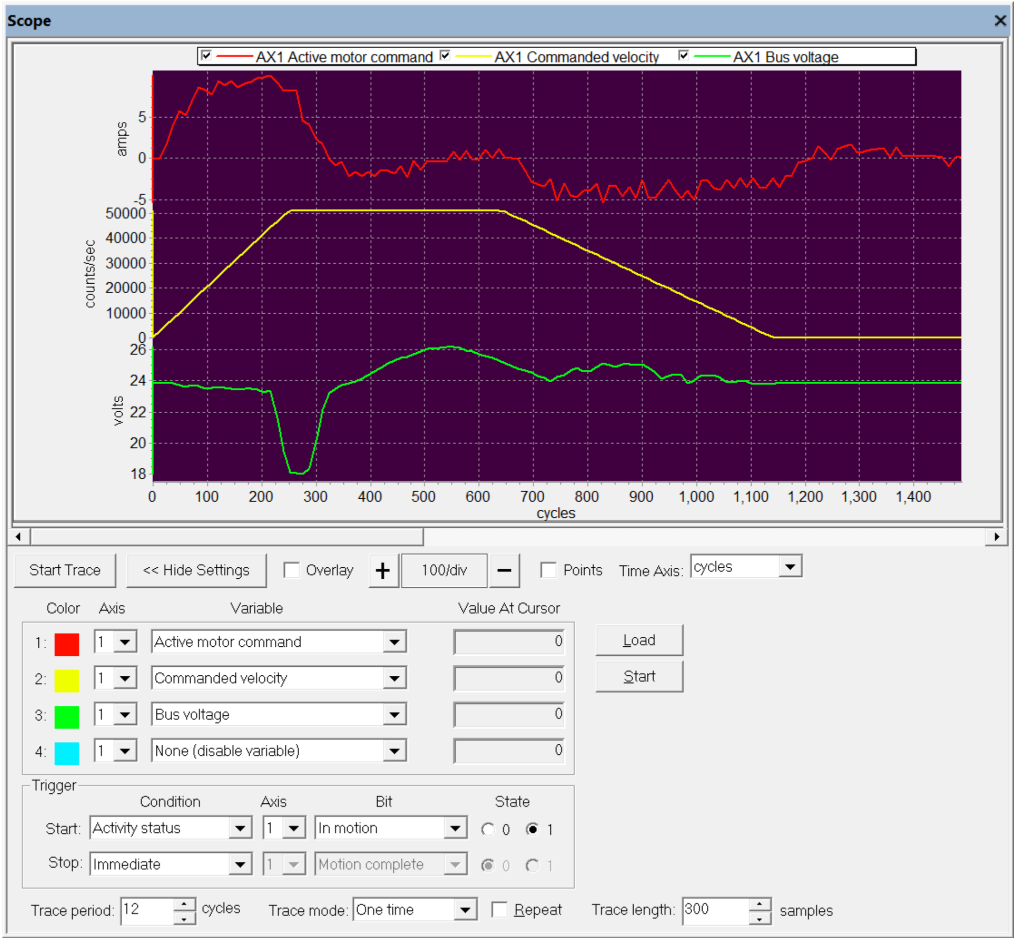
The following trace shows a baseline, stable point-to-point trapezoidal move with the resulting position loop output current command and measured HV DC Bus voltage.



In this stable voltage scenario the main feature of interest is that as the motor accelerates and decelerates, although there is a variation in the 24V supply voltage, its magnitude is small - in the above trace between 23.2 and 25.3 volts. This level of variation is acceptable in nearly all motion systems, resulting in little stress on the power supply, and in no meaningful impact on the controller's ability to maintain accurate position or torque control through the move.

3.11.1.2 Undervoltage Condition During Motion

In the Pro-Motion scope trace below the same motor setup and load used with the scope trace above is driven at a higher acceleration, resulting in a significant drop of the supply voltage from 24V to 18V. The trajectory's coasting and deceleration phase allow the power supply to return to a stable voltage.



Why does the voltage decrease during trajectory acceleration? The reason is that during acceleration the motion controller rapidly commands more current to the motor, which in turns means more current is 'requested' from the DC supply. The power supply cannot provide this dramatic increase in current and the result is a drop in voltage, an effect similar to an electrical grid brownout.

The current commanded by the motor controller is directly proportional to the amount of torque needed to accelerate the motor and load, with the torque being proportional to acceleration times the moment of rotational inertia. For example if the acceleration doubles and the load stays the same the amount of torque (and therefore required current) doubles. The significance of this is that to establish the supply voltage stability for a given mechanism the worst case current demand should be investigated. For a single axis machine this would be the highest acceleration move with the heaviest load, and for multi-axis machines this would be the same, but including all axes that move simultaneously.

Why is a drop in voltage a problem? There are a number of reasons why a voltage drop may be a problem. The first is that the power supply may become overstressed. Depending on its design, a request for current that exceeds its output capacity may overheat or damage the power supply.

Another problem is that servo stability may be impacted. In extreme cases of voltage drop the current control loop or the position control loop may become unstable. Finally, since motion controllers generally derive their internal logic power from the main supply voltage, if that supply voltage drops too far the controller may cease functioning. Modern motion controllers (including PMD's products) have a feature called undervoltage protection which automatically disables the motor drive and shuts down the trajectory generator. When an undervoltage event occurs, although the move is aborted, the more dangerous situation where the motion controller completely shuts down is avoided.

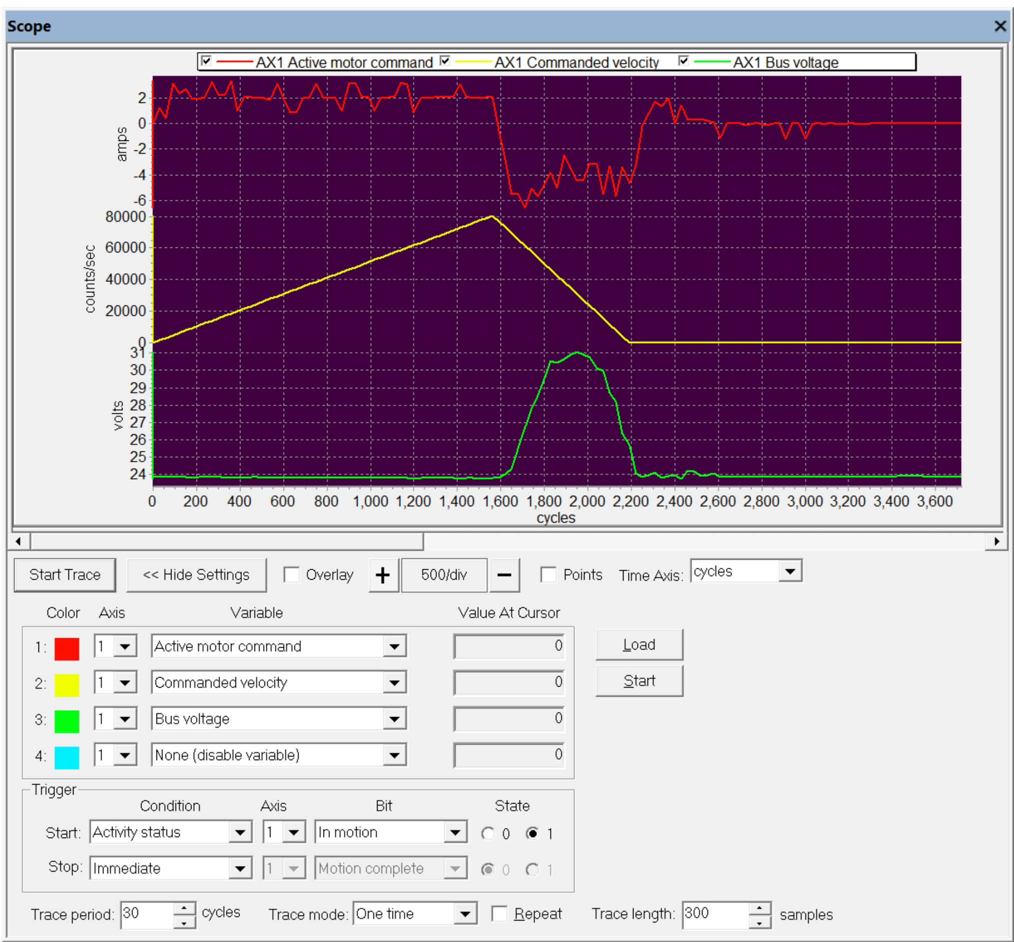
What are the remedies to voltage drop during motor acceleration? The simplest solution is to reduce the acceleration rate. This will directly reduce the amount of current the power supply needs to deliver.

If reducing the acceleration is not an option the next two options to consider are switching to a power supply with greater current output or adding capacitance to the DC supply line. Especially to provide short bursts of additional current during acceleration, adding capacitance is often the preferred option.

One more option for reducing supply voltage sag during motor acceleration is to switch to a motor that can output more torque for a given current. Note that this might also be accompanied by the need for an increase in the operating supply voltage.

3.11.1.3 Overvoltage Condition During Motion

The scope trace below shows a setup similar to the one above except with a higher deceleration trajectory resulting in a significant increase in DC supply voltage. Overvoltage conditions are potentially dangerous because they may damage the power supply, motion controller, or motor, or may lead to instability of the motion controller's servo control function.



Why does the voltage increase during trajectory deceleration? The reason that the voltage increases during deceleration is that the motor acts like a generator. During deceleration the mechanical energy stored in the rotation of the motor and attached load is converted to electrical energy which must be absorbed somewhere. This negative current output has the effect of increasing the supply voltage because most regulated power supplies have very little capacity to absorb energy from the device they are powering.

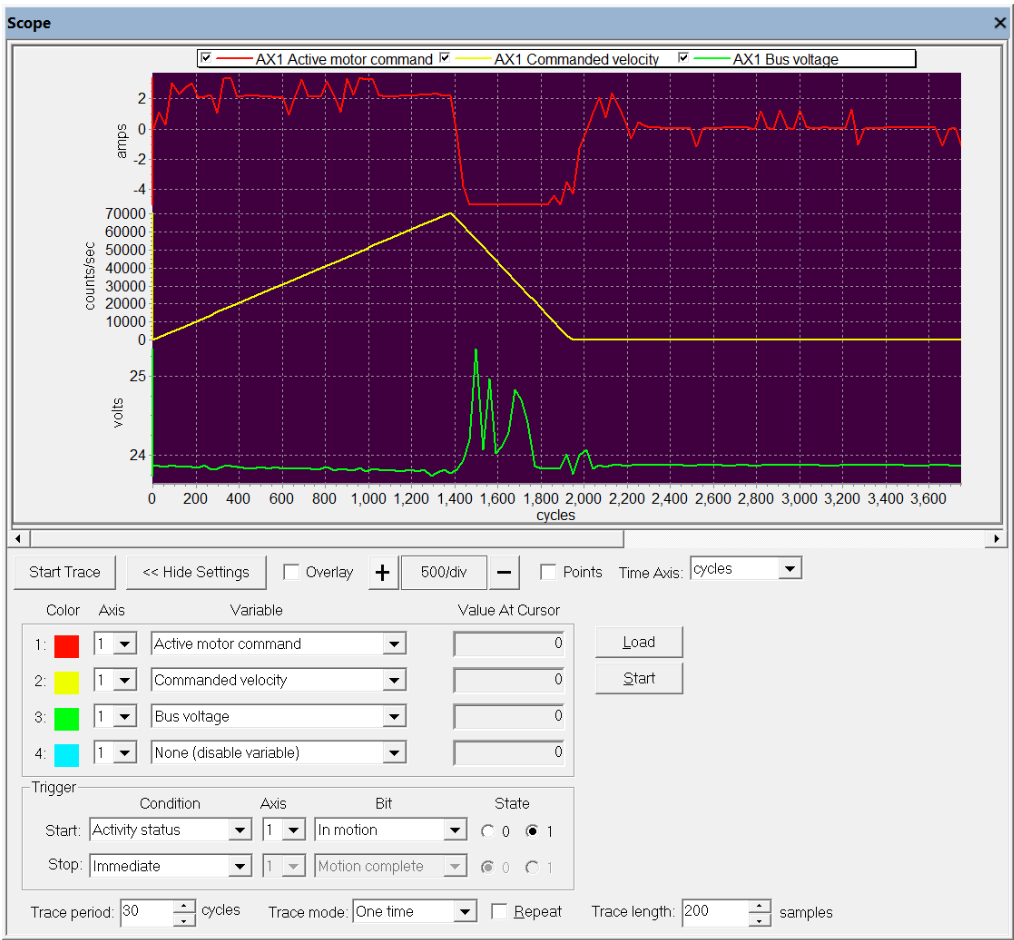
Does trajectory deceleration always generate negative current flow? No. The motor becoming a net generator occurs at higher deceleration rates for motors & loads with larger rotational inertia. An additional factor that affects this is the amount of friction in the system. Predicting when a mechanism will become a net generator is complicated so most engineers exercise the system and directly observe the supply voltage.

What are the remedies to voltage rise during motor acceleration? If an undesirable voltage rise is seen in the deceleration phase of the trajectory the simplest solution is to reduce the deceleration rate. This will reduce or eliminate the power supply needing to absorb current. Another option is to add capacitance to the HV DC supply. This is especially effective to absorb short bursts of current.

A third option is called shunt regulation. Shunt regulation functions by continually comparing the DC bus voltage to a user programmed threshold (usually set to a few volts above the supply voltage) and 'shunting' or diverting power directly from HV to GND when the threshold is exceeded. A resistor and diode in the shunt circuit path controls the rate of current flow and ensures that current flow is always in the correct direction. The power shunted through the resistor causes the resistor to heat up, therefore care should be taken to ensure that the resistor and any heat sinking elements it is mounted on can dissipate the worst case heat load.

See Section 4.3, "HV Shunt Regulation Operation" for more information on using this PMD controller's shunt function.

In the trace below the same exact motion profile as above is executed with a shunt function in place with the voltage threshold set for 25.0V. As can be seen shunt does an excellent job of regulating the voltage, lowering the maximum voltage rise from 31V to 25.3V.



3.12 Troubleshooting Suggestions

If the first-time verification sequence detailed in Section 2.3, "Step #3 — First-Time System Verification" was successful problems while using Pro-Motion are not common. Nevertheless below is a list of suggested corrections for issues users may encounter while exercising their motion hardware.

The Motor Stops Responding. If a motor previously moving and responding to your commands stops responding the most likely reason is that the motion controller's operating mode is not set correctly. Although most such 'transitions' from normal operation to an error/protection state result in a dialog box appearing, to check the operating mode select the Operating Mode box in the Axis Control window and adjust the settings if needed. See Section 3.4, "Axis Control Window" for more on the Operating Mode dialog box.

Events Aren't Being Reported. As the MC7x112 IC executes its control settings some occurrences representing a change in the control system state may occur. These changes are called events and include items such as motion error, over temperature, breakpoints, and more. To check the Pro-Motion settings for which events will result in a dialog box popping up click the Event Manager box in the lower right of the Axis Control window. For detailed information on events refer to the *MC7x112 Torque Control IC User Guide*.

An Overvoltage Event Occurred. If during a motion profile an overvoltage event is indicated the most likely reason is that during deceleration of the motor the inertia of the motor and load resulted in net generation of energy. Depending on the design of the power supply you are using this can result in the HV voltage rising and eventually exceeding the overvoltage limit. Remedies are to reduce the profile's rate of deceleration, add more capacitance to the HV supply, or use the controller's shunt regulation function if it has one.

An Undervoltage Event Occurred. If during a motion profile an undervoltage event is indicated the most likely reason is that the power supply was not able to deliver the commanded amount of current, or was not able to deliver an increase in current fast enough. Undervoltage (and overvoltage) events may also occur if the current loop is unstable. Remedies are to increase the current output limit/rating of the power supply, add more capacitance to the HV supply, or lower the MC7x112's current limit setting.

An Overcurrent Event Occurred. If an overcurrent event is indicated the most likely reason is that during a rapid trajectory deceleration the inertia of the motor and load resulted in the HV voltage rising rapidly, which, in combination with a motor with very low coil resistance can sometimes result in excess bus current. Remedies are to reduce the profile's rate of deceleration, use the shunt feature of the PMD controller (if it has one), or add more capacitance to the HV supply.

Communication Errors Are Occurring. Communication errors between Pro-Motion and the controller are rare, but depending on the connection type and motion setup may occur. The 3-pin serial connection port for example, used with some MC7x112 IC DKs and N-Series ION DKs, is susceptible to noise if high motor currents are commanded because it uses a UART signal scheme which is not a differential signal. If signal noise may be at fault consider switching to a link that uses a more robust signaling scheme such as RS232, CAN, or Ethernet, if available.

This page intentionally left blank.

4. Operations Guide

4

In This Chapter

- Commanding Torque with Hardware Signals
- Changing the Active Serial Settings
- HV Shunt Regulation Operation
- Operating with an External (Remote) Amplifier

In this chapter we provide information on how various common and useful operations can be achieved using the MC7x112 developer kit boards and Pro-Motion.

In addition to Chapter 5, *DK73112N Board Reference*, and Chapter 6, *DK78113 Board Reference*, when undertaking these operations you may find it useful to refer to the *MC7x112 Torque Control IC User Guide.*

4.1 Commanding Torque with Hardware Signals

The first time system verification process described in Chapter 2 uses the MC7x112 torque control IC's ramp generator to specify torque values and generate motion of the motor. This is convenient because it allows motion to be manually controlled from the Pro-Motion program.

However in the production application many applications will use the MC7x112's analog or digital hardware signal inputs to command the torque. This section will provide information on how to make signal connections and how to use Pro-Motion to set the command source to these hardware signals.

Depending on whether you plan to use the AnalogCmd signal or SPI bus hardware interface you should refer to one of the following two sections for connection information.

4.1.1 AnalogCmd Signal Connections

The following table shows the connections that are made to connect external analog command signals to the DK73112N or DK78113 board. In both cases the input command signals have a +/–10 V differential input format. These signals are typically generated by a user-created motion controller or circuit, by an off-the-shelf third party motion controller, or by one of PMD's motion control ICs such as the Magellan MC58000 series chipset.

Signal Name	Connector	Pin #
DK73112N Board		
AnalogCmd+	J2	6
AnalogCmd–	J2	5
GND*	J2	4
DK78113 Board		
AnalogCmd+	J5	1
AnalogCmd–	J5	3
GND*	J5	5

**For the differential analog inputs a ground connection is not required, but is recommended to limit the common mode voltage.*

For additional information on analog signal processing, for the DK73112N board see Section 5.6.3, "Analog Command Input" and for the DK78113 board see Section 6.7.3, "Analog Command Input."

For additional information on the connectors in the above table refer to either Section 5.7.3, "I/O Signals Connector (J2)" or Section 6.9.7, "Analog Command Connector (J5)."

4.1.2 SPI Bus Signal Connections

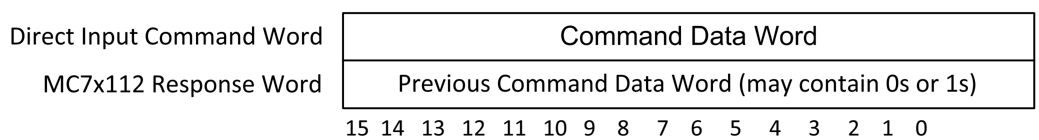
The following table shows the connections that are made to connect an external SPI bus to the DK73112N or DK78113 board. These external signals may be generated by a user-created motion controller, user-created circuit, or by one of PMD's motion control ICs such as the Magellan MC58000 series chipset.

Signal Name	Connector	Pin #
DK73112N Board		
SPIXmt	J5	1
SPIRcv	J5	2
SPIClock	J5	3
SPIEnable	J5	4
GND	J5	5
DK78113 Board		
SPIXmt	J2	1
SPIRcv	J2	2
SPIClock	J2	3
SPIEnable	J2	4
GND	J2	7

For additional information on the connectors in the above table refer to either Section 5.7.5, "SPI Connector (J5)" or Section 6.9.9, "SPI Connector (J2)."

The above signals are all TTL level and represent the standard SPI bus enable, chip select, clock, and data functions. The MC7x112 IC acts as an SPI slave, and the host processor acts as an SPI master. The SPI port operation supported is a write by the external circuitry of a 16-bit torque command data word. The write format is shown in Figure 4-1. The external circuitry serves as the SPI master generating the clock and the enable and transmitting the 16-bit data word data. New SPI command writes may occur at any rate not exceeding one write per 50 µSecs.

Figure 4-1: Direct Input SPI Format

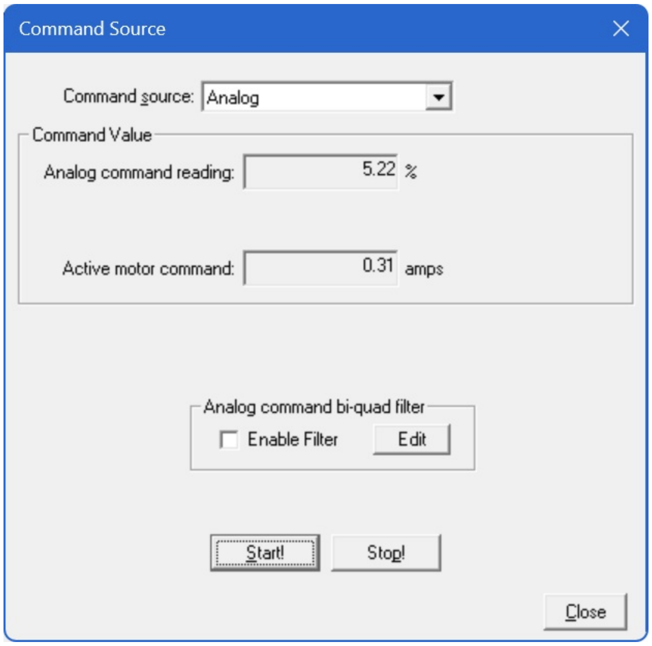


4.1.3 Using Pro-Motion to Select AnalogCmd Signals

Here is the Pro-Motion sequence to set the hardware command source to the AnalogCmd signals. Before starting this sequence the analog command source should be connected and outputting a +/– 10 V voltage of 0 volts, in other words a current command request of no current.

1. With the DK board powered up and with control settings from the "quick start" Axis Wizard setup session in place and active, select the "Command Source" box of the Axis Control Window.
2. From the Command Source field select "Analog". The Analog command reading field continually shows the value being read at the MC7x112's AnalogCmd input signal. The displayed value should be 0% give or take 1 or 2%. If the value deviates significantly from this confirm that the AnalogCmd+ and AnalogCmd– signals are outputting the expected voltage levels and are connected correctly to the DK board.

3 With the input signal value showing the expected command level select the Start! Button. Whatever analog voltage command is being input will now result in the Active Motor Command field displaying a proportional current command which is the value being continuously sent to the MC7x112's current loop. An example of this dialog box is shown in the image below:



Once the command source has been set and the Start! button activated the displayed current and current command sent to the MC7x112's current loop will track whatever the input voltage is from the AnalogCmd signals. This is true even if you exit the Command Source dialog box.

To disable torque command from the AnalogCmd signals go to the Command Source window and set the command source to something other than Analog, or for a faster method you can select the Pro-Motion Stop icon which is in the top icon bar.

4.1.3.1 AnalogCmd Biquad Filter

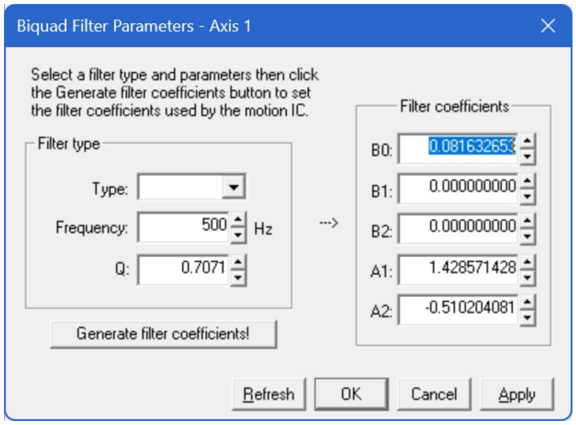
The MC7x112's AnalogCmd signal input function includes a biquad filter. Although often left at its default state of not enabled, some applications may benefit from filtering of the AnalogCmd signals, particularly if these signals are noisy.

An alternative use of the biquad filter is to correct for the 50% scaling of the AnalogCmd signal so that the full commandable current range of the MCx7112 IC can be exercised. For details on AnalogCmd scaling refer to the *MC7x112 Torque Control IC User Guide*.

To make this scaling change the biquad is programmed to double its input value without making any other changes. This is accomplished by setting the B0 gain to 2.0, and the other biquad gains (B1, B2, A1, A2) to 0.0.

Whether for filtering or for rescaling, the sequence below shows how to do this:

1. From the Command Source dialog box select the Analog command biquad filter Edit button. The following dialog box will display:



2. From the left hand side select the filter type desired, the frequency, and the Q value desired. When ready, select Generate filter coefficients! and then select OK. Alternatively, you can directly enter the B0, B1, B2, A1, and A2 gain settings in the fields to the right of the dialog box. Once the gain values have been entered select OK.
3. Use the Enable Filter radio button to Enable the biquad filter, which once enabled will activate the biquad parameters you entered.

For more information on the function of the biquad filter provided by the MC7x112 ICs refer to the *MC7x112 Torque Control IC User Guide.* For general information on biquad filters and how to use them to achieve various desired filtering effects refer to online resources such as www.octave.org.

4.1.3.2 AnalogCmd Signal Calibration

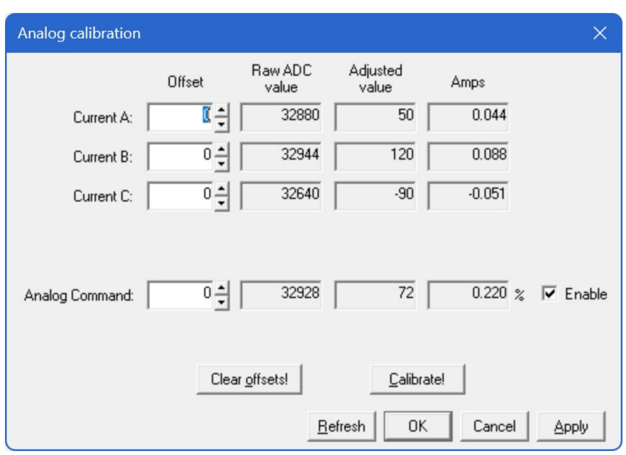
All of the MC7x112's analog signal inputs can be calibrated, meaning they can generate an internal offset so that their digital reading is zero when presented with a zero voltage at the signal.

Calibrating the MC7x112's AnalogCmd signal is recommended, especially if the motion application requires high current accuracy. The procedure below shows how to calibrate the AnalogCmd signal as well as the leg current inputs for the motor being controlled.

Before executing this procedure the motor should be at rest, and the torque commanded at the AnalogCmd+ and AnalogCmd– pins should be zero.

To perform a calibration:

1. Select the Current Loop box of the Axis Control window
2. From the Current Loop dialog box select Analog calibration. The following dialog box will appear:



3 Make sure that the Enable check box is selected and then select Calibrate!

4 After a pause of about one second the Offset column will display the calibration offset values determined for each signal.

Typical offset values for the Analog command input are in the range of −400 to +400. If the derived offset is greater than this you should check the input voltage at the DK board's AnalogCmd pins to confirm a zero command is being commanded.

5 Select OK to activate the offsets and return to the Current Loop dialog box.

For more information on calibrating analog input signals including how to do this in the production application refer to the *MC7x112 Torque Control IC User Guide*.

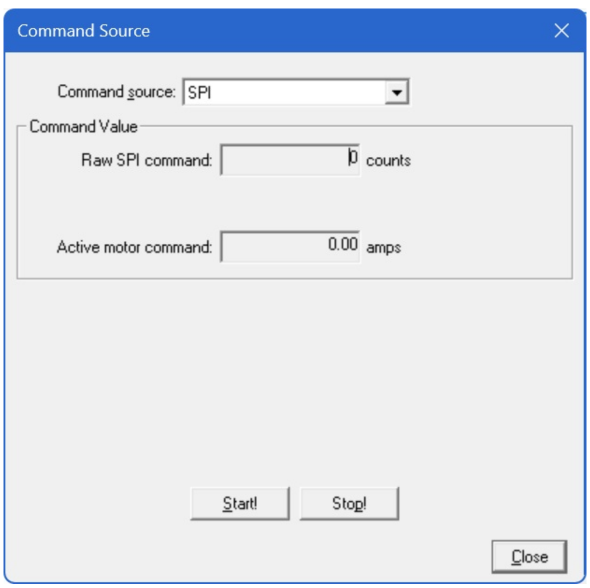
4.1.4 Using Pro-Motion to Select the SPI Bus

Here is the Pro-Motion sequence to set the hardware torque command source to the SPI Bus. Before starting this sequence the controller that is outputting the SPI commands should be connected and sending an output value of 0.

1 With the DK board powered up and with control settings from the "quick start" Axis Wizard setup session in place and active, select the "Command Source" box of the Axis Control Window.

2 From the Command source field select "SPI".

3 The "Raw SPI Command" field will now continually show the value being read at the MC7x112's SPI bus. The displayed value should be equal to the value being sent by the external SPI bus controller. If this is not the case double check signal connections. The image below shows this dialog box with SPI selected:



4 With the Raw SPI command field showing the expected command value select the Start! Button. Whatever command is being input by the MC7x112 IC will now result in the Active Motor Command field displaying a proportional current command which is the value being continuously sent to the MC7x112's current loop.

Once the command source has been set and the Start! button activated the displayed current and current command sent to the MC7x112's current loop will track whatever values are sent on the SPI bus. This is true even if you exit the Command Source dialog box.

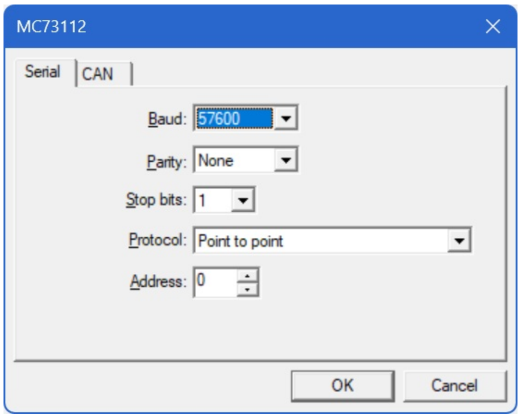
To disable torque command from the SPI bus you can go to the Command Source window and set the command source to something other than SPI, or for a faster method you can select the Stop icon which is in the top icon bar of Pro-Motion.

4.2 Changing the Active Serial Settings

Both the DK73112N and the DK78113 board provide serial connections. The DK73112N board provides a UART serial connection at its 3-pin J1 connector and the DK78113 provides an RS232 connection at its DB9 J3 connector as well as a UART connection at its 3-pin J14 connector. When using the DK78113 the DB9 RS232 connection is recommended. For additional information on the serial functions supported by the MC7x112 IC refer to the *MC7x112 Torque Control IC User Guide.*

Here is the Pro-Motion sequence to change the active serial parameter settings in the MC7x112 IC, and in Pro-Motion.

1. With Pro-Motion communications via the serial port functioning properly, click the Device Control toolbar icon. The Device Control window appears.
2. Click Network I/O. The Network I/O Defaults dialog box appears.
3. Click the Serial tab. This window appears with data entry fields for the default serial port setting such as the baud rate. This is shown below.



4. Enter the desired new comm settings in the corresponding data fields. The Protocol field should be set to point-to-point and the Address field can be left unchanged.
5. Click OK to set as the MC7x112 IC's and Pro-Motion's active serial communication settings.

Note that these changes were made to the active settings and are not stored in the MC7x112's NVRAM. Therefore upon a power-up or reset the MC7x112's active settings will revert to default values (or values that were stored in the motion IC's NVRAM).

4.2.1 Changing Serial NVRAM Settings

For serial changes to be permanently stored in the MC7x112 IC these settings must be saved in NVRAM. This is accomplished with Command window scripts.

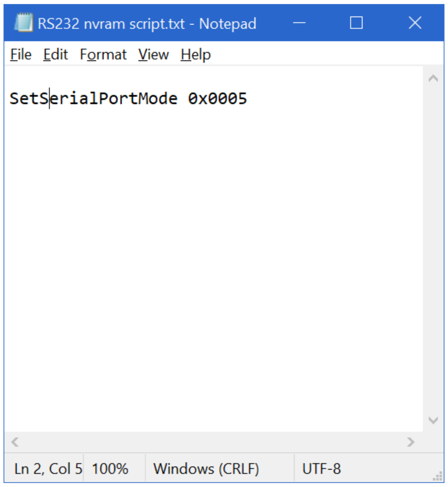
Serial NVRAM settings are specified using a script with the command **SetSerialPortMode**. The complete format of this command is detailed in the *Juno Velocity & Torque Control IC Programming Reference*, but for convenience the table below provides script line entries to program various common serial settings.

Baud Rate	# Data Bits-Parity-#Stop bits	Protocol	Address	Corresponding Script Line
57,600	8-N-1	Point-to-point	N/A	SetSerialPortMode 0x0004
115,200	8-N-1	Point-to-point	N/A	SetSerialPortMode 0x0005
230,400	8-N-1	Point-to-point	N/A	SetSerialPortMode 0x0006
460,800	8-N-1	Point-to-point	N/A	SetSerialPortMode 0x0007

Here are the steps for storing serial settings into the MC7x112's NVRAM using scripts and activating these settings in the unit:

1. Compose a text script file containing the command line that provides the desired serial settings.

   An example of such a script file in Windows Notepad is shown here:



2. With Pro-Motion communicating successfully to the MC7x112 via serial, select the NVRAM button of the Pro-Motion Device Control window. Specify the file name to download, click Download!, and when the download is completed click Close
3. Disconnect Pro-Motion's existing serial communication link using the Disconnect toolbar icon
4. Power down the MC7x112 DK board, wait a few seconds, and repower.

If the serial communication settings used by the MC7x112 IC have been changed the settings used by Pro-Motion must be set to the same settings. The next section describes how to do this.

4.2.2 Changing Pro-Motion Serial Settings

To set Pro-Motion's serial parameters:

1. With no serial connection to the PMD controller operating, click the Connect toolbar icon.
2. Select Serial from the list of options.
3. Select the comm port that the serial connection is located at and select OK.
4. Enter the same serial settings as were programmed into the MC7x112's NVRAM previously.
5. Click OK.

If serial communication is successful a graphical icon representing the MC7x112 will be loaded into the Project window. If communication is not successful, a Communications Timeout Error dialog box appears. If this happens recheck the connections and retry to establish communications with the MC7x112 IC.

4.3 HV Shunt Regulation Operation

Both the DK73112N and DK78113 boards provide a connection for a shunt resistor and diode that may be used to regulate overvoltage conditions on the DC bus. Such conditions can occur during deceleration of a motor with a large inertia.

The DK73112N board provides shunt resistor connections at its J4 connector. Wiring a resistor and diode to this connector is shown in Figure 5-5. The DK78113 provides shunt resistor connections at its J12 connector. Wiring a resistor and diode to this connector is shown in Figure 6-5.

Choosing a shunt resistor and diode with sufficient current capacity and heat sinking is important to insure the shunt circuit functions correctly and isn't damaged during operation. For both the DK73112N and DK78113 boards the shunt resistor connected should have a resistance such that the current flow through the on-board Shunt MOSFET does not exceed 10 amps. For example with an HV supply of 48Volts, this means a resistance of 4.8 ohms or greater.

The diode, which is connected in parallel to the resistor, should have a voltage and current rating at least equal to those of the MOSFET. For the DK73112N and DK78113 boards this means a voltage and current rating of 100 volts and 10 amps or higher.

For applications below these limits, the actual resistance used is application specific and depends on the nature of the anticipated over voltage generating conditions, the power supply used, the wattage rating of the resistor, and the output duty cycle of the shunt bypass specified by the user.

Example: a shunt resistor with a resistance of 10 ohms is installed and a comparison value of 51Volts and a PWM duty cycle of 75% are specified by the user. When the +HV voltage exceeds 51.0Volts, HV will be connected to GND via the shunt resistor resulting in an effective average current flow of (51.0 V * .75)/10 ohms = 3.825 amps. Therefore in this application the shunt the resistor and diode should support capacities comfortably above 3.8 A.

An additional consideration for systems with very large motor inertias or very high spin rates during deceleration is managing the total heat generated in the resistor. Even if the resistor's current capacity is not exceeded, if the shunt is active for a long period of time, without proper heat sinking the resistor may overheat.

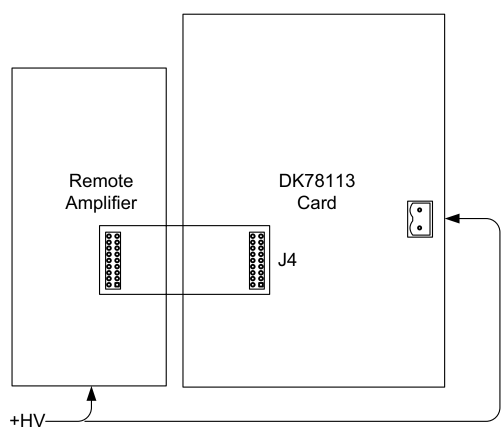
To program the MC7x112 ICs shunt function both a voltage comparison value and a PWM output duty cycle are specified. Using Pro-Motion this is accomplished by selecting the Drive Safety box in the Axis Control window and entering the desired values in the Shunt fields box which is located toward the bottom of the dialog box. Once these two values have been specified the shunt function can be enabled using the Enable radio button. When ready enter OK to have these settings become active.

4.4 Operating with an External (Remote) Amplifier

The DK78113 board supports the ability to drive a remote switching amplifier via its J4 connector. Development of a user-designed amplifier connected directly to the DK78113 board may be useful to service higher power motors, or motors or actuators that require special bridge configurations.

To operate the DK78113 board with a remote amplifier jumpers JP4 and JP5 must be set to the 2-3 position, thereby disabling the DK78113's on-board amplifier.

Figure 4-2: DK78113 to Remote Amplifier Board Connections



The overall connection scheme for a remote amplifier is shown in Figure 4-2. Note that use of the remote amplifier's HV supply to also power the DK78113 board is recommended as long as HV is within the DK78113 board's input range, which is 12-56V. Doing so allows use of the DK78113's over and undervoltage check feature. If the remote amplifier is to be operated at a voltage below 12V or above 56V then the remote amplifier power supply cannot be used to also power the DK78113 board.

> Operating the DK78113 board with a remote switching amplifier requires that jumpers JP4 and JP5 be set to the 2-3 position, thereby disabling the DK78113's on-board amplifier. Failure to do so may result in damage to the DK78113 board, the remote amplifier, and the controlled motor.

4.4.1 Supported Features

Not all of the features associated with the DK78113 on-board amplifier are supported when a remote amplifier is used. This is shown in the table below:

Feature	Supported?	Comments
PWM signal generation	Yes	Supported for all PWM modes.
Analog current feedback	Yes	Supported with PWM High/Low control mode only.
I^2t	Yes	Requires that current control be active.
Overtemperature protection	No	
Supply overcurrent detect	No	
Return overcurrent detect	Yes	Supported. Requires that current control be active.
DC Bus overvoltage detect	Yes	Supported if remote amplifier and DK 78113 board share common HV supply.
DC Bus undervoltage detect	Yes	Supported if remote amplifier and DK 78113 board share common HV supply.
Shunt Control	No	

4.4.2 Control Settings

When a remote amplifier circuit is ready for testing the MC7x112 IC that will drive it must be programmed with the correct control settings. The table below provides a list of amplifier control settings that should be set when PWM High/Low output mode is used. PWM High/Low is the MC7x112 IC amplifier output mode setting when using current control:

Setting	Comments
PWM Switching Frequency	This setting is amplifier and motor-specific. Higher inductance motors should be set for 20 kHz. Lower inductance motors may use 40 or 80 kHz to reduce current ripple and minimize heat generation if the amplifier supports those higher switching rates.
PWM Dead Time PWM Refresh Time PWM Refresh Period PWM Signal Sense Minimum Current Read Timing	These settings are amplifier design specific and vital to safe and correct operation of the amplifier bridge.
Continuous Current Limit Foldback Energy Limit	
Overvoltage Limit Undervoltage Limit	Can be used with all PWM output modes.
Leg Current Scale	Specifies relationship between ADC current input reading to actual current in amps. Optional, but allows Pro-Motion to correctly display current in amps.

From the above table the PWM settings and Minimum Current Read Timing setting can be made from Pro-Motion using the Motor Output box of the Axis Control window. The I^2t settings and over and under voltage settings are made by selecting the Drive Safety box of the Axis Control window. The leg current scale setting is made from this same Drive Safety dialog box by selecting the Drive signal scaling… button.

4.4.3 User-Designed Amplifier Bring-Up

To bring up an untested remote amplifier design Pro-Motion can be used for programming the above settings, commanding the amplifier manually, and creating scope displays of the amplifier function.

Once the critical switch control settings have been entered such as PWM dead time, PWM signal sense, etc… a typical next step is to set the MC7x112's Operating Mode to Motor Output only. This is done using the Operating Mode box of Pro-Motion's Axis Control window. Enable Motor Output and disable the current loop.

The Motor Control box in the Axis Control window allows you to specify specific PWM duty cycle settings and then check with an oscilloscope whether the signal output from the switcher appears as expected. Be aware that multiphase motors such as Brushless DC motors will distribute a given specified motor command to multiple phases. To determine the exact command values sent to each phase Pro-Motion's scope trace function can be used.

As you build up more layers of amplifier function you can eventually enable the current loop by checking the corresponding radio button in the Operating Mode box and by setting gain values using the Current Loop box of the Axis Control window.

Pro-Motion Axis Wizard should not be used when initializing the control settings of a remote amplifier. Axis Wizard assumes that you are using the DK's on-board amplifier and therefore may program parameter values not appropriate for a user-designed remote amplifier. To set these parameters for a remote amplifier use the Pro-Motion Axis Control window.

5. DK73112N Board Reference

5

In This Chapter

- DK73112N Internal Block Diagram
- Communication Port
- Switching Motor Amplifier
- Drive Protection and Control Signals
- DC Bus
- Miscellaneous Signal Processing
- DK73112N Board Connector Reference
- Absolute Maximum Ratings
- Environmental and Electrical Ratings
- DK73112N Control Settings Quick Reference

5.1 DK73112N Internal Block Diagram

The DK73112N Developer Kit board provides a complete functioning MC71112N and MC73112N IC exerciser and development system. It directly interfaces to a host computer using USB to serial communication, and provides all power, motor drive, control, and feedback signals required to drive a DC Brush or Brushless DC motor.

The DK73112N consists of several subsystems including the MC71112N or MC73112N ICs themselves, a high performance MOSFET-based motor amplifier, a DC Bus conditioning and monitoring system, and various other circuitry.

The following sections describe the major sections of the DK73112N board.

For a complete description of the MC7x112 and MC7x112N ICs, see the *MC7x112 Torque Control IC User Guide*.

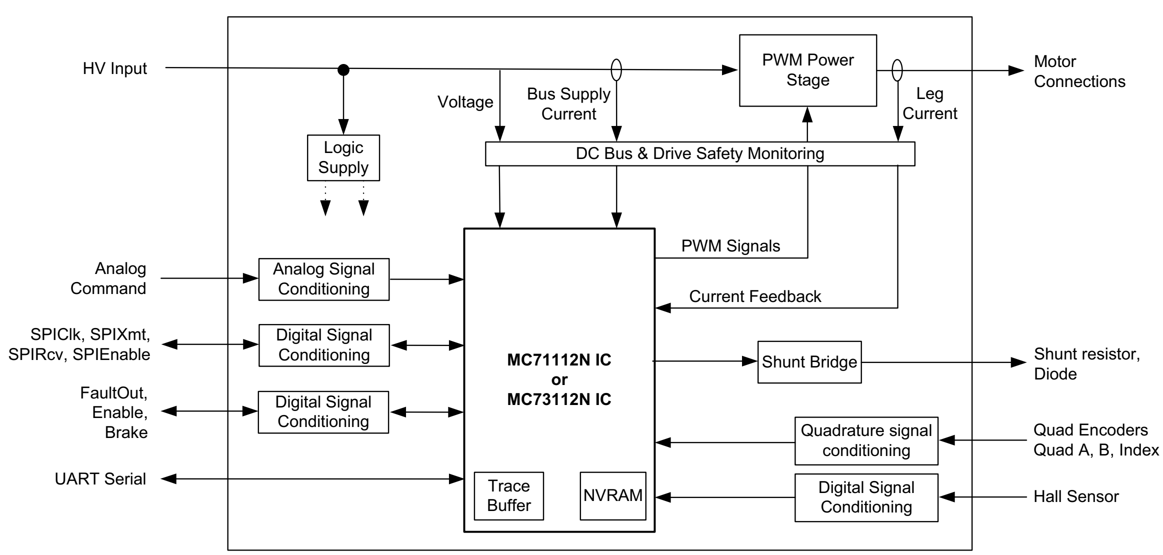


Figure 5-1: DK73112N Internal Block Diagram

5.2 Communication Port

The DK73112N supports a UART serial interface via its J1 connector. This port is used for communicating with Pro-Motion for setup of the MC71112N or MC73112N's control parameters or to command the torque.

The serial interface can be operated at the communication settings shown in the following table:

Parameter	Range	Default
Baud rate	1,200 to 460,800	57,600
Parity	None, even, odd	None
# data bits	8	8
# stop bits	1, 2	1

All DK73112N communication functions are controlled by the MC71112N or MC73112N IC. For more information on serial port function see the *MC7x112 Torque Control IC User Guide*. In addition, see Section 4.2, "Changing the Active Serial Settings" which shows how to change the serial port settings using Pro-Motion.

5.3 Switching Motor Amplifier

The DK73112N contains a high-efficiency MOSFET power stage with PWM input control and leg current feedback. Brushless DC motors are driven with three half-bridges, one for each phase, for a total of 6 MOSFETs and 3 leg current sensors. DC Brush motors are driven in an H-Bridge configuration, for a total of 4 MOSFETs and 2 leg current sensors.

The following sections provide more details.

5.3.1 Brushless DC Motor Drive

Figure 5-2 shows the arrangement of the DK73112N's amplifier stage when used with Brushless DC motors.

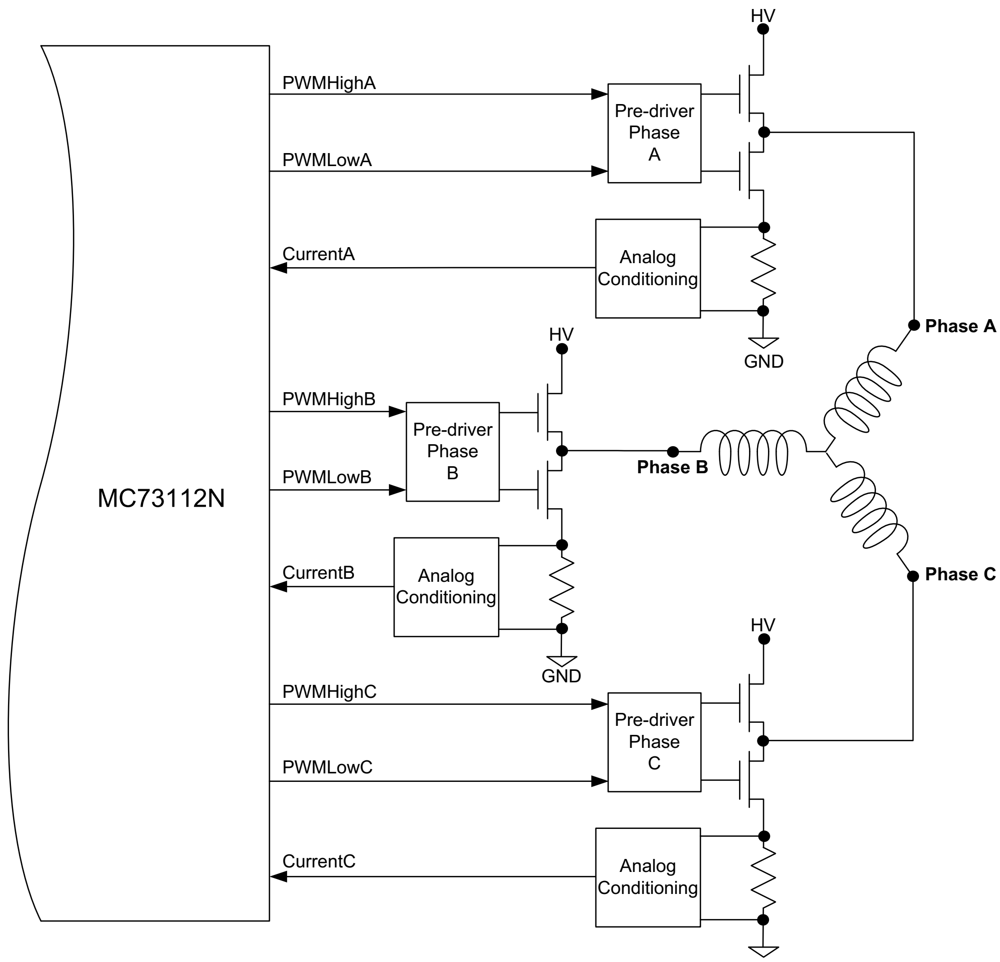


Figure 5-2: Brushless DC Motor Bridge Configuration

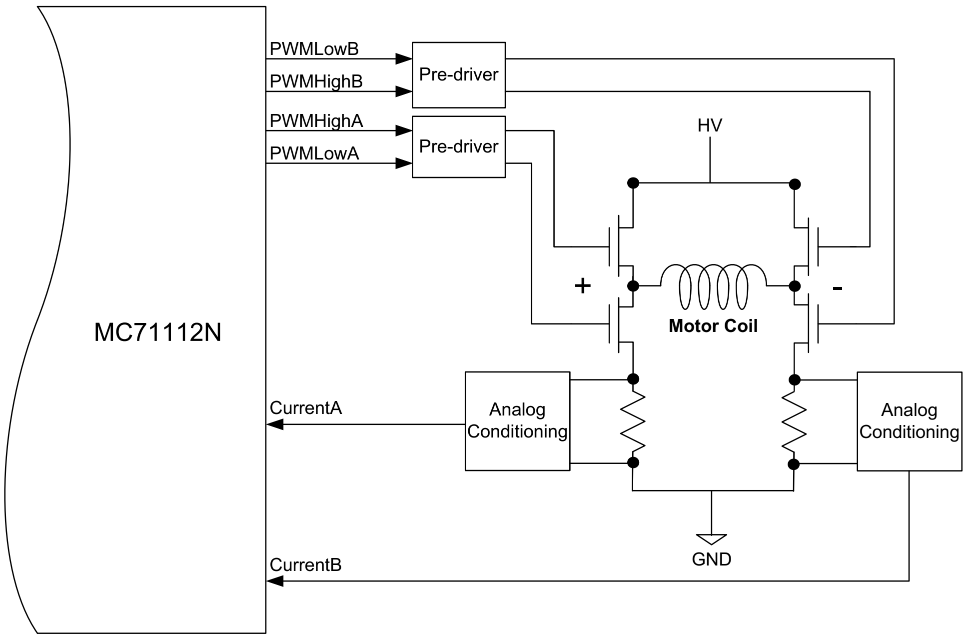
As shown in the table below six PWM output signals and three analog feedback signals interface between the MC73112N IC and the DK73112N's switching amplifier.

Signal	Description
PWMHighA	Digital high side drive output for motor phase A, positive coil terminal
PWMLowA	Digital low side drive output for motor phase A, positive coil terminal
PWMHighB	Digital high side drive output for motor phase B, negative coil terminal
PWMLowB	Digital low side drive output for motor phase B, negative coil terminal
PWMHighC	Digital high side drive output for motor phase C, positive coil terminal
PWMLowC	Digital low side drive output for motor phase C, positive coil terminal
CurrentA	Analog input containing the current flow through the positive leg of phase A bridge
CurrentB	Analog input containing the current flow through the negative leg of phase B bridge
CurrentC	Analog input containing the current flow through the positive leg of phase C bridge

5.3.2 DC Brush Motor Drive

Figure 5-2 shows the arrangement of the DK73112N's amplifier stage when used with DC Brush motors.

Figure 5-3: DC Brush Motor Bridge Configuration



As shown in the table below four PWM output signals and two analog feedback signals interface between the MC71112N IC and the DK73112N's switching amplifier.

Signal	Description
PWMHighA	Digital high side drive output for positive coil terminal
PWMLowA	Digital low side drive output for positive coil terminal
PWMHighB	Digital high side drive output for negative coil terminal
PWMLowB	Digital low side drive output for negative coil terminal
CurrentA	Analog input containing the current flow through the positive leg of the bridge
CurrentB	Analog input containing the current flow through the negative leg of the bridge

5.3.3 Amplifier-Related Settings

There are a number of MC71112N or MC73112N IC settings which are used to set or control various aspects of the DK73112N's on-board switching amplifier and related current sense circuitry.

The following table shows the default values and recommended (or required) settings for amplifier-related parameters:

Parameter	Value & Units	Comment
Motor Output Mode	PWM High/Low	Sets the motor output mode to PWM High/Low for operation with the on-board amplifier.
PWM Switching Frequency	20 kHz	This setting is motor-specific. Higher inductance motors should be set for 20 kHz. Lower inductance motors may use 40, 80, or 120 kHz to reduce current ripple and minimize heat generation.
PWM Dead Time	540 nSec	To ensure correct operation of the DK73112N's on-board switching amplifier this parameter must be set to this value.

Parameter	Value & Units	Comment
PWM Refresh Time	2,000 nSec	To ensure sufficient time to recharge the DK73112N's on-board amplifier high side switches this parameter must be set to this value.
PWM Refresh Period	8 cycles	To ensure sufficient time to recharge the DK73112N's on-board amplifier high side switches this parameter must be set to this value.
PWM Signal Sense, high	Active High	For correct operation of the DK73112N's on-board amplifier all high PWM outputs must be set to active high.
PWM Signal Sense, low	Active Low	For correct operation of the DK73112N's on-board amplifier all low PWM outputs must be set to active low.
Minimum Current Read Time	2,000 nSec	To ensure sufficient minimum current read time with BLDC motors using the DK73112N's on-board amplifier this parameter must be set to this value.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board all of these required settings will be made automatically.

5.3.4 DK73112N Current Scale Value

To control or trace motor current in units of amps it is useful to know the DK73112N's amplifier-specific current conversion factor. The following table shows this:

Parameter	Value & Units	Comments
Leg Current Conversion	.733 mA/count	This value is used to calculate current in amps.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board the above conversion factor(s) will be used automatically.

5.4 Drive Protection and Control Signals

5.4.1 I^2t Current Foldback Protection

The MC71112N and MC73112N use current feedback to implement I^2t current limiting. This feature protects the on-board amplifier by controlling its ability to operate above specific selected continuous current ratings.

When the current loop is enabled and the I^2t energy limit is exceeded, the MC71112N and MC73112N will automatically fold back the phase currents to a programmable continuous current limit value. Alternatively, the MC71112N and MC73112N can be configured to fault and disable the output stage when the I^2t energy limit is exceeded

5.4.1.1 I^2t Limits

The following I^2t limits are required to safely operate the DK73112N board.

Parameter	Value & Units	Comments
Foldback Continuous Current Limit	5.0 A	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or lower.
Foldback Maximum Energy Limit	50 A^2sec	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or lower.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board all of these required settings will be made automatically.

For motors with current and energy limits lower than those specified above reducing the settings of these parameters is recommended.

These limits are designed to be safe for operation of the DK73112N's on-board amplifier when cooled with a 110 CFM (or higher) fan and with an ambient room temperature of 25 degrees C. For use without a fan, or in higher ambient temperature environments, these values should be lowered.

If special heatsinking is used the DK73112N board is capable of driving motors at 10 amps continuously or higher, in which case these limits can be increased. Contact PMD for details.

5.5 DC Bus

Figure 5-4: Juno Torque Control ICs DC Bus Monitoring Circuitry

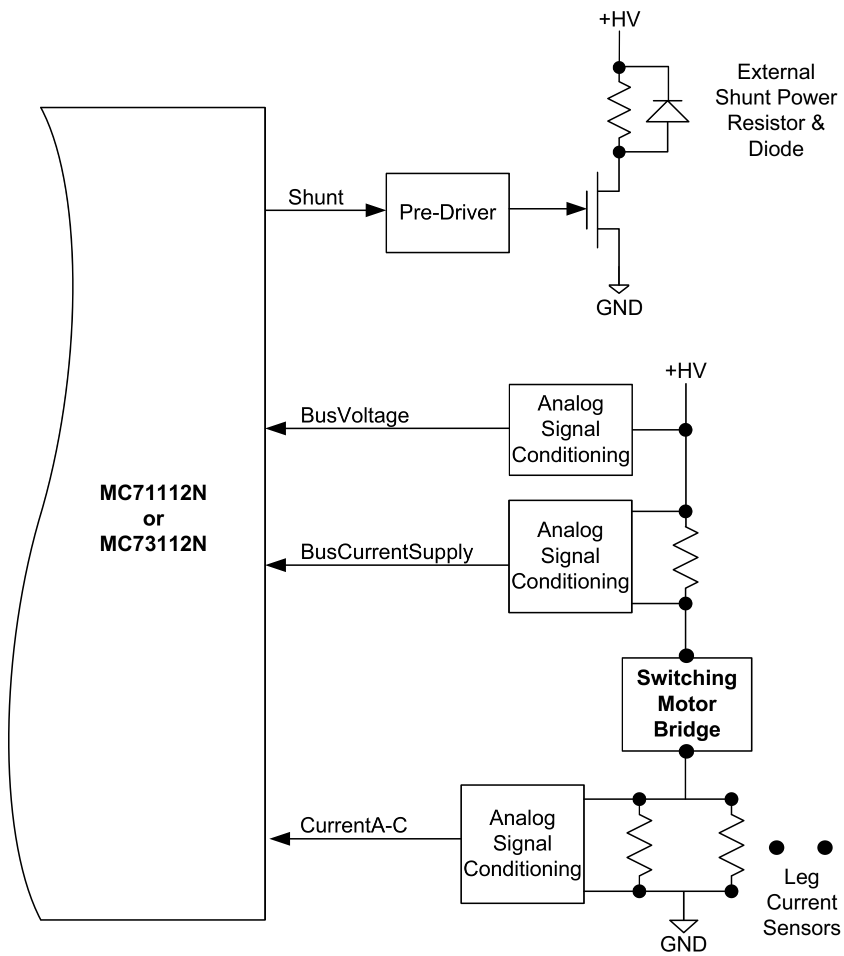


Figure 5-4 shows the DC bus monitoring circuitry used with the DK73112N board. This circuitry is designed to work with the MC71112N and MC73112N's DC bus management and protection logic. These functions include overcurrent protection, over and under voltage detection, and ground fault detection.

5.5.1 DC Bus Current Monitoring

DK73112N monitors both the supply and return DC bus current to detect overcurrent conditions including: line-to-line, line-to-power supply, and line-to-ground short circuits.

5.5.1.1 DC Bus Current Monitoring Scale Values

The following DC bus current monitoring scale values for the DK73112N board are required to specify or display current in amps:

Parameter	Value & Units	Comments
Leg Current Return Conversion	.733 mA/count	This value is used to calculate current in amps.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board the above conversion factor(s) will be used automatically.

5.5.1.2 DC Bus Current Limits

The following DC bus current limits are required to safely operate the DK73112N board:

Parameter	Value & Units	Comments
Bus Current Return Limit	20 A	To ensure safe operation of the DK73112N this parameter must be set to this value.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board these required settings will be made automatically.

5.5.2 DC Bus Voltage Monitoring

DK73112N monitors the main DC bus voltage for overvoltage and undervoltage conditions. These thresholds are user-settable within the voltage operating range of the drive.

5.5.2.1 DC Bus Voltage Monitoring Scale Values

To monitor the DC bus voltage in units of volts it is necessary to know the DC bus voltage scale factor. The following table provides this value:

Parameter	Value & Units	Comments
Bus Voltage Conversion	1.424 mV/count	This value is used to calculate voltage in volts.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board the above conversion factor(s) will be used automatically.

5.5.2.2 DC Bus Voltage Limits

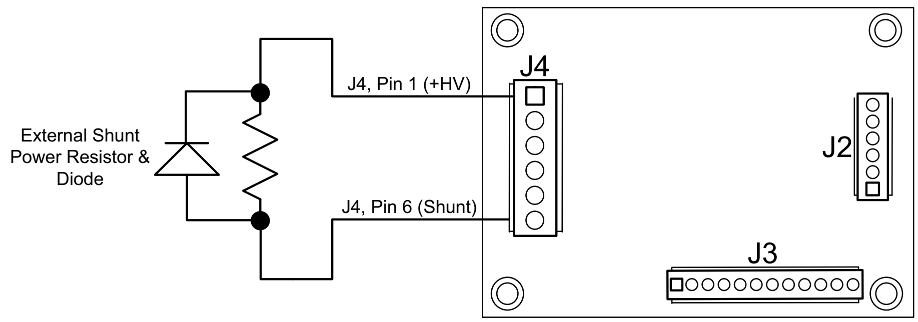
The following DC bus voltage limits are required to safely operate the DK73112N board:

Parameter	Value & Units	Comments
Undervoltage Limit	10.0V	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or higher.
Overvoltage Limit	52.0V	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or lower.

While the above settings determine safe limits for the on-board amplifier, most applications will set these limits according to the supplied HV bus voltage. A general guideline is to set these values above and below the planned supply voltage by approximately 20 %. For example if the supply voltage is 24 V, under and voltage limits of 18 V and 30 V respectively may be good values to use.

5.5.3 Shunt Resistor & Diode

Figure 5-5: Wiring to External Shunt Resistor & Diode



J4 provides a connection for a shunt resistor and diode that may be used to regulate overvoltage conditions on the DC bus. Such conditions can occur during deceleration of a motor with a large inertia. The wiring connections to the DK73112N for these external shunt components is shown in Figure 5-5.

For more information on using the shunt function see Section 4.3, "HV Shunt Regulation Operation."

5.6 Miscellaneous Signal Processing

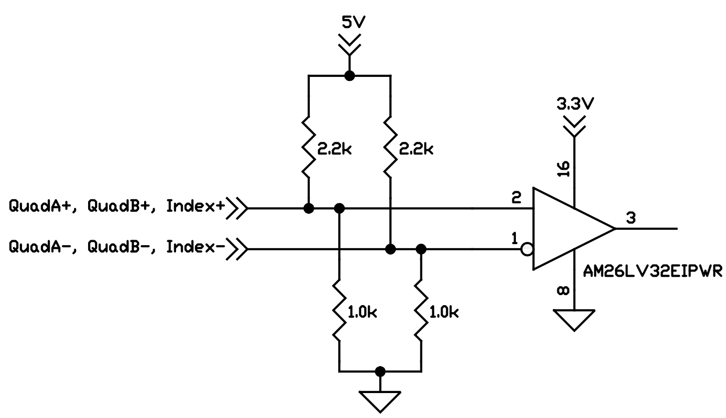
The following sections detail the signal processing circuitry provided by the DK73112N board for various signal connections.

5.6.1 Encoder Inputs

The DK73112N's encoder input signals provide processing of quadature encoder A, B feedback along with an Index signal. By default a valid index is recognized when Index is low however the interpretation of this signal as well as the A, B quadrature signals can be user programmed.

The differential input circuitry for the encoder A, B and Index signals is shown in Figure 5-6. This circuit accepts both differential and single-ended signals in the range of 0 – 5V. For single-ended operation, only the positive connection is used and the negative connection is left unconnected.

Figure 5-6: Main Encoder Input Circuitry



5.6.2 Hall Inputs

Hall signals are used with Brushless DC motors. They are used to directly commutate the motor in 6-step commutation mode or to provide an absolute phase reference for sinusoidal commutation. By default Hall sensors are defined as being on when their signal is high however this can be user programmed.

The input buffer for the Hall A, B and C signals is shown in Figure 5-7. This circuit accepts signals in the range of 0 – 24V and has TTL compatible, Schmitt trigger thresholds. It has a pull-up to 5V to allow direct interfacing to open collector sources without the need for an external pull-up resistor and an R-C low pass filter to reject noise.

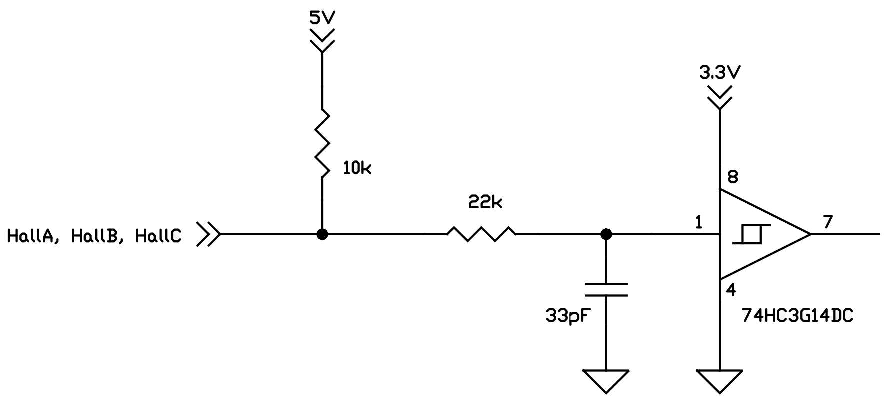


Figure 5-7: Hall Input Circuitry

5.6.3 Analog Command Input

AnalogCmd+ and AnalogCmd– are differential +/–10V analog inputs that provide a direct torque command to the MC71112N or MC73112N.

The analog signal conditioning circuit is shown in Figure 5-8. This circuit accepts +/–10V differential signal and scales it between 0 and 3.3V with an offset of 1.65V representing a zero input. The 680pF capacitors provide filtering with a bandwidth of 23.4kHz.

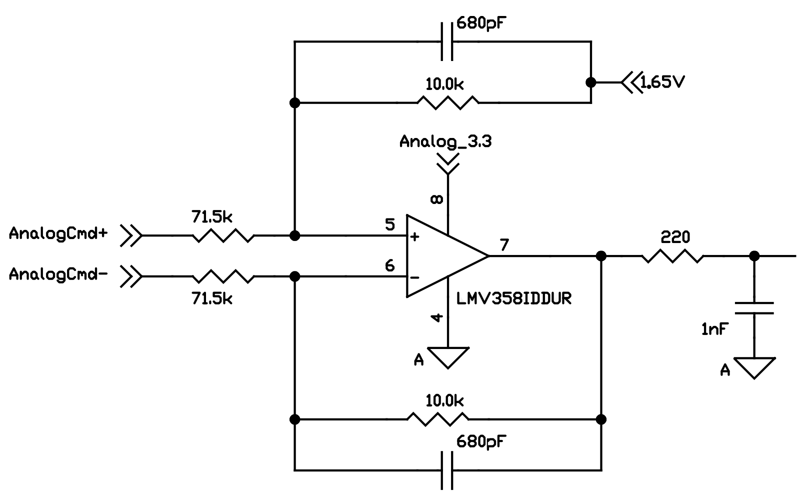


Figure 5-8: AnalogCmd Input Circuitry

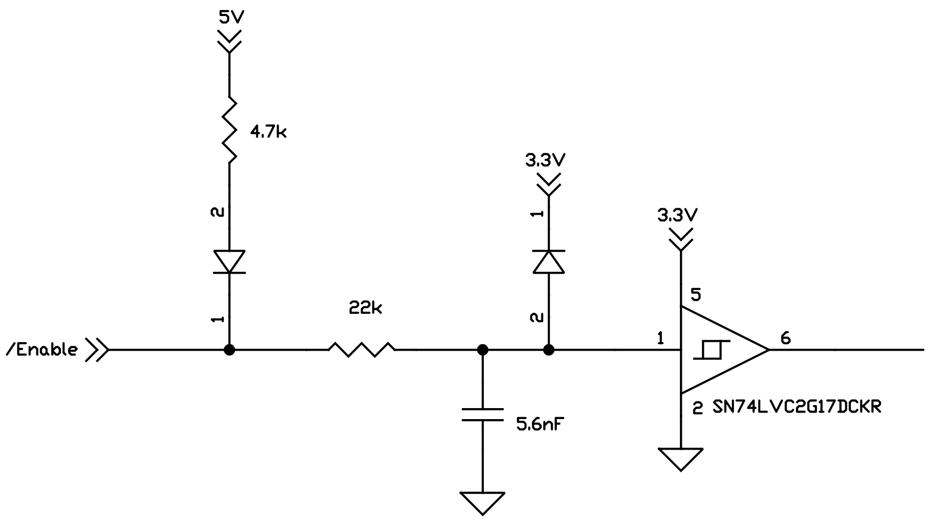
5.6.4 Enable & FaultOut

These dedicated signals are typically used to implement a safety interlock between the DK73112N board and other control portions of the system. /Enable is an active-low input that must be tied or driven low for the DK73112N output stage to be active.

FaultOut indicates a serious problem. When DK73112N is operating properly FaultOut is inactive. The polarity of this signal is fixed and cannot be user programmed.

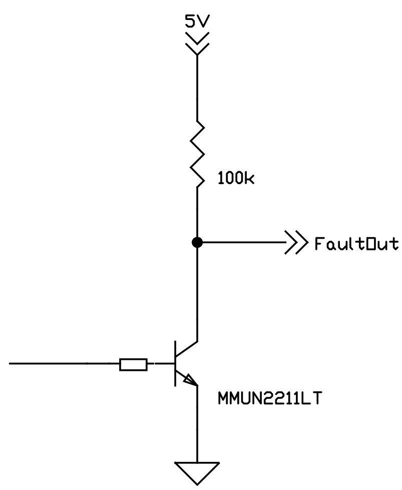
The input buffer for the /Enable input is shown in Figure 5-9. This circuit accepts signals in the range of 0 – 24V and has TTL compatible, Schmitt trigger thresholds. It has a pull-up to 5V to allow direct interfacing to open collector enable sources without the need for an external pull-up resistor and a 1.3 kHz R-C low pass filter to reject noise.

Figure 5-9: Enable Input Circuitry



The output driver for FaultOut is shown in Figure 5-10. This circuit can continuously sink over 100 mA.

Figure 5-10: FaultOut Circuitry



5.6.5 SPI (Serial Peripheral Interface)

The DK73112N board supports an SPI (Serial Peripheral Interface) connection for direct digital torque command input via its J5 connector.

The circuit below shows the DK73112N's SPI signal processing circuitry.

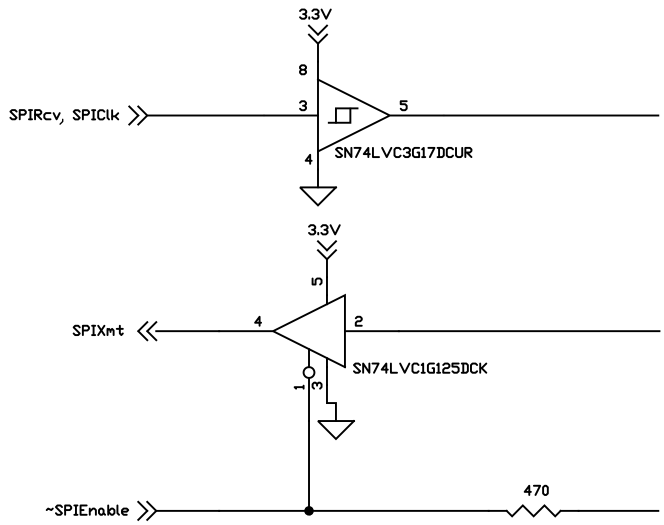


Figure 5-11: SPI Signal Circuitry

5.7 DK73112N Board Connector Reference

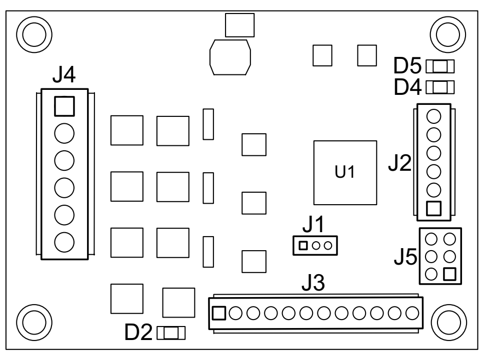


Figure 5-12: DK73112N Board Component Location

There are five user-accessible connectors on the DK73112N board. See Figure 5-12 for the specific locations of the connectors. The connectors are identified in the following table:

Label	Description
J4	HV Power & Motor Drive Connector
J3	Feedback Connector
J2	I/O Signals Connector
J1	Serial Connector
J5	SPI Connector
D2, D4, D5	AmplifierEnable, Fault and power LED indicators (respectively)

5.7.1 HV Power & Motor Drive Connector (J4)

The DK73112N connector (J4) accepts HV input voltage in the range of +12 to 48 VDC, provides the motor winding connections, and provides a shunt power resistor connection. This connector is a terminal block.

Pin #	Signal Name	Description
	J4 - HV Power & Motor Drive Connector	
1	HV	Provides DC power to the board and on-board switching amplifier
2	GND	Ground
3	Motor A	Motor output signal A+
4	Motor B	Motor output signal A–
5	Motor C	Motor output signal B+
6	Shunt	Shunt output

5.7.2 Feedback Connector (J3)

The Feedback Connector (J3) provides connections to the motor encoder and Hall sensors. The Feedback Connector consists of a 2.54 mm pitch 12-circuit terminal block.

Pin #	Signal Name	Description
	J3 - Feedback Connector	
1	Vcc	+5V output
2	GND	Ground
3	QuadA+	Quadrature A+ encoder input
4	QuadA–	Quadrature A– encoder input
5	QuadB+	Quadrature B+ encoder input
6	QuadB–	Quadrature B– encoder input
7	Index+	Index+ input
8	Index–	Index– input
9	HallA	HallA signal input
10	HallB	HallB signal input
11	HallC	HallC signal input
12	GND	Ground

5.7.2.1 Notes on Encoder Connections

Encoder inputs may be connected differentially, with two wires for QuadA, QuadB, and Index signals, or with just one wire per signal. If single-ended encoders are used, connect encoder signals to the positive encoder input only. The negative input may remain unconnected.

The following tables show this:

Encoder connections when using differential encoder input:

Signal	J3 Feedback Connector
QuadA+	J3-3
QuadA-	J3-4
QuadB+	J3-5
QuadB-	J3-6
Index+	J3-7
Index-	J3-8
Vcc	J3-1
GND	J3-2

Encoder connections when using single-ended encoder input:

Signal	J3 Feedback Connector
QuadA +	J3-3
QuadB +	J3-5
Index +	J3-7
Vcc	J3-1
GND	J3-2

5.7.3 I/O Signals Connector (J2)

The I/O Signals Connector provides the analog AnalogCmd inputs along with the Enable input, the Brake input, and the FaultOut output. This connector is a 2.54 mm pitch 8-circuit terminal block.

Pin #	Signal Name	Description
		J2 - I/O Signals Connector
1	Enable	Active low Enable digital input signal
2	FaultOut	Active high digital FaultOut output signal
3	Brake	Brake input signal
4	GND	Ground
5	AnalogCmd−	AnalogCmd− torque command input
6	AnalogCmd +	AnalogCmd + torque command input

5.7.4 Serial Connector (J1)

The 3-pin Serial Connector (J1) provides a UART serial connection to the MC71112N or MC73112N ICs for purposes of NVRAM programming, diagnostics, or application development. This connector directly mates with the 3-pin cable, P/N: Cable-USB-3P, included in each DK73112N Developer Kit. This connector is a 3-pin 2 mm single-row header.

Pin #	Signal Name	Description
		J1 - Serial Connector
1	SrlXmt	Serial transmit output
2	SrlRcv	Serial receive input
3	GND	Ground

5.7.5 SPI Connector (J5)

The SPI Connector (J5) provides connection to the SPI (Serial Peripheral Interface) bus. This bus may be used to command a torque value in a digital 16-bit format. This connector is a 6-pin .1 inch dual-row header.

Pin #	Signal Name	Description
		J5 - SPI Connector
1	SPIXmt	SPI bus synchronous transmit signal output
2	SPIRcv	SPI bus synchronous receive signal input
3	SPIClock	SPI bus synchronous clock input
4	SPIEnable	Host SPI bus active low enable signal input
5	GND	Ground
6	NC	No Connect

5.8 Absolute Maximum Ratings

HV voltage range:	0V to +52V

5.9 Environmental and Electrical Ratings

Storage temperature:	−40 to +125 degrees C (−40° F to +257°F)
Operating temperature:	0 to +70 degrees C (32° F to +158°F)
HV power requirement:	+12V to + 48V operating range
Motor amplifier continuous current limit*:	5.0 A
Motor amplifier peak current limit:	10.0 A
Digital input voltage range:	0V to 5V, TTL thresholds

* *Current rating at 25 C ambient and with 110 CFM air flow on board. Significantly higher currents are possible with additional heat sinking. Contact your PMD representative for details.*

5.10 DK73112N Control Settings Quick Reference

To use the DK73112N circuitry safely and to view results in physical units such as amps and volts via Pro-Motion various limits and conversion constants must be specified.

These parameters are automatically provided when using Pro-Motion's Axis Wizard setup sequence, however they are listed below for reference.

Parameter	Value & Units	Comments
Motor Output Mode	PWM High/Low	Sets the motor output mode to PWM High/Low for operation with the on-board amplifier.
PWM Switching Frequency	20 kHz	This setting is motor-specific. Larger motors (some NEMA 23 and most NEMA 34) should be set for 20 kHz. Smaller motors may use 40, 80, or 120 kHz to maximize current control accuracy and minimize heat generation.
PWM Dead Time	540 nsec	To ensure correct operation of the DK73112N's on-board switching amplifier this parameter must be set to this value.
PWM Refresh Time	2,000 nSec	To ensure sufficient time to recharge the DK73112N's on-board amplifier high side switches this parameter must be set to this value.
PWM Refresh Period	8 cycles	To ensure sufficient time to recharge the DK73112N's on-board amplifier high side switches this parameter must be set to this value
PWM Signal Sense, high	Active High	To ensure correct operation of the DK73112N's on-board amplifier all high PWM outputs must be set to active high.
PWM Signal Sense, low	Active Low	To ensure correct operation of the DK73112N's on-board amplifier all low PWM outputs must be set to active low.
Minimum Current Read Time	2,000 nSec	To ensure sufficient minimum current read time with the DK73112N's on-board amplifier when driving BLDC motors this parameter must be set to this value.
Leg Current Conversion	.733 mA/count	This value specifies the conversion factor so that the leg current can be traced and displayed in amps.
Foldback Continuous Current Limit	5.0 A	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or lower.

Parameter	Value & Units	Comments
Foldback Maximum Energy Limit	50 A^2sec	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value.
Bus Current Return Limit	20 A	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or lower.
Bus Voltage Conversion	1.424 mV/count	This value specifies the conversion factor so that the DC bus voltage can be traced and displayed in volts.
Undervoltage Limit	10.0V	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or higher.
Overvoltage Limit	52.0V	To ensure safe operation of the DK73112N's on-board amplifier this parameter must be set to this value or lower.

This page intentionally left blank.

6. DK78113 Board Reference

6

In This Chapter

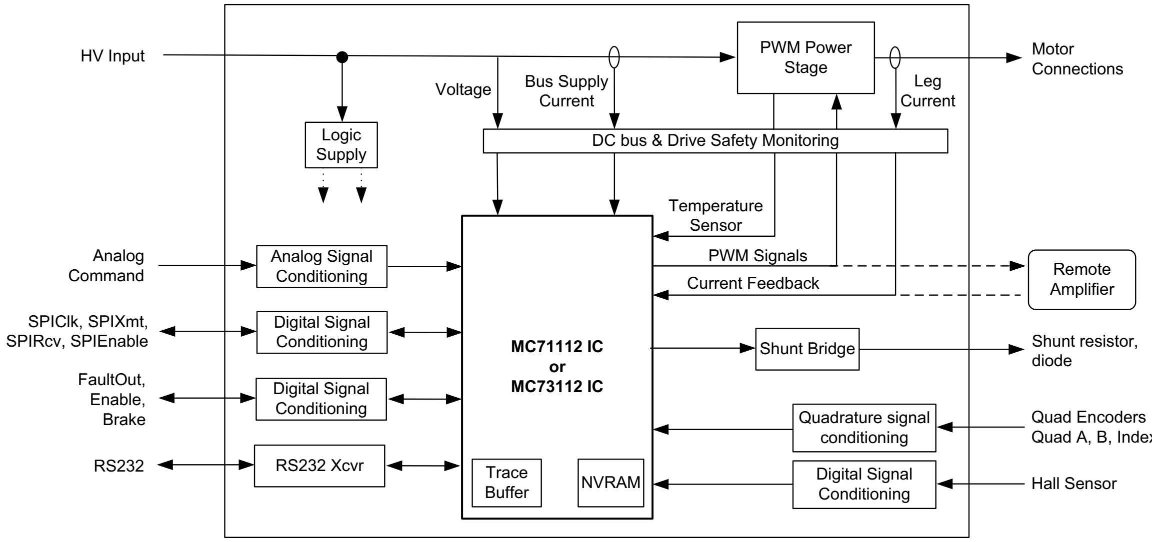
6.1 DK78113 Internal Block Diagram

The DK78113 Developer Kit board provides a complete functioning MC71112 and MC73112 IC exerciser and development system. It directly interfaces to a host computer using USB to serial communication, and provides all power, motor drive, control, and feedback signals required to drive a DC Brush or Brushless DC motor.

The DK78113 consists of several subsystems including the MC71112 or MC73112 ICs themselves, a high performance MOSFET-based motor amplifier, a DC Bus conditioning and monitoring system, and various other circuitry.

The following sections describe the major sections of the DK78113 board when used with the MC71112 or MC73112 Torque Control ICs. For a complete description of the MC7x112 and MC7x112N ICs, see the *MC7x112 Torque Control IC User Guide*.

Figure 6-1: DK78113 Internal Block Diagram



Note that the internal block diagram above reflects the functions of the DK78113 board when a MC71112 or MC73112 Torque Control IC is installed. For an internal block diagram of the DK78113 board when a MC71113 or MC73113 Velocity Control IC is installed refer to the *MC7x113 Developer Kit User Manual*.

6.2 Communication Port

The DK78113 supports the RS232 signaling standard at its J3 RS232 Connector.

The serial interface can be operated at the communication settings shown in the following table:

Parameter	Range	Default
Baud rate	1,200 to 460,800	57,600
Parity	None, even, odd	None
# data bits	8	8
# stop bits	1, 2	1

All DK78113 communication functions are controlled by the MC7x112 IC. For information on serial port functions see the *MC7x112 Torque Control IC User Guide*. In addition, see Section 4.2, "Changing the Active Serial Settings" which shows hot to change the serial port settings using Pro-Motion.

6.3 Switching Motor Amplifier

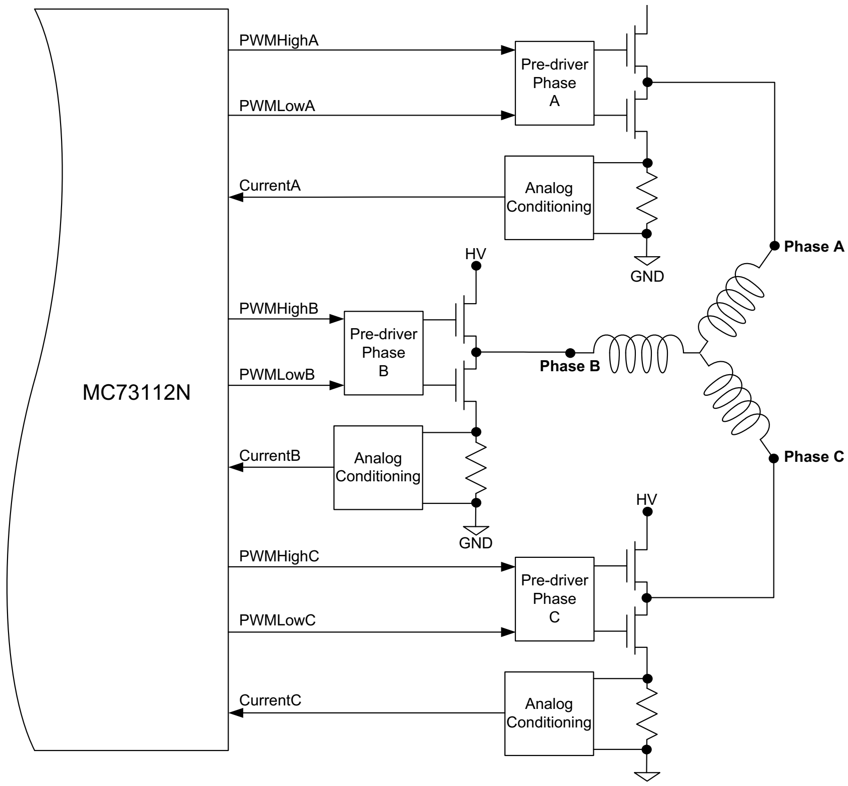
The DK78113 contains a high-efficiency MOSFET power stage with PWM input control and leg current feedback. Brushless DC motors are driven in a 3-phase bridge configuration consisting of 6 MOSFETs and 3 leg current sensors. DC Brush motors are driven in an H-Bridge configuration consisting of 4 MOSFETs and 2 leg current sensors

To operate the DK78113's on-board amplifier the JP4 and JP5 jumpers must be installed in the 1-2 position.

6.3.1 Brushless DC Motor Drive

Figure 6-2 shows the arrangement of the DK78113's amplifier stage when used with Brushless DC motors.

Figure 6-2: Brushless DC Motor Bridge Configuration



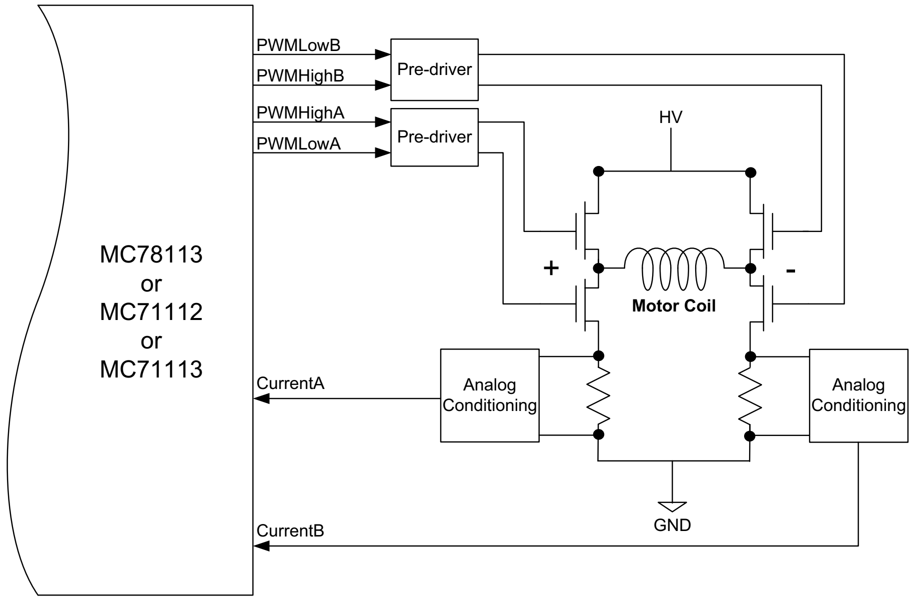
As shown in the table below six PWM output signals and three analog feedback signals interface between the MC7x112 IC and the DK78113's switching amplifier.

Signal	Description
PWMHighA	Digital high side drive output for motor phase A
PWMLowA	Digital low side drive output for motor phase A
PWMHighB	Digital high side drive output for motor phase B
PWMLowB	Digital low side drive output for motor phase B
PWMHighC	Digital high side drive output for motor phase C
PWMLowC	Digital low side drive output for motor phase C
CurrentA	Analog input containing the current flow through the low side of the switching bridge for phase A.
CurrentB	Analog input containing the current flow through the low side of the switching bridge for phase B.
CurrentC	Analog input containing the current flow through the low side of the switching bridge for phase C.

6.3.2 DC Brush Motor Drive

Figure 6-3 shows the arrangement of the DK78113's amplifier stage when used with DC Brush motors.

Figure 6-3: DC Brush Motor Bridge Configuration



As shown in the table below four PWM output signals and two analog feedback signals interface between the MC78113 IC and the DK78113's switching amplifier.

Signal	Description
PWMHighA	Digital high side drive output for positive coil terminal
PWMLowA	Digital low side drive output for positive coil terminal
PWMHighB	Digital high side drive output for negative coil terminal
PWMLowB	Digital low side drive output for negative coil terminal
CurrentA	Analog input containing the current flow through the positive leg of the bridge
CurrentB	Analog input containing the current flow through the negative leg of the bridge

6.3.3 Amplifier-Related Settings

There are a number of MC7x112 IC settings which are used to set or control various aspects of the DK78113's on-board switching amplifier and related current sense circuitry. If a remote amplifier (via the J4 connector) is connected the value of these settings will likely change.

The following table shows the recommended (or required) settings for amplifier-related parameters:

Parameter	Value & Units	Comment
Motor Output Mode	PWM High/Low	Sets the motor output mode to PWM High/Low for operation with the on-board amplifier. For remote amplifier operation PWM High/Low or sign/magnitude PWM may be selected.
PWM Switching Frequency	20 kHz	This setting is motor-specific. Higher inductance motors should be set for 20 kHz. Lower inductance motors may use 40, 80, or 120 kHz to reduce current ripple and minimize heat generation.
PWM Dead Time	540 nSec	To ensure correct operation of the DK78113's on-board switching amplifier this parameter must be set to this value.
PWM Refresh Time	2,000 nSec	To ensure sufficient time to recharge the DK78113's on-board amplifier high side switches this parameter must be set to this value.

Parameter	Value & Units	Comment
PWM Refresh Period	8 cycles	To ensure sufficient time to recharge the DK78113's on-board amplifier high side switches this parameter must be set to this value.
PWM Signal Sense	Active High	To ensure correct operation of the DK78113's on-board amplifier all PWM outputs must be set to active high.
Minimum Current Read Time	2,000 nSec	To ensure sufficient minimum current read time with BLDC motors using the DK78113's on-board amplifier this parameter must be set to this value.

Note that if Pro-Motion Axis Wizard is used for setup of the DK78113 board all of the required settings will be made automatically.

6.3.4 DK78113 Current Scale Value

To control or trace motor current in units of amps it is useful to know the DK78113's amplifier-specific current conversion factor. The following table shows this:

Parameter	Value & Units	Comments
Leg Current Conversion	.733 mA/count	This value is used to calculate current in amps.

Note that if Pro-Motion Axis Wizard is used for setup of the DK78113 board the above conversion factor(s) will be used automatically to correctly display physical units such as amps.

6.4 Drive Protection and Control Signals

6.4.1 I^2t Current Foldback Protection

MC7x112 ICs use current feedback to implement I^2t current limiting. This feature protects the on-board amplifier by controlling its ability to operate above specific selected continuous current ratings.

When the current loop is enabled and the I^2t energy limit is exceeded, the MC7x112 will automatically fold back the phase currents to a user programmable continuous current limit value. Alternatively, the IC can be configured to fault and disable the output stage when the I^2t energy limit is exceeded.

6.4.1.1 I^2t Limits

The following I^2t limits are required to safely operate the DK78113 board.

Parameter	Value & Units	Comments
Foldback Continuous Current Limit	5.0 A	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value or lower
Brushless DC motor: Foldback Total Energy Limit	125 A^2sec	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value or lower

Note that if Pro-Motion Axis Wizard is used for setup of the DK78113 board the above conversion factor(s) will be used automatically to correctly display physical units such as amps.

For use with motors that have current and energy limits lower than those specified above, it is recommended that lower settings be used.

These limits are designed to be safe for operation of the DK78113's on-board amplifier when cooled with a 110 CFM (or higher) fan and with an ambient room temperature of 25 degrees C. For use without a fan, or in higher ambient temperature environments, these values should be lowered.

If special heatsinking is used the DK78113 board is capable of driving motors at 7 amps continuously or higher, in which case these limits can be increased. Contact PMD for details.

If a remote amplifier (via the J4 connector) is used the value of these settings will likely change.

6.4.2 Overtemperature Protection

The DK78113 uses a temperature sensor to continuously monitor the temperature of the on-board power MOSFETs.

6.4.2.1 Converting Temperature Readings into Degrees C

The MC7x112 inputs temperature readings via its Temperature analog input and performs related functions such as over temperature checking without converting the readings from the attached temperature sensor into degrees C. It does this by comparing temperature in units of 'counts,' meaning the numerical value of the Temperature signal converted by the MC7x112's on-chip A/D.

While using Pro-Motion however, it is convenient to be able to view and trace the amplifier temperature in units of degrees C. This is accomplished using a conversion table, which is called up automatically from the Axis Wizard. If using Pro-Motion with a user-designed board a conversion table file specific to that amplifier design can be created and manually specified. For details see Appendix C, "Temperature Conversion Tables."

6.4.2.2 Temperature Limit

The following temperature limit is required to safely operate the DK78113 board.

Parameter	Value & Units	Comments
Temperature limit	75.0 C	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value.

Note that if Pro-Motion Axis Wizard is used for setup of the DK78113 board all of the required settings will be made automatically.

The DK78113 board's temperature sensor is located on the DK78113 board and therefore will not function correctly when a remote amplifier is used.

6.5 DC Bus

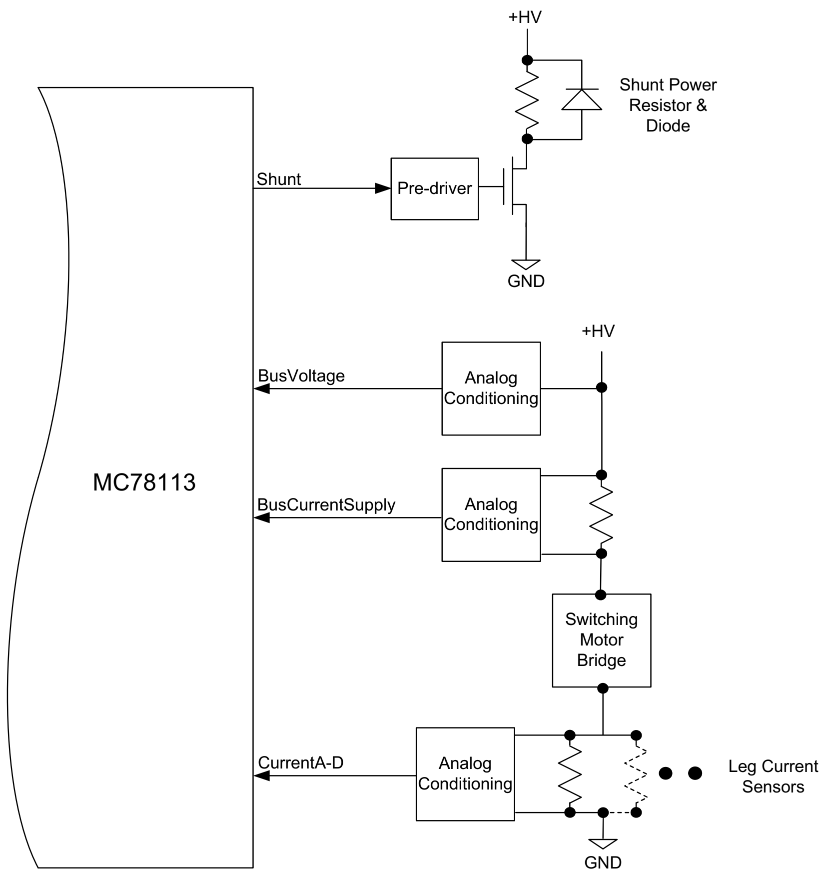


Figure 6-4: DC Bus Monitoring Circuitry

Figure 6-4 shows the DC bus monitoring circuitry used with the DK78113 board. This circuitry is designed to work with the MC7x112's DC bus management and protection logic. These functions include overcurrent protection, over and under voltage detection, overcurrent detection, and shunt resistor control.

6.5.1 DC Bus Current Monitoring

The DK78113 monitors both the positive and negative DC bus current to detect overcurrent conditions including: line-to-line, line-to-power supply, and line-to-ground short circuits.

6.5.1.1 DC Bus Current Monitoring Scale Values

The following DC bus current monitoring scale values for the DK78113 board are required to specify or display current in amps:

Parameter	Value & Units	Comments
Bus current supply conversion	.505 mA/count	This value is used to calculate current in amps.
Leg current return conversion	.733 mA/count	This value is used to calculate current in amps.

Note that if Pro-Motion Axis Wizard is used for setup of the DK78113 board the above conversion factor(s) will be used automatically to correctly display physical units such as amps.

6.5.1.2 DC Bus Current Limits

The following MC7x112 DC bus current limits are required to safely operate the DK78113 board:

Parameter	Value & Units	Comments
Bus Current Supply Limit	20.0 A	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value.
Bus Current Return Limit	20.0 A	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value.

Note that if Pro-Motion Axis Wizard is used for setup of the DK78113 board all of the required settings will be made automatically.

6.5.2 DC Bus Voltage Monitoring

DK78113 monitors the main DC bus voltage for overvoltage and undervoltage conditions. These thresholds are user-settable within the voltage operating range of the drive.

6.5.2.1 DC Bus Voltage Monitoring Scale Values

To monitor the DK78113 DC bus voltage in units of volts it is necessary to know the DC bus voltage scale factor. The following table provides this value:

Parameter	Value & Units	Comments
Bus Voltage Conversion	1.424 mV/count	This value is used to calculate voltage in volts.

Note that if Pro-Motion Axis Wizard is used for setup of the DK73112N board the above conversion factor(s) will be used automatically to correctly display physical units such as amps.

6.5.2.2 DC Bus Voltage Limits

The following DC bus voltage limits are required to safely operate the DK78113 board:

Parameter	Value & Units	Comments
Undervoltage Limit	10.0V	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value or higher.
Overvoltage Limit	60.0V	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value or lower.

While the above settings determine safe limits for the on-board amplifier, most applications will set these limits according to the supplied HV bus voltage. A general guideline is to set these values above and below the planned supply voltage by approximately 20%. For example if the supply voltage is 24 V, under and voltage limits of 18 V and 30 V respectively may be good values to use.

6.5.3 Shunt Resistor & Diode

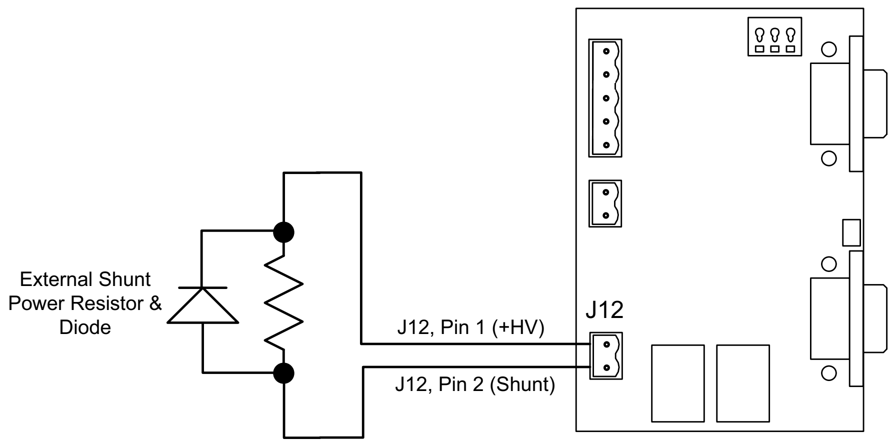


Figure 6-5: Wiring to External Shunt Resistor & Diode

J12 provides a connection for a shunt resistor and diode that may be used to regulate overvoltage conditions on the DC bus. Such conditions can occur during deceleration of a motor with a large inertia. The wiring connections to the DK78113 board for these external shunt components is shown in Figure 6-5.

For more information on using the shunt function see Section 4.3, "HV Shunt Regulation Operation."

6.6 Connecting to a Remote Amplifier

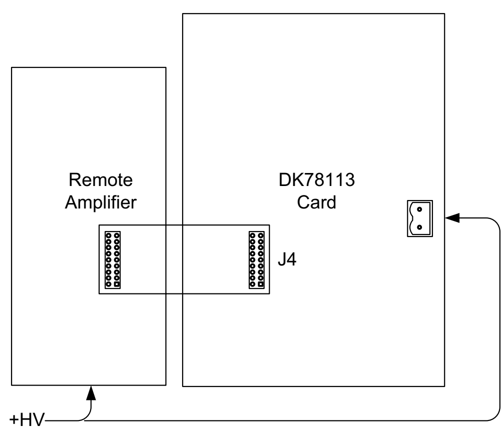
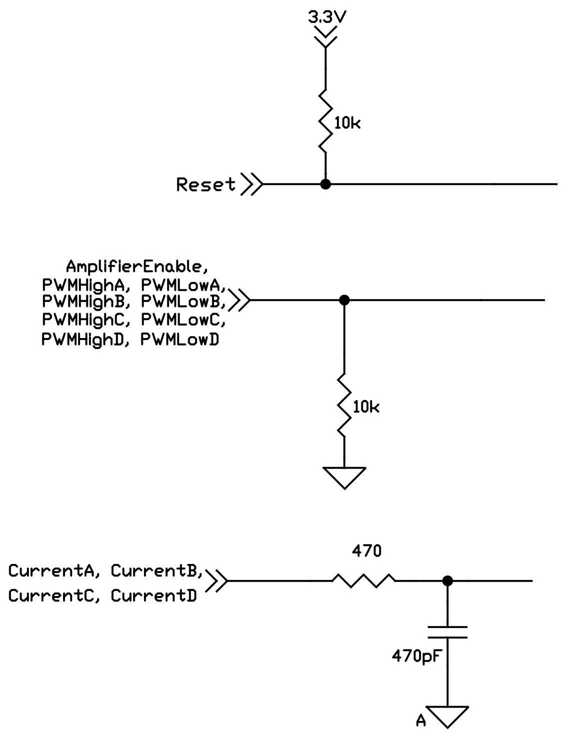


Figure 6-6: DK78113 to Remote Amplifier Board Connections

In addition to the DK78113's on-board switching amplifier, J4 provides connections for an external remote switching amplifier. This connector is typically used to interface with user-designed amplifiers that service higher power motors, or motors or actuators that require a special bridge configuration. Such an external amplifier connection is shown in Figure 6-6. To operate a remote amplifier jumpers JP4 & JP5 must be in the 2-3 position.

The DK78113's Remote Amplifier Connector signal processing circuitry is shown in Figure 6-7. There is minimal processing other than a low pass filter with a frequency of approximately 700 kHz on the analog input circuitry.

Figure 6-7: Remote Switcher Circuits



For a description of setting up and operating the DK78113 with a remote amplifier see Section 4.4, "Operating with an External (Remote) Amplifier."

> For remote amplifier operation JP4 and JP5 must be in the 2-3 position.

6.7 Miscellaneous Signal Processing

The following sections detail the signal processing circuitry provided by the DK78113 board for various signal connections.

6.7.1 Encoder Inputs

MC7x112's encoder input signals are input via the DK78113 board's J7 Feedback Connector and provide processing of quadature encoder A, B feedback along with an Index signal. By default a valid index is recognized when Index is low however the interpretation of this signal as well as the A, B quadrature signals can be user programmed.

The differential input circuitry for the encoder A, B and Index signals is shown in Figure 6-8. This circuit accepts both differential and single-ended signals in the range of 0 – 5V. For single-ended operation, only the positive connection is used and the negative connection is left unconnected.

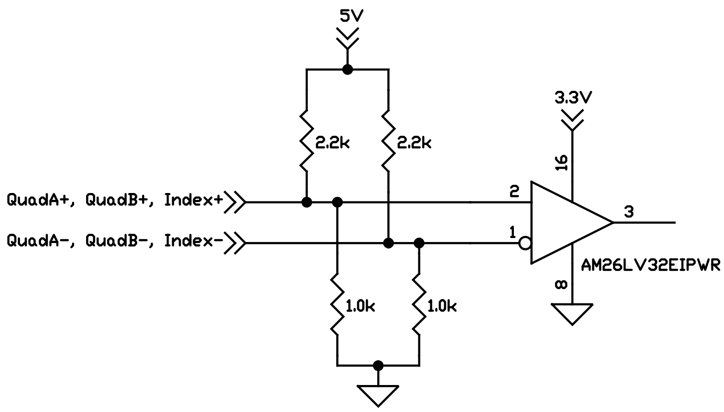


Figure 6-8: Main Encoder Input Circuits

6.7.2 Hall Inputs

Hall signals are used with Brushless DC motors. They are used to directly commutate the motor in 6-step commutation mode or to provide an absolute phase reference for sinusoidal commutation. By default Hall sensors are defined as being on when their signal is high however this can be user programmed.

The DK78113 board inputs Hall sensor signals via its J7 Feedback Connector. The input buffer for the Hall A, B and C signals is shown in Figure 6-9. This circuit accepts signals in the range of 0 – 24V and has TTL compatible, Schmitt trigger thresholds. It has a pull-up to 5V to allow direct interfacing to open collector sources without the need for an external pull-up resistor and an R-C low pass filter to reject noise.

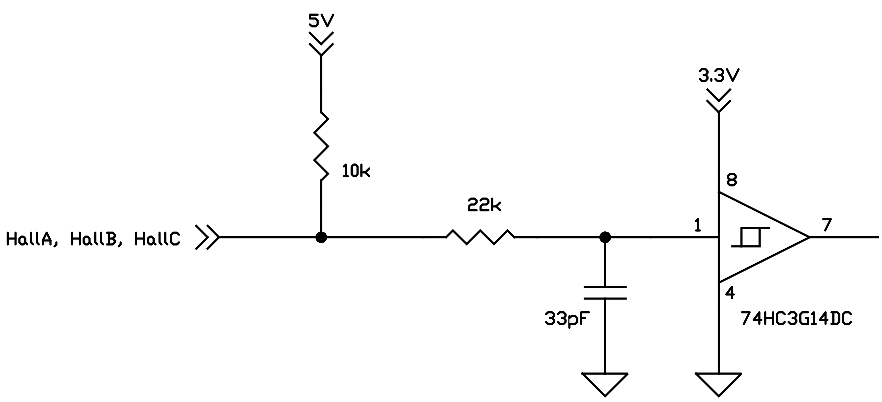


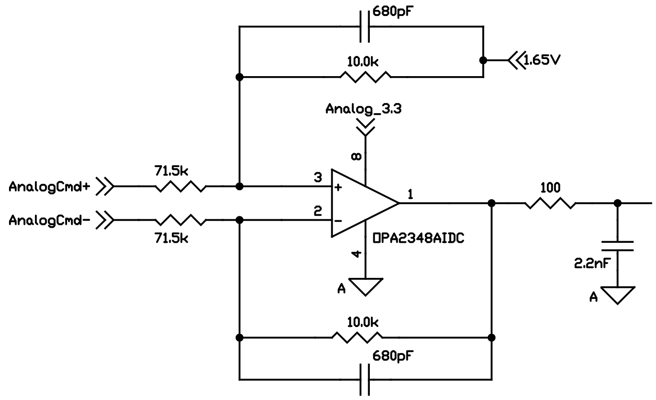
Figure 6-9: Hall Input Processing Circuits

6.7.3 Analog Command Input

AnalogCmd, input via the J5 Analog Command Connector, is a differential +/–10V analog input that provides a direct torque command to the MC7x112 IC.

The analog signal conditioning circuit for AnalogCmd is shown in Figure 6-10. This circuit accepts +/–10V differential signal and scales it between 0 and 3.3V with an offset of 1.65V representing a zero input. The 680pF capacitors provide filtering with a bandwidth of 23.4kHz.

Figure 6-10: Analog Command Inputs Processing Circuit



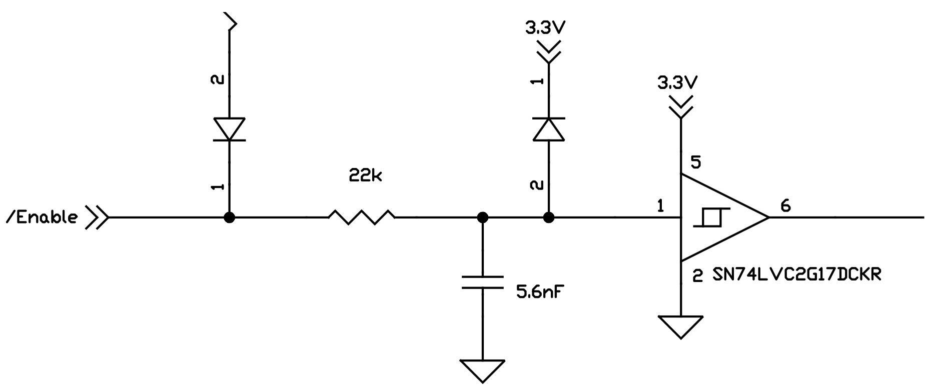
6.7.4 Enable & FaultOut

These dedicated signals are typically used to implement a safety interlock between the DK78113 board and other control portions of the system. /Enable is an active-low input that must be tied or driven low for the DK78113 output stage to be active.

FaultOut indicates a serious problem. When the DK78113 is operating properly FaultOut is inactive. The polarity of these signals is fixed and cannot be changed via user command.

The Enable and FaultOut signals are located on the J9 I/O Signals Connector. The input buffer for the /Enable input is shown in Figure 6-11. This circuit accepts signals in the range of 0 – 24 V and has TTL compatible, Schmitt trigger thresholds. It has a pull-up to 5V to allow direct interfacing to open collector enable sources without the need for an external pull-up resistor and a 1.3 kHz R-C low pass filter to reject noise.

Figure 6-11: Enable Input Circuit



The output driver for FaultOut is shown in Figure 6-12. This circuit can continuously sink over 100 mA and source 4mA from a pull-up resistor to 5 V. The diode in series with the pull-up resistor allows loads powered from up to 24 VDC to be switched. The FET driver is internally protected from shorts up to 30V.

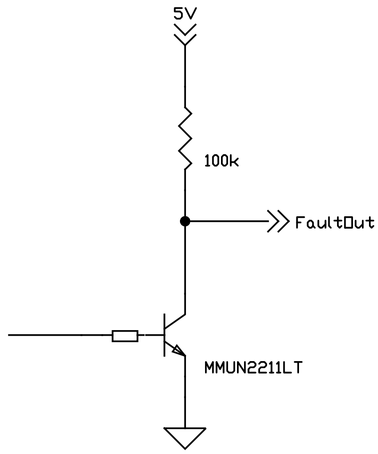


Figure 6-12: FaultOut Circuit

6.8 Jumpers

The following table details the available DK78113 jumper settings when a MC7x112 IC is installed:

Jumper ID	Factory Default Setting	Setting &Description	
JP4, JP5	1-2 (on-board amplifier)	1-2	Installing jumpers at 1-2 for JP4 and JP5 configures the DK78113 for operation of the on-board amplifier.
		2-3	Installing jumpers at 2-3 for JP4 and JP5 disables the on-board amplifier, and configures the DK78113 for operation with a user-designed amplifier via the J4 Remote Amplifier Connector.

6.9 DK78113 Board Connector Reference

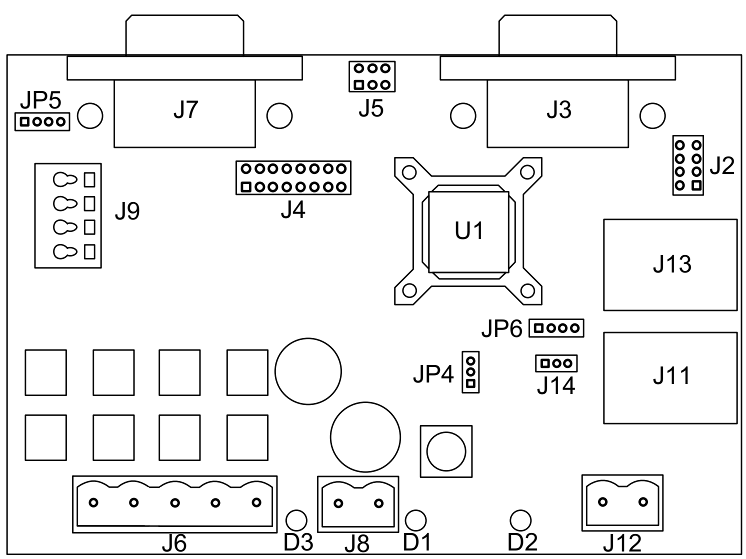


Figure 6-13: DK78113 Board Component Location

There are 10user-accessible connectors on the DK78113 board when used with MC7x112 ICs. See Figure 6-13 for the specific locations of the connectors on the board. The connectors and other components are identified in the following table:

Label	Description
J8	HV Power Connector
J6	Motor Drive Connector
J7	Feedback Connector
J9	I/O Signals Connector
J4	Remote Amplifier Connector
J5	Analog Command Connector
J12	Shunt Connector
J3	RS232 Connector
J2	SPI Connector
J14	Serial Connector
D1, D2, D3	Power, fault, and amplifier enabled LED indicators (respectively)
JP4, JP5	Remote amplifier enable jumpers

6.9.1 HV Power Connector (J8)

The DK78113 board uses a dedicated 2-pin HV power connector (J8) that accepts input voltage in the range of +12 to 56 VDC. This connector is a Phoenix Contact 2-circuit pluggable terminal block connector.

Pin #	Signal Name	Description
	J8 - HV Power Connector	
1	HV	Provides DC power to the board and on-board switching amplifier
2	GND	Ground

6.9.2 Motor Drive Connector (J6)

The Motor Drive Connector (J6) provide motor output signals for use with Brushless DC or DC Brush motors. This is a Phoenix Contact 5-circuit pluggable terminal block connector.

Pin #	Signal Name	Description
	J6 - Motor Drive Connector	
1	Motor A	Motor coil A
2	Motor B	Motor coil B
3	Motor C	Motor coil C
4	NC	No Connect
5	GND	Ground

6.9.3 Feedback Connector (J7)

The Feedback Connector (J7) provides connections to various motor feedback signals. The Feedback Connector uses a 15-pin high density DB connector, which can be connected to the PMD MC-HW-05 breakout board accessory included with DK78113 board developer kits.

6.9.3.1 Feedback Connector

Pin #	Signal Name	Description
	J7 - Feedback Connector	
1	QuadA+	Quadrature A+ encoder input
2	QuadA–	Quadrature A– encoder input
3	QuadB+	Quadrature B+ encoder input
4	QuadB–	Quadrature B– encoder input
5	GND	Ground
6	Index+	Index+ input
7	Index–	Index– input
8	HallA	Hall A input
9	HallB	Hall B input
10	HallC	Hall C input
11-13	NC	No Connect
14	Vcc	+5V output
15	NC	No Connect

6.9.3.2 Notes on Encoder Connections

Encoder inputs may be connected differentially, with two wires for QuadA, QuadB, and Index signals, or with just one wire per signal. If single-ended encoders are used, connect encoder signals to the positive encoder input only. The negative input may remain unconnected.

The following tables show this:

Encoder connections when using differential encoder input:

Signal	J7 Feedback Connector
QuadA+	J7-1
QuadA–	J7-2
QuadB+	J7-3
QuadB–	J7-4
Index+	J7-6
Index–	J7-7
Vcc	J7-14
GND	J7-5

Encoder connections when using single-ended encoder input:

Signal	J7 Feedback Connector
QuadA	J7-1
QuadB	J7-3
Index	J7-6
Vcc	J7-14
GND	J7-5

6.9.4 I/O Signals Connector (J9)

The I/O Signals Connector provides an Enable input to the MC7x112 ICs as well as the FaultOut from the MC7x112.

Pin #	Signal Name	Description
	J9 - I/O Signals Connector	
1	Brake	Active low brake signal input
2	Enable	Active low Enable digital input signal
3	FaultOut	Active high digital FaultOut output signal
4	GND	Ground

6.9.5 Remote Amplifier Connector (J4)

This connector provides signals that allow an external switching amplifier to be connected to the DK78113. This connector is not used if the on-board amplifier is used. J4 is a 16-pin 0.1 inch dual-row header (P/N Samtec TSW-108-07-G-D).

If a remote amplifier is used jumpers JP4 and JP5 must be in the 2-3 position.

Pin #	Signal Name	Description
	J4 - Remote Amplifier Connector	
1	AmplifierEnable	Active high AmplifierEnable output signal
2	Reset*	Active low Reset input signal
3	PWMHighA	PWMHighA output signal
4	PWMLowA	PWMLowA output signal
5	PWMHighB	PWMHighB output signal
6	PWMLowB	PWMLowB output signal
7	PWMHighC	PWMHighC output signal
8	PWMLowC	PWMLowC output signal
9	NC	No Connect
10	NC	No Connect
11	GND	Ground
12	AGND	Analog Ground
13	CurrentA	CurrentA analog input signal
14	CurrentB	CurrentB analog input signal
15	CurrentC	CurrentC analog input signal

* *This reset signal pin is located on this connector and may be useful during development, but is not typically used by the remote amplifier.*

6.9.6 Shunt Connector (J12)

The Shunt Connector (J12) provides a dedicated high current connection to a shunt resistor or similarly functioning component in applications where it may be desirable to remove excess voltage from the DC bus.

This connector is a Phoenix Contact 2-circuit pluggable terminal block.

Pin #	Signal Name	Description
	J12 - Shunt Connector	
1	HV	+HV
2	Shunt	Switched connection to ground

6.9.7 Analog Command Connector (J5)

The Analog Command Connector (J5) provides differential +/–10V analog command inputs. This connector is a 6-pin 0.1 inch dual-row header (P/N Samtec TSW-103-07-G-D).

Pin #	Signal Name	Description
	J5 - Analog Command Connector	
1	AnalogCmd+	Command+ input
2	NC	No Connect
3	AnalogCmd–	Command– input
4	NC	No Connect
5	AGND	Analog Ground
6	AGND	Analog Ground

6.9.8 RS232 Connector (J3)

The RS232 Connector (J3) provides connections to an RS232 serial port.

J3 is a standard 9-pin female D-sub connector. The pinouts for the J3 connector are as follows:

Pin #	Signal Name	RS232
	J3 - RS232 Connector	
1	NC	No Connect
2	SrlXmt	Serial transmit output
3	SrlRcv	Serial receive input
4	NC	No Connect
5	GND	Ground
6	NC	No Connect
7	NC	No Connect
8	NC	No Connect
9	NC	No Connect

6.9.9 SPI Connector (J2)

This Host SPI Connector (J2) provides host SPI (Serial Peripheral Interface) signals used to specify a torque command. This connector is an unshrouded 8-position double-row male header, .1" spacing.

Pin #	Signal Name	Description
		J2 - SPI Connector
1	SPIXmt	SPI bus synchronous transmit signal.
2	SPIRcv	SPI bus synchronous receive signal to MC78113.
3	SPIClock	SPI bus synchronous clock input.
4	SPIEnable	Host SPI bus active low enable signal input.
5	HostInterrupt	Active low Host interrupt output can be programmed to indicate an event requiring attention from the host.
6	NC	No Connect
7	GND	Ground

6.9.10 Serial Connector (J14)

The 3-pin Serial Connector (J14) provides a UART serial connection to the MC7x112 ICs for purposes of NVRAM programming, diagnostics, or application development. This connector directly mates with the 3-pin serial cable, P/N: Cable-USB-3P, included in each DK78113 Developer Kit. This connector is a 3-pin 2 mm single-row header.

Pin	Signal	Description
		J14 - Serial Connector
1	SrlXmt	Serial transmit output
2	SrlRcv	Serial receive input
3	GND	Ground

Although the J14 Serial Connector may be used to communicate by serial to the MC7x112 IC on the DK78113 board, more commonly the J3 RS232 Connector is used for this purpose.

6.9.11 Connector Parts Reference

The following table is supplied as a reference only.

Label	Connector Name	Connector Part Number	Connector Mate
J8	HV Power	Phoenix Contact 2-circuit pluggable terminal block p/n 1924305	Phoenix Contact 2-circuit mating terminal, 5.08mm pitch. p/n 1912401
J6	Motor Drive	Phoenix Contact 5-circuit pluggable terminal block p/n 1924334	Phoenix Contact 5-circuit mating terminal, 5.08mm pitch. p/n 1912430
J7	Feedback	High density D-sub 15-position, female	High density D-sub 15-position, male
J9	I/O Signals	Phoenix Contact 3-circuit pluggable terminal blockp/n 1985205	N/A
J4	Remote Amplifier	2 by 8 un-shrouded header, male, .1" spacing	Samtec socket, 16-position, .1′′ spacing. p/n: ISDM-08
J12	Shunt	Phoenix Contact 2-circuit pluggable terminal block, p/n 1924305	Phoenix Contact 2-circuit mating terminal, 5.08mm pitch. p/n 1912401
J3	RS232	DB9, female	DB9, male
J2	SPI	2 by 4 unshrouded header, male, .1" spacing	Samtec socket, 8-position, .1′′ spacing. p/n: ISDM-05
J14	Serial	3-pin 2mm single row header	Samtec SQT-103-01-L-S-RA

6.10 Absolute Maximum Ratings

HV voltage range:	0V to +60V
+5V voltage range:	−0.3V to +5.5V

6.11 Environmental and Electrical Ratings

Storage temperature:	−40 to +125 degrees C (−40° F to +257°F)
Operating temperature:	0 to +70 degrees C (32° F to +158°F)
HV power requirement:	+12V to + 56V operating range
Motor amplifier continuous current limit*:	5.0 A
Motor amplifier peak current limit:	10.0 A
Digital input voltage range:	0V to 5V, TTL thresholds
Serial communications:	RS232

** Current rating at 25 C ambient and with 110 CFM air flow on board. Significantly higher currents are possible with additional heat sinking. Contact your PMD representative for details.*

6.12 DK78113 On-Board Amplifier Quick Reference

To use the DK78113 circuitry safely and to view results in physical units such as amps and volts via Pro-Motion various limits and conversion constants must be specified.

These parameters are automatically provided when using Pro-Motion's Axis Wizard setup sequence, however they are listed below for reference.

Parameter	Value & Units	Comments
Motor Output Mode	PWM High/Low	Sets the motor output mode to PWM High/Low for operation with the on-board amplifier. For remote amplifier control operation PWM High/Low or sign/ magnitude PWM may be selected.
PWM Switching Frequency	20 kHz	This setting is motor-specific. Larger motors (some NEMA 23 and most NEMA 34) should be set for 20 kHz. Smaller motors may use 40, 80, or 120 kHz to maximize current control accuracy and minimize heat generation.
PWM Dead Time	540 nsec	To ensure correct operation of the DK78113's on-board switching amplifier this parameter must be set to this value.
PWM Refresh Time	2,000 nSec	To ensure sufficient time to recharge the DK78113's on-board amplifier high side switches this parameter must be set to this value.
PWM Refresh Period	8 cycles	To ensure sufficient time to recharge the DK78113's on-board amplifier high side switches this parameter must be set to this value.
PWM Signal Sense	Active High	To ensure correct operation of the DK78113's on-board amplifier all PWM outputs must be set to active high.
Minimum Current Read Time	2,000 nSec	To ensure sufficient minimum current read time with the DK78113's on-board amplifier when driving BLDC motors this parameter must be set to this value.
Leg Current Conversion	.733 mA/count	This value specifies the conversion factor so that the leg current can be traced and displayed correctly in amps.

Parameter	Value & Units	Comments
Foldback Continuous Current Limit	5.0 A	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value or lower.
Foldback Total Energy Limit	125 A^2sec	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value or lower.
Temperature Limit	75.0 C	To ensure safe operation of the DK78113's on-board amplifier this parameter must be set to this value.
Bus Current Supply Conversion	.505 mA/count	This value specifies the conversion factor so that the DC bus current supply can be traced and displayed in amps.
Bus Current Supply Limit	20.0 A	To ensure safe operation of the DK78113's on-board amplifier this parameter should be set to this value.
Bus Current Return Limit	20.0 A	To ensure safe operation of the DK78113's on-board amplifier this parameter should be set to this value.
Bus Voltage Conversion	1.424 mV/count	This value specifies the conversion factor so that the DC bus voltage can be traced and displayed in volts.
Undervoltage Limit	10.0V	To ensure correct operation of the DK78113's on-board amplifier this parameter should be set to this value or higher.
Overvoltage Limit	60.0V	To ensure correct operation of the DK78113's on-board amplifier this parameter should be set to this value or lower.

7. Software Development

In This Chapter

- Overview of C-Motion
- Getting Started With C-Motion Magellan
- PC User Code Development
- Embedded User Code Development with MC7x112 ICs

This chapter provides a general introduction to creating software for PMD controller products. The focus will be on C-Motion, PMD's C-language libraries. PMD also provides .NET language support which enables software to be written in C# or VisualBasic.

For a complete description of PMD's software language support, tools, and design examples refer to the *C-Motion Engine Development Tools Manual*.

> MC7x112 ICs may be controlled by software running on a host microcontroller, or control parameters may be stored in the IC's NVRAM and torque values commanded through analog or digital hardware interfaces. The latter does not require a host command in which case you may skip this chapter.

7.1 Overview of C-Motion

The C-language libraries used to develop user application code for PMD products is called C-Motion. C-Motion is a set of callable C-language routines that provide many features including:

- Axis virtualization
- Ability to communicate to multiple PMD motion ICs, boards, or modules
- Ability to communicate via RS232, RS485, CAN, Ethernet, SPI (Serial Peripheral Interface), or 8/16 bit parallel bus
- Provided as source code, allowing easy compilation & porting onto various run-time environments including PCs, user-designed boards, or C-Motion Engines

Broadly speaking there are two different ways user application code written in C-Motion can be used to control a PMD controller; the user application program can run on a host separate from the PMD controller, or the user application program can run directly on the PMD controller in a code execution module called the C-Motion Engine. All PMD products can support user application code running on a separate host, while only PMD products with a "/CME" designation in their product name contain a C-Motion Engine.

Here is more information on these two different ways of executing the user application code.

7.1.1 Host-based Execution of User Code

When located on a host controller the user code communicates via one of the available host interfaces to the PMD controller. Depending on the PMD product being used this may be point-to-point serial, multi-drop serial, CAN, Ethernet, or SPI (Serial Peripheral Interface). The format of these communications is one of two packet based protocols depending on the PMD product being used; PRP protocol, which is short for PMD Resource Access Protocol, or Magellan protocol. The user need not be concerned with the packet format however because these details are handled automatically when code is written using C-Motion libraries.

For a list of PMD products and the packet protocol used with each see Section 7.1.4, "C-Motion SDKs."

For more information on the Magellan protocol refer to the *C-Motion Magellan Programming Reference*. For more information on the PRP protocol refer to the *C-Motion PRP Programming Reference*.

7.1.2 C-Motion Engine-based Execution of User Code

When executed on the PMD controller's C-Motion Engine the user code communicates internally to the resources available on the controller such as the Magellan Motion Control IC. This has speed advantages both in communicating with those resources and in real time code execution predictability. The software tools used to compile and debug C-Motion code when run on the C-Motion Engine are contained in the SDK (Software Development Kit) provided by PMD.

Executing the code directly on the C-Motion Engine allows the controller to function as a fully standalone controller. In this mode a host controller communication link is not needed, and one or more of the PMD controller's communication ports or digital I/O ports may be used to interface to user-operated buttons or a touch screen.

Alternatively, code can be executed on the PMD controller's C-Motion Engine that processes commands which are received from a host network, thus forming an application-specific local controller within a larger system. For example for a device with a C-Motion Engine controlling a three-axis gantry a host may send high level commands which are interpreted to mean "move the gantry to location X, Y, Z". The user code executing on the C-Motion Engine parses these incoming commands and generates axis-specific motions for each of the three controlled motion axes to execute the high level host command.

7.1.3 PMD Products & User Code Execution Options

The following table shows specific types of PMD products and options for running the user application code:

PMD Product Type	Application Code Runs On	System Description
PMD Magellan or Juno Motion Control IC	Microcontroller	**User-designed board.** A microcontroller sends commands to a PMD motion control IC, both of which are located on a user-designed board. Allows standalone operation.
PMD Magellan or Juno Motion Control IC	PC, user-designed controller, or other controller	**Host-connected user-designed board.** A PC, user-designed controller, or other controller sends commands via a network connection to a PMD motion control IC which is located on a user-designed board.
PMD ION Drive or Prodigy Board	PC, user-designed controller, or other controller	**Host-connected ION drive or Prodigy board.** A PC, user-designed controller, or other controller sends commands via a network connection to a PMD ION drive or Prodigy board.
PMD ION/CME * Drive or Prodigy/CME Board	C-Motion Engine	**ION/CME drive or Prodigy/CME board.** User application code runs on the PMD controller's C-Motion Engine. Allows standalone operation.

** PMD products which support a C-Motion Engine have a "/CME" in their product name, for example ION/CME N-Series Drive.*

7.1.4 C-Motion SDKs

There are three different C-Motion SDKs; C-Motion Magellan, C-Motion PRP, and C-Motion PRP II. All of these SDKs are available from the PMD website at http://www.pmdcorp.com/resources/software. Here is more information on each:

- **C-Motion Magellan SDK** – an SDK for creating host-based user applications for PMD products that utilize a Magellan or Juno formatted protocol.
- **C-Motion PRP SDK** – an SDK for creating host-based and downloadable C-Motion Engine-based user code for systems utilizing either a PRP (PMD Resource Access Protocol) protocol device or a Magellan/Juno protocol device. The C-Motion PRP SDK is also used in motion applications that will use the .NET (C#, VB) programming languages.
- **C-Motion PRP II SDK** – This SDK is similar to C-Motion PRP but is used with ION/CME N-Series Digital Drives. Compared to standard C-Motion PRP, C-Motion PRP II supports additional features such as multi-tasking, mailboxes, mutexes, and enhanced event management.

For reference the following table shows the packet protocol and C-Motion SDKs that can be used with each PMD product family:

Product Family	Packet Protocol	C-Motion SDKs
Magellan MC5x113 Family ICs & DKs	Magellan	C-Motion Magellan, C-Motion PRP*
Magellan MC5x000 Family ICs & DKs	Magellan	C-Motion Magellan, C-Motion PRP*
Juno MC7x113 ICs & DKs	Magellan**	C-Motion Magellan, C-Motion PRP*
Juno MC7x112 ICs & DKs	Magellan**	C-Motion Magellan, C-Motion PRP
ION/CME N-Series	PRP	C-Motion PRP II
ION 500 Drive (except Ethernet)	Magellan	C-Motion Magellan, C-Motion PRP*
ION 500 Drive with Ethernet interface	PRP	C-Motion PRP
ION/CME 500 Drive	PRP	C-Motion PRP
ION 3000 Drive	Magellan	C-Motion Magellan, C-Motion PRP*
Prodigy/CME Machine-Controller	PRP	C-Motion PRP

** With this product C-Motion PRP typically only used for .NET support, or if a mix of Magellan protocol and PRP protocol devices are attached.*
*** Although the Juno IC command set is somewhat different than the Magellan IC command set, the overall packet format is the same and therefore the Juno packet command set also referred to as a Magellan packet protocol.*

From C-Motion Magellan to C-Motion PRP to C-Motion PRP II, each 'higher' level is a functional superset of the previous, and so when using multiple PMD products the highest required SDK should be used. For example for a PC-based application that communicates both with ION/CME N-Series drives (which by themselves require the C-Motion PRP II SDK) and with Prodigy/CME Machine-Controllers (which by themselves require the C-Motion PRP SDK) the C-Motion PRP II SDK should be used.

Because higher SDKs are a superset of the lower level SDKs there may be situations where more than one SDK could be chosen and successfully used in the application. In subsequent sections, when this is the case, we will provide guidance to help with this choice.

7.2 Getting Started With C-Motion Magellan

The C-Motion Magellan SDK is used to develop user application code for PMD controllers that use a Magellan/Juno packet protocol, including MC7x112 Torque Control ICs. While the details of how Magellan packets are formatted is not important for most C-Motion library users, a general understanding of the underlying architecture is helpful.

Magellan packets may be transmitted via serial, CAN, SPI (Serial Peripheral Interface) and for some products parallel-words. A basic communication to a Magellan packet format device consists of a command word and for some communications additional data words. For example for the command **SetPosition** the data words contain the 32-bit desired position value. Each packet sent to the PMD controller contains a checksum field.

After a Magellan format communication is sent a return communication is sent by the controller which consists of any requested data as well as a checksum field. Commands sent to the controller that do not request data return just a checksum word.

The Magellan packet protocol is a master/slave system. The host functions as the master and initiates communication sequences which the addressed PMD controller responds to. There is one exception to this however which can occur with CAN using an 'Event' Node ID that must be different than the main communication channel Node ID.

In the Magellan/Juno architecture access to all available functions occurs via communication channels connected to the motion IC itself. This includes access to internal functions of the IC such as quadrature position tracking, trajectory generation, PWM generation, etc.…

Magellan MC5x113 and MC5x000 ICs support at least two different communication interfaces, and some support as many as three. Command packet processing is simultaneously supported on each communication interface. For example a CAN communication channel can be used to command motion profiles while a serial connection can be used to simultaneously retrieve status information during the move.

MC7x113 Juno velocity control ICs support three different communication interfaces (serial, SPI, and CAN) while MC7x112 Juno torque control ICs support only one (Serial). Therefore comments in subsequent sections regarding the use of multiple interfaces only apply to MC7x113 ICs.

7.2.1 Accessing Magellan/Juno Functions with C-Motion

C-language handles are used to reference each axis supported by a particular Magellan or Juno motion control IC. This handle type is referred to as an Axis handle. For multi-axis products such as MC5x000 ICs an Axis handle will be created for each active axis, up to four. For single-axis Magellan products such as MC58113 ICs two Axis handles are created, one for the primary axis and one for the auxiliary axis which provides just encoder input. For Juno ICs, a single axis handle is created.

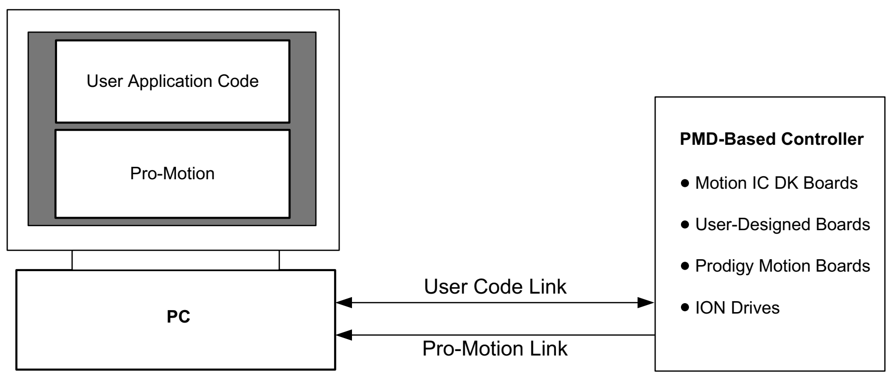
Once a C-language Axis handle is obtained, that handle is used as an argument in subsequent C-Motion library calls, thereby identifying which specific axis is being commanded. This is true even if the motion system being controlled has multiple communication channels in use. Subsequent commands using that Axis handle will be automatically routed to the correct motion control IC via the correct communication channel.

7.3 PC User Code Development

PC user code means user code that runs on the PC. This is a popular approach for controlling off-the-shelf PMD board, module, and IC developer kits products, and also for controlling user-designed boards that contain PMD motion ICs or PCB-mountable drive modules.

In addition to running user code written with C-Motion libraries, running on the PC is the most popular approach when coding in C# and other .NET languages.

Figure 7-1: Typical Connection Scheme for PC-Based User Code Development



If available, two separate communication links can be used with PC-based code development as shown in Figure 7-1:

User Code Link

When the PC runs the user application code it uses a communication link to send commands to the PMD controller. This communication channel is called the User Code link. For PRP devices, the User Code link will carry PRP-formatted packets so that the PRP device can correctly interpret them and respond accordingly. For Magellan protocol devices the User Code link will carry Magellan-formatted packets. See Section 7.1.4, “C-Motion SDKs” for a list of packet protocol types for various PMD products.

Pro-Motion Link (optional)

The Pro-Motion link uses a separate connection to allow Pro-Motion to communicate with the PMD controller. Although optional, running Pro-Motion and utilizing a Pro-Motion link as part of the code development setup has significant benefits because Pro-Motion can be used to investigate the status of the PMD controller before, during, and after execution of user application code sequences. For more information on Pro-Motion refer to Chapter 3, *Going Further with Pro-Motion*.

Note that for IC products such as MC7x112 ICs which have only one communication link a simultaneous Pro-Motion link cannot be used.

> If both the user code link and Pro-Motion link are operating care should be taken to avoid using Pro-Motion to command motions or make changes to the PMD controller that could conflict with commands being sent by the user application code. In addition, status polling in Pro-Motion should be disabled. This can be done via the View menu.

7.3.1 Choosing the SDK For PC-Based User Code Development

In most setups where the host code will run on the PC in the production application there won't be a choice of C-Motion library SDK. For example if one or more of the devices being commanded are N-Series IONs, the SDK used for the PC-based user code must be C-Motion PRP II.

The scenario where a choice may exist is if the setup consists only of a Magellan packet format device, for example a user-designed board with MC7x112 ICs on it. In this case both C-Motion Magellan SDK and C-Motion PRP SDK are viable choices.

In this situation some developers may opt for the C-Motion PRP SDK. The reason is that if in the future a PRP device is added to the controller setup there will be no need to switch SDK types. In other words since C-Motion PRP provides a superset function of C-Motion Magellan it is more 'future proof'.

Alternatively, a good reason for choosing C-Motion Magellan is that the size of the C-Motion source code libraries is significantly smaller. While not a major consideration if the user code runs on the PC in the production application, smaller and simpler source code libraries may be important if the user application code will eventually be ported to run on a microcontroller on a user-designed board.

7.3.2 PC-Based Code Development Process

Figure 7-2 shows a flowchart of the recommended process for developing PC-based user application code with PMD products.

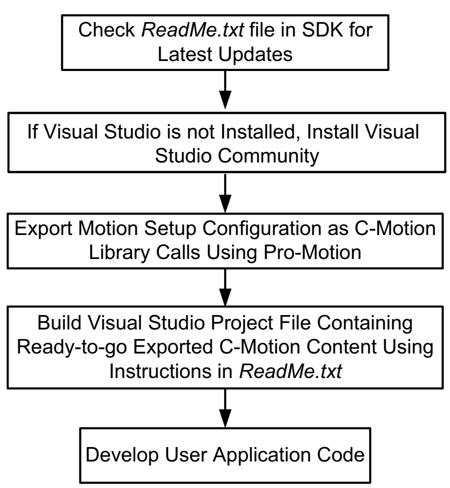


Figure 7-2: Recommended Sequence for PC Code Development

A first step is to review the *ReadMe.txt* file that is included at the top level directory of the SDK you are using. This file will provide up-to-date information on example Visual Studio projects, source code examples, and other resources provided by the SDK.

The next step is to install Visual Studio Community, if you haven't installed Visual Studio already. Visual Studio is a software development environment made by Microsoft Corporation.

Next, with Pro-Motion connected to the PMD controller(s) in the motion setup and with all other configuration settings such as gain settings and safety settings in place for your motion setup, execute a configuration export to C-Motion as detailed in Section 3.10, "Configuration Export to C-Motion." The output of this operation will be one or more C-language source files that will be incorporated into your initial user application code Visual Studio project.

The instructions for the final step to building your initial user code development Visual Studio project can be found in the *ReadMe.txt* file for the SDK you are using. This step combines the C-Motion source code library files with your exported configuration source code file(s) into a ready-to-compile and run Visual Studio project.

As you proceed with development of your application code the following manuals will be useful. For detailed information on C-Motion PRP refer to *C-Motion PRP Programming Reference*. For detailed information on Magellan Motion Control IC commands refer to the *C-Motion Magellan Programming Reference*. For detailed information on Juno Velocity & Torque Control IC commands refer to the *Juno Velocity & Torque Control IC Programming Reference*. For more detailed information on all aspects of developing software for PMD products refer to the *C-Motion Engine Development Tools Manual.*

7.4 Embedded User Code Development with MC7x112 ICs

Embedded user code means user code that runs on a microcontroller on a user-designed board that contains PMD motion control ICs such as Magellan or Juno ICs. This section provides information on developing embedded user code with MC7x112 ICs.

Figure 7-3: Typical Connection Scheme for Embedded User Code Development

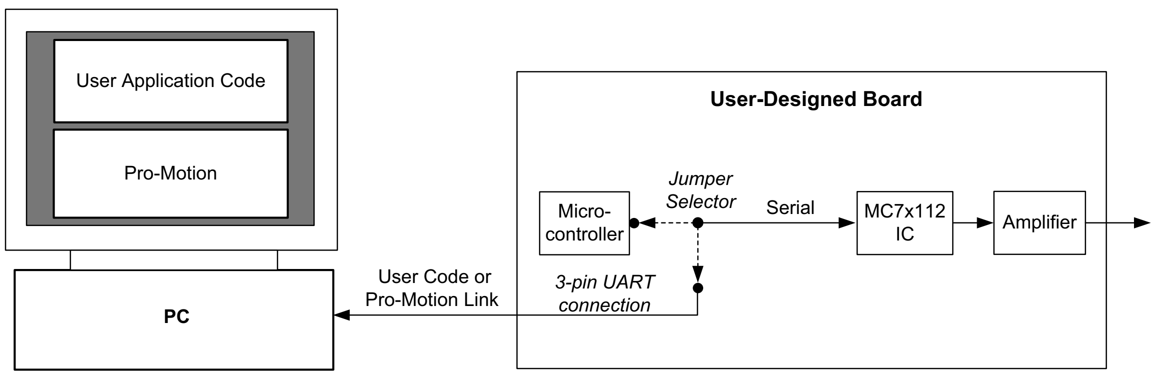


Figure 7-3 shows a typical development setup for embedded code development with a user-designed board using MC7x112 ICs.

When a microcontroller sends commands to the MC7x112 it uses the MC7x112's serial communication channel. For MC7x112 ICs there is one available link, serial point-to-point, and for MC7x113 ICs there are three available links; SPI (Serial Peripheral Interface), point-to-point and multi-drop serial, and CAN.

When more than one link is available it is frequently useful to utilize simultaneous User Code and Pro-Motion links. The Pro-Motion link can be used for board check-out and exercising before the user application code is completed, and it can also be used while the application code is being developed to help confirm user code commands are being received and having the desired motion results.

Although MC7x112 ICs only support a single point-to-point serial link a PC link can still be utilized using jumpers designed into the board which select between a serial connection to the on-board microcontroller and a serial connection to the 3-pin UART serial cable (PMD P/N Cable-USB-3P). Although this approach doesn't allow simultaneous Pro-Motion monitoring of the user code function it still supports initial board check-out and exercising the motion sub-system via Pro-Motion.

7.4.1 Choosing the SDK for Embedded User Code Development

The recommended C-Motion SDK when controlling MC7x112 or MC7x113 ICs and when the target for code execution is a microcontroller running on a user-designed board is C-Motion Magellan.

Although C-Motion PRP SDK is also a viable choice since it supports both PRP and Magellan packet communications, C-Motion Magellan is recommended because the Magellan source code libraries are smaller for the Magellan libraries than for the PRP libraries, and because the source code libraries are simpler, smaller, and only use "C". This will make porting into the microcontroller environment easier, particularly if some C-Motion code sequences are executed from ISRs (Interrupt Service Routines).

For more information on C-Motion SDKs see Section 7.1.4, "C-Motion SDKs."

7.4.2 Typical Embedded Code Development Session

Figure 7-4 shows a popular development sequence for building user code that will run on the microcontroller of a user-designed board.

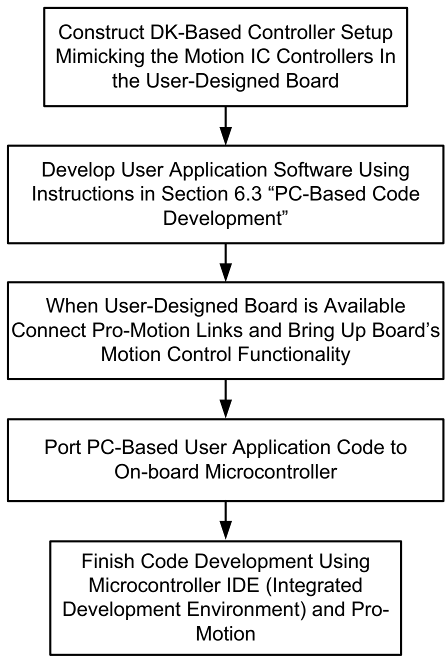


Figure 7-4: Recommended Sequence for Embedded Code Development

The recommended process for developing embedded user application code starts with code being developed on the PC and then later moves code execution to the microcontroller. Starting application development on the PC provides access to resources such as Visual Studio which should make initial code development easier. It also, at least initially, avoids complications that may occur from porting the C-Motion application code to run on the microcontroller, a process that will be discussed further in Section 7.4.4, "Porting PC-Based User Code to the Microcontroller."

While initially developing code on the PC has advantages, it should be pointed out that beginning code development directly on the embedded board target may also be a good alternative approach, especially if the user has familiarity with the microcontroller IDE (Integrated Development Environment) and experience with C-Motion libraries. Readers who will use this approach can skip to Section 7.4.3, "User-Designed Board Bring-Up."

Developing and running user application code on the PC usually means that you will use developer kit boards to mimic the architecture of the user-designed board. In the case of an MC7x112-based board this means either a DK7x112 or DK7x112N developer kit.

For additional information on developing used code on the PC refer to Section 7.3, "PC User Code Development."

7.4.3 User-Designed Board Bring-Up

If, as shown in Figure 7-3, a PC connection to the MC7x112 IC was designed into the board, the motion-specific portions of the board can be exercised even before the microcontroller user code is up and running. Using Pro-Motion you can confirm communication with the motion IC and then check out proper functioning of each portion of motion control circuitry including the amplifier, encoder feedback, and motion peripheral signals such as home and limit switch inputs.

Once the motion control circuitry is working Pro-Motion can be used to exercise the motors driven by the user-designed board and if desired collect trace data to generate optimized motion IC control parameter settings for the user-designed board.

7.4.4 Porting PC-Based User Code to the Microcontroller

At some point after bring-up of the user-designed board the code running on the PC will be ready to port to the microcontroller. This is not an automatic process and you should check the readme.txt file included with the SDK you are using for the latest information and design examples to assist with this process.

The C-Motion SDK source code library content, as delivered, uses drivers that access the PC's COM ports. To port C-Motion to your microcontroller similar drivers for your microcontroller need to be present. PMD provides drivers for some microprocessors and Linux in the C-Motion SDK. Check the C-Motion SDK content to determine if this is the case for your micro.

If not, you will need to write these routines yourself. To get started on this begin with a review of the C-Motion source modules that provide library calls for the communication ports you plan to use. PMDW32ser.c provides drivers for serial functions on Windows-based PCs, PMDCAN.c and PMDIXXATCAN.c provide CAN functions, and PMDNIspi.c provides SPI functions.

PMDsys.h should also be looked at because it provides execution system specific information that you may need to change. Beyond communication functions, other C-Motion functions that may need to be rewritten include time functions such as the **PMDTaskWait(ms)** C-Motion call.

When ready, start with a software example provided by the microcontroller vendor for the serial/UART interface. If available, a development setup which includes connections to a microcontroller IDE (Integrated Development Environment) is recommended. The IDE can be used to debug executing code.

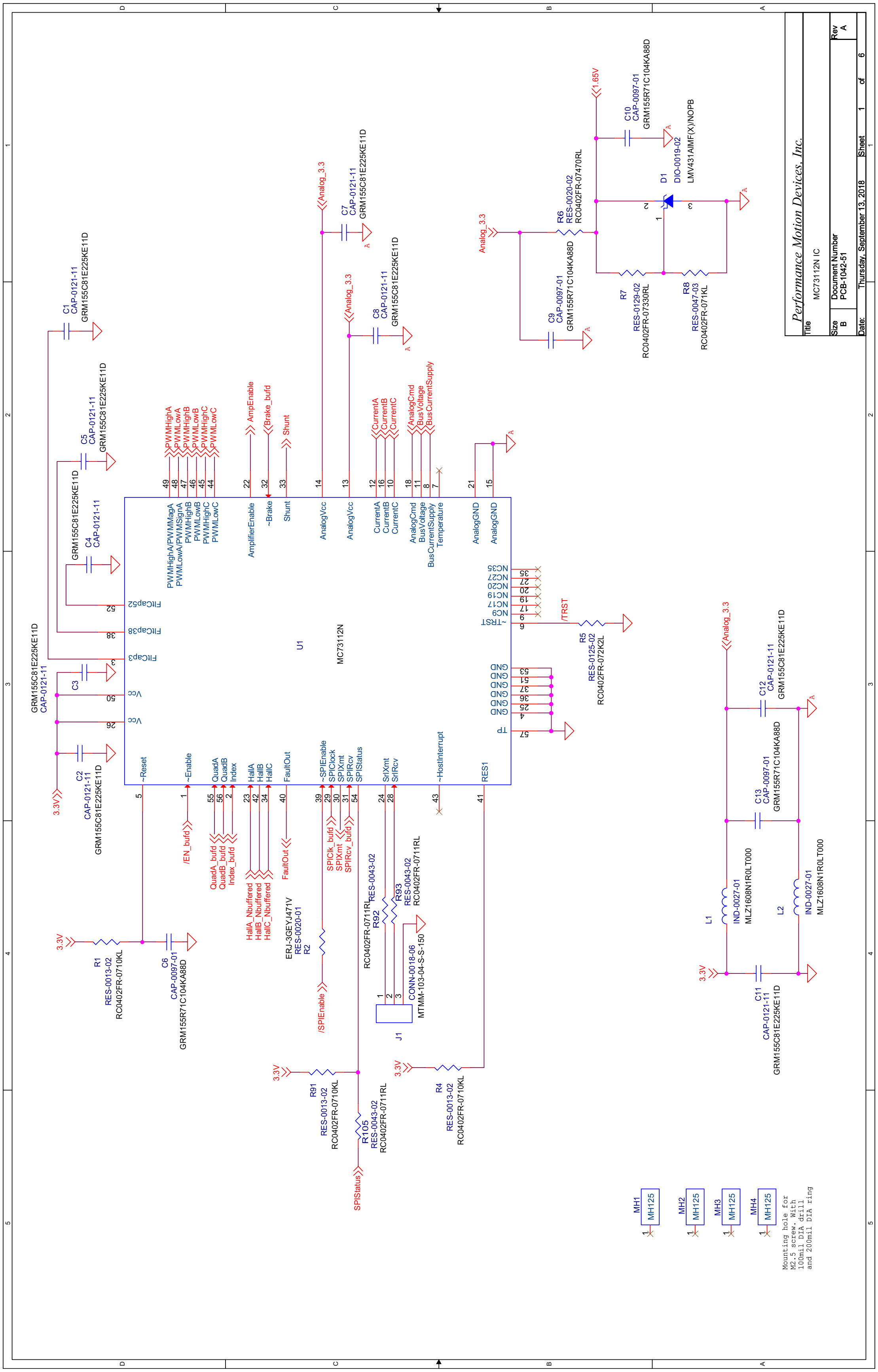
Since the motion control circuitry of the board has already been checked out the next steps in the porting process are to confirm that communications are functioning between the microcontroller and the on-board motion control ICs, and that the code changes to allow porting of the user code from the PC environment to the microcontroller environment are working correctly.

Once these elements have been confirmed development of the final user application code running on the user-designed board can be completed.

Appendix A. DK73112N Board Schematics

In This Appendix

Figure A-1: DK73112N Board Schematic, MC73112N IC



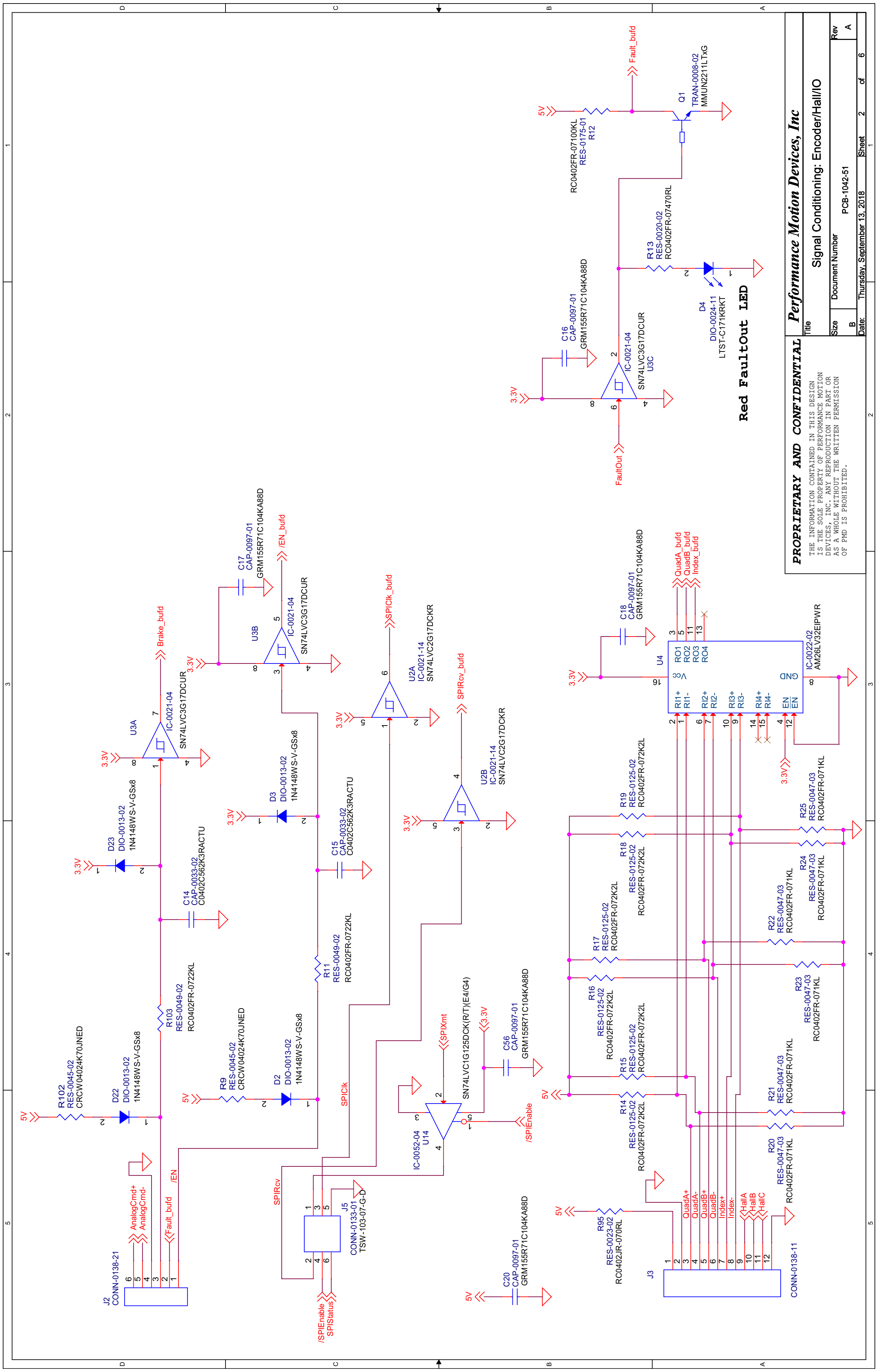
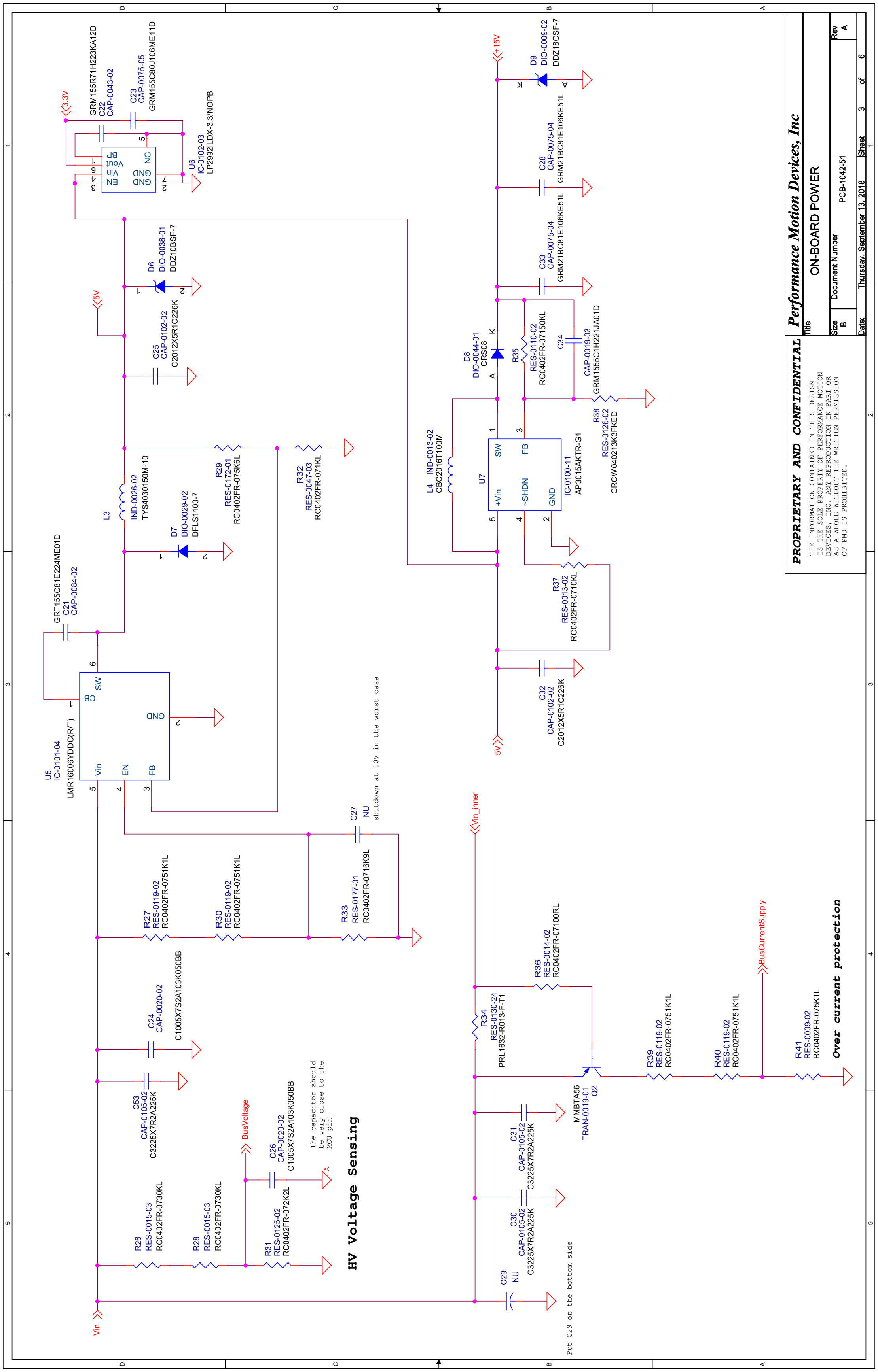


Figure A-2: DK73112N Board Schematic, Signal Conditioning: Encoder/Hall/IO

Figure A-3: DK73112N Board Schematic, On-Board Power



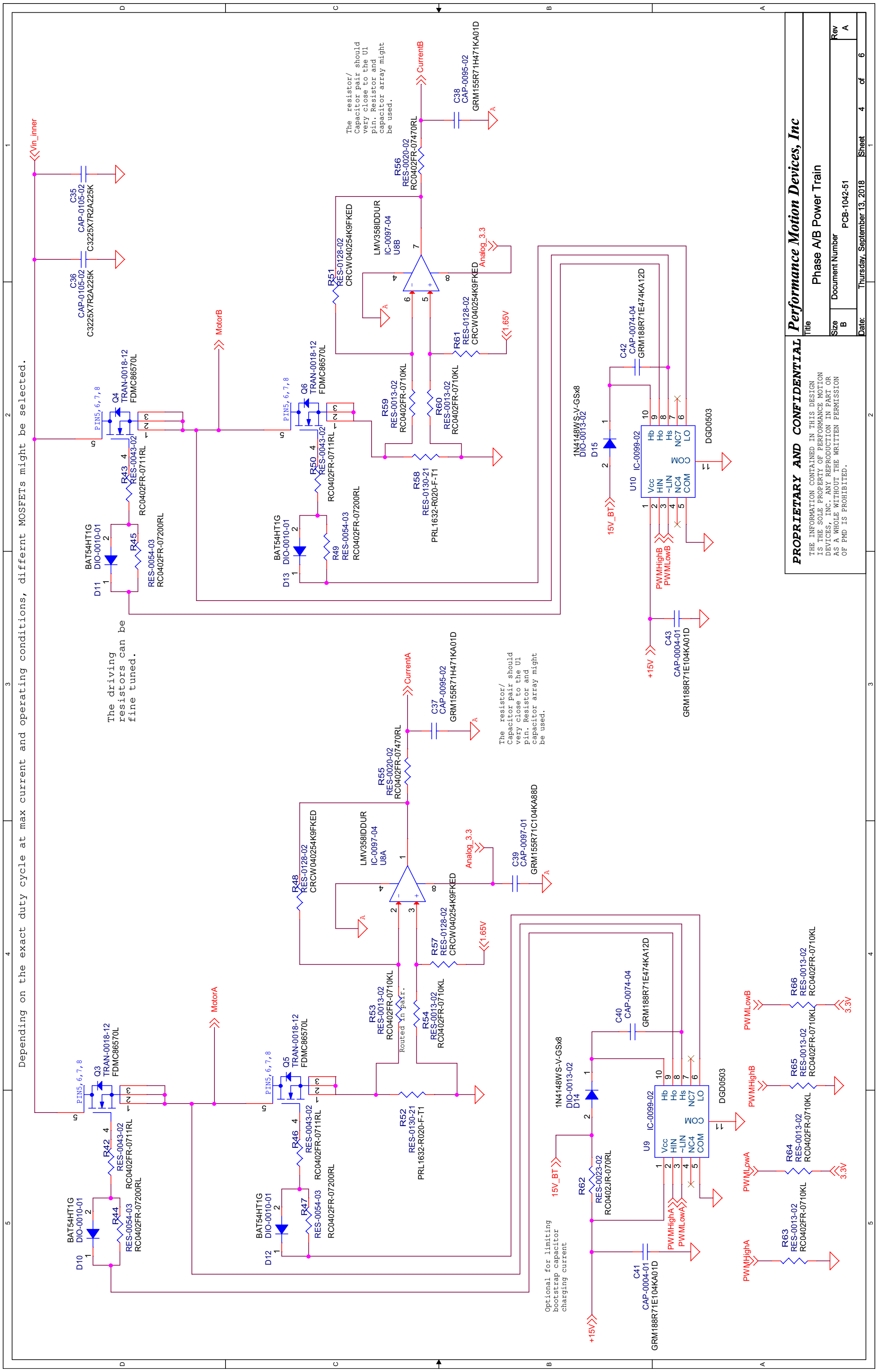
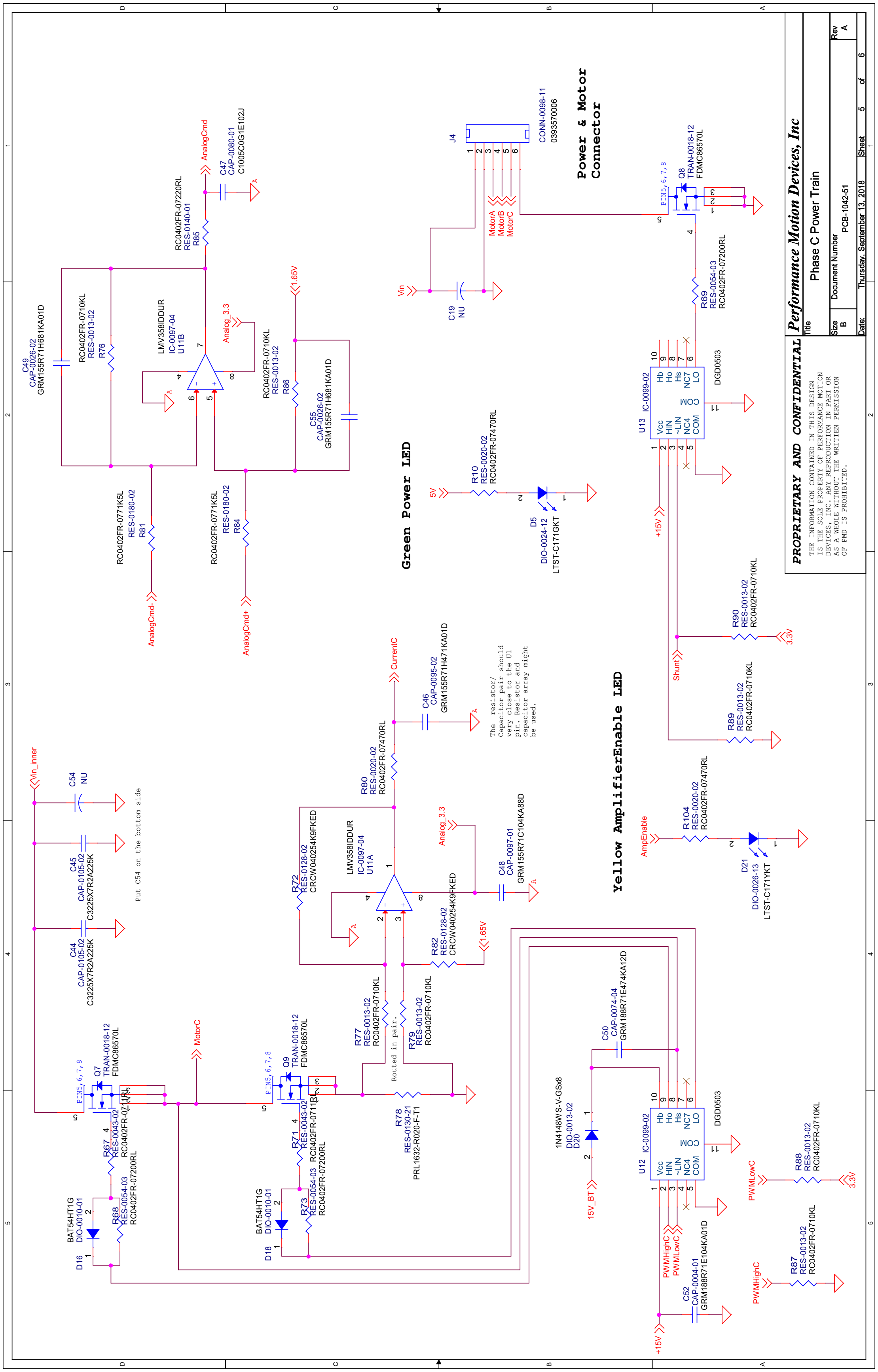


Figure A-4: DK73112N Board Schematic, Phase A/B Power Train

Figure A-5: DK73112N Board Schematic, Phase C Power Train



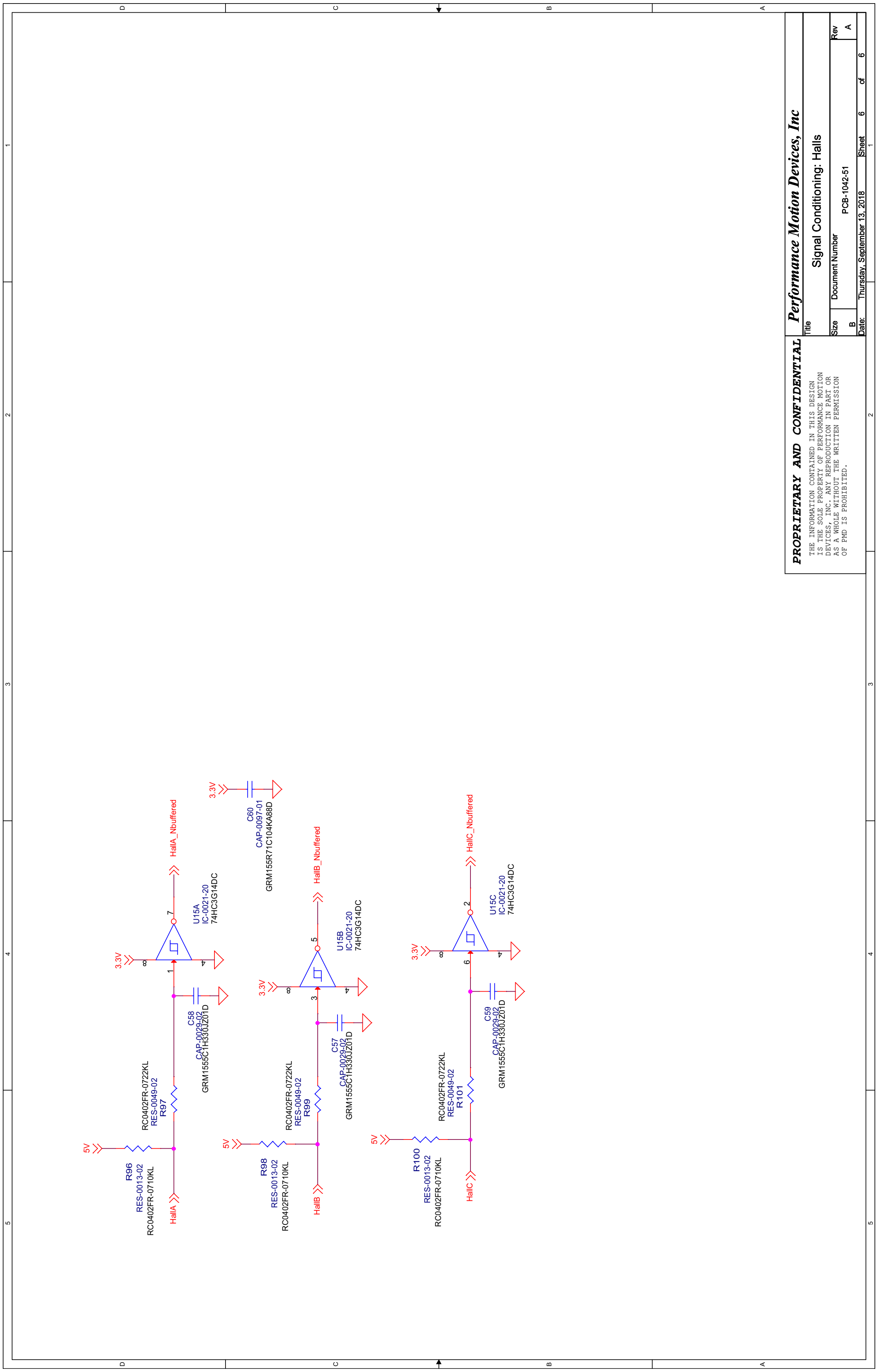


Figure A-6: DK73112N Board Schematic, Signal Conditioning: Halls

This page intentionally left blank.

Appendix B. DK78113 Board Schematics

In This Appendix

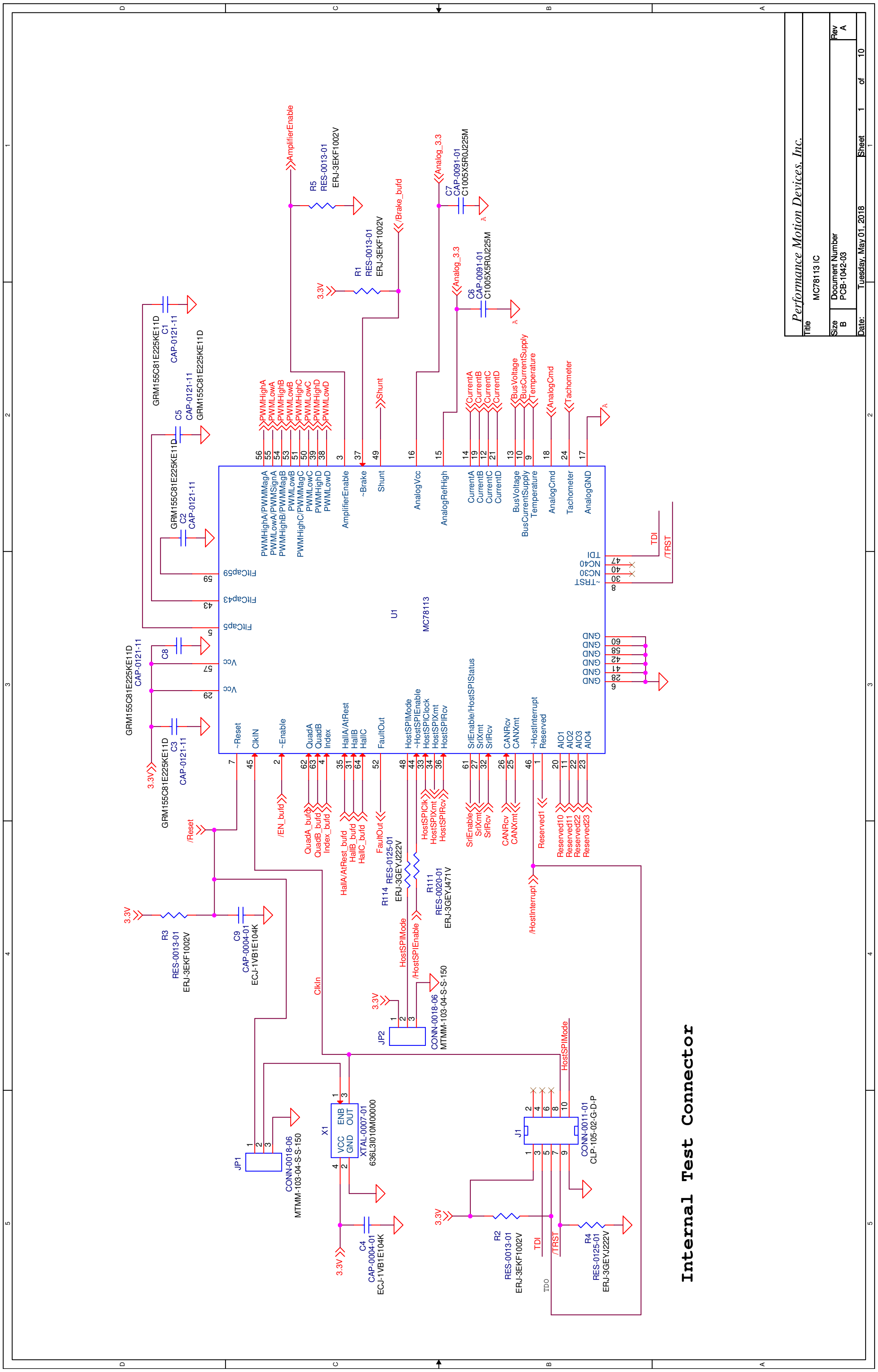
- Internal Test Connector
- SPI and Serial Ports
- Signal Conditioning: Encoder/Hall/IO
- Analog Signal Conditioning
- Phase A/B Power Train
- Phase C/D Power Train
- Signal Monitoring
- On-Board Power
- Connector & LED
- Jumpers & Connectors

The complete schematic for the DK78113 board is shown on the following pages.

Be aware that when MC7x112 ICs are installed in this board there are functions implemented on the board that are not supported by the MC7x112 ICs. In general these functions support features provided by the Juno MC7x113 Velocity Control ICs, but in some cases may support custom versions of the Juno ICs not accessible in the standard MC7x112 or MC7x113 ICs.

Therefore if using the DK78113 board schematic as a reference for a MC7x112-based user-designed board this unused circuitry should be removed.

Figure B-1: DK78113 Board Schematic, Internal Test Connector



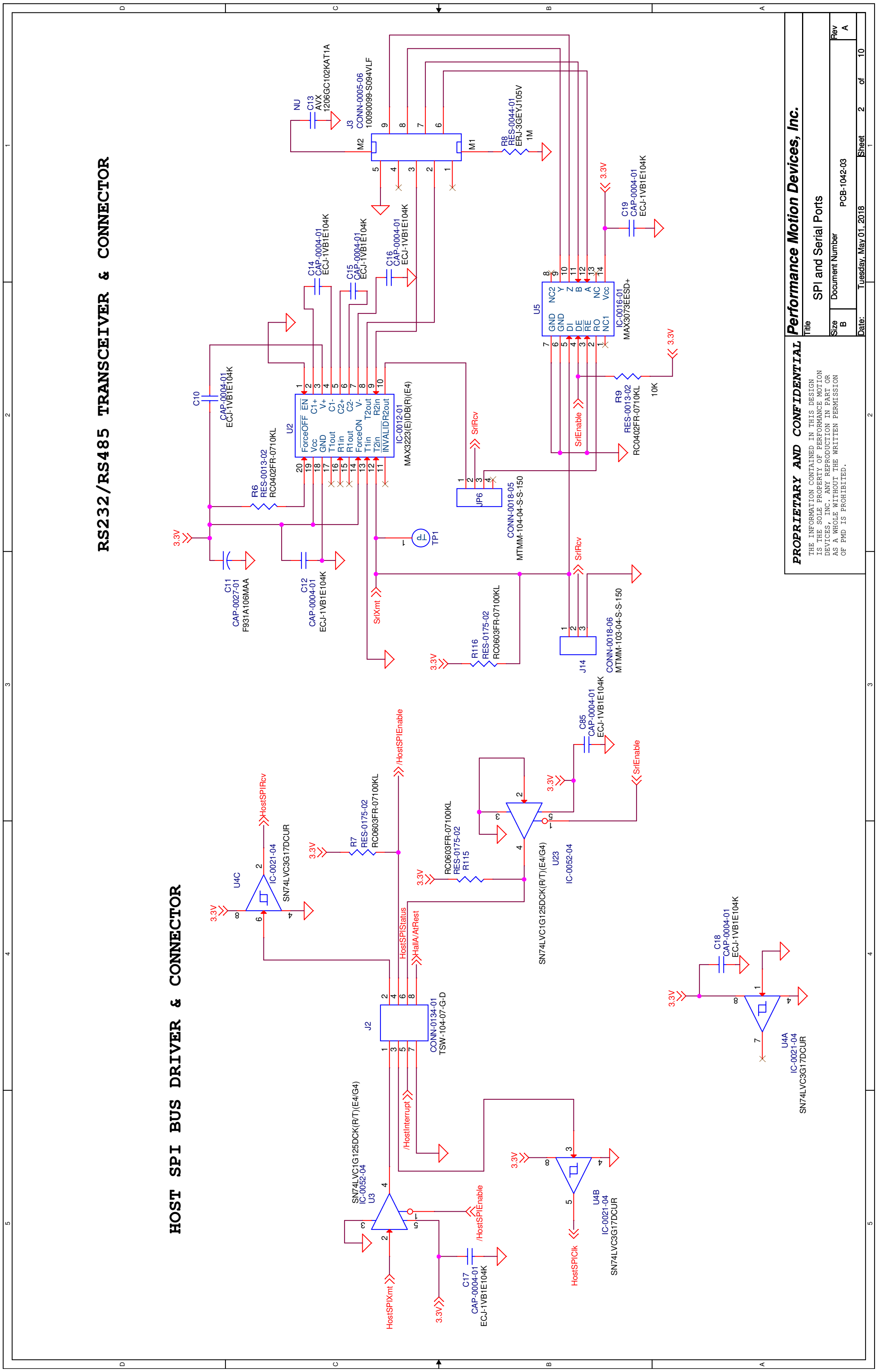
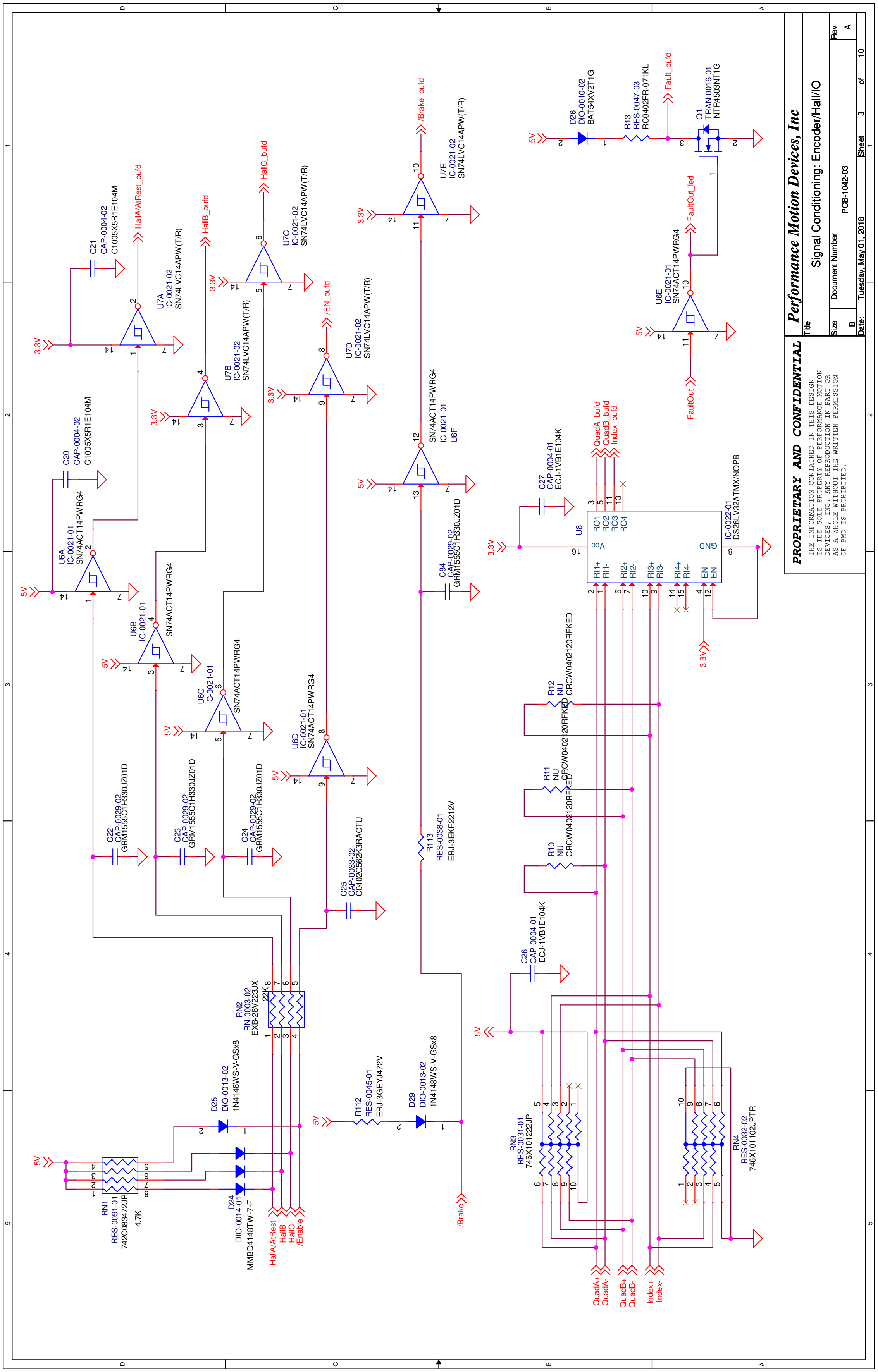


Figure B-2: DK78113 Board Schematic, SPI and Serial Ports

Figure B-3: DK78113 Board Schematic, Signal Conditioning: Encoder/Hall/IO



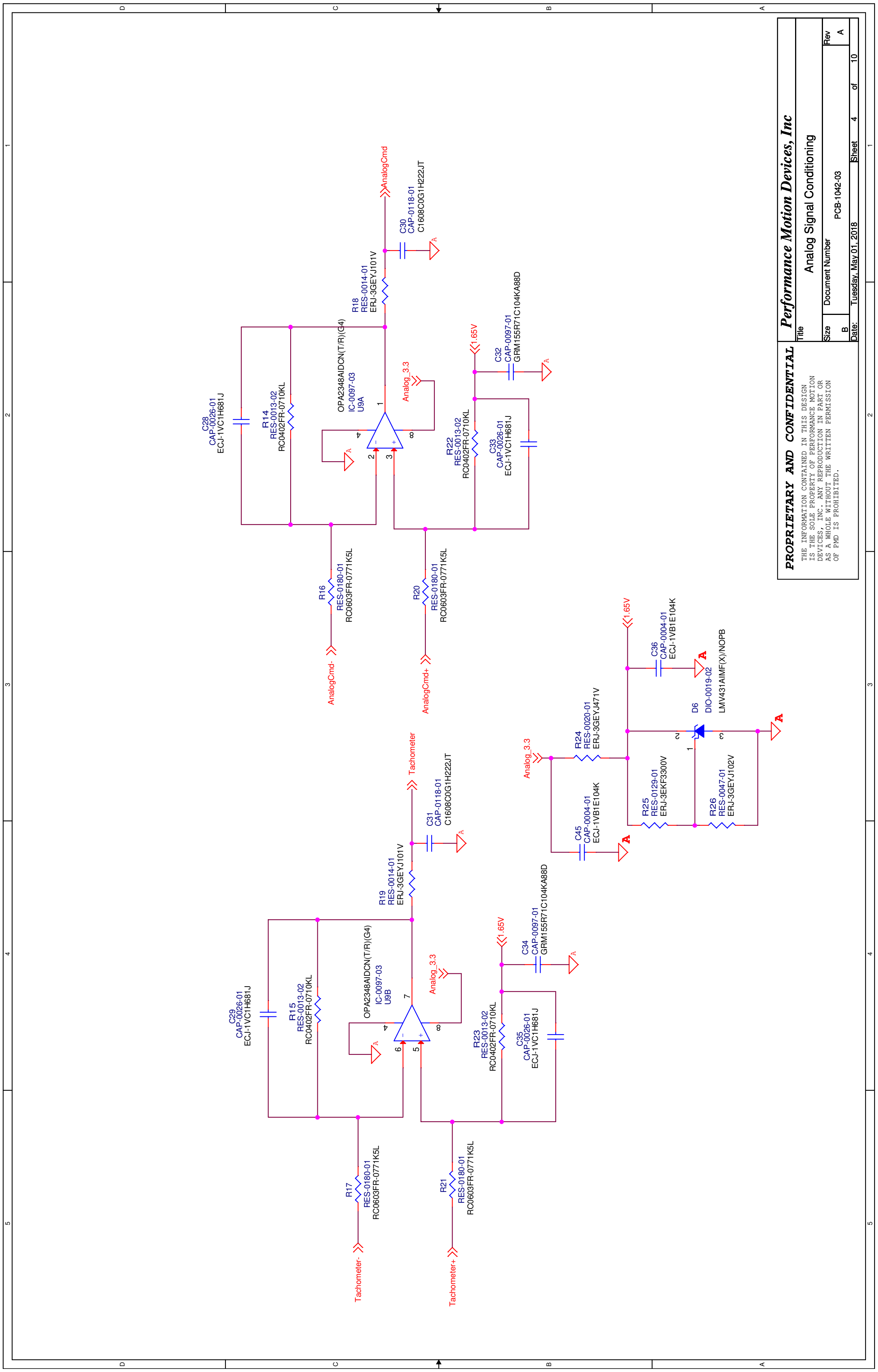
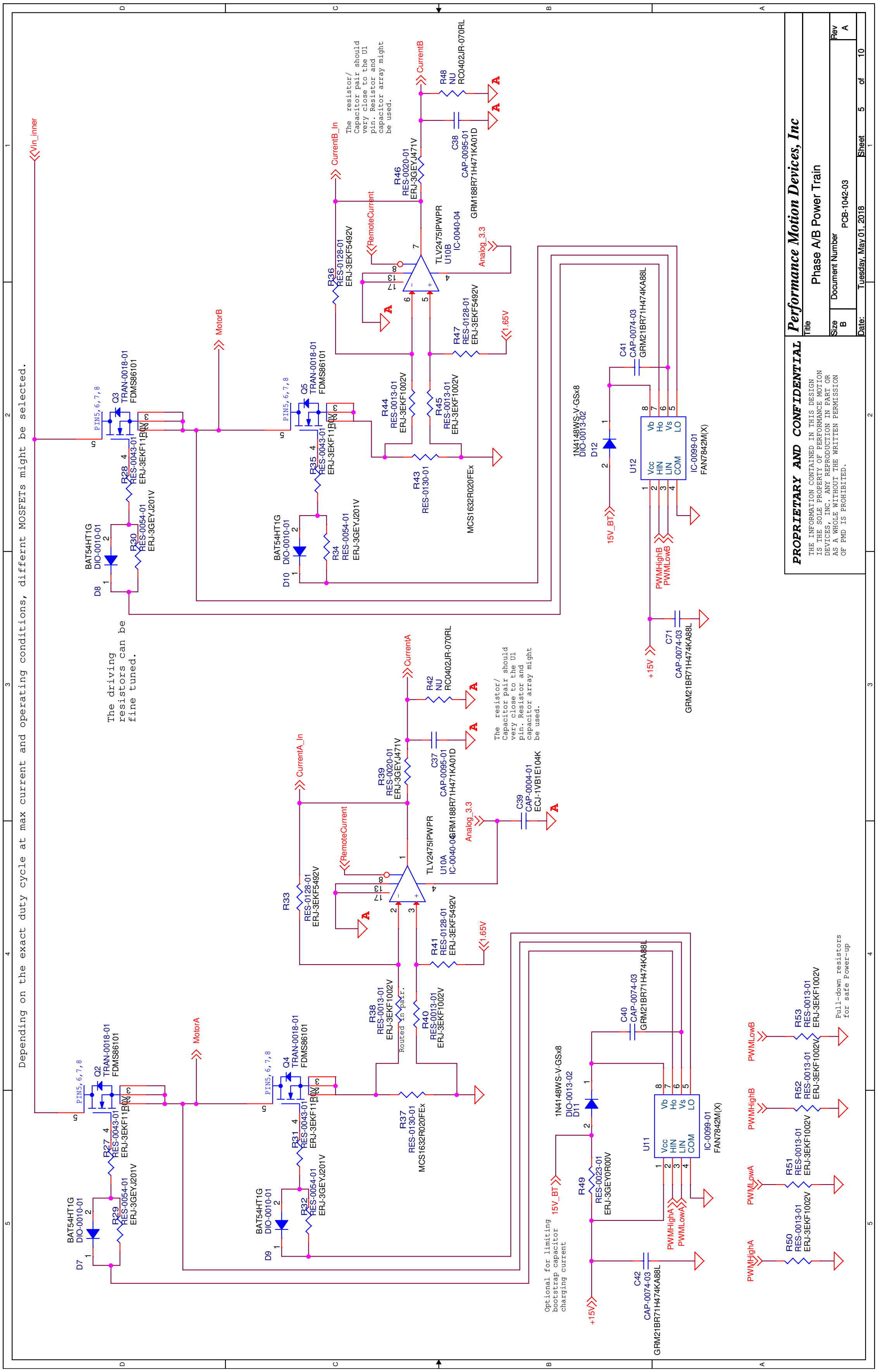


Figure B-4: DK78113 Board Schematic, Analog Signal Conditioning

Figure B-5: DK78113 Board Schematic, Phase A/B Power Train



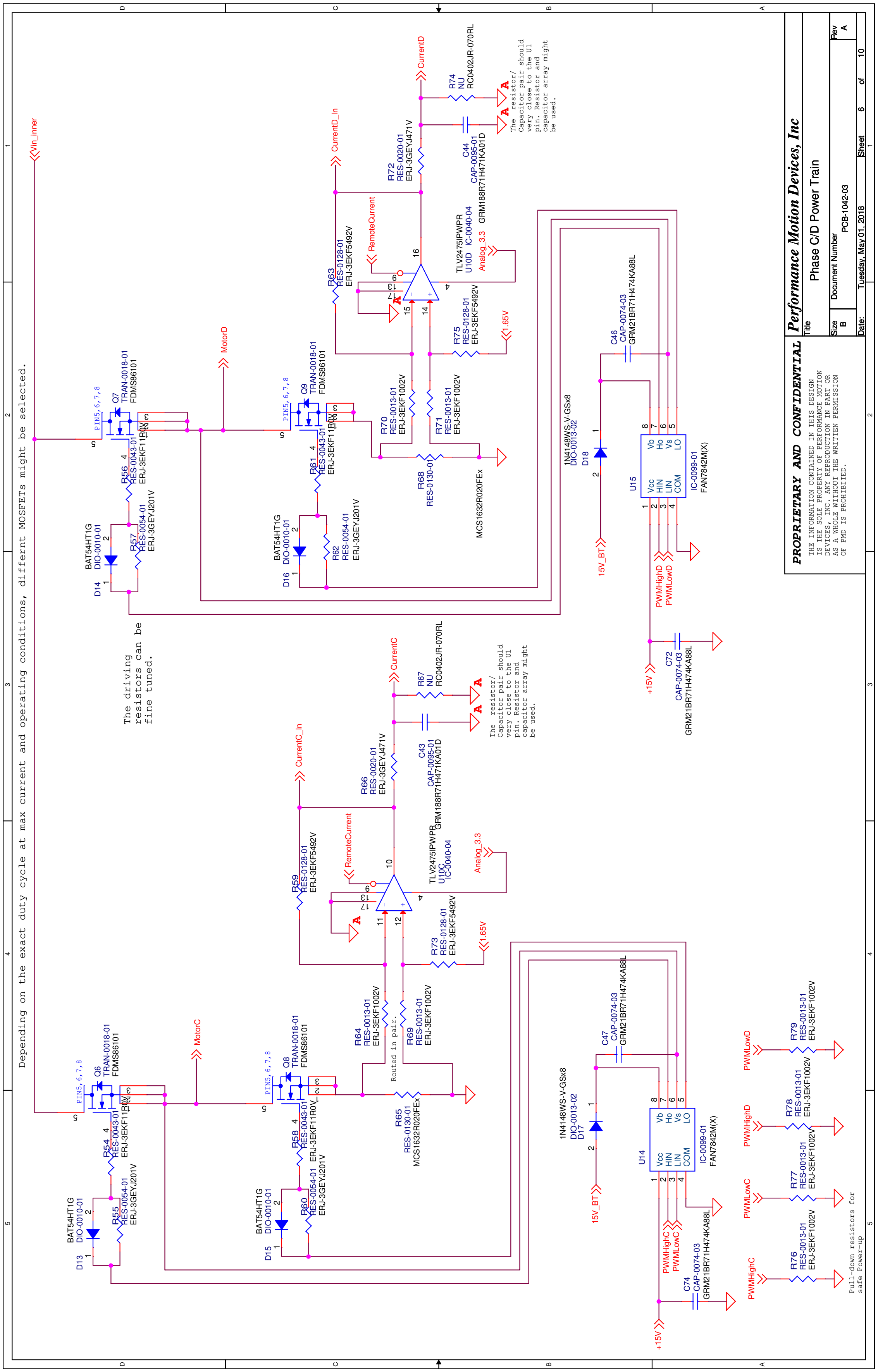
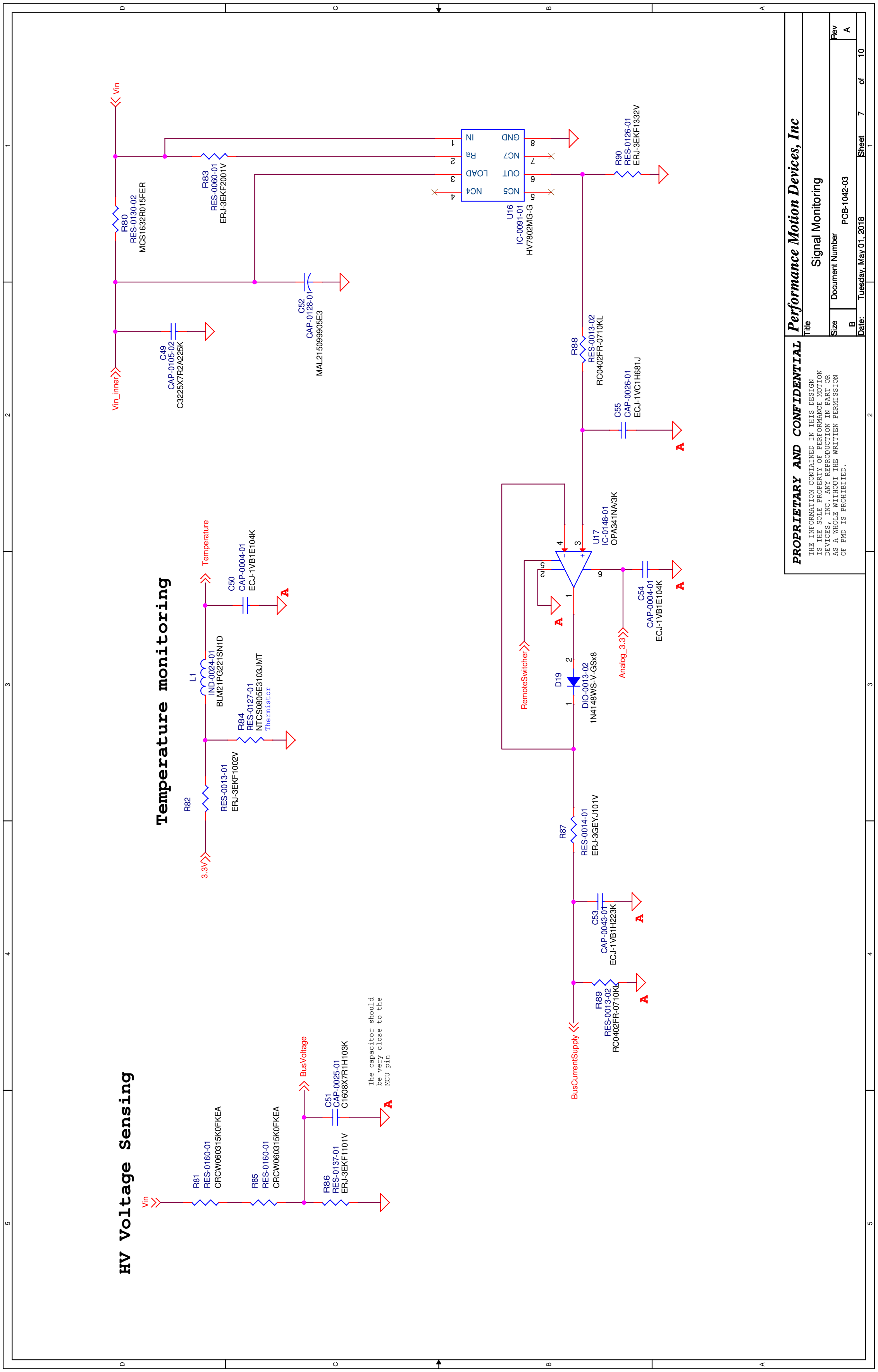


Figure B-6: DK78113 Board Schematic, Phase C/D Power Train

Figure B-7: DK78113 Board Schematic, Signal Monitoring



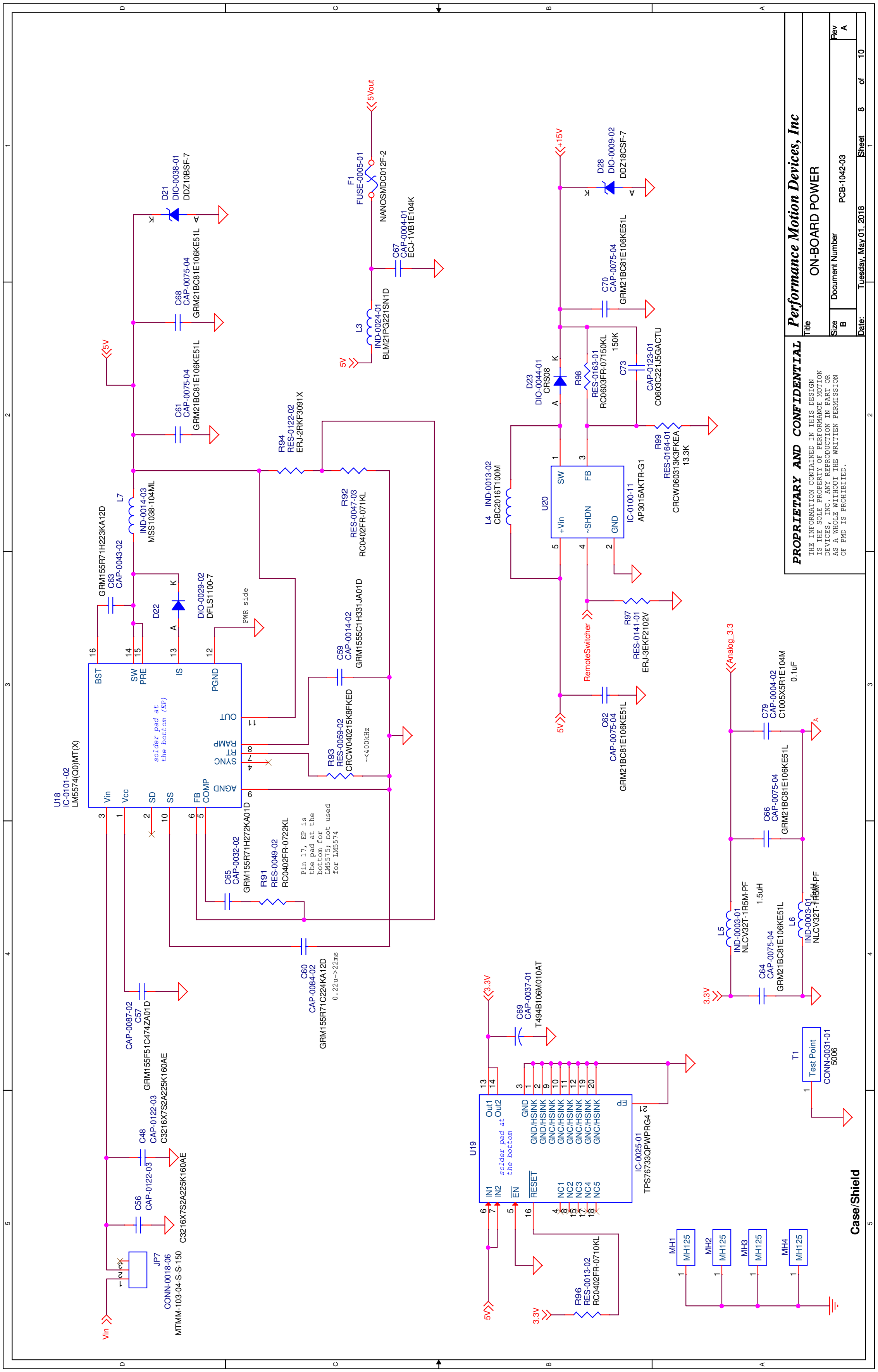
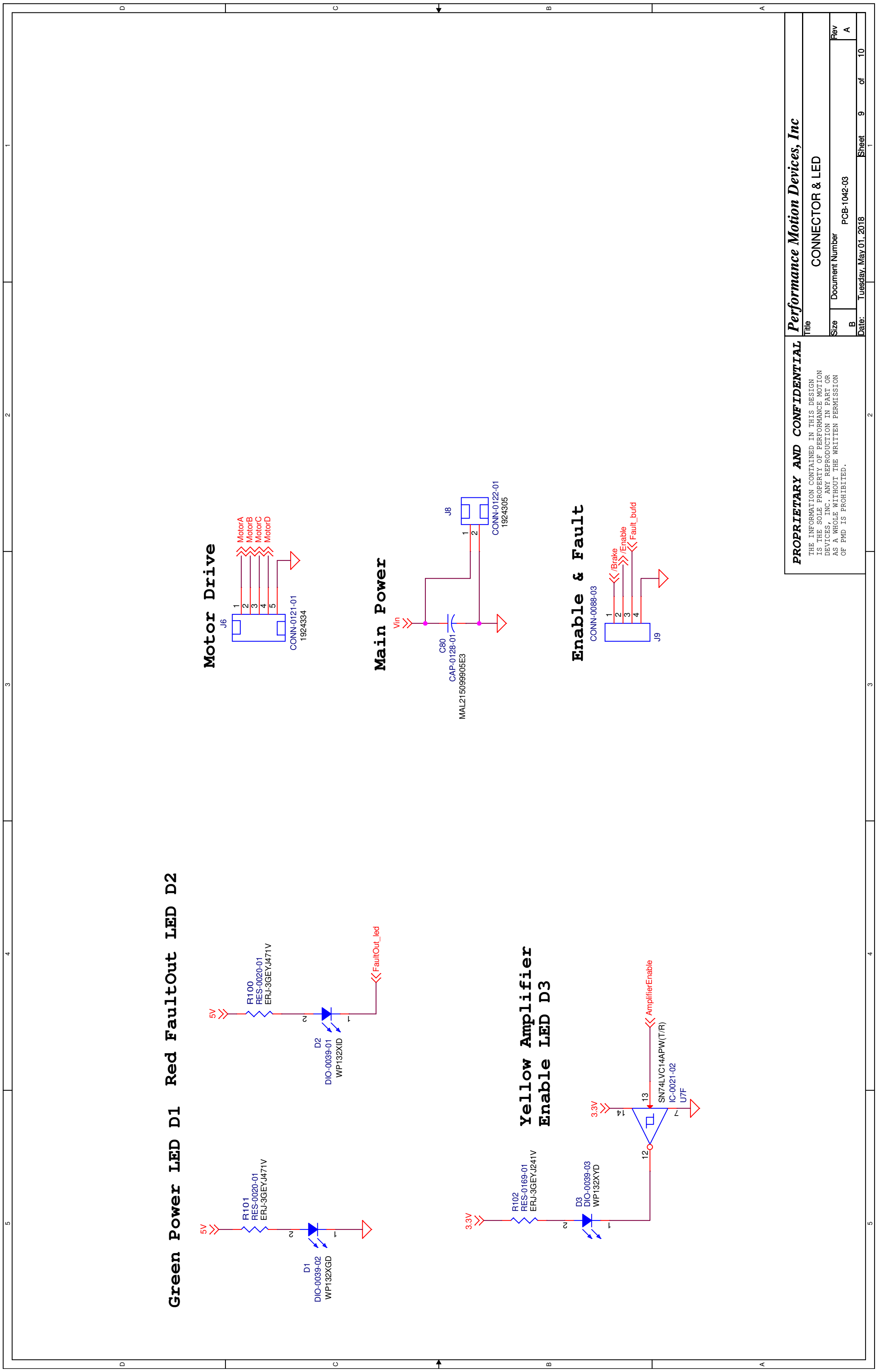


Figure B-8: DK78113 Board Schematic, On-Board Power

Figure B-9: DK78113 Board Schematic, Connector & LED



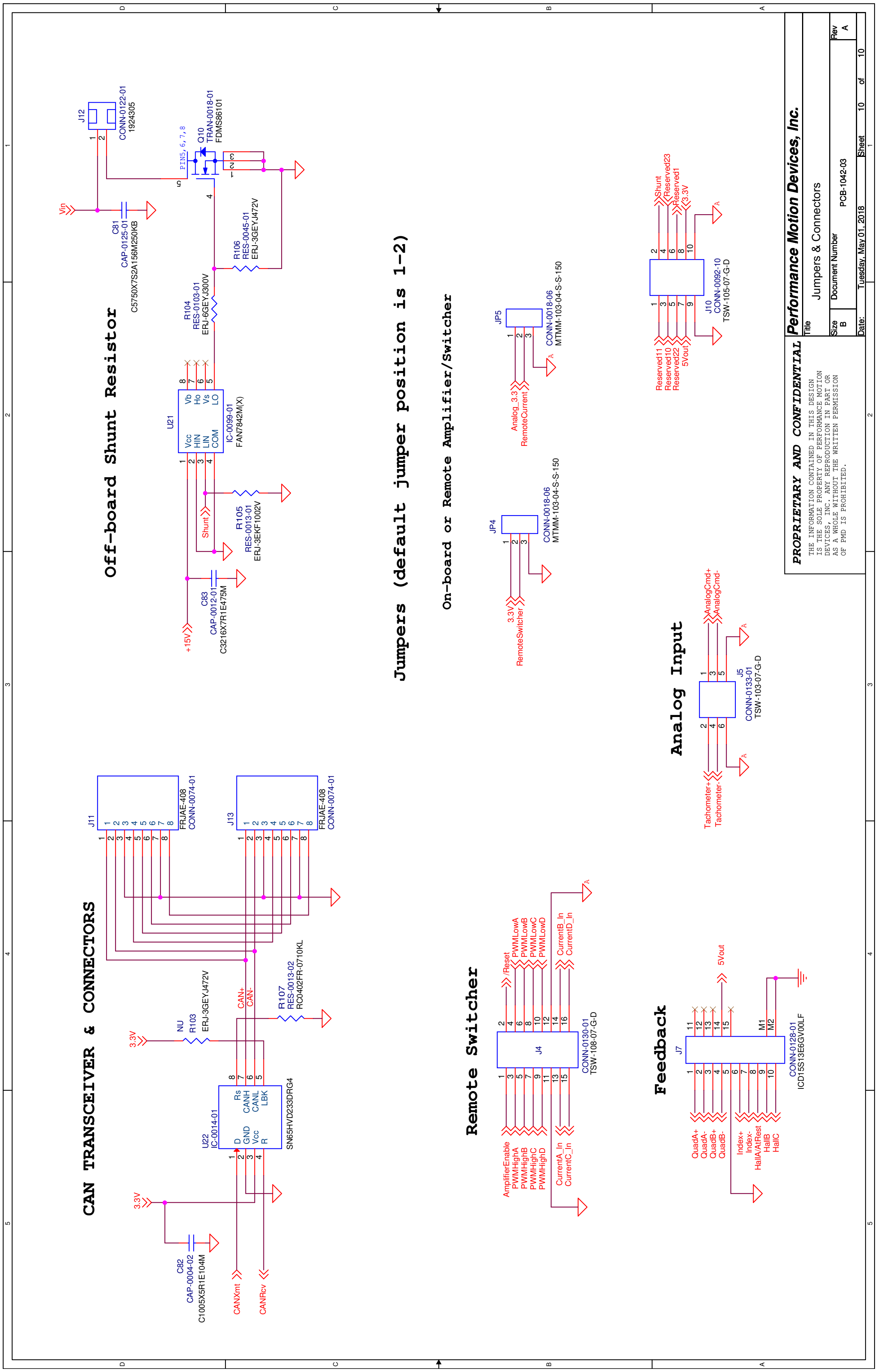


Figure B-10: DK78113 Board Schematic, Jumpers & Connectors

This page intentionally left blank.

Appendix C. Temperature Conversion Tables

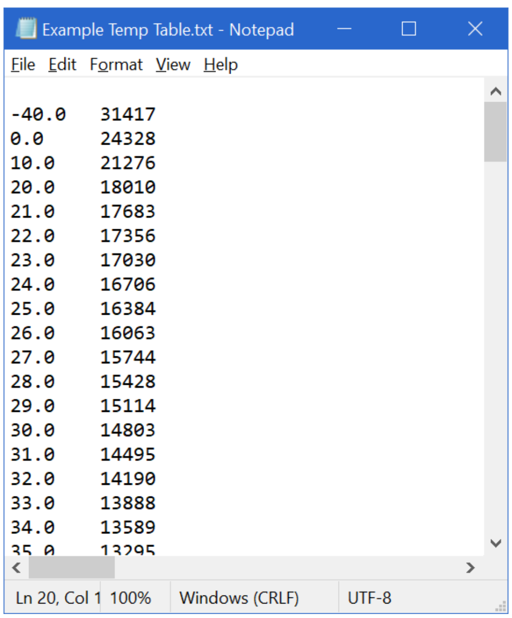
In This Appendix

- Thermistor Function and Table Verification

Pro-Motion has the ability to input an ASCII text file containing a table allowing it to convert ADC (Analog to Digital Converter) readings from the MC7x112's analog Temperature signal input into degrees C.

When the DK78113 board is initialized using the Pro-Motion Axis Wizard a temperature conversion file is automatically loaded so that readings from the temperature sensor on the DK78113 board are displayed correctly in degrees C. Pro-Motion can load other conversion files as long as their content follows the correct format. To load a conversion table manually use the Drive Safety button in Pro-Motion's Axis Control window and click the Drive Signal Scaling button to enter the file name.

The content of the conversion file, an example of which is shown below, consists of a series of entries with two numbers per line. The first is a floating point number with units of degrees C, and the second is an integer representing the corresponding ADC value from 0 to 32,767. The first number is separated from the second by a <Tab> character, and each line is terminated by a <NEWLINE> character.


Example Temp Table.txt - Notepad
File Edit Format View Help

-40.0	31417
0.0	24328
10.0	21276
20.0	18010
21.0	17683
22.0	17356
23.0	17030
24.0	16706
25.0	16384
26.0	16063
27.0	15744
28.0	15428
29.0	15114
30.0	14803
31.0	14495
32.0	14190
33.0	13888
34.0	13589
35.0	13295

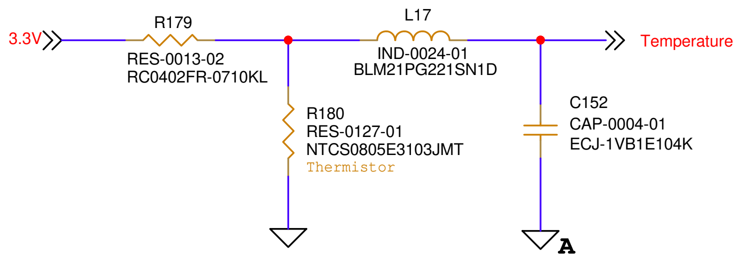
Ln 20, Col 1 100% Windows (CRLF) UTF-8

Degree entries do not need to be exactly one degree apart or even at equal temperature intervals. The only requirement is that the temperature entries be monotonic and increasing (the first entry should be the lowest temperature value and each entry after should be a higher temperature than the value immediately preceding it). Pro-Motion parses the table and linearly interpolates the values from the ADC to convert to degrees C.

Both 'increasing resistance with increasing temperature' and 'decreasing resistance with increasing temperature' thermistors can be handled by Pro-Motion. Either way the table should just list the ADC value associated with a series of increasing temperature values.

A typical circuit for processing resistance values in the thermistor into a voltage at the MC7x112's Temperature signal input is shown below. This signal input expects a voltage in the range of 0V to 3.3V (unless otherwise set via the AnalogRefHigh and AnalogRefLow signals) representing the sensed temperature, with corresponding ADC value table entries in the range of 0 to 32,767.

Figure C-1: Typical Thermistor Processing Circuit



If the thermistor's data sheet shows the correspondence of temperature to resistance it is generally straightforward to construct a conversion table. For this circuit, which uses a 10 Kohm voltage divider, with a thermistor resistance of R at a particular temperature the ADC value entry TE is expressed by:

$TE = 32{,}767 * R/(10{,}000 + R)$

C.1 Thermistor Function and Table Verification

Confirming that a newly developed thermistor circuit and conversion table are working correctly is highly recommended. Accurate temperature readings are vital for the correct functioning of the MC7x112's over temperature detect safety feature.

There may be several ways to do this, but in the description below a calibrated oven with a settable temperature, an operating version of the thermistor signal processing circuit connected to the MC7x112, and Pro-Motion is used.

The circuitry should be powered on and located inside the oven, and Pro-Motion should be in communication with the MC7x112 IC using the serial host connection. The Pro-Motion Scope window should be setup to trace two variables at the same time:

- the converted drive temperature (specify the *Temperature* variable in the Drive data category)
- the raw temperature signal input (specify the *Temperature (raw)* variable in the Analog Inputs (raw) data category

Select a relatively slow capture rate with a Trace Period of perhaps 20,000 cycles, and select a trace length of perhaps 1,000. Set the oven to the initial temperature and when it stabilizes select "Start Trace" to begin the trace. With a trace period of 20,000 cycles one data point per second will be collected. With the trace still running command the oven to the next temperature (perhaps 5 degC higher) and when the temperature stabilizes repeat for increasing temperatures until the highest desired temperature is reached. To stop the trace (if it hasn't stopped already) select "Stop Trace". Once the trace is completed you may find it useful to store the collected trace data in a spreadsheet. This can be done using the Export Trace function of Pro-Motion's File Menu item.

The advantage to simultaneously capturing both the converted temperature readings and the corresponding raw analog values is that it may help you diagnose any problems with the circuit or table if the converted temperature readings are incorrect.

An alternative, simpler approach is to just view the Pro-Motion Status window which continuously displays the converted temperature reading from the thermistor. You can visually confirm that the oven temperature and reported temperature in the MC7x112 Status Window are acceptably close over the desired range of temperature readings.