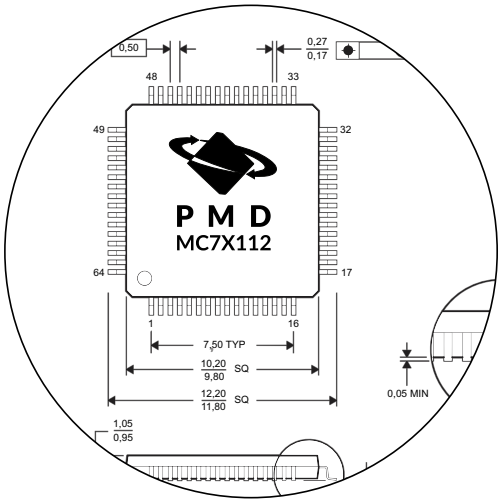
PERFORMANCE MOTION DEVICES

MOTION CONTROL AT ITS CORE



Juno™ MC7x112 Torque Control IC

User Guide

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 Developer Kit User Manual

How to install and configure the MC7x112 Developer Kit. Documents both the 56-pin VQFN and 64-pin TQFP developer kits for the MC7x112 torque control ICs including MC71112, MC71112N , MC73112, and MC73112N.

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

This page intentionally left blank.

List of Figures

1. Introduction

1

In This Chapter

- Juno IC Family Overview
- Part Numbers and Configurations
- Juno Torque Control IC Developer Kits
- Juno IC Family Overview
- Juno Torque Control ICs Typical Applications

1.1 Juno IC Family Overview

This manual describes the MC71112, MC71112N, MC73112, and MC73112N ICs from Performance Motion Devices, Inc. These devices comprise the torque control ICs of PMD's Juno Velocity & Torque Control IC family. Additional members of the family are the MC71113, MC73113, and MC78113 ICs for velocity control of DC Brush and Brushless DC motors.

MC7x112 torque control ICs are ideal for a wide range of applications including precision liquid pumping, laboratory automation, scientific automation, flow rate control, pressure control, high speed spindle control, and many other robotic, scientific, and industrial applications.

These ICs provide full four quadrant motor control and directly input quadrature encoder, index, and Hall sensor signals. They interface to external bridge-type switching amplifiers utilizing PMD's advanced current and switch signal technology for ultra smooth, ultra quiet motor operation.

MC7x112 torque control ICs can be operated with SPI (Serial Peripheral Interface) or direct analog command input. Alternatively they can interface to a host processor via point-to-point asynchronous serial. Additional features include motion trace, programmable event actions, FOC (Field Oriented Control), external shunt resistor DC bus voltage supply control, and more.

1.2 Part Numbers and Configurations

There are four different MC7x112 torque control ICs providing DC Brush motor control and Brushless DC motor control, each provided in a 64-pin TQFP package and a 56-pin VQFN package.

The following table shows the available part numbers:

P/N	Package	Motor Supported	Comments
MC71112	64-pin TQFP	DC Brush	Torque Control
MC71112N	56-pin VQFN	DC Brush	Torque Control
MC73112	64-pin TQFP	Brushless DC	Torque Control
MC73112N	56-pin VQFN	Brushless DC	Torque Control

1.3 Juno Torque Control IC Developer Kits

Four different Juno torque control IC developer kits are available. 64-pin TQFP package Juno ICs and 56-pin VQFN package ICs are supported via two different boards. The DK part numbers also differ in the specific type of IC that is installed.

Developer Kit P/N	IC Installed	Motor supported	Comments
DK71112	MC71112	DC Brush	64-pin TQFP package
DK73112	MC73112	Brushless DC	64-pin TQFP package
DK71112N	MC71112N	DC Brush	56-pin VQFN package
DK73112N	MC73112N	Brushless DC	56-pin VQFN package

Each developer kit includes:

- Standalone board with easy to access connectors for fast setup and testing
- Autotuning and axis wizard setup software
- Extensive application schematic examples

1.4 Juno IC Family Overview

The following table summarizes the operating modes and control interfaces supported by the complete Juno IC family.

Note that the MC78113 IC allows the motor type to be selected by the user. It provides all of the operating modes indicated for the MC71113 and MC73113 ICs.

	MC71112 MC71112N	MC71113 MC78113	MC73112 MC73112N	MC73113 MC78113
Motor Type & Control Mode				
Motor Type	DC Brush	DC Brush	Brushless DC	Brushless DC
Velocity		✓		✓
Torque/current	✓	✓	✓	✓
Position & outer loop		✓		✓
Host Interface				
Serial point-to-point	✓	✓	✓	✓
Serial multi-drop		✓		✓
SPI		✓		✓
CANbus		✓		✓
Command Input				
Analog velocity or torque	✓	✓	✓	✓
SPI velocity or torque	✓	✓	✓	✓
Pulse & direction		✓		✓
SPI position increment				✓
Motion I/O				
Quadrature encoder input		✓	✓	✓
Hall sensor input			✓	✓
Tachometer input		✓		✓
AtRest input				
FaultOut output	✓	✓	✓	✓
HostInterrupt output	✓	✓	✓	✓
Amplifier Control				
PWM High/Low	✓	✓	✓	✓
PWM Sign/Magnitude	✓	✓		
DC Bus & Safety				

	MC71112 MC71112N	MC71113 MC78113	MC73112 MC73112N	MC73113 MC78113
Shunt	✓	✓	✓	✓
Overcurrent detect	✓	✓	✓	✓
Over/undervoltage detect	✓	✓	✓	✓
Temperature input	✓	✓	✓	✓
Brake	✓	✓	✓	✓

1.5 Juno Torque Control ICs Typical Applications

1.5.1 Torque Control of Brushless DC and DC Brush Motors Using Analog Command

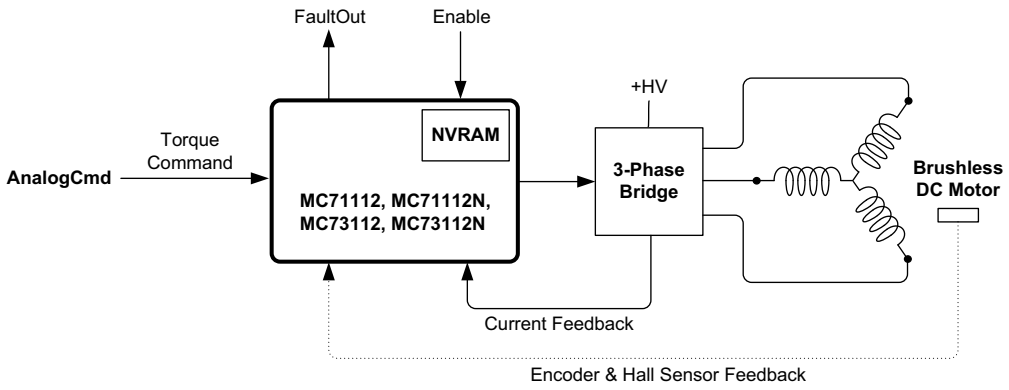


Figure 1-1: Analog Command Torque Control

Applications: General purpose amplifier, spindle control, centrifuge control, drug infusion, precision liquid pumping, turbine control.

In this configuration the MC71112, MC71112N, MC73112, or MC73112N IC receives direct analog commands representing the instantaneous desired torque. For Brushless DC motors Hall sensors normally provide commutation feedback. If encoder signals are available however Halls are not required as long as the motor can move freely during startup (using an algorithmic phase initialization procedure). For DC Brush motors neither Hall sensor or encoder signals are used for torque control.

In the diagram above a Brushless DC motor is shown but similar torque control can be provided for DC Brush motors.

1.5.2 Torque Control of Brushless DC and DC Brush Motors Using SPI Command

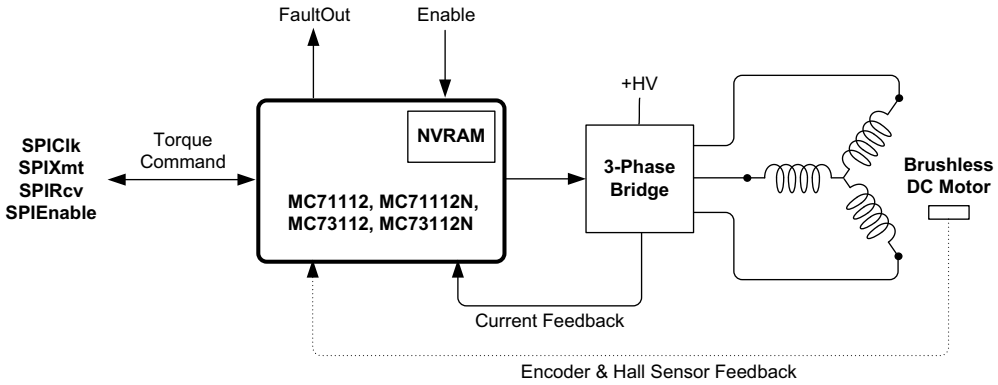


Figure 1-2: SPI Command Torque Control

Applications: General purpose amplifier, spindle control, centrifuge control, drug infusion, precision liquid pumping, turbine control.

In this configuration the MC71112, MC71112N, MC73112, or MC73112N IC receives 16-bit digital SPI (Serial Peripheral Interface) commands representing the instantaneous desired torque. For Brushless DC motors Hall sensors normally provide commutation feedback. If encoder signals are available however Halls are not required as long as the motor can move freely during startup (using an algorithmic phase initialization procedure). For DC Brush motors neither Hall signals nor encoder signals are used.

The above diagram shows a Brushless DC motor but similar torque control can be provided for DC Brush motors.

1.5.3 Torque Control of Brushless DC and DC Brush Motors Using Serial Host Command

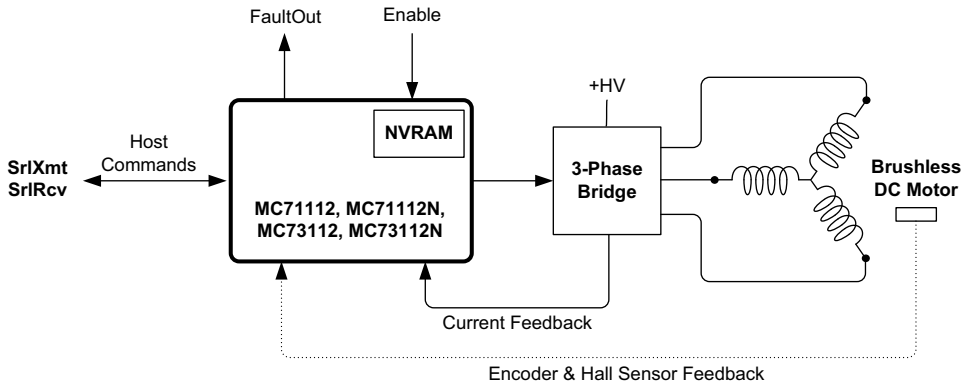


Figure 1-3: Serial Host Command Torque Control

Applications: Microprocessor controlled servo motor drive, programming of MC7x112 IC NVRAM.

In this configuration a microprocessor or PC communicates via the MC7x112's serial host command port to control the motor torque. For Brushless DC motors Hall sensors normally provide commutation feedback. If encoder signals are available however Halls are not required as long as the motor can move freely during startup (using an algorithmic phase initialization procedure). For DC Brush motors neither Hall signals nor encoder signals are used.

Another use of this configuration is during set up to store the production configuration of the MC7x112 IC into its NVRAM. For more information on NVRAM setup refer to Section 6.1, "Loading the NVRAM."

1.5.4 Torque Control Of Servo Motors with PMD Magellan Motion Control IC

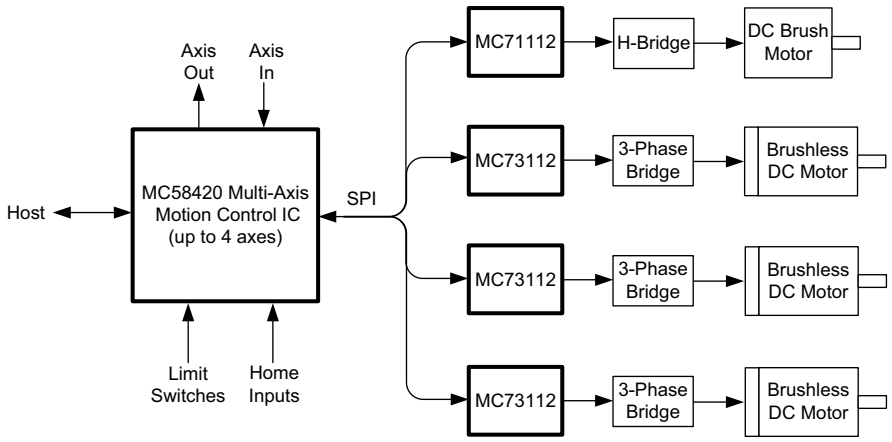


Figure 1-4: Torque Control with Magellan MC5x000 Motion Control IC

Applications: General purpose multi-axis motion control, laboratory automation, scientific instruments, XY stages, multi-dimensional contouring, semiconductor equipment.

PMD's MC5x000 Magellan Motion Control ICs provide up to four axes of profile generation, position servo loop control, pulse & direction signal generation, and numerous other synchronization and control features. In this application one or more MC7x112 ICs (one per motor axis) connect to the Magellan IC via SPI and provide high performance current control and motor amplifier management for Brushless DC or DC Brush motors.

For more information on MC5x000 positioning control ICs refer to the *Magellan Motion Control IC User Guide*.

This page intentionally left blank.

2. Functional Characteristics

In This Chapter

2.1 Configurations, Parameters, and Performance

Control command sources	AnalogCmd	Torque command is provided via external direct analog input.
	SPI	Torque command is provided via external SPI (Serial Peripheral Interface) direct input.
	Serial port	Torque command is provided via host command using point-to-point serial port.
Host communication modes	Point to point asynchronous serial	
Serial port baud rate range	1200 baud to 460800 baud (1200, 2400, 9600, 19200, 57600, 115200, 230400, 460800)	
Motor output modes	PWM High/Low	Individual high and low drive signals for each bridge switch.
	Sign/Magnitude PWM	Sign and magnitude drive signals for an H-bridge.
Commutation rate	39.06 kHz	
Current loop rate	19.53 kHz	
Current command input sampling rate	9.765 kHz	
Current measurement resolution	12 bits	
PWM resolution & rates	1:1,536 @ 20 kHz 1:768 @ 40 kHz 1:384 @ 80 kHz 1:256 @ 120 kHz	
DC Bus & safety signals	*Brake, BusVoltage, BusCurrentSupply, Temperature, Shunt*	
Amplifier output signals	*AmplifierEnable, PWMHighA, PWMLowA, PWMHighB, PWMLowB, PWMHighC, PWMLowC*	
Serial communication signals	*SrlXmt, SrlRcv*	
Analog command signals	*AnalogCmd*	
SPI command signals	*SPIXmt, SPIRcv, SPIClk, SPIEnable*	
SPI frequency range	1.0 MHz - 10.0 MHz	
Encoder input signals	*QuadA, QuadB, Index*	
Position range	-2,147,483,648 to +2,147,483,647 counts	
Hall sensor signals	*Hall A-C*	
Miscellaneous control signals	*Enable, FaultOut, HostInterrupt, Reset*	
Drive safety functions	Over current detect, over temperature detect, overvoltage detect, undervoltage detect, I^2t current foldback.	
Brake input modes	Passive braking, full disable	
Output limiting	I^2t, current, and voltage limit	

Maximum encoder rate	40 Mcounts/sec
Cycle time range	102.4 microseconds to 1.114 seconds
Position-capture triggers	*Index* signal
Internal RAM	6,144 16-bit words
Maximum number of simultaneous trace variables	4
NVRAM storage size	1,024 16-bit words

2.2 Physical Dimensions, 64-PIN TQFP Package

All dimensions are in millimeters.

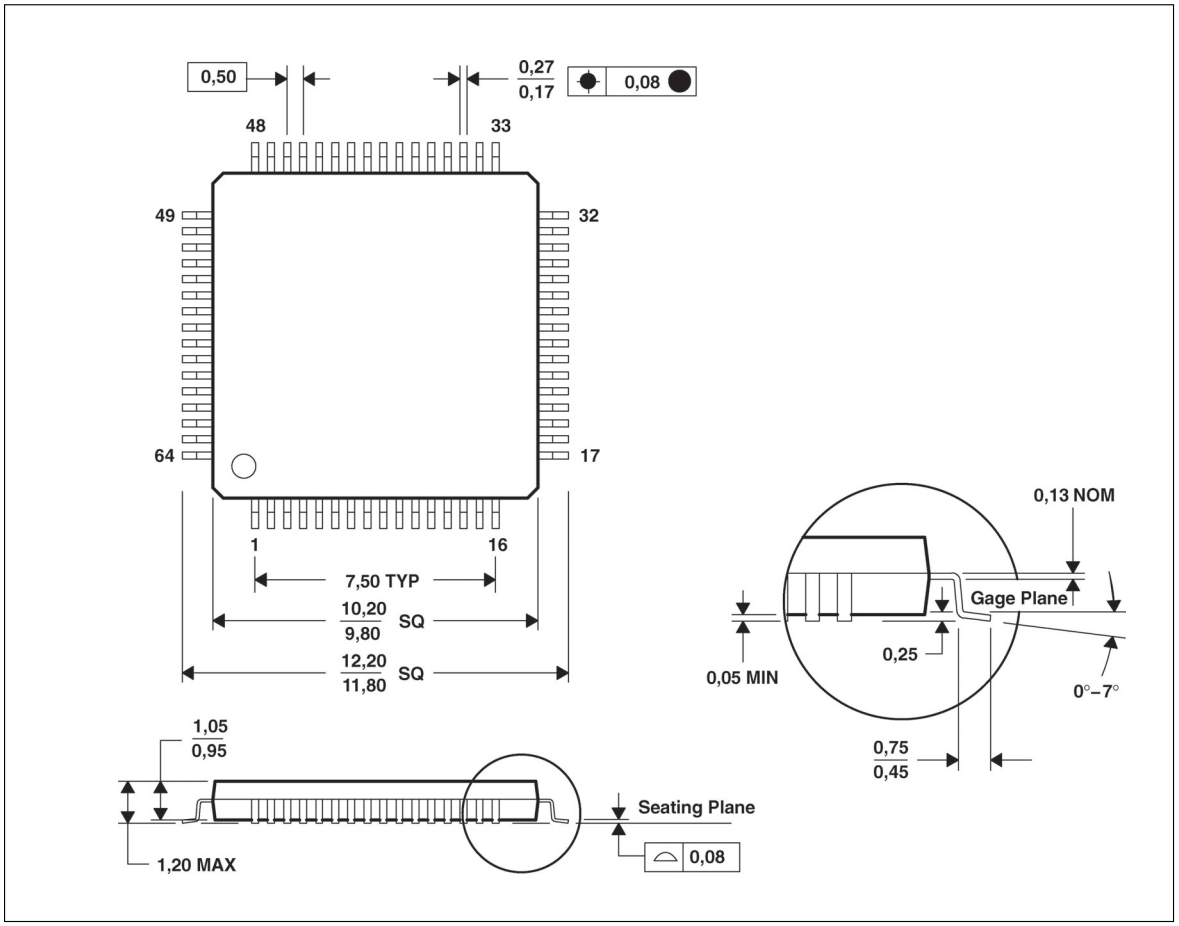


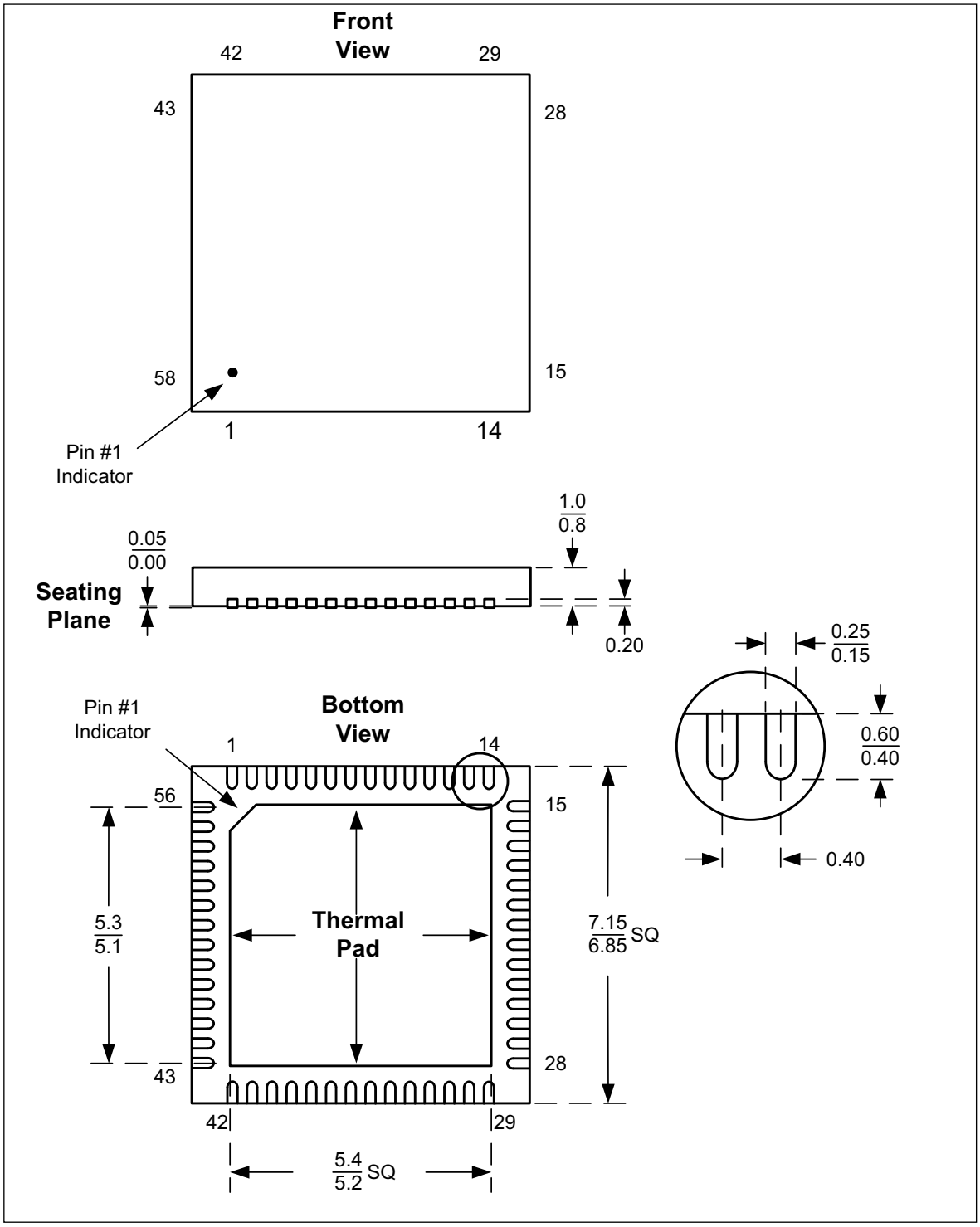
Figure 2-1: 64-Pin TQFP Physical Dimensions

Notes:

1. *MC7x112 ICs are RoHS compliant and free of Bromine and Antimony based flame retardants.*
2. *Moisture sensitive level: MSL 3*

2.3 Physical Dimensions, 56-PIN VQFN Package

Figure 2-2: 56-Pin VQFN Physical Dimensions



Notes:

1 *MC7x112 ICs are RoHS compliant and free of Bromine and Antimony based flame retardants.*

2 *Moisture sensitive level: MSL 3*

2.4 Absolute Maximum Environmental and Electrical Ratings[1]

Supply voltage (Vcc)	-0.3V to +4.6V
FltCap pin voltage range	-0.3V to 2.5V
Analog voltage range with respect to AnalogGND (AnalogVcc)	-0.3V to +4.6V
Input voltage (Vi)	-0.3V to +4.6V
Output voltage (Vo)	-0.3V to +4.6V
Input clamp current (Ii,clamp), peak:	+/-20 mA
continuous:	+/-2 mA
Output clamp current (Io,clamp)	+/-20 mA
Package thermal impedance, 64-pin TQFP package (θ_{JA})	56.5°C/W
Package thermal impedance, 56-pin VQFN package (θ_{JA})	34.8°C/W
Junction temperature range (Tj)[(2)]	-40°C to 150°C
Storage temperature range (Ts)	-65°C to 150°C
Nominal clock frequency (FcIkin)	10.0 MHz

(1) Unless otherwise noted, all voltages are with respect to GND
(2) Please refer to "Design tips" section for more information

This page intentionally left blank.

3. Electrical Characteristics

3

In This Chapter

- DC Characteristics
- AC Characteristics

3.1 DC Characteristics

(Vcc and Ta per operating ratings, Fclkin = 10.0MHz)

Symbol	Parameter	Minimum	Maximum	Conditions
V_{cc}	Supply voltage	2.97V	3.63V	With respect to GND
I_{dd}	Supply current		115 mA	All I/O pins are floating
AnalogV_{cc}	Analog input supply voltage	2.97V	3.63V	With respect to AnalogGND
AnalogI_{dd}	Analog supply current		18 mA	
T_a	Operating free-air temperature	-40°C	105°C	Note 2
T_j	Operating junction temperature	-40°C	150°C	
Input Voltage				
V_{ih}	Logic 1 input voltage	2.0V	Vcc + 0.3V	
V_{il}	Logic 0 input voltage	-0.3V	0.8V	
Output Voltage				
V_{oh}	Logic 1 Output Voltage	2.4V		Io = -4 mA
		Vcc-0.2V		Io = -50 µA
V_{ol}	Logic 0 Output Voltage		0.4V	Io = 4 mA
Other				
I_{out}	Tri-state output leakage current	-2 µA	2 µA	Vo = 0 or Vcc
I_{in}	Input current	2 µA	-205 µA	Vcc = 3.3V with internal pullup
$I_{in,\sim RESET}$	Input current for ~RESET pin	2 µA	-375 µA	Vcc = 3.3V
C_i	Input capacitance		2 pF	typical
V_{fltcap}	FltCap voltage		1.9V	typical
V_{reset}	Vcc BOR trip point	2.50V	2.96V	Falling Vcc
$V_{reset,hys}$	Vcc BOR hysteresis		35 mV	typical
T_{reset}	BOR reset release delay time	400 µs	800 µs	Time from removing reset to ~RESET release

Symbol	Parameter	Minimum	Maximum	Conditions
Analog Input				
V_{analog}	Analog input voltage range	0	3.3V	With respect to AnalogGND
C_{ai}	Analog input capacitance		5 pF	typical
E_{dnl}	Differential nonlinearity error. Difference between the step width and the ideal value. No missing codes.	-1	1	LSB
E_{inl}	Integral nonlinearity error. Maximum deviation from the best straight line through the ADC transfer characteristics, excluding the quantization error.	-4	4	LSB
E_{zo}	Zero-offset error	-4	4	LSB

Notes:
(1) V_{cc} and AnalogV_{cc} should be within 0.3V from each other.
(2) Please refer to design tips in the Application Notes of this manual for thermal design considerations.

3.2 AC Characteristics

Refer to Chapter 4, *Timing Diagrams*, for timing diagrams.

3.2.1 Quadrature Encoder Input

Timing Interval	No.	Min	Max
Encoder pulse width	T4	33.3 nSec	
Dwell time per state	T5	16.7 nSec	
~Index active pulse time	T6	33.3 nSec	

Notes:
(1) ~Index is defaulted as active low, which is shown here. It can be configured to be active high.

(2) Rising edge of Index signal must occur 25 nSec or more before AB count state end to guaranteed capture occurs on same count. Otherwise capture may register on next count.

3.2.2 SPI

Timing Interval	No.	Min	Max
SPIClock clock cycle time	T23	100 nSec	1,000 nSec
Pulse duration, SPIClock high	T24	33 nSec	
Pulse duration, SPIClock low	T25	33 nSec	
~SPIEnable low to first SPIClock high	T26	25 nSec	
SPIClock high to SPIXmt valid delay time	T27		21 nSec
SPIXmt data valid time after SPIClock low	T28	T25	
SPIRcv setup time before SPIClock low	T29	25 nSec	
SPIRcv valid time after SPIClock low	T30	25 nSec	
Last SPIClock low to ~SPIEnable high	T31	25 nSec	

3.2.3 Power-on Reset

Timing Interval	No.	Min	Max
Power on pulse duration driven by device (typical)	T32		600 μSec
Device ready/ outputs initialized (typical)	T33		2 mSec

Note: The device will generate a ~*Reset* pulse upon power on. An external ~*Reset* signal is optional.

3.2.4 Warm Reset

Timing Interval	No.	Min	Max
~Reset low duration for warm reset	T35	570 nSec	

This page intentionally left blank.

4. Timing Diagrams

In This Chapter

For the values of *Tn*, please refer to the table in Section 3.2, "AC Characteristics" for more information

4.1 Quadrature Encoder Input

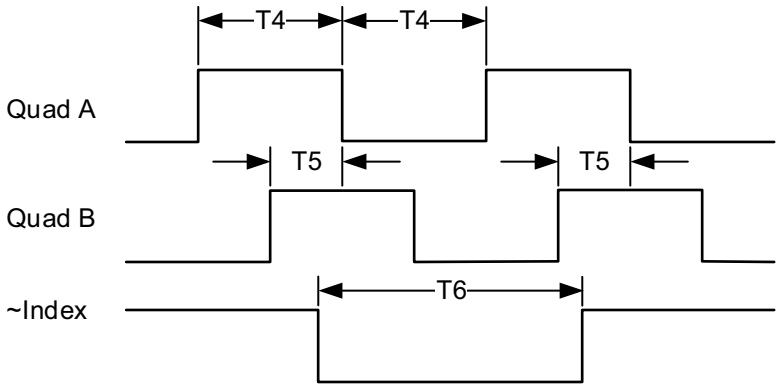
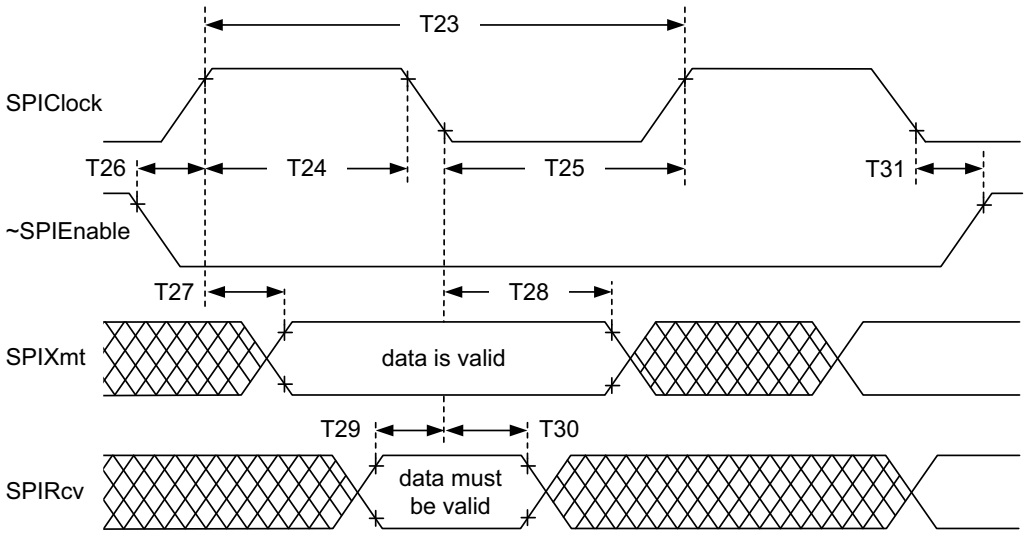


Figure 4-1: Quad Encoder Timing

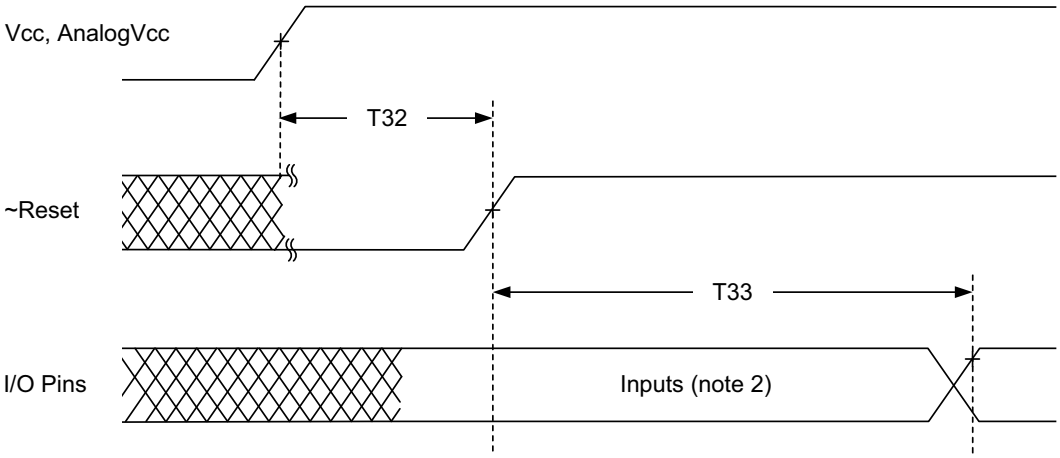
4.2 SPI Timing

Figure 4-2: SPI Timing



4.3 Power On Timing

Figure 4-3: Power On Timing



4.4 Warm Reset

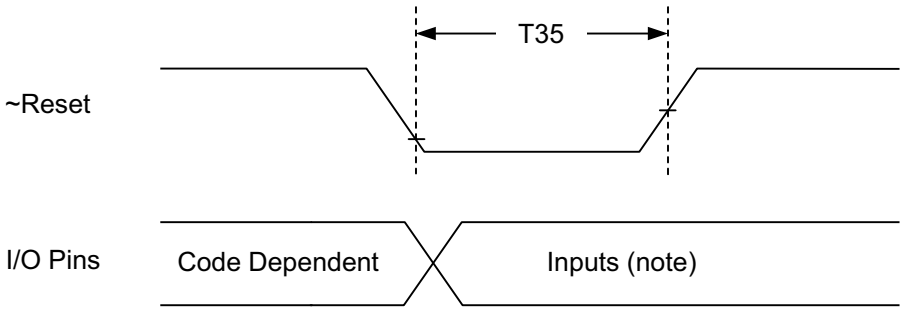


Figure 4-4: Warm Reset Timing

This page intentionally left blank.

5. Pinouts and Pin Descriptions

In This Chapter

5.1 Pinouts for the MC71112

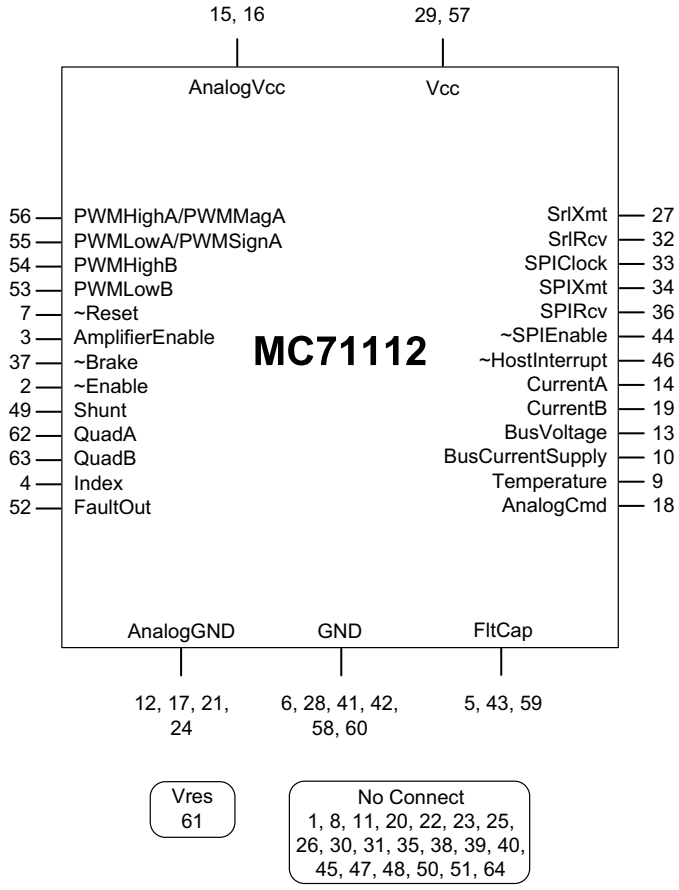
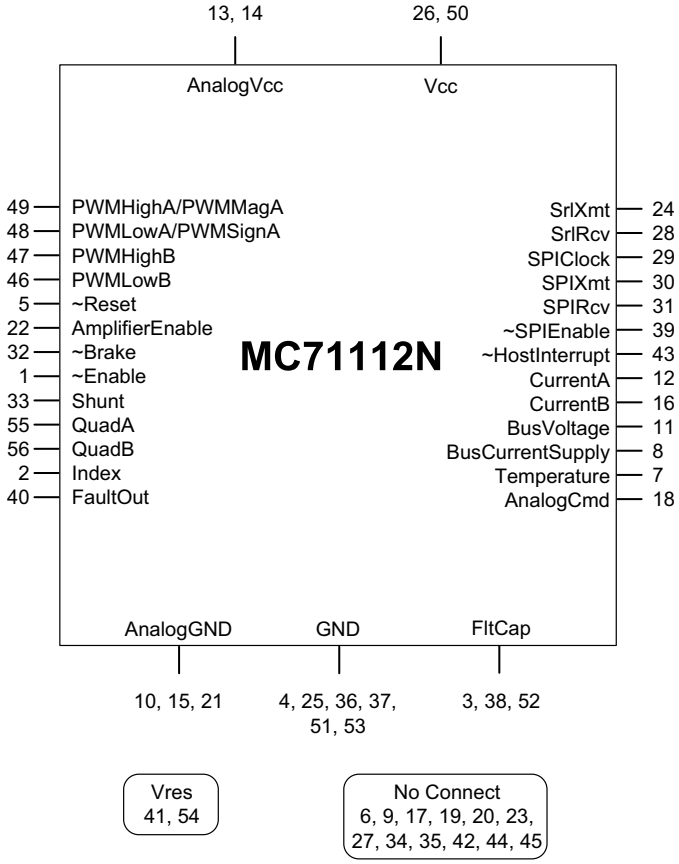


Figure 5-1: MC71112 Pinouts

5.2 Pinouts for the MC71112N

Figure 5-2: MC71112N Pinouts



5.3 Pinouts for the MC73112

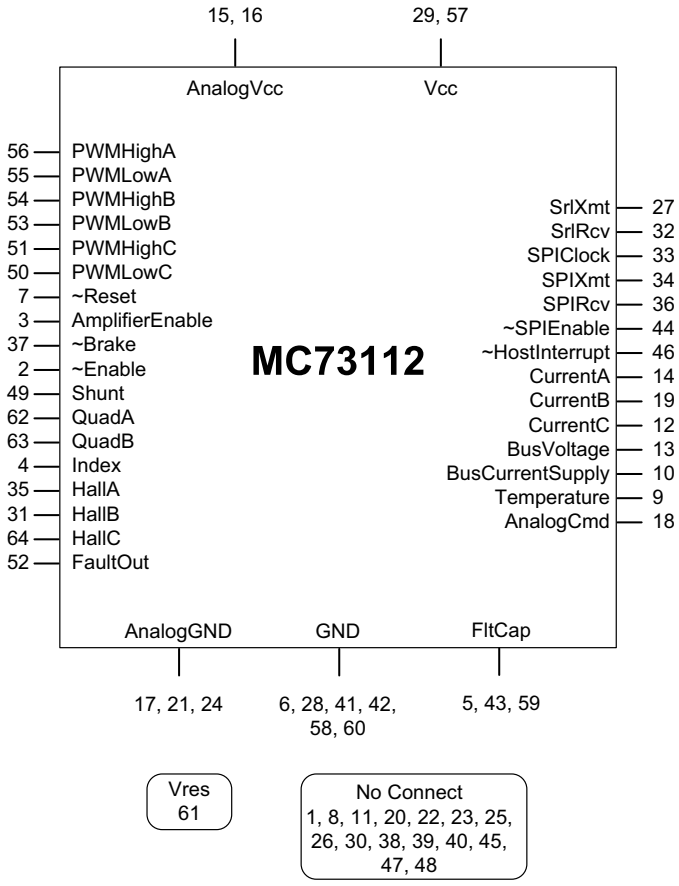
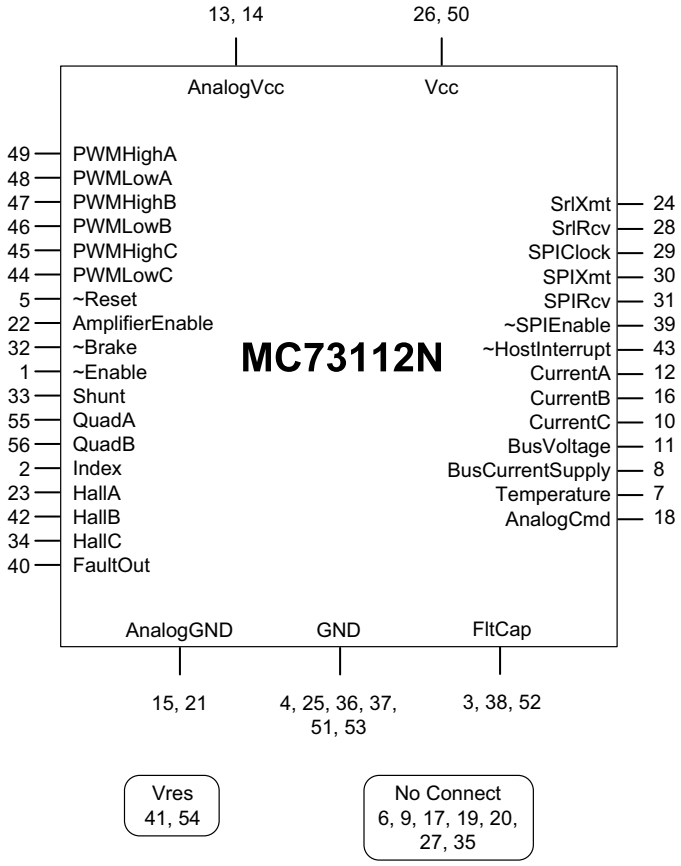


Figure 5-3: MC73112 Pinouts

5.4 Pinouts for the MC73112N

Figure 5-4: MC73112N Pinouts



5.5 MC71112, MC71112N, MC73112, MC73112N Pin Descriptions

The following table details the pinouts for the MC71112, MC71112N, MC73112, and MC73112N ICs.

Name	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Direction	Description
~Reset	7	5	in/out	This pin is the master reset input. Although during powerup it is not necessary to provide a reset because MC7x112 ICs generate an internal reset upon powerup, if a manual reset is desired this signal may be temporarily brought low and then restored to high. During a power-on or reset condition this pin is driven low by the MC7x112 IC. If this pin is driven manually, it must be driven with an open drain output device. For correct reset operation a 10k pull-up resistor should be added between Reset and Vcc. In addition, a 100nF or smaller capacitor is recommended between Reset and GND.
PWMHighA/ PWMMagA PWMLowA/ PWMSignA PWMHighB PWMLowB PWMHighC PWMLowC	56 55 54 53 51 50	49 48 47 46 45 44	out	Depending upon the selected motor type and output mode, these pins have the following functions: *PWMHighA/B/C* signals encode the high side drive for a switching bridge with separate high/low controls. The default encoding is active high, however this can be user programmed. *PWMLowA/B/C* signals encode the low side drive for a switching bridge with separate high/low controls. The default encoding is active high, however this can be user programmed. *PWMMagA* signals encode the magnitude of a pulse width modulated output for use with switching bridges with sign/magnitude controls. The default encoding is active high, however this can be user programmed. *PWMSignA* encode the sign of the pulse width modulated output for use with switching bridges with sign/magnitude controls. The default encoding is active high, however this can be user programmed.
AmplifierEnable	3	22	out	This pin provides an amplifier enable signal that may be useful for some amplifier connection schemes. A high signal indicates that amplifier output should be enabled and a low indicates that amplifier output should be disabled. If this signal is used an external 4.7 Kohm pull down resistor is recommended to avoid glitches during reset.
~Brake	37	32	in	This pin inputs a high speed PWM output disable that may be useful for a braking function with some PWM amplifier schemes. When low, PWM output is overridden to execute a brake function or if so programmed, to be disabled. PWM operation is normal when this signal is high.
~Enable	2	1	in	This pin is an enable input. To allow normal operations a low signal is asserted. When a high signal is asserted a programmable response may disable motor control operations, although communications and various other operations are still available.

Name	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Direction	Description
Shunt	49	33	out	This pin provides a PWM-based shunt signal that may be used with an external switching circuit and high wattage resistor or other energy dissipating device to regulate supply bus overvoltage conditions. The default encoding is active high however this setting can be user programmed.
QuadA QuadB Index	62 63 4	55 56 2	in	These pins input the A and B quadrature signals along with the *Index* signal for an incremental encoder. By default a valid index pulse is recognized when ~Index transitions low, however the interpretation of this signal can be user programmed. Note: Index capture is not conditioned by the QuadA and QuadB signals. If such conditioning is not provided by the encoder then external circuitry should be used if such conditioning is desired. If unused, these pins may be left unconnected.
HallA HallB HallC	35 31 64	23 42 34	in	These pins input Hall-encoded phasing inputs for brushless DC motors. The A, B, and C signals encode six valid states as follows: A on, A and B on, B on, B and C on, C on, C and A on. By default a sensor is defined as being on when its signal is high, however this signal interpretation can be user programmed. Note: Some Hall sensors require a pull up resistor to 3.3V on each signal to establish a proper high signal. Check your Hall sensor's electrical specification. If unused, these pins may be left unconnected.
FaultOut	52	40	out	This pin provides a general purpose fault indicator that can be programmed to indicate a number of conditions including a motion error, drive exception, or various other conditions. A high indicates that a fault condition is present. If this signal is used an external 10 Kohm pull down resistor is recommended to avoid glitches during reset.
SrlXmt	27	24	out	This pin outputs serial data from the asynchronous serial port.
SrlRcv	32	28	in	This pin inputs serial data to the asynchronous serial port. If unused, this pin may be left unconnected.
SPIClock	33	29	in	*SPIClock* inputs the clock signal used with synchronous serial transfer on the SPI bus. If unused, this pin may be left unconnected.
SPIXmt	34	30	out	This pin transmits synchronous serial data on the SPI bus to the host processor.
SPIRcv	36	31	in	This pin inputs synchronous serial data for the SPI bus. If unused, this pin may be left unconnected.
~SPIEnable	44	39	in	This pin inputs an enable signal for the host communication SPI bus. This signal is active low, meaning it should be low when an SPI communication is occurring, and high at all other times. If used, this pin must be controlled in series with a 470 ohm resistor. If unused, this pin may be left unconnected.

Name	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Direction	Description
~HostInterrupt	46	43	out	This pin provides a host interrupt signal. When low, it signals an interrupt to the host processor. **Whether this pin function is used or not this pin must be connected to a 10 Kohm pullup resistor.**
CurrentA CurrentB CurrentC	14 19 12	12 16 10	in	These pins input analog voltages representing leg current flow through the low sides of the switching bridges. DC Brush motors use the A and B inputs, and Brushless DC motors use the A, B, and C inputs. These signals are only accessible when the PWM output mode is set to PWM High/Low. These signals are used when a current loop is used or when I2t current monitoring is desired. These signals are sampled by an internal A/D converter. The A/D resolution is 12 bits. The allowed signal input range is zero to 3.3V. If unused, these signals should be connected to AnalogGND.
BusVoltage	13	11	in	This pin inputs an analog voltage representing the DC bus voltage. The allowed signal input range is zero to 3.3V. If unused, this signal should be connected to AnalogGND.
BusCurrentSupply	10	8	in	This pin inputs an analog voltage representing the current through the supply terminal of the DC bus. The allowed signal input range is zero to 3.3V. If unused, this signal should be connected to AnalogGND.
Temperature	9	7	in	This pin inputs an analog voltage representing the temperature of the amplifier or other monitored circuitry. The allowed signal input range is zero to 3.3V. If unused, this signal should be connected to AnalogGND.
AnalogVcc	16 15	13 14	N/A	These pins are connected to the analog input supply voltage, which must be in the range of 3.0V to 3.6V. If analog inputs are not used, these pins should be tied to Vcc.
AnalogCmd	18	18	in	This pin is used for commanding either voltage or torque using an analog signal. If unused, this pin should be tied to AnalogGND.
AnalogGND	17 21 24	15 21	N/A	This pin should be connected to the analog input power supply return. Any unused analog inputs (CurrentA-C, BusVoltage, BusCurrentSupply, Temperature or AnalogCmd pins) should be tied to AnalogGND.
Vcc	29 57	26 50	N/A	All of these pins must be connected to the digital supply voltage, which should be in the range of 3.0V to 3.6V.
FltCap	5 43 59	3 38 52	N/A	Each of these pins must be connected to a 1.2 µF (or higher) filtering capacitor which in turn connects to GND.
Vres	61	41 54		Each of these pins must be connected to Vcc via a 10K resistor.

Name	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Direction	Description
GND	6 28 41 42 58 60	4 25 36 37 51 53	N/A	All of these pins must be connected to the digital power supply return.
No Connect	1 8 11 20 22 23 25 26 30 38 39 40 45 47 48	6 9 17 19 20 27 35	N/A	These pins must be left unconnected.
Thermal pad	N/A	T. Pad	N/A	Thermal pad on bottom of 56-pin VQFN IC package must be connected to GND. For 64-pin TQFP package there is no thermal pad.

6. IC Configuration in the Production Application

In This Chapter

- Loading the NVRAM
- Analog Signal Calibration in the Production Application

Each MC7x112 IC, before undertaking torque control, must be programmed with control parameter settings appropriate for the application that it will be used in. These control parameters include quantities such as PWM (Pulse Width Modulation) frequency, current gains, safety thresholds, and more.

Correct values for these parameters are most easily determined by PMD's Pro-Motion software, specifically via the Axis Wizard setup sequence. The axis wizard steps the user through a series of set up and verification pages, finally resulting in control parameter settings tailored for that user application.

Once the control parameters are determined the user has two choices for how they can be loaded into the active control registers of the MC7x112 IC when it is in the production PCB; they can be stored permanently into the IC's internal NVRAM (non-volatile memory) or they can be loaded at each power-up by an on-board microprocessor connected to the IC via its serial port.

Storing the NVRAM data into the MC7x112 IC is discussed in the next section, Section 6.1, "Loading the NVRAM." Applications that load the control registers via an on-board microprocessor use specially formatted host commands sent over the serial port. For more information refer to Section 7.4, "Host Commands."

6.1 Loading the NVRAM

6.1.1 NVRAM Programming via DK IC Socket

The socketed version of the DK78113 Juno Developer Kit (P/N DK78113S) can be used to program the NVRAM on 64-pin TQFP MC7x112 ICs prior to soldering into the user's production PCB. Pro-Motion as well as a more compact programming executive available from PMD supports script files to program the IC NVRAM. For more information on PMD script files refer to Section 7.4.1, "Host Command Script Files."

> The 56-pin VQFN MC7x112 IC DK does not have a socket and therefore cannot be used to program the NVRAM of production 56-pin VQFN MC7x112 ICs.

6.1.2 NVRAM Programming via 3-pin UART Cable

An alternate NVRAM programming approach is by communicating to the MC7x112 IC via serial port after it is installed in the production PCB. This approach requires that each installed IC have a 3-pin connector installed on the production board. A technician plugs into this connector and performs the NVRAM download. To be programmed the IC must receive power, so this generally means the PCB power is turned on during this procedure.

To facilitate this approach PMD provides a dedicated USB to 3-pin UART programming cable (PMD Part # Cable-USB-3P) with each MC7x112 DK. This programming cable plugs into the PC's USB port on one end and into a 1x3 3-pin 2 mm header component on the other. A representative PCB mounted header component is Samtec MTMM-103-04-x-S-150.

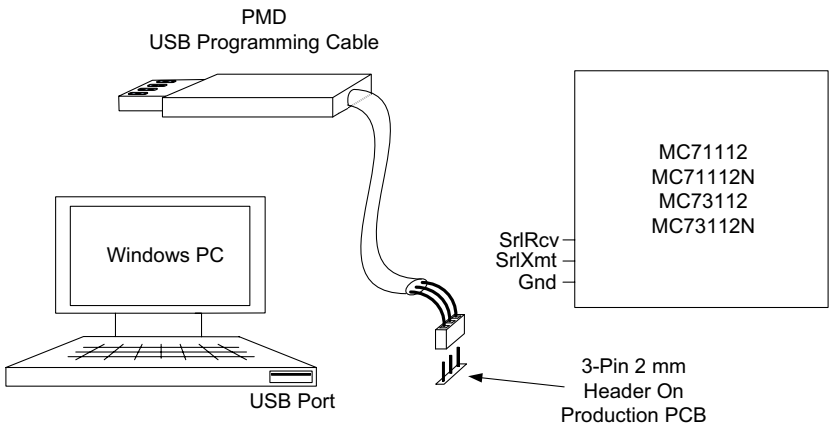


Figure 6-1: NVRAM Programming Via 3-pin UART Programming Cable

The table below shows the required wiring for the on-board connector if it is to be used with the PMD programming cable:

Connector Pin #	Pin Name	MC7x112 Pin # (64-pin TQFP Package)	MC7x112 Pin # (56-pin VQFN Package)
1	SrlXmt	27	24
2	SrlRcv	32	28
3	GND	28 (or other ground signal)	25

6.1.3 Purchasing Pre-Configured Parts

Some PMD distributors and sales outlets provide an NVRAM programming service for MC7x112 ICs. Contact your PMD sales representative for availability, terms, and conditions.

6.2 Analog Signal Calibration in the Production Application

After integration into a PCB, it is recommended that the external analog signal processing circuitry that inputs to the MC7x112 IC be calibrated. While some applications will not need these calibrations, for applications where the quietest, smoothest, and most accurate motion is desired, calibration of the analog inputs is recommended. The signals that can be calibrated are *AnalogCmd* and *CurrentA-CurrentC*. For more information refer to Section 9.3.3, "Signal Processing."

When a microprocessor is on the user PCB, generally this microprocessor is used to send the serial host commands needed to calibrate the analog inputs as part of the power up sequence.

Another approach is to have the MC7x112 IC execute the calibration procedure during its NVRAM-based initialization startup. This is done by embedding an initialization sequence that executes the calibration during the NVRAM startup at each power cycle. This approach can only be used when the startup condition of the PCB and connected motors is controllable. For example if the calibration occurs when the motors are still spinning, the calibration will not give accurate results.

A third approach that has the benefit of eliminating the need for calibration at each power cycle is to execute the calibration on the assembled PCB using a 3-pin UART programming cable. The derived calibration offsets are stored into NVRAM and recalled automatically thereafter at each power-up. For more information see Section 6.1.2, "NVRAM Programming via 3-pin UART Cable."

PMD's Pro-Motion software program provides a convenient set up facility for selecting NVRAM startup options. In addition, advanced users can directly edit start up scripts using a text editor. Refer to Section 7.4.1, "Host Command Script Files" for more information.

7. Functional Overview

7

In This Chapter

- Internal Block Diagram
- Signal Connections Overview
- Control Flow Overview
- Host Commands

7.1 Internal Block Diagram

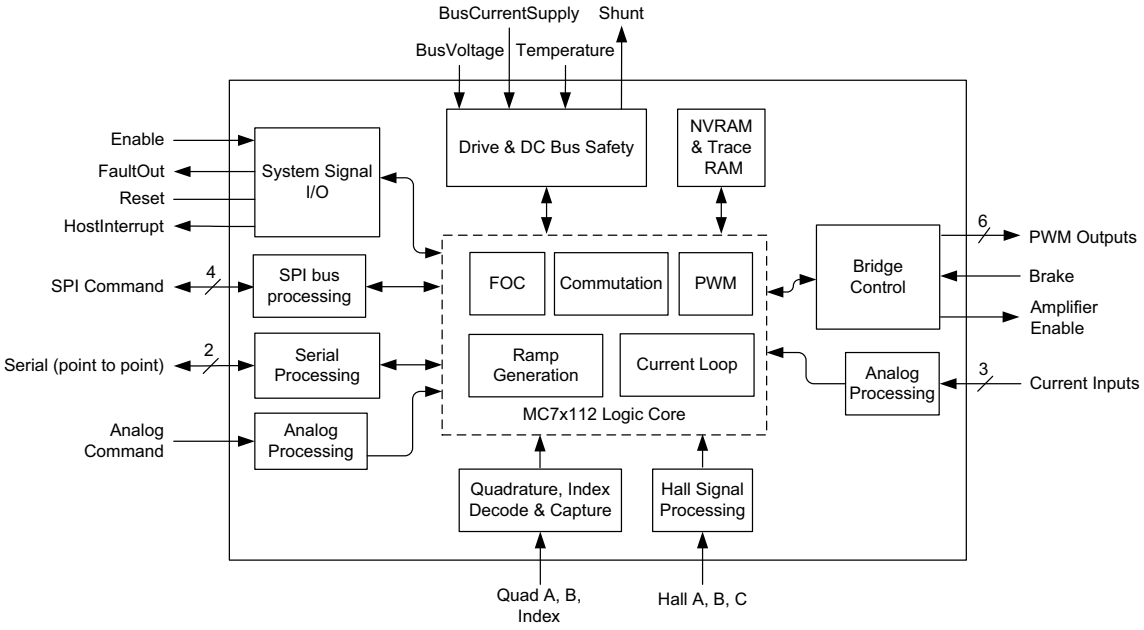


Figure 7-1: MC7x112 Internal Block Diagram

MC7x112 ICs are single-axis devices for torque or voltage-mode control of three-phase brushless DC and DC Brush motors.

At power-up or reset, the MC7x112 IC checks for the presence of stored configuration commands in its NVRAM. If NVRAM is programmed, the stored configuration commands are read into the chip, providing parameter information that will be used during operation. If no initial configuration is stored in NVRAM, then default values are used and information will then be sent by serial from a host device such as a microprocessor.

Depending on how the control loop has been configured an external analog signal may serve as the torque command value, or an SPI (Serial Peripheral Interface) data stream may be used for this command value. Alternatively an internal ramp generator commanded by a host microprocessor via the serial port may be used to generate torque command values.

Current control is performed via direct input of analog signals representing the instantaneous current through the motor coils. These signals are typically derived from external dropping resistors or Hall sensors at the amplifier circuitry. This analog current information is then combined with the desired current for each phase to generate PWM signals.

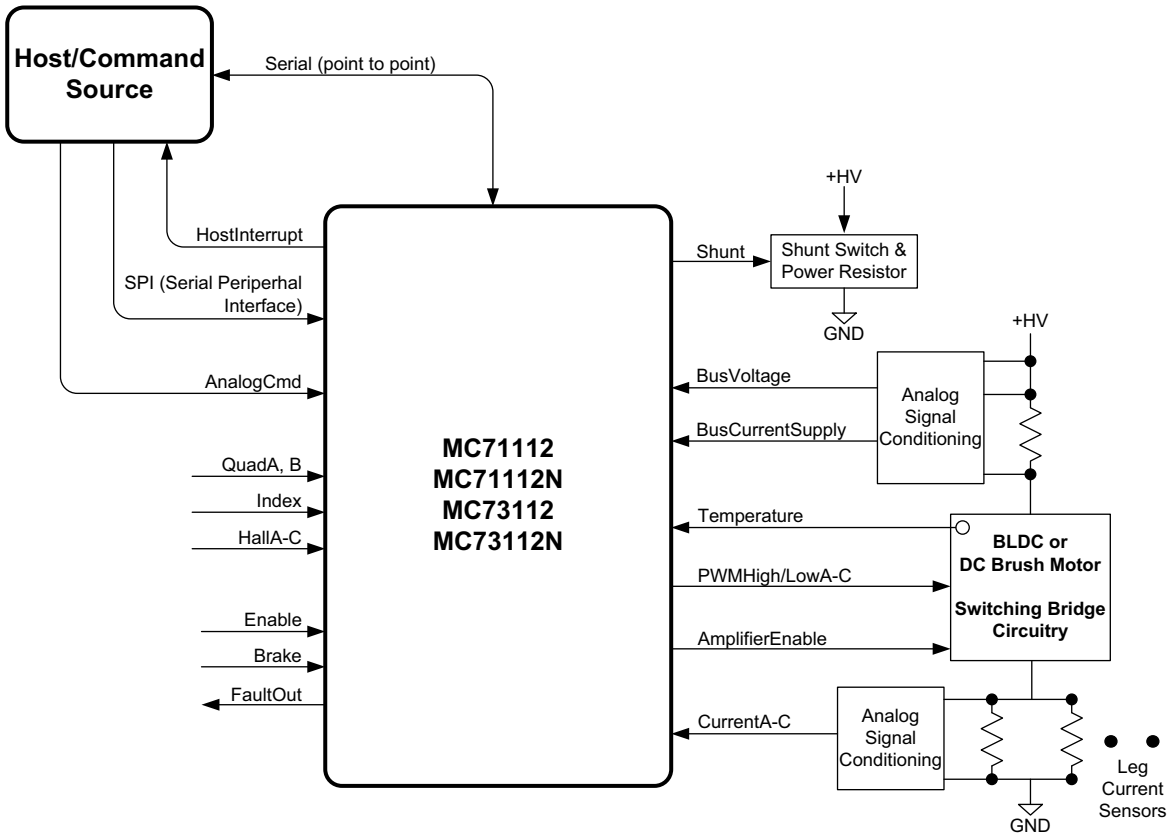
To create a complete torque controller MC7x112s are connected to switching amplifiers, typically MOSFET or IGBT-based. A programmable dead time function and other timing control parameters ensure that switch synchronization and control is optimal over the entire operating range of the driven motor.

A number of safety features are incorporated into the MC7x112 ICs including shunt control, I^2t current limiting, brake signal input, DC bus overvoltage and undervoltage detect, overcurrent detect, and overtemperature detect.

7.2 Signal Connections Overview

Figure 7-2 shows an overview of the connections used with MC7x112 ICs.

Figure 7-2: MC7x112 IC Connections Overview



7.3 Control Flow Overview

Figure 7-3 provides a control flow overview for the MC7x112 torque control ICs. It shows how a final motor command is generated starting with the command source and ending with the motor output module that generates amplifier control signals.

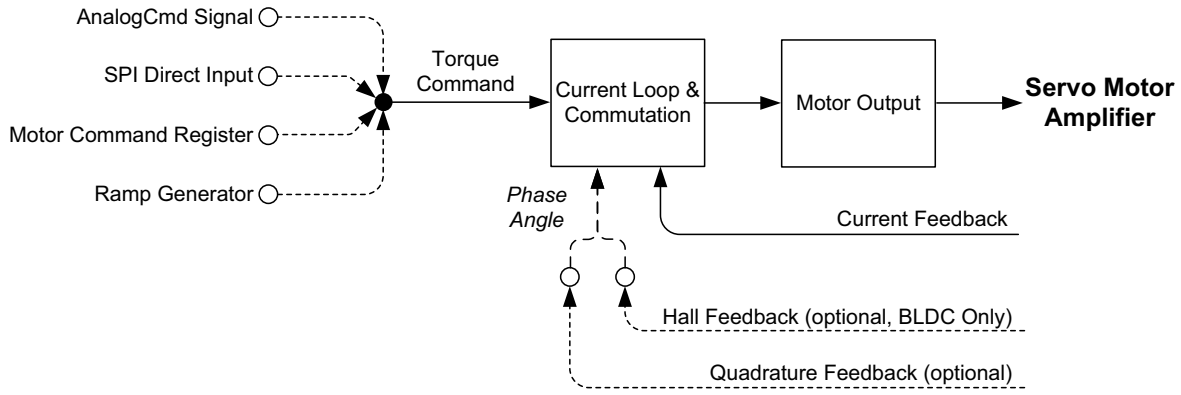


Figure 7-3: MC7x112 Control Flow Overview

When controlling a Brushless DC motor encoders are optional, but if used enable encoder-based (sinusoidal) commutation which generally provides smoother motion than Hall-based commutation. Hall signals are used with Brushless DC motors only and although not required if an encoder is connected, should be used if available. Encoders are not used with DC Brush motors.

The following table provides a brief description of the control modules shown in the diagram:

Module Name	Function
Commutation	This module is used with Brushless DC motors only and splits a single current command into phased current commands according to the selected control method used.
Current loop	This module inputs the commanded phase current along with the actual measured motor phase current and uses a PI filter to generate voltage commands for each motor phase. For Brushless DC motors operating in FOC mode the commutation and current loop module are combined into one functional module.
Motor Output	This module inputs the voltage commands for each motor phase and generates the appropriate PWM (Pulse Width Modulated) signal waveforms to drive the switching amplifier according to the selected amplifier signal output format.

As will be discussed in subsequent sections each of the above modules have control settings associated with them to allow the IC's operation to be optimized for the motor and application being controlled.

7.4 Host Commands

MC7x112 ICs have more than 40 user programmable control parameters for tailoring its control to a specific application. Upon power-up these programmable settings are at their default values.

The serial port may be used to program these control parameters using what are known as host commands. Host commands are data packets that follow a particular format and protocol. Alternatively the MC7x112's internal NVRAM may be used to auto-load the control parameters upon power-up. The format of the stored NVRAM data closely resembles the serial host command packets.

Most host commands specify a single parameter but some specify two or even three. Parameters may be signed or unsigned integers, may be bit encoded, or may be a fixed code specifying one of a list of available values. Throughout this manual we will show the NVRAM script file mnemonics associated with host commands in table form at the end of each chapter.

Below is an example of such a command mnemonic showing the variable name, associated mnemonic code, and range of settable values:

Parameter	Host Command Mnemonic	Range & Description
PWM frequency	SetDrivePWM	Two parameter command, the first must be 3 and the second is a fixed code value which sets the PWM frequency to 20 kHz, 40 kHz, 80 kHz, or 120 kHz.

For a complete description of each command supported by MC7x112s refer to the *Juno Velocity & Torque Control IC Programming Reference*.

For a complete list of MC7x112 IC default settings refer to Appendix A, "MC7x112 Default Values."

For more information on how host commands and the serial interface see Chapter 15, "Host Communication."

For more information on MC7x112 startup and NVRAM programming see Chapter 14, "Power-up, Configuration Storage & NVRAM."

7.4.1 Host Command Script Files

Figure 7-4: Sample Pro-Motion Script File

```
#ScriptVersion 1
:DESC "Motor 2 settings"
:CVER 1.3
SetDrivePWM 1 561
SetDrivePWM 2 0x80ff
SetDrivePWM 4 8
SetDrivePWM 5 2013
SetDrivePWM 6 2013
SetOutputMode 7
SetMotorCommand 0
SetSignalSense 0x0001
SetPhaseParameter 0 0
SetCurrentControlMode 1
SetFOC 512 680
ETC...
```

MC7x112 ICs store NVRAM host commands in their native 'machine' packet format consisting of a series of hexadecimal numbers. Pro-Motion however can use an ASCII text file format to store parameter settings. This file is known as a script file and an example is shown in Figure 7-4. Script files are convenient because they are human readable and editable.

In a typical motion application development sequence control setting optimization would be conducted via PMD's Windows-based Pro-Motion application development software program. When complete, if a script containing these settings is desired, a Pro-Motion menu function is called to automatically generate the script. This script file can then be used to program the MC7x112's NVRAM via a different Pro-Motion menu function, or can be edited by hand if desired using an application such as Windows Notepad.

For detailed instructions on using Pro-Motion and generating script files refer to the developer kit user manual for the specific MC7x112 IC being used.

Most users will not directly edit script files and will instead rely on Pro-Motion to create a script file. Pro-Motion has features for exporting and importing the IC configuration settings to a script file, or loading or uploading data to/from the MC7x1112 IC's NVRAM.

8. Current Loop

8

In This Chapter

- Selecting the Current Control Mode
- Selecting the Command Source
- Current Limit
- PI Filter
- Motor Current Feedback
- AnalogCmd Signal Processing
- SPI Signal Interfacing
- Watchdog Timer
- Enabling and Disabling the Current Loop Module
- Related Host Commands

Figure 8-1, Figure 8-2 and Figure 8-3 show the control flow of the MC7x112's current loop controller. Figure 8-1 shows FOC (Field Oriented Control) used with Brushless DC motors which combines the process of current control and commutation, Figure 8-2 shows an alternate control scheme called A/B control used with Brushless DC motors, and Figure 8-3 shows single phase control used with DC Brush motors and with Brushless DC motors operated in third leg floating mode.

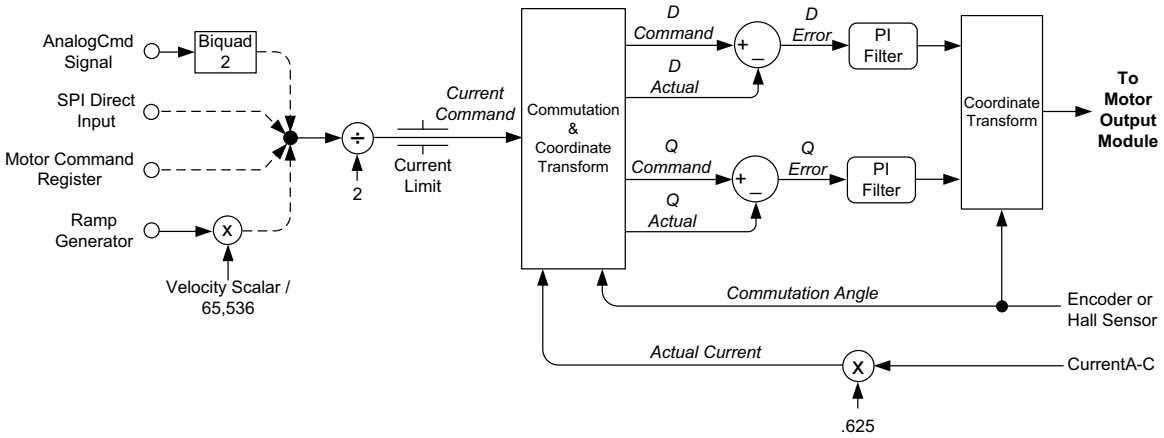
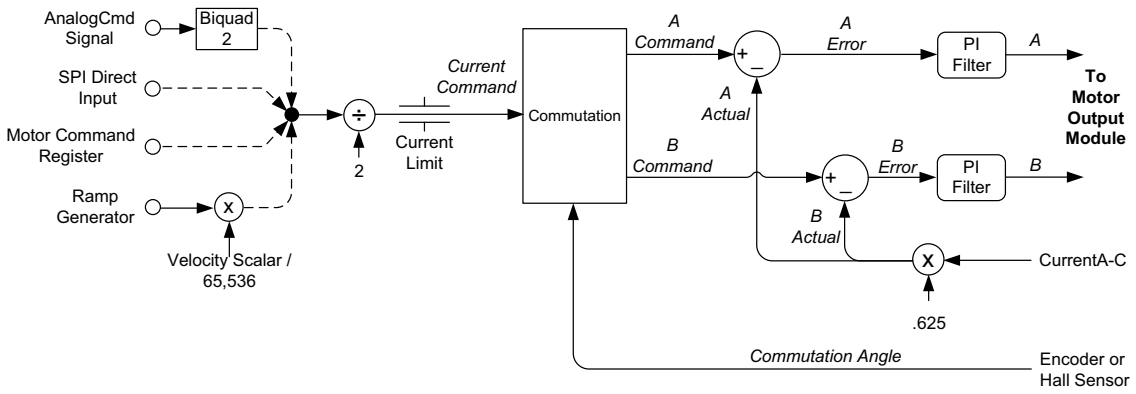


Figure 8-1: FOC Current Loop & Commutation Control Flow

Current control is a technique for controlling the current (and therefore the torque) through each winding of the motor. By controlling the current, response times improve, motor efficiency is higher, and motion smoothness increases. The MC7x112 current loops provide a high level of performance utilizing direct analog signal input of the actual measured current for each winding of the motor.

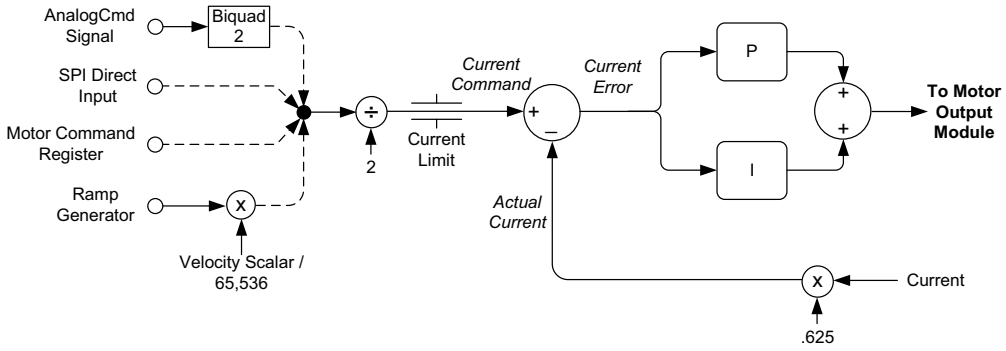
For Brushless DC motors using FOC control the current loop operates in a rotating coordinate frame known as the D and Q axes. The commanded and actual D and Q currents are subtracted and an error is generated, which in turn is passed through a PI (proportional, integral) filter. The output of the D and Q PI filters are converted back into the 3-phase non-rotating reference frame and output as motor commands for each phase of the motor.

Figure 8-2: A/B Current Loop Control Flow



A/B current control shown in Figure 8-2 operates directly on the commanded current for each phase and the measured current for each phase, but otherwise similarly subtracts the actual current from the measured current to generate a current error which in turn is passed through a PI (proportional, integral) filter. Finally, single phase control shown in Figure 8-3 also operates directly on the commanded and measured current but only for a single phase.

Figure 8-3: Single-Phase Current Loop Control Flow



MC7x112 ICs use a discrete time sampling method to perform current control. This current loop update rate is fixed at 20 kHz and cannot be changed. Command source inputs, which are discussed in more detail in Section 8.2, "Selecting the Command Source" are sampled at a rate of 10 kHz but do not have to be synchronized to the current loop update rate. This gives users the freedom to update the command rapidly, which is common when the torque command comes from a microcontroller executing a position servo loop, or to update at a lower rate in applications where the command torque only needs to change from time to time.

8.1 Selecting the Current Control Mode

For DC Brush motors the current control method is always single phase and does not need to be set by the user. Brushless DC motors may use either FOC control, A/B control, or third leg floating. The large majority of applications will use FOC. FOC usually provides the highest top speeds and more energy efficient operation of the motor. FOC may be used when the position feedback is an encoder or when it is a Hall sensor.

For Brushless DC motors third leg floating can sometimes provide a higher top speed than FOC. In this mode only a single phase of current control is performed. Third leg floating control mode is also generally used when there is no encoder and only Hall sensors are available for position feedback. See Chapter 12, *Brushless DC Motor Control*, for more information on third-leg floating control.

The table below summarizes the available current control mode settings for driving DC Brush and three-phase Brushless DC motors with various feedback peripherals present:

Motor Type	Single Phase	A/B	FOC	Third Leg Floating
DC Brush	Yes* *	No	No	No
Brushless DC, Halls & encoder	No	Yes	Yes *	Yes
Brushless DC, encoder only	No	Yes	Yes *	No
Brushless DC, Halls only	No	Yes	Yes	Yes*

* *Recommended current control method*
* * *DC Brush motors always use single phase control*

The recommended current control method is generally best for most applications, but the alternate control schemes may also be considered. In particular third-leg floating can sometimes deliver higher top spin rates than FOC and may be considered even if an encoder is present.

8.2 Selecting the Command Source

The current control module allows one of four command sources to be selected. For direct signal input either the SPI port or the *AnalogCmd* signal can be used to specify a torque command from external circuitry. For command via serial port the ramp generator can be selected as the current loop command or a signed direct 16-bit motor command register can be selected.

For information on how to select one of these four command sources, see Section 8.11, "Related Host Commands."

8.2.1 AnalogCmd Signal Command Source

When the *AnalogCmd* signal is selected as the torque command source the analog voltage presented at the pin is converted by an A/D into a signed 16-bit value such that a voltage of 0.0V represents the most negative torque command having a value of -16,383 and a voltage of +3.3V represents the most positive torque command having a value of +16,383.

Note that relative to the other command sources the *AnalogCmd* command values are one half in value. To achieve a fully scaled value, if desired, the biquad filter can be used to double the output of the *AnalogCmd* signal. To achieve this the B0 gain should be set to a value of 2.0 (2,147,483,647) and all the other gains (B1, B2, A1, A2) should be set to zero.

Numerical current command values do not have units of amps. The current that is commanded, in amps, is determined by the switching amplifier's leg current sense resistors used and the gain of the phase A-C current signal input processing circuit. See Section 8.5, "Motor Current Feedback" for more information and detailed examples.

8.2.1.1 Biquad Filtering

The *AnalogCmd* signal, should it be selected as the command source, includes a biquad filtering capability labeled Biquad 2. While not necessary for many applications, biquad filtering may be useful for reducing noise or correcting for specific dynamic system responses, or doubling the scale of the *AnalogCmd* output. By default the Biquad 2 filter is not active.

The exact scaling and control flow for the MC7x112 biquad filter is shown in Figure 8-4 below.

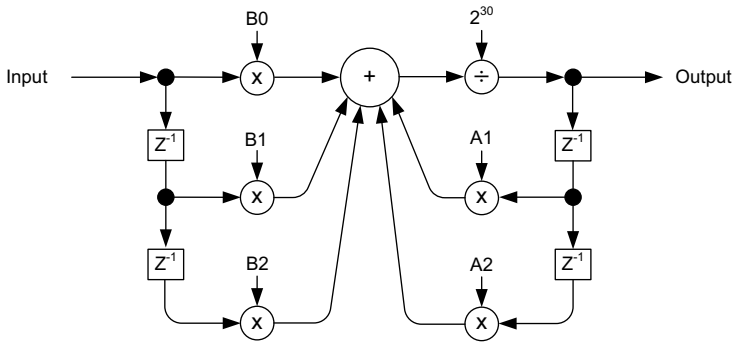


Figure 8-4: Biquad Calculation Flow

8.2.1.2 Related Host Commands

There are a number of parameters which are used to activate and set the parameters for the Biquad 2 filter. In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences:

Parameter	Host Command	Description
Biquad 2 enable	SetLoop GetLoop	Two parameter command, the first must be a value of 16 and the second is a 1 to enable, or 0 to disable Biquad 2.
Biquad 2 coefficient A1	SetLoop GetLoop	Two parameter command, the first must be a value of 20 and the second is a signed 32 bit value with a range of -2,147,483,648 to +2,147,483,647.
Biquad 2 coefficient A2	SetLoop GetLoop	Same format & scale as A1 setting except first parameter is 21.
Biquad 2 coefficient B0	SetLoop GetLoop	Same format & scale as A1 setting except first parameter is 17.
Biquad 2 coefficient B1	SetLoop GetLoop	Same format & scale as A1 setting except first parameter is 18.
Biquad 2 coefficient B2	SetLoop GetLoop	Same format & scale as A1 setting except first parameter is 19.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

For more information on setting the biquad filter to achieve specific filtering functions please refer to Section A.3, "Setting Biquad Filter Values."

Before changing biquad coefficient values the biquad should be disabled. Once the entire set of new bi-quad coefficients have been loaded the biquad can then be safely re-enabled.

8.2.2 SPI Direct Command Source

When SPI Direct Input is selected as the torque command source the SPI bus transmits the torque command as a signed 16-bit number, with -32,767 representing the most negative torque command and a value of +32,767 representing the most positive torque command. See Section 8.8, "SPI Signal Interfacing" for detailed information on SPI signal interfacing and transmitted data word format.

8.2.3 Motor Command Register Command Source

When the motor command register is selected the host serial port is used to specify a signed 16-bit torque command value, with -32,767 representing the most negative torque command and +32,767 representing the most positive torque command.

Caution should be applied when using this command source because the specified values are applied immediately without ramping or smoothing. Nevertheless, this choice of command source is sometimes useful for testing or calibration during system setup.

8.2.4 Ramp Generator Command Source

The most popular approach to commanding a desired torque via the host serial port is using the ramp generator. This command source allows the rate of change of the torque command to be controlled, thereby avoiding abrupt changes in torque that might overly stress the electronics or mechanics of the controlled system. Both a rate of increase and a rate of decrease can be specified, and changes in ramp rate and command torque can be made on-the-fly.

For complete details on the MC7x112 IC's ramp generator see Chapter 10, *Ramp Generator*..

8.2.4.1 Velocity Scalar

The output of the rate generator is scaled by a register called Kvel. Kvel is an unsigned 32 bit number with 1/65,536 scaling meaning that a scale factor of 1.0 (unity scaling) is expressed with a value of 65,536.

When used with the current loop Kvel is most commonly left at the default unity value of 65,536.

8.2.4.2 Reading Commanded Current Values

For diagnosing a problem or for calibration it is sometimes useful to read the commanded current command that will be input into the current loop. This is accomplisied with host command **GetActiveMotorCommand,** which returns the commanded current regardless of the command source setting.

If desired, to read the input value from each specific command source the information in the table below can be used:

Command Source	Host Command	Comments
AnalogCmd signal	ReadAnalog	The ReadAnalog command allows the A/D value reading for all analog signal inputs to be queried. See Section 8.11, "Related Host Commands" for details.
SPI direct	GetTraceValue	The GetTraceValue command with a trace variable code of 105 returns the most recent value input on the SPI bus.
Motor command register	GetMotorCommand	
Ramp generator	GetCommandedVelocity	See Section 10.4, "Related Host Commands" for more information.

8.3 Current Limit

A settable limit to the magnitude of the commanded current is provided via a register called Currentlimit. Currentlimit is an unsigned 16 bit number with a range 0 to 16,383. The default value is 16,383 meaning no limiting is applied. The current limit functions by capping the magnitude of the commanded current to the specified value. For example if the specified current limit is 10,000 an incoming command of +12,345 would be set to +10,000 and an incoming command of -12,345 would be set to -10,000.

Limiting the commanded current is a useful safety feature for insuring that the physical or electrical limitations of the actual system are not exceeded.

The current limit only limits the commanded current. Whether or not the motor current is actually limited to this threshold is a function of whether or not the current loop is functioning properly.

8.4 PI Filter

As shown in Figure 8-1, Figure 8-2, and Figure 8-3 the current loop uses a PI filter to control the current loop response. Three parameters are specified; Kp, Ki, and Ilimit. Two of these are gain factors for the PI (proportional, integral) controller, one is a limit for the integral contribution.

Determining correct Kp, Ki, and Ilimit parameters for the current loop controller gains can be done in a number of ways but the easiest is to utilize the auto-tuning facility provided within PMD's Pro-Motion software package.

Depending on the current control mode selected different PI filter variable gain settings are specified and different host commands are used. This is shown in the table below:

Motor Type	Current Control Mode	Variable	Command
DC Brush	Single-phase	Q	SetFOC
Brushless DC	FOC	D, Q	SetFOC
Brushless DC	A/B	A, B	SetCurrentLoop
Brushless DC	3rd-leg floating	Q	SetFOC

The parameter ranges, formats, and interpretations of the PI filter settable parameters are shown in the following table.

Term	Name	Representation & Range
Kp	Proportional Gain	unsigned 16 bits (0 to 32,767)
Ki	Integral Gain	unsigned 16 bits (0 to 32,767)
Ilimit	Ingreation Limit	unsigned 32 bits (0 to 2,147,483,647)

8.5 Motor Current Feedback

8.5.1 Current Feedback Signals

The following table shows the signals that are used to input the motor coil current measurements:

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
CurrentA	14	12	Measured Current for Leg current input A
CurrentB	19	16	Measured Current for Leg current input B
CurrentC	12	10	Measured Current for Leg current input C

Signals representing the instantaneously measured current of each coil leg are input in a voltage range of 0.0 to 3.3V with a voltage of 0.0 representing the largest possible negative measured current, a voltage of 1.65V representing a measured current of 0, and a voltage of 3.3V representing the largest possible positive measured current.

Current sensors consist of sense resistors, as shown in Figure 8-5, or linear Hall sensors. If sense resistors are used ground-referenced operational amplifiers may be used.

MC7x112 ICs sample current inputs at a rate of 20 kHz and these signals should be filtered to minimize noise. A low pass filter with a rolloff of 200 kHz - 1,000 kHz is recommended, with 500 kHz being a typical value for most applications. When operating at 20 kHz PWM frequency in high noise environments a rolloff on the lower end of the frequency range is recommended. When operating with PWM frequencies of 40 kHz, 80 kHz, or 120 kHz operating in low or normal noise environments a rolloff on the higher end of this range is recommended.

8.5.2 Typical Current Signal Processing Circuitry

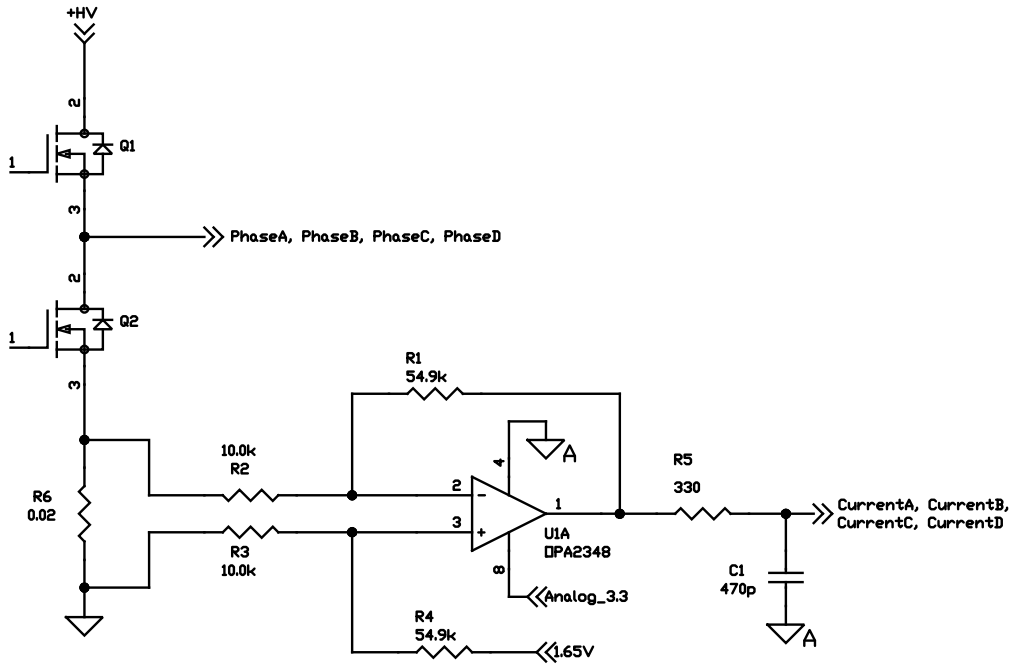


Figure 8-5: Typical Current Signal Processing Circuitry

Figure 8-5 shows a typical leg sensing processing circuit for the current signal inputs. Q1 and Q2 are the half-bridges for one motor phase, and R6 is the current sensing resistor. U1A with R1 ~ R4 is a differential amplifier for signal conditioning; it is capable of measuring bidirectional current and has an output of 1.65V at zero current. R5 and C1 form a low pass filter, and they should be placed close to the pins on the IC. R1 ~ R4 should be 1% or higher grade. The power rating of R6 should match the winding current with 50% power margin recommended.

See Section B.9, "PWM High/Low Motor Drive with Leg Current Sensing/Control" for complete example schematics for various amplifier designs with current control.

Current control only functions when the PWM output mode is set to PWM High/Low. For information on motor output modes, refer to Chapter 9, *Motor Output*.

8.5.3 Current Signal Scaling

The value of the user's external sense resistors and analog conditioning circuitry determine the overall controllable current range of the amplifier. The overall current sense range should be 25% to 50% above the largest expected peak current. The user commandable current range is 80% of the current sense range.

Example

A Brushless DC application will require a peak current in each phase of 7.5 amps. What is an appropriate scaling for the external current sense circuiting and what is the numerical value to command a current of 4.75 amps?

The total current sense is selected as +/- 10.0 amps, which gives a commandable range of +/- 8.0 amps (80% of the total current sense range). A sense resistor and op amp are used to generate +/- 1.65 volts for the desired current range of +/- 10 amps, presenting a voltage of 0.0V for a reading of -10.0A, 1.65V for 0.0A, and 3.3V for +10.0A at the *CurrentA-C* pins.

To determine the numerical value for the user specified current command we multiply by 1/.8 reflecting the 80% scaling within the MC7x112's current loop. So given a desired current in amps, in this example, the command numerical value is value = I * 1.25 * 32,767 / 10.0A and conversely, given a commanded numerical value the equivalent commanded current in amps is I = value * 10.0A * .80 / 32,767. Plugging in the desired command of 4.75 A, the numerical command value is 4.75 A * 1.25 * 32,767/10.0 A = 19,455.

Note that this counts to amps conversion constant can be more simply expressed as amps = .24415 mA / count or counts = 4,096 / amp.

8.5.4 Minimum Current Read Time

When controlling Brushless DC motors with FOC (field oriented control) current control mode MC7x112s require a minimum current read time to be specified. The minimum current read time setting determines the minimum amount of off time for two of the three PWM output channels for three phase output in PWM High/Low output mode. The specified value has units of nSecs.

The minimum current read time setting is determined by the external analog current signal processing circuitry. The recommended setting consists of the electrical time constant of this external current input circuitry multiplied by 2. For example from the design for current signal input filtering provided in Section 8.5.1, "Current Feedback Signals" assuming a low pass filter of 500 kHz is implemented, the associated minimum read time setting would be 2 * 1/500,000 Hz = 4.0 μSec or 4,000 nSecs.

The default value for minimum current read time effectively disables PWM output and therefore must be set by the user with a value appropriate for the current sensor circuitry being used.

8.6 AnalogCmd Signal Processing

MC7x112 torque control ICs support input of an *AnalogCmd* direct input signal to provide a bipolar torque command for the motor or actuator being controlled.

The table below shows the *AnalogCmd* signal.

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
AnalogCmd	18	18	Analog torque command input

The *AnalogCmd* input range is 0.0V to 3.3V with the maximum positive command represented with a value of 3.3V, the maximum negative command having a value 0.0V, and a command value of 0 having a value of 1.65V.

On-board circuitry that directly outputs an analog signal in a compatible format may directly connect to the *AnalogCmd* input pin. External connections via cable require on-board signal conditioning to insure proper scaling and filtering. In both cases the voltage presented at the *AnalogCmd* pin must not exceed 3.30V.

The *AnalogCmd* input is sampled at a rate of 20 kHz and therefore a low pass filter with a roll off at 50 kHz - 200 kHz is recommended. In addition to any hardware filtering of this signal input, a biquad filter is available for more complex filter control at frequencies below 5 kHz should that be desired. Refer to Section 8.2.1.1, "Biquad Filtering" for more information.

8.6.1 Typical AnalogCmd Processing Circuitry

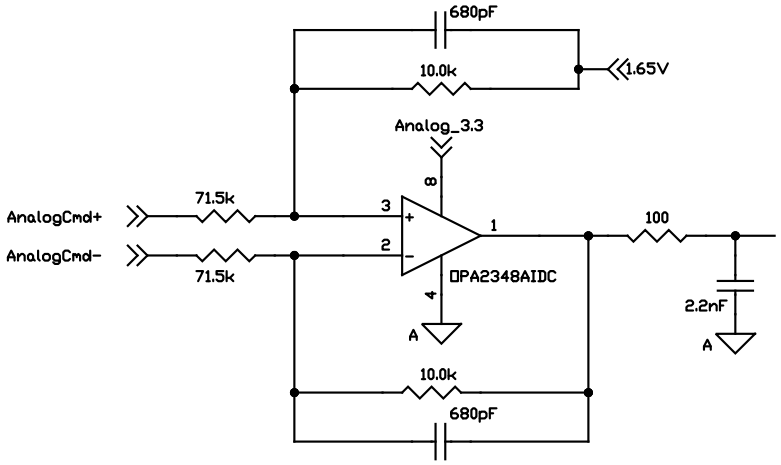


Figure 8-6: Example +/- 10V Input AnalogCmd Circuitry

Figure 8-6 shows a typical circuit for processing a +/-10V differential input command signal. The output has an offset of 1.65V corresponding to a zero command. The bandwidth of this circuit is 23.4 kHz. The 100 ohm resistor and 2.2nF capacitor should be placed close to the *AnalogCmd* input pin. All resistors used should be 1% or higher grade. All capacitors should be X7R or higher grade.

8.6.2 AnalogCmd Scaling

The example below illustrates how the *AnalogCmd* signal generates a specific command in amps for a particular current control setup.

Example

A user's circuit board inputs a +/-10V analog torque command signal to drive the current controller in the example in Section 8.5.3, "Current Signal Scaling." What torque is commanded when a voltage of -7.0 V is applied to the +/-10V input circuitry?

Analog processing circuitry such as shown in Figure 8-6 converts a +/- 10V signal to the full scale of the 0 to 3.3V input range of the *AnalogCmd* input signal. An incoming voltage of -7.0V is received, representing a request for 70% of available maximum negative torque. Using $V_{AnalogCmd} = 1.65V + Vin * 3.3 / 20$ and plugging in Vin = -7.0V, this incoming signal results in a voltage of 0.495V at the *AnalogCmd* pin which results in numerical command value of -22,937.

We convert this numerical value to commanded amps using the scaling determined from the example in Section 8.5.3, "Current Signal Scaling" of .24415 mA/count. This gives I = -22,937 * .24415 mA/count = -5.6A.

8.7 Analog Signal Calibration

After integration into a particular PCB (printed circuit board), for best performance, it is generally recommended that if used, the *CurrentA-C* and *AnalogCmd* signal analog input offsets be calibrated so that their zero value readings are as close to 1.65V as possible.

Whether or not calibration is needed in a particular application depends on the external analog signal processing circuitry used and the extent to which the absolute best motion performance, particularly motor smoothness and quietness, is important. Higher precision external circuitry or use of external offsetting circuitry may obviate the need for internal calibration. Conversely the MC7x112's internal analog offset calibration procedure may allow less expensive circuitry to be used.

For the *Temperature* and *BusVoltage* signals, offsetting is allowed but generally not necessary. The *BusCurrentSupply* signal does not support an offset calibration nor is one needed for proper functioning.

Calibration of MC7x112 analog inputs should occur when the board is powered up, with no analog torque command asserted, with no motor output command asserted, and with the physical motor axis stationary. Two overall calibration methods are provided. The simplest method is to send a **CalibrateAnalog** command. This command will automatically measure and set the offsets so that the Current A-C and *AnalogCmd* inputs are zeroed

out. Because a number of samples are taken and averaged, 100 mSec should be allowed for this command to complete. In addition, the MC7x112 operating mode should be set to motor output only before this command is executed.

The second method is to directly read each analog input via the **ReadAnalog** command and then write the same values for the corresponding analog offsets using the **SetAnalogCalibration** command. When using this manual calibration method it is recommended that a number of analog reads of each signal are averaged together to improve the offset accuracy.

Regardless of how the analog offsets are determined, unless explicitly stored into NVRAM they will not be retained after a reset or power cycle. For more information on NVRAM configuration storage see *Chapter 14, Power-up, Configuration Storage & NVRAM*,.

Calibration of the MC7x112's CurrentA-C and *AnalogCmd* signal inputs, if used, is recommended to achieve the full extent of smooth and quiet motion that MC7x112 IC's are capable of.

8.8 SPI Signal Interfacing

The MC7x112 SPI interface utilizes three digital input pins *SPIEnable*, *SPSIClock* and *SPIRcv*, and one digital output pin *SPIXmt*. These signals 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 table below provides a summary of the SPI related signals:

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
SPIRcv	36	31	SPIRcv inputs synchronous serial data for the SPI bus
SPIXmt	34	30	SPIXmt transmits synchronous serial data for the SPI bus
SPIClock	33	29	SPIClock inputs the clock signal used with synchronous serial transfer on the SPI bus
SPIEnable	44	39	SPIEnable inputs an enable for SPI bus communications

For electrical timing information for SPI bus operation see Section 3.2.2, "SPI."

All of the SPI-related signals are digital TTL level signals. Some applications may directly connect microprocessors, FPGAs, or other on-board logic to these signals. Other applications, depending on signal lengths and noise level may benefit from external drivers to improve signal integrity of the SPI bus.

SPI is a serial hardware bus intended for communications within a PCB substrate only. It is not recommended for board to board communications. It is the responsibility of the user to insure the signal integrity of the SPI bus for the application in which it is used.

The only SPI port operation supported by the MC7x112 torque control ICs is a write by the external circuitry of a 16-bit torque command data word.The write format is shown in Figure 8-7. 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.

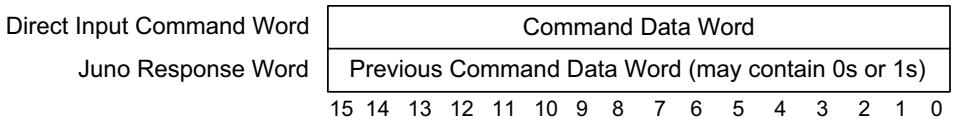


Figure 8-7: Direct Input SPI Format

The word returned by the MC7x112 ICs on the SPIXmt line is the previous command word received. It is recommended, but not required, that the external circuitry read this returned word and confirm that it matches the previously transmitted word.

8.9 Watchdog Timer

MC7x112 ICs provide a facility for detecting when external commands which arrive on a regular basis unexpectedly stop. The ability to detect this condition, known as a watchdog timer, is useful for safely shutting down an axis.

The watchdog feature functions with the ramp generator, the SPI direct input command, and the motor command register command sources. It does not function when *AnalogCmd* is selected as the command source.

In each case the user selects a watchdog countdown time in units of cycles. The default value for the watchdog countdown timer is 0 which indicates no watchdog function is active. If a lack of command activity occurs for more than the watchdog countdown period a watchdog error occurs, resulting in the drive exception flag of the Event Status register being set. For more information see Section 11.1.1, "Event Status Register."

The MC7x112 ICs can be programmed to take various actions when a watchdog timeout occurs such as disabling the motor output. The mechanism to program and process these functions is called event handling and described in detail in Section 11.3, "Event Action Processing."

8.10 Enabling and Disabling the Current Loop Module

The current loop module may be explicitly enabled or disabled by the user. If disabled, the value from the selected command source will pass directly to the motor output stage with no current control being performed. The most common use of MC7x112 operation with the current loop not functioning is to drive the motor in voltage mode, which may be useful in some motor control applications, or may be useful for calibration or testing of the amplifier.

Whether the current loop module is enabled or disabled is controlled by the Operating Mode register. See Section 8.10.2, "Operating Mode Register" for more information. To enable the current loop a host command can be sent via the serial port, or a similar command can be stored as part of the MC7x112's NVRAM initialization sequence. If the current loop was disabled as part of an automatic safety event-related action operation can be restored either by serial port host command or via external signal manipulation. See Section 11.3, "Event Action Processing" for details.

The default condition of the current loop module is disabled, therefore it must be enabled for the MC7x112 to provide motor control with current control.

8.10.1 Voltage Mode Operation

The MC7x112 ICs may be operated with the current loop module disabled, in which case the operating mode is referred to as voltage mode. In voltage mode the command sources shown in Figure 8-1 and Figure 8-2 are still available, however downstream current loop elements are bypassed. Refer to Chapter 9, *Motor Output*, for details.

Operation without a current loop warrants special precautions. Depending on the motor being controlled, the degree to which the motor is moving, and the voltage being commanded, excessive current flow may occur resulting in a DC supply undervoltage condition, damage to the motor, or damage to the amplifier.

8.10.2 Operating Mode Register

Some of the major control modules provided by the MC7x112 ICs can be turned off or turned on by the user. The register used for this purpose is called the Operating Mode register. This register also displays status information connected with operating mode changes that occur as a result of event actions. For more information on event actions see Section 11.3, "Event Action Processing."

The table below shows the available settings or status indicators in the Operating Mode register.

Function	Default Value	Description
Motor Output Module	0	Setting of 1 is module enabled, setting of 0 is disabled.
Current Loop Module	0	Setting of 1 is module enabled, setting of 0 is disabled.
Command Source	0	Setting of 1 means command source is either the *AnalogCmd* signal, the SPI port, or the ramp generator. Setting of 0 means the command source is the motor command register.
Passive Braking	0	Status indicator only, can not be directly set by user. Value of 1 means passive braking is active, value of 0 means it is not.

8.11 Related Host Commands

Host commands may be sent via the serial port, or may be loaded into the MC7x112's NVRAM for automatic execution at startup. Refer to Chapter 14, "Power-up, Configuration Storage & NVRAM" for more information. In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

Parameter	Host Command Mnemonic	Range & Description
Current control mode	SetCurrentControlMode GetCurrentControlMode	Specified value is a fixed code selecting FOC, A/B, or third leg floating control mode.
Operating mode register	SetOperatingMode GetOperatingMode	Specified value is a bit-oriented mask determining the state of the command source, current loop, and motor output enable/disable flags.
Current operating mode value	GetActiveOperatingMode	This command returns the instantaneous actual state of the operating mode register. By comparison the GetOperatingMode command returns the value set via the SetOperatingMode command. These two may differ as a result of event actions.
Select command Source	SetDriveCommandMode GetDriveCommandMode	Specified value is a fixed code selecting either AnalogCmd, SPI, or the ramp generator as the command source. For these command sources to be active the Command Source bit of the operating mode register must be set to 1.

Parameter	Host Command Mnemonic	Range & Description
D or Q Kp	SetFOC GetFOC	Three parameter command, the first specifies whether the provided gain applies to the D loop, the Q loop, or both. The second must be 0, and the third has a range of 0 to 32,767 and specifies the proportional gain value.
D or Q Ki	SetFOC GetFOC	Three parameter command, the first specifies whether the provided gain applies to the D loop, the Q loop, or both. The second must be 1, and the third has a range of 0 to 32,767 and specifies the integral gain value.
D or Q Ilimit	SetFOC GetFOC	Three parameter command, the first specifies whether the provided gain applies to the D loop, the Q loop, or both. The second must be 2, and the third has a range of 0 to 32,767 and specifies the integrator sum limit.
Phase A or B Kp	SetCurrentLoop GetCurrentLoop	Three parameter command, the first specifies A, B, or both. The second must be 0 and the third has a range of 0 to 32,767 and specifies the proportional gain value.
Phase A or B Ki	SetCurrentLoop GetCurrentLoop	Three parameter command, the first specifies A, B, or both. The second must be 1 and the third has a range of 0 to 32,767 and specifies the integral gain value.
Phase A or B Ilimit	SetCurrentLoop GetCurrentLoop	Three parameter command, the first specifies A, B, or both. The second must be 2 and the third has a range of 0 to 32,767 and specifies the integrator sum limit.
Current limit	SetCurrentLimit GetCurrentLimit	Specified value has a range of 0 to 16,383 and specifies the maximum allowed current command.
Motor command register	SetMotorCommand GetMotorCommand	Specified value has a range of -32,767 to 32,767 and specifies a command current value. For the motor command register to operate as the command source the Command Source bit of the operating mode register must be set to 0.
Kvel	SetLoop GetLoop	Two parameter command, the first must be 64. The second has a range of 0 to 2,147,483,647 and specifies the velocity scalar.
Minimum current read time	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 6. The second has a range of 0 to 32,767 and specifies the minimum current read time in units of nSecs.
Watchdog countdown time	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command, the first must be 3. The second has a range of 0 to 32,767 and has units of cycles.
AnalogCmd gain	SetAnalogCalibration GetAnalogCalibration	Two parameter command, the first must be 519, the second specifies the AnalogCmd gain and has a range of 0 to 32,767.
Calibrate analog signals	CalibrateAnalog	Specified value is a fixed code selecting either all leg current signals or the AnalogCmd signal for calibration.
Set analog signal calibration offset values	SetAnalogCalibration GetAnalogCalibration	Two parameter command, the first is a fixed code selecting either the CurrentA, CurrentB, CurrentC, CurrentD, or AnalogCmd signal offset. The second has a range of -32,767 to 32, 767 and specifies the offset value.

Parameter	Host Command Mnemonic	Range & Description
Read analog signals	ReadAnalog	To read an analog pin voltage a channel # is specified. The returned value is scaled from 0 to 65,536 with 0 representing 0.0V, and 65,536 representing 3.3V. Here are the channel #s for each readable pin: 0 – CurrentA 1 – CurrentB 2 – CurrentC 4 – Temperature 5 – BusCurrentSupply 6 – BusVoltage 7 – AnalogCmd

For more information on MC7x112 host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

9. Motor Output

In This Chapter

- Selecting the Command Source
- PWM High/Low Motor Output Mode
- Sign/Magnitude PWM Output Mode
- AmplifierEnable
- Brake

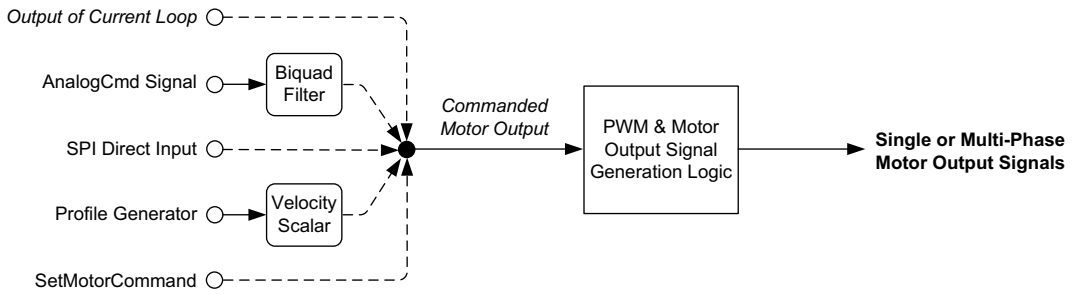


Figure 9-1: Motor Output Control Flow

The purpose of the motor output module is to generate PWM (Pulse Width Modulation) and related motor output signals for use by external switching amplifier circuitry.

MC7x112 ICs provide two different PWM motor output methods, PWM High/Low, and Sign/Magnitude PWM. The PWM output method that is used when the MC7x112's current control facility is utilized is PWM High/Low mode.

The table below shows the available PWM generation modes for each motor type.

PWM Mode	DC Brush	Three-phase Brushless DC	Used with Current Control ?
PWM High/Low	Yes	Yes	Yes
Sign/Magnitude PWM	Yes	No	No

To minimize the chance of unexpected motor movement during startup, by default the PWM output method is "none", and motor output module is disabled. To enable motor output first the PWM output method must be set, then the motor output flag of the Operating Mode register must be enabled.

To activate motor output first the PWM output method must be set, then the motor output flag of the Operating Mode register must be enabled.

9.1 Selecting the Command Source

The motor output module allows several command sources to be selected. Most common by far is that the 'upstream' current control loop is enabled in which case the current loop module provides the motor output command, and a motor output module command source does not need to be selected.

If the current loop module is not enabled the MC7x112 operates in voltage mode (no current loop active) and one of four command sources can be selected; the *AnalogCmd* signal, the SPI Direct Input, the motor command register,

and the ramp generator. The function of these input sources and how to select them as the command source is the same as for the current loop module, and is described in Section 8.2, "Selecting the Command Source."

If the current loop is enabled then the command source for the Motor Output module is the command output of the current loop module.

9.1.1 Motor Output Module Command Scaling

The command input to the motor output module is a signed number with a value from -32,768 to +32,767, with -32,768 representing a commanded negative voltage of 100% of the DC supply voltage, and 32,767 representing a positive voltage of 100% of the DC supply voltage.

Example: A system with a DC supply voltage of 48V commands the motor output module with a value of +4,489 (decimal). What is the resultant voltage output by the switching amplifier?

A command value of 4,489 is 4,489/32,767 or 13.7% of full scale, resulting in a voltage output of +0.137 x 48V = +6.58V. Note that depending on the PWM output mode selected and the associated switch signal generation control settings, the actual voltage may vary slightly from this value.

9.2 PWM High/Low Motor Output Mode

The MC7x112 ICs can control high-efficiency MOSFET or IGBT power stages with individual high/low switch input control. DC Brush motors are driven in an H-Bridge configuration consisting of 4 switches, while Brushless DC motors are driven in a triple half-bridge configuration consisting of 6 switches.

In PWM High/Low mode each signal carries a variable duty cycle PWM signal. A zero desired motor command results in the high side and low side being active for the same amount of time. Positive motor commands are encoded as a high-side duty cycle greater than 50%, and negative motor commands are encoded as a duty cycle less than 50%. This is shown in Figure 9-2.

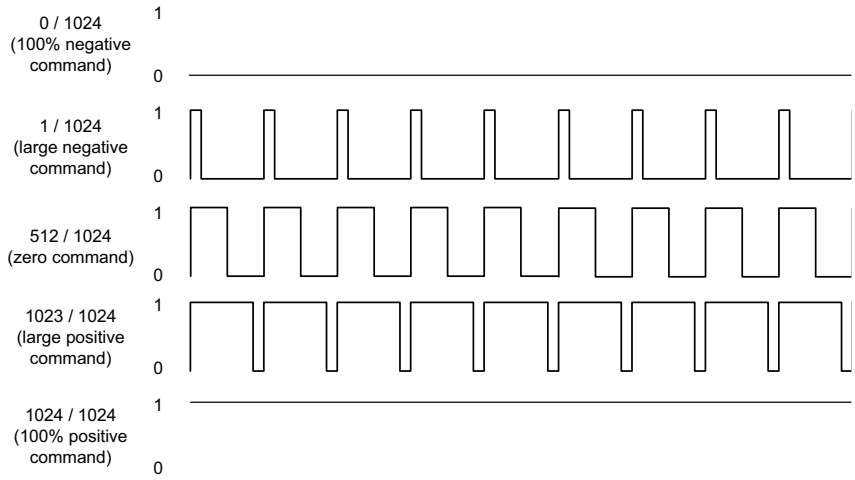


Figure 9-2: PWM High/Low Encoding

In PWM High/Low mode two output pins are used per motor or per motor phase, allowing separate high-side/low-side control of each bridge switch. In this scheme, as Figure 9-3 shows, the high side output and the low side output are never active at the same time, and there is generally a period of time when neither output is active. This period of time is called the dead time, and provides a shoot through protection function for MOSFET or IGBT switches.

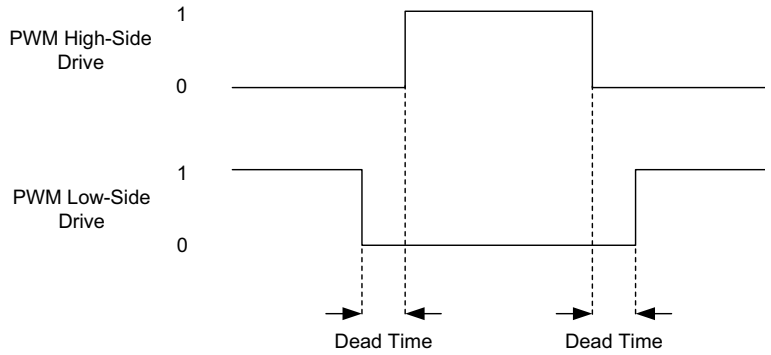


Figure 9-3: PWM High/Low Signal Generation

The dead time is specified in nSecs. The correct value can generally be determined from the MOSFET or IGBT IC manufacturer's data sheet, or you can call PMD technical support if you have questions. If only high side bridge control switches are used then the dead time can be set to zero, creating what is known as a PWM 50/50 scheme. This scheme may sometimes be useful to connect with integrated bridge circuitry which requires just a single level shifted control signal per phase.

In addition to dead time, some high side switch drive circuitry requires a minimum amount of off time to allow the charge pump circuitry to refresh. This parameter is known as the refresh time and has units of nSecs. The related parameter of refresh time period, which is the time interval between these off time refreshes has units of current loop cycles.

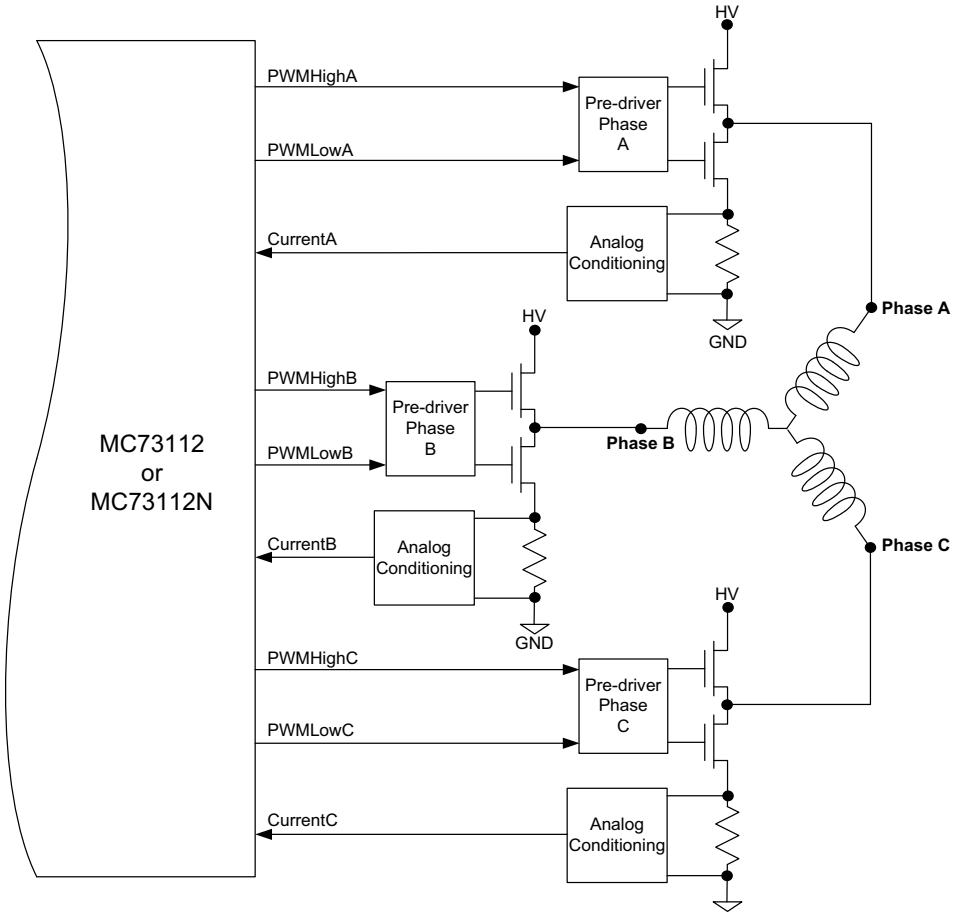
It is also possible to control the maximum allowed PWM duty cycle. This may be useful to limit the effective voltage presented to the motor windings, or to provide some other needed off-time for the switching amplifier circuitry.

The MC7x112 torque control ICs default values for dead time, refresh time, and refresh period are set to be maximally safe but are not appropriate to drive typical switching hardware. For proper amplifier function these values must be set with values appropriate for the connected switching circuitry.

9.2.1 PWM High/Low Brushless DC Motor Drive

Figure 9-4 shows the typical amplifier stage arrangement when a MC73112 IC or MC73112N IC is used in PWM High/Low mode with current control.

Figure 9-4: Brushless DC Motor Bridge Configuration with Current Control



As shown in the table below six PWM output signals and, if used, three analog feedback signals for current control interface between the MC73112 IC and the switching amplifier circuitry.

Signal	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Description
PWMHighA	56	49	Digital high side drive output for motor phase A
PWMLowA	55	48	Digital low side drive output for motor phase A
PWMHighB	54	47	Digital high side drive output for motor phase B
PWMLowB	53	46	Digital low side drive output for motor phase B
PWMHighC	51	45	Digital high side drive output for motor phase C
PWMLowC	50	44	Digital low side drive output for motor phase C
CurrentA	14	12	Analog input containing the current flow through the low side of the switching bridge for phase A
CurrentB	19	16	Analog input containing the current flow through the low side of the switching bridge for phase B
CurrentC	12	10	Analog input containing the current flow through the low side of the switching bridge for phase C

9.2.1.1 Multi-Phase PWM Signal Generation

In addition to differing in their approach to overall current control, A/B current control, FOC (Field Oriented Control), and third-leg floating control also differ in how the PWM bridge control signals are generated when driving BLDC motors.

The table below summarizes this:

Control Method	Bridge Control Method	Comment
A/B	Sine modulated PWM	
FOC	Space vector PWM	
Third-leg floating*	Standard PWM	Standard PWM is also used for single-phase devices such as DC Brush motor.

** Third-leg floating, when the current loop is enabled, uses only a single current loop.*

The above bridge control schemes apply whether current control is enabled or not. For example If FOC is the selected control scheme with the current loop disabled, the PWM bridge control scheme is still space vector PWM.

The detailed differences between sine-modulated PWM and space vector PWM will not be described in this manual, but the main practical difference is in the maximum effective voltage that can be achieved. With space vector PWM this is approximately 15% higher than for sine-modulated PWM.

9.2.2 PWM High/Low DC Brush Motor Drive

Figure 9-5 shows the typical amplifier stage arrangement when a MC71112 IC or MC71112N IC is used in PWM High/Low mode.

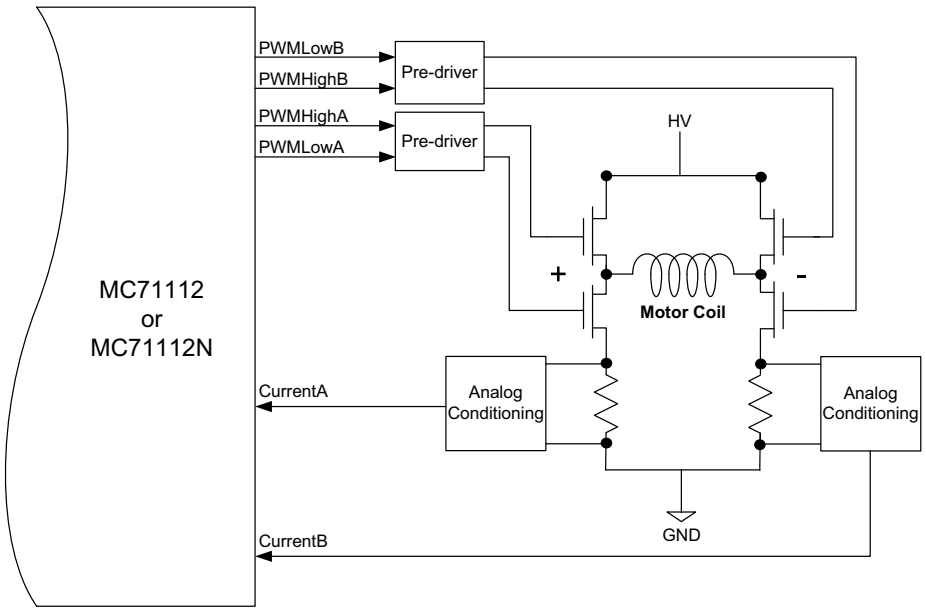


Figure 9-5: DC Brush Motor Bridge Configuration

As shown in the table four PWM output signals and, if used, two analog feedback signals for current control interface between the MC71112 IC and the switching amplifier circuitry.

Signal	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Description
PWMHighA	56	49	Digital high side drive output for the positive coil terminal
PWMLowA	55	48	Digital low side drive output for the positive coil terminal
PWMHighB	54	47	Digital high side drive output for the negative coil terminal
PWMLowB	53	46	Digital low side drive output for the negative coil terminal

Signal	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Description
CurrentA	14	12	Analog input containing the current flow through the positive leg of the bridge
CurrentB	19	16	Analog input containing the current flow through the negative leg of the bridge

9.2.3 Low Pass PWM Signal Filtering

Some integrated amplifier ICs expect an analog command input. This can be accomplished by low pass filtering the PWM output signal thereby generating an analog signal. Depending on the input voltage required, additional analog processing circuitry may be needed. Note also that depending on the amplifier command format expected, low pass filtering of the Sign/Magnitude PWM signal may be preferred versus the PWM high/low format signal.

As was the case for interconnection to single-input bridges, when analog input integrated bridges are used, current control, if desired, must be provided by the external amplifier.

9.2.4 Related Host Commands

There are a number of parameters which are used to set up or control the switching amplifier circuitry when the motor output mode is set to PWM High/Low. In the table below both Set and Get commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

Parameter	Host Command Mnemonic	Range & Description
Operating mode register	SetOperatingMode GetOperatingMode	Specified value is a bit-oriented mask determining the state of the command source, current loop, and motor output enable/disable flags.
Current operating mode value	GetActiveOperatingMode	This command returns the instantaneous actual state of the operating mode register. By comparison, the GetOperatingMode command returns the value set via the SetOperatingMode command. These two may differ as a result of event actions.
PWM switching frequency	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 3 and the second is a fixed code value setting the PWM frequency to either 20, 40, 80, or 120 kHz.
PWM dead time	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 1 and the second has a range of 0 to 16,383 and determines the dead time in nSecs.
PWM refresh time	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 5 and the second has a range of 0 to 32,767 and determines the refresh time in nSecs.
PWM refresh period	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 4 and the second has a range of 1 to 32,767 and determines the refresh period in current loop cycles which are 51.2 μSec in duration.
PWM limit	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 0 and the second has a range of 0 to 16,384 and determine the maximum allowed PWM duty cycle in units of 16,384/163.84 percent.
PWM high/low signal sense	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 2 and the second is a 16-bit mask specifying the PWM signal output sense.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

9.3 Sign/Magnitude PWM Output Mode

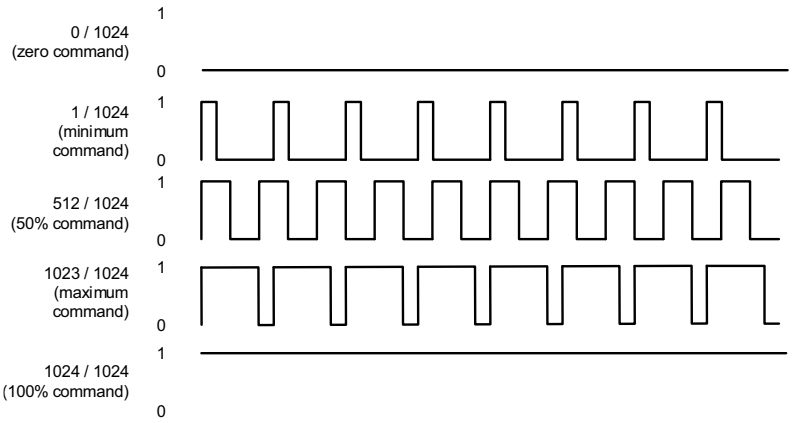


Figure 9-6: Sign/ Magnitude PWM Encoding

In Sign/Magnitude PWM mode two pins are used to output the motor command information. One pin carries the PWM magnitude, which ranges from 0 to 100% as shown in Figure 9-6. A high signal on this pin means the motor coil should be driven with voltage. A second pin outputs the sign of the motor command by going high for positive sign, and low for negative.

PWM sign/magnitude output is only used with DC Brush motors. PWM sign/magnitude is not used with Brushless DC motors.

9.3.1 Sign/Magnitude PWM DC Brush

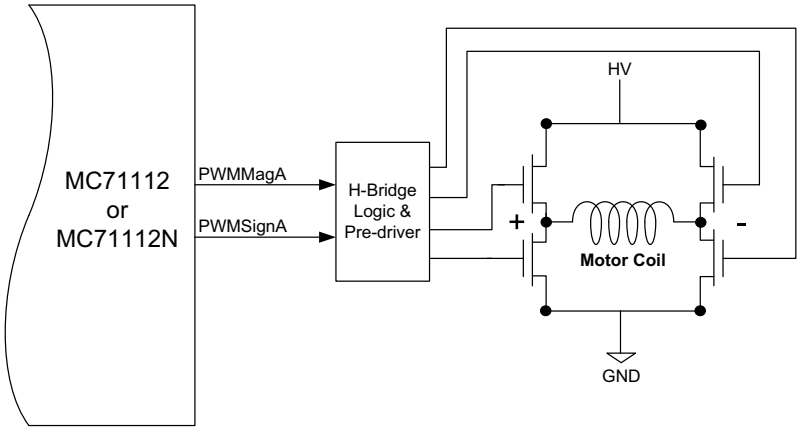


Figure 9-7: DC Brush PWM Sign/ Magnitude Bridge Configuration

Figure 9-7 shows a typical connection for a DC Brush motor when Sign/Magnitude PWM motor output mode is used. Note that in this mode the MC7x112 IC does not provide current control, and therefore if current control is desired this capability must be provided by the amplifier bridge's circuitry.

9.3.2 Related Host Commands

There are a number of parameters which are used when the motor output mode is set to sign/magnitude PWM. In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

Parameter	Host Command Mnemonic	Range & Description
Operating mode register	SetOperatingMode GetOperatingMode	Specified value is a bit-oriented mask determining the state of the command source, current loop, and motor output enable/disable flags.

Parameter	Host Command Mnemonic	Range & Description
Current operating mode value	GetActiveOperatingMode	This command returns the instantaneous actual state of the operating mode register. By comparison the GetOperatingMode command returns the value set via the SetOperatingMode command. These two may differ as a result of event actions.
PWM switching frequency	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 3 and the second is a fixed code value sets the PWM frequency to either 20, 40, 80, or 120 kHz.
PWM refresh time	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 5 and the second has a range of 0 to 32,767 and determines the refresh time in nSecs.
PWM refresh period	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 4 and the second has a range of 1 to 32,767 and determines the refresh period in current loop cycles which are 51.2 µSec in duration. When PWM sign/magnitude mode is used the refresh period must be set to zero.
PWM limit	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 0 and the second has a range of 0 to 16,384 and determine the maximum allowed PWM duty cycle in units of 16,384/163.84 percent.
PWM signal sense	SetDrivePWM GetDrivePWM	Two parameter command, the first must be 2 and the second is a 16-bit mask specifying the PWM signal output sense. Only the PWM Magnitude signal sense can be changed. The PWM Sign signal sense is not changeable.

9.3.3 Signal Processing

As shown in the table below one PWM magnitude and one PWM sign signal are output to interface between the MC71112 IC and the amplifier circuitry:

Signal	64-Pin TQFP Pin #	56-Pin VQFN Pin #	Description
PWMMagA	56	49	Digital PWM magnitude output for the H-bridge switching amplifier.
PWMSignA	55	48	Digital sign output for the H-bridge switching amplifier.

For examples of PWM signal magnitude amplifiers above see Chapter B, *Application Notes*.

9.4 AmplifierEnable

The MC7x112 ICs provide an *AmplifierEnable* signal output that indicates whether the external amplifier circuitry should be active or not. While not all external amplifiers will require or provide such an input control, this signal is useful for general safety purposes, as well as to simplify the task of ensuring startup without jogging the motor after power-up.

The output of this signal is affected by whether the motor output module is enabled and whether the brake function is active. By default the motor output module is disabled and must be enabled by the user via serial host command or NVRAM initialization command. During operations the motor output may become disabled by safety related event processing. For more information on event processing refer to Section 11.3, "Event Action Processing."

If either the motor output module is enabled or if the brake function is active than the *AmplifierEnable* signal is active. If both the motor output module is disabled and the brake function is inactive than the *AmplifierEnable* signal is inactive.

9.4.1 Signal Processing

The table below shows the AmplifierEnable signal pin:

Signal	64-p[in TQFP Pin #	56-Pin VQFN Pin #	Description
AmplifierEnable	3	22	AmplifierEnable output signal used with some amplifiers. A high signal indicates the amplifier is enabled and a low indicates amplifier is disabled. If this signal is used a 4.7 Kohm external pull down resistor is recommended to prevent glitches during reset.

9.5 Brake

The MC7x112 IC's *Brake* signal input provides a high speed PWM output disable that may be useful for safety protection when the motor output mode is set to PWM High/Low. When this input is active PWM output is driven to one of two user programmable states; disable motor output or passive braking.

In the motor output disabled state all amplifier bridge switches for the selected motor type are open, meaning that all high and low control signals are driven inactive and the *AmplifierEnable* signal is inactive. The result is that the motor will "free-wheel" to a stop. In the passive braking state all of the high-side switch control signals are driven inactive, all of the low-side switch signals are active, and the *AmplifierEnable* signal is active, thereby closing the lower side switches. The result is that current will circulate through the motor winding resulting in back-EMF generation and a more rapid deceleration of the motor.

If a brake signal event occurs this will be reflected in the status reported by the active Operating Mode register either by showing motor output disabled, or passive braking enabled. Note in addition to brake signal events there are other event actions that may result in the PWM output being driven to the passive braking state or the motor output disabled state.

To re-enable normal motor operation after a brake signal event, the brake signal must be de-asserted, the drive exception bit of the Event Status register must be cleared, and the MC7x112's operating mode must be restored. If the brake signal is de-asserted without also clearing the Event Status register bit the MC7x112 will behave as if the brake signal is still asserted. It is recommended, but not required, that in addition the Drive Fault Status register be queried and the Brake bit of the Drive Fault Status register be cleared. For more information on these status registers refer to Section 11.1.1, "Event Status Register" and Section 11.1.4, "Drive Fault Status Register."

For more information on event processing see Section 11.3, "Event Action Processing."

The passive braking function is only available with the PWM control mode set to PWM High/Low. When the output mode is Sign/Magnitude PWM the Brake signal can only execute a motor output disable function.

9.5.1 Signal Processing

The table below shows the Brake signal pin:

Signal	64-p[in TQFP Pin #	56-Pin VQFN Pin #	Description
Brake	37	32	Brake input signal.

This page intentionally left blank.

10. Ramp Generator

10

In This Chapter

- Ramp Generator Cycle Time
- Specifying Ramp Parameters
- Current Ramp Profile Stop Events
- Related Host Commands

MC7x112 ICs include an internal ramp generator. The ramp generator is used in conjunction with serial host commands to specify torque (current) commands and rates of change (ramp rate) of torque commands.

To control the ramp generator the user specifies a desired target current, a current increase rate, and a current decrease rate. Using these parameters the ramp generator performs calculations to determine the commanded current at any given moment.

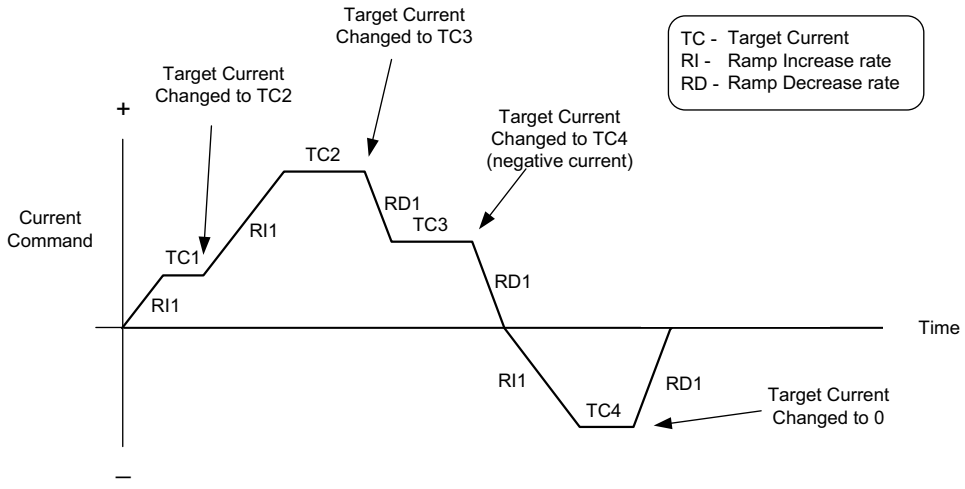


Figure 10-1: Example Commanded Current Ramp Profile

The specified target current may be positive or negative in value. The ramp is executed by continuously changing the commanded current at the user-specified ramp increase rate until the target current is reached.

The magnitude of the commanded current decreases at the user-specified current decrease rate when a new target current is specified with a smaller value (in magnitude) than the present target current, or when a new target current has a sign that is opposite to the present one. When a ramp profile crosses through a current of zero and reverses sign the ramp generator will apply the ramp increase rate rather than the ramp decrease rate.

Specified ramp increase and ramp decrease rates must always be positive. If the current decrease rate is not specified and left at a value of zero the current increase rate will be used for the current decrease rate.

Figure 10-1 illustrates a complex profile in which the specified target current and the current increase rate changes several times.

10.1 Ramp Generator Cycle Time

MC7x112 ICs calculate ramp generator updates on a regular interval known as the ramp generator cycle time. For MC7x112 torque control ICs this nominal cycle time value is 102.4 μSeconds, however this cycle time can be increased if desired.

10.2 Specifying Ramp Parameters

The ramp parameters use an encoding denoted "X.Y" with X indicating the number of bits representing the integer portion and Y indicating the number of bits used to represent the fractional component.

Name	Format	Representation, Range & units
target current	16.16	signed 32 bits (-32,768 to +32,767.9998 counts/cycle/2^{16})
current increase rate	8.24	unsigned 32 bits (0 to +127.99999994 counts/cycle^2/2^{24})
current decrease rate	8.24	unsigned 32 bits (0 to +127.99999994 counts/cycle^2/2^{24})

Specified ramp parameters are applied immediately. Whether or not these new parameters result in an immediate ramp profile change depends on the state of the ramp being generated. For example if a new current decrease rate is programmed while the current ramp is increasing this new value will not be applied until the ramp enters a decreasing current command phase.

Example

A current profile is desired that ramps the commanded torque (current) from zero to 7.5 amps in 120 mSecs. The velocity scalar (Kvel) and the cycle time have been left at their default values of 65,536 (unity), and 102.4 µSec respectively. What are the control parameters needed to command this ramp?

To convert the command in amps to a correctly scaled numerical value we use the result of the current scale factor calculation as detailed in Section 8.5.3, "Current Signal Scaling." Assuming a current scale factor of 2,500 counts per amp, the current command value is 7.5 amps * 2,500 counts/amp = 18,750.

The ramp increase setting is calculated using current command value * 2^{32}/ Velocity Scalar = 18,750 * 2^{32}/ 65,536 = 1,228,800,000. With the default cycle time of 102.4 µSec, 120 mSecs is equivalent 120,000 µSecs / 102.4 µSec/cycle = 1,172 cycles. The desired ramp increase setting is therefore 1,228,800,000 counts/cycle / 1,172 cycles = 1,048,464 counts/cycle^2. However the scaling format of the ramp rate registers is 8.24 rather than 16.16, and therefore the final commanded ramp rate value is 1,048,464 * 256 = 268,406,784.

10.3 Current Ramp Profile Stop Events

MC7x112 ICs provide an event action mechanism that allows automatic stopping of the motor axis under various conditions such as motion error, over temperature condition, and disable. Two types of controlled stops are provided; a smooth stop and an abrupt stop. In a smooth stop the current command ramps down at the user-specified value until it reaches a current command of zero. In an abrupt stop the current command is instantaneously set to zero without a ramp down phase.

When the ramp generator is active both a smooth stop and an abrupt stop will result in the user specified target current being set to zero. This means that to restart a profile the target current has to be reloaded. For an abrupt stop, in addition to the target current being set to zero the instantaneous commanded current is also set to zero.

For more information on setting up and recovering from event actions refer to Section 11.3, "Event Action Processing."

Abrupt or smooth ramp generator stops may or may not stop the motor smoothly, abruptly, or even at all. This is because it is the current command ramp profile that is being smoothly or abruptly stopped, not the motor motion itself.

10.4 Related Host Commands

In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

The following table shows the settable and readable ramp generator-related control parameters and commands:

Parameter	Host Command Mnemonic	Description
Target Current	SetVelocity* GetVelocity	32-bit signed number with 16.16 representation.
Current increase rate	SetAcceleration* GetAcceleration	32-bit unsigned number with 8.24 representation.
Current decrease rate	SetDeceleration* GetDeceleration	32-bit unsigned number with 8.24 representation.
Instantaneous current being commanded	GetCommandedVelocity	32-bit signed number with 16.16 representation indicating actual current being commanded by the ramp generator.
Instantaneous current increase/decrease being commanded	GetCommandedAcceleration	32-bit signed number with 8.24 representation indicating actual current increase or decrease rate being commanded by the ramp generator.
Set ramp profile cycle time	SetSampleTime GetSampleTime	32-bit unsigned number representing cycle time in microseconds. Default value is 102.4 μSec which is also the smallest settable value.

** The MC7x112 current command ramp generator uses the same profile generator also used with Juno ICs that create velocity profiles. For MC7x112 ICs however the resultant profile output commands current rather than velocity.*

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

This page intentionally left blank.

11. Motion Monitoring & Control

In This Chapter

11.1 Status Registers

There are five bit-oriented status registers that provide a continuous report of the state of MC7x112 ICs and the controlled axis. These five 16-bit registers are Event Status, Activity Status, Drive Status, Drive Fault Status, and Signal Status. Reading these registers is optional and is done via serial host commands.

11.1.1 Event Status Register

The Event Status register is designed to record latched events. That is, events which occur due to some condition monitored by the MC7x112. As such, each bit in this register is set by the MC7x112 and cleared by the host.

The Event Status register is defined in the following table:

Bit	Name	Description
0	*Reserved*	*May contain 0 or 1*.
1	Position wraparound	Set when the encoder position exceeds 2,147,483,647 (the most positive position), and wraps to -2,147,483,647 (the most negative position), or vice versa.
2	*Reserved*	*May contain 0 or 1*.
3	Capture received	Set when the Index position capture hardware acquires a new position value.
4-6	*Reserved*	*May contain 0 or 1*.
7	Instruction error	Set when an instruction error occurs.
8	Disable	Set when the user drives the enable signal inactive.
9	Overtemperature fault	Set when an overtemperature fault occurs.
10	Drive exception	Set when one of a number of drive exceptions, such as overcurrent, bus overvoltage or undervoltage fault occurs.
11	Commutation error	Set when a commutation error occurs.
12	Current foldback	Set when current foldback occurs.
13	Run time error	Set when a runtime error occurs.
14-15	*Reserved*	*May contain 0 or 1.*

The Event Status register may be used to generate a host interrupt signal.

11.1.1.1 Instruction Errors

Bit 7 of the Event Status register indicates an instruction error. Such an error occurs when an incorrect opcode is sent, when an argument is out of bounds, or when some other host command instruction error occurs.

Should an instruction error occur, the invalid parameters are ignored, and the Instruction error indicator in the Event Status register is set.

To determine the nature of the instruction error a **GetInstructionError** command is sent. The returned word provides the first two errors recorded after the previous execution of the **GetInstructionError** command. Refer to the documentation for the **GetInstructionError** command in the *Juno Velocity & Torque Control IC Programming Reference* for a list of instruction error codes. In addition, some commands have command-specific instruction error codes. In this case check the documentation for the specific command that was sent to the MC7x112 IC which resulted in the instruction error.

11.1.2 Activity Status Register

Activity Status register bits are not latched, they are continuously set and reset to indicate the status of the corresponding conditions.

The Activity Status register is defined in the following table:

Bit	Name	Description
0	Phasing initialized	Set 1 when the motor's commutation hardware has been initialized, cleared when an InitializePhase command is received.
1	At target current	Set 1 when the ramp generator command current is equal to the target current specified by the host. Cleared 0 if it is not. This bit only functions in conjunction with the ramp generator and is not set if the command source is set to anything other than ramp generator.
2-8	*Reserved*	*May contain 0 or 1.*
9	Position capture	Set 1 when a new position value is available to read from the Index capture hardware. Cleared 0 when a new value has not yet been captured. A serial host command retrieves the captured position value and clears this bit, thus allowing additional captures to occur. While this bit is set, no new values will be captured.
10	Ramping indicator	Set 1 when the ramp generator commanded current is non-zero. Cleared 0 when the commanded current is zero.
11-15	*Reserved*	*May contain 0 or 1.*

11.1.3 Drive Status Register

The specific status bits provided by the Drive Status register are defined in the following table. Like the Event Activity Status Register these bits are not latched.

Bit	Name	Description
0	Calibration completed	Set 1 when an analog input calibration procedure is completed. Cleared by a **CalibrateAnalog** command.
1	In foldback	Set 1 when in foldback, cleared 0 if not in foldback.
2	Overtemperature	Set 1 when the axis is currently in an overtemperature condition. Cleared 0 if the axis is currently not in an overtemperature condition.
3	Shunt active	Set 1 when shunt request is active. Cleared 0 if not.
4	*Reserved*	*May contain 0 or 1.*
5	Overvoltage	Set 1 when the axis is currently in an overvoltage condition. Cleared 0 if the axis is currently not in an overvoltage condition.
6	Undervoltage	Set 1 when the axis is currently in an undervoltage condition. Cleared 0 if the axis is currently not in an undervoltage condition.
7-11	*Reserved*	*May contain 0 or 1.*
12	Output Clipped	Set 1 when the amplifier current command can not be met because of output clipping.
13	*Reserved*	*May contain 0 or 1.*

Bit	Name	Description
14	Initializing	Set 1 when the MC7x112 is in the process of initializing from NVRAM commands. Set 0 when initializing is complete.
15	*Reserved*	*May contain 0 or 1.*

11.1.4 Drive Fault Status Register

The following table indicates the contents of the Drive Fault Status register. Like the Event Status Register these bits are latched. Unlike the Event Status register they can only all be cleared at once. They are set by the MC7x112 and cleared by the user.

Bit	Name	Description
0	Overcurrent	Set 1 to indicate a fault due to a short circuit or overload in the drive output.
1-4	*Reserved*	*May contain 0 or 1*
5	Overvoltage	Set 1 to indicate an overvoltage condition of the external bus voltage input.
6	Undervoltage	Set 1 to indicate an undervoltage condition of the external bus voltage input.
7	*Reserved*	*May contain 0 or 1*
8	Foldback	Set 1 to indicate that a current foldback event has occurred.
9, 10	*Reserved*	*May contain 0 or 1*
11	Watchdog	Set 1 to indicate that a watchdog event has occurred.
12	*Reserved*	*May contain 0 or 1*
13	Brake	Set 1 to indicate that the Brake signal input pin has gone active.
14, 15	*Reserved*	*May contain 0 or 1*

11.1.5 Signal Status Register

The Signal Status register provides real-time signal levels for various I/O pins. The Signal Status register is defined in the following table:

Bit	Name	Description
0	A encoder	A signal of quadrature encoder input.
1	B encoder	B signal of quadrature encoder input.
2	Index encoder	Index signal of quadrature encoder input.
3-6	*Reserved*	*May contain 0 or 1.*
7	HallA	Hall affect sensor input A.
8	HallB	Hall affect sensor input B.
9	HallC	Hall affect sensor input C.
10-12	*Reserved*	*May contain 0 or 1.*
13	/Enable	Enable signal input.
14	FaultOut	Fault signal output.
15	*Reserved*	*May contain 0 or 1.*

All Signal Status register bits are inputs except bit 14 (*FaultOut*).

The input bits in the Signal Status register represent the actual hardware signal level combined with the state of the signal sense mask described in the next section. That is, if the signal level is high, and the corresponding signal mask bit is 0 (do not invert), then the bit will be 1. Conversely, if the signal mask for that bit is a 1 (invert), then a high signal on the pin will result in a read of 0.

The output bits in the Signal Status register are not affected by the signal sense mask. For these signals a 1 indicates an active condition and a 0 indicates a non active condition.

11.1.5.1 Signal Sense Mask

The bits in the Signal Status register represent the high/low state of various signal pins. It is possible to invert the incoming signal to match the signal interpretation of the user's hardware.

The default value of the signal sense mask is "not inverted." The bits of the signal sense mask register are defined in the following table:

Bit	Name	Interpretation
0	A encoder	Set 1 to invert quadrature A input signal. Clear 0 for no inversion.
1	B encoder	Set 1 to invert quadrature B input signal. Clear 0 for no inversion.
2	Index encoder	Set 1 to invert, clear 0 for no inversion. This means that for active low interpretation of index signal, set to 0; and for active high interpretation, set to 1.
3-6	*Reserved*	
7	HallA	Set 1 to invert HallA signal. Clear 0 for no inversion.
8	HallB	Set 1 to invert HallB signal. Clear 0 for no inversion.
9	HallC	Set 1 to invert HallC signal. Clear 0 for no inversion.
10-11	*Reserved*	
12	Motor Direction	Set 1 to invert Motor Direction. Clear 0 for no inversion.
13–15	Reserved	

11.1.6 Related Host Commands

There are a number of commands that may be used with monitoring or signal sense registers. In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

Function	Host Command Mnemonic	Description
Read Event Status register	GetEventStatus	Returns 16-bit value of the Event Status register.
Clear Event Status register bits	ResetEventStatus	Takes a 16-bit mask as an argument allowing one or more bits of the Event Status register to be cleared.
Read Activity Status register	GetActivityStatus	Returns 16-bit value of the Activity Status register.
Read Drive Status register	GetDriveStatus	Returns 16-bit value of the Drive Status register.
Read Drive Fault Status register	GetDriveFaultStatus	Returns 16-bit value of the Drive Fault Status register.
Clear Drive Fault Status register	ClearDriveFaultStatus	Clears (sets to zero) all bits of the Drive Fault Status register. Note that for this command to take effect the Drive Fault Status register must first be read using the **GetDriveFaultStatus** command.
Read Signal Status register	GetSignalStatus	Returns 16-bit value of the Signal Status register.
Program Signal Sense mask	SetSignalSense GetSignalSense	Specified value is a 16-bit mask with each '1' bit value indicating invert, and each '0' bit value indicating don't invert.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

11.2 FaultOut Signal

The MC7x112's *FaultOut* signal is used to indicate an occurrence of a fault condition. This signal is always active high and its sense cannot be changed. Any bit condition of the Event Status register may be used to trigger activation of this signal via a user-programmable mask value.

The bit conditions specified in this mask are logically ANDed with the Event Status register. Any resultant non-zero value will cause the *FaultOut* signal to go active. See Section 11.1.1, "Event Status Register" for information on the Event Status register.

Example

Programming the fault out mask with a value of 1,032 (0x0x408) configures the *FaultOut* signal to be driven high upon an *Index* capture received (bit #3) or a drive exception error (bit #10).

11.3 Event Action Processing

MC7x112s provide a programmable mechanism for automatically reacting to various safety or performance-related events. This mechanism is called event action processing.

Each monitored event condition may have an associated event action defined for it. The following table lists each of the event conditions that MC7x112 ICs monitor along with the default event actions that will occur if no alternate user specified event actions are specified:

Event Condition Name	Default Action	Description
Current foldback	Disable Motor Output	Occurs when high amplifier current output causes a foldback condition.
Encoder Position Capture	No Action	Occurs when the quadrature encoder Index signal has triggered a position capture.
Overtemperature	Disable Motor Output	Occurs when an overtemperature condition is detected.
Disabled	Disable Motor Output	Occurs when the Enable signal goes inactive. The programmed event action must be Disable Motor Output or Passive braking.
Overcurrent	Disable Motor Output	Occurs when an overcurrent condition is detected. The programmed event action must be Disable Motor Output or Passive braking.
Overvoltage	Disable Motor Output	Occurs when an overvoltage condition is detected.
Undervoltage	Disable Motor Output	Occurs when an undervoltage condition is detected.
Watchdog timeout	No Action	Occurs when a watchdog timeout condition is detected.
Brake signal	Passive braking	Occurs when the Brake signal goes active. The programmed event action must be Disable Motor Output or Passive braking.

Unless otherwise noted above the default event action may be changed by the user. The following table describes the event actions that can be programmed:

Action Name	Description
No Action	No action taken.
Smooth Stop	Causes current ramp profile to transition to a commanded current of zero at the programmed current decrease rate.
Abrupt Stop	Causes current ramp profile to instantaneously change to a commanded current of zero.
Disable Current Loops	Disables ramp generator and current loop.
Disable Motor Output	Disables ramp generator, current loop, and motor output.
Passive Braking	Turns the passive braking function on and disables the ramp generator, current loop, and motor output.

11.3.1 Event Processing

Upon power-up and initialization completion MC7x112 ICs begin to continuously monitor the event conditions and execute the programmed event action if they occur. When the programmed action is executed, related actions may occur such as setting the appropriate bit in the Event Status register.

To recover from an event action, the cause of the event occurring should be investigated and corrected. To process events using serial host commands a **ResetEventStatus** is first sent, clearing the Event Status register associated with the event condition. The command **RestoreOperatingMode** is then sent to restore the operating mode previously specified using a **SetOperatingMode** command. In addition it is recommended, but not required, that

event conditions that affect the Drive Fault Status Register clear this register using a **GetDriveFaultStatus** command followed by a **ClearDriveFaultStatus** command. These event conditions are the Overcurrent, Overvoltage, Undervoltage, and Brake Signal events. Note that if the event condition is still present, then the event action will immediately occur again.

It is the responsibility of the user to safely and thoroughly investigate the cause of event-related events, and only restart motion operations when appropriate corrective measures have been taken.

Once programmed, an event action will be in place until reprogrammed. The occurrence of the event condition does not reset the programmed event action.

11.3.2 Enable Signal-Based Event Recovery

The MC7x112's serial port in conjunction with host commands can be used to recover from events such as overtemperature, current foldback, etc. However a second method that can be executed via simple external circuitry can be used to recover from events. This method uses the MC7x112's *Enable* signal.

Enable-based recovery, also called automatic recovery, relies on the *FaultOut* signal to indicate a fault condition. After the *FaultOut* signal goes active, external logic must delay a minimum of 150 μSec, but thereafter may request that the MC7x112 attempt to recover by deasserting, and then asserting, the *Enable* signal. The *Enable* signal must be in the deasserted state for at least 150 μSec for the request to be recognized.

When an automatic recovery request is recognized by the MC7x112 it behaves as though the command sequence **ResetEventStatus 0** and **RestoreOperatingMode** has been sent to it by a host controller. As is the case when these commands are sent by the host controller, if the fault condition is still present when recovery is attempted, the IC will immediately again experience an event action, and a recovery procedure must once again be commanded. If the fault has been corrected however a recovery request will result in resumption of normal operation.

Example

An application uses a thermistor mounted on the motor body and programs the fault out mask to generate a *FaultOut* signal when the motor gets too hot. The MC7x112's default event action is left unchanged which results in motor output being disabled. External logic monitors the *FaultOut* signal and when it goes active, the external logic delays 200 μSec, deasserts the *Enable* signal for 200 μSec, and then restores *Enable* to an asserted condition. This process is repeated until the temperature has dropped sufficiently so that the *FaultOut* signal will stay deasserted and normal operation is restored.

11.4 Host Interrupts

Interrupts allow a host microprocessor or other external circuitry to be automatically notified if a special condition occurs. For this purpose MC7x112s provide a *HostInterrupt* signal. *HostInterrupt* functions similarly to *FaultOut* but provides a separate programmable mask.

Any or all of Event Status register bits may be programmed to cause an interrupt. If a 1 is stored in the mask, then a 1 in the corresponding bit of the Event Status register will cause an interrupt to occur. MC7x112s continually and simultaneously scans the Event Status register and interrupt mask to determine if an interrupt has occurred. When an interrupt occurs, the *HostInterrupt* signal is made active.

To recover from an interrupt serial host commands are used. Once a valid recovery sequence is sent by external circuitry the MC7x112 IC will attempt to clear the Event status register along with the *HostInterrupt* signal.

11.5 Related Host Commands

There are a number of commands related to the functions described in the above sections. In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

Parameter	Host Command Mnemonic	Description
Fault out mask	SetFaultOutMask GetFaultOutMask	Specified value is a 16-bit mask with each '1' bit value indicating that the corresponding bit in the Event Status register, if active, will drive the *FaultOut* signal active.
Event action processing	SetEventAction GetEventAction	Two parameter command. The first is a fixed value that specifies the event condition for which an action is being defined, the second is a fixed value defining the action. 10 separate event conditions are defined and therefore this command could be called up to 10 times, one for each condition.
Event recovery method	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 2, the second is a fixed code value specifying either Enable-based (automatic) event recovery or commanded event recovery.
Restore the operating mode	RestoreOperatingMode	This commands restores the operating mode register to its settings before an event action occurred. Before executing this command all bits in the Event Status register must be cleared.
Host interrupt mask	SetInterruptMask GetInterruptMask	Specified value is a 16-bit mask with each '1' bit value indicating that the corresponding bit in the Event Status register, if active, will trigger a host interrupt and drive the *HostInterrupt* signal active.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

11.6 Signal Processing

The table below shows the location of the HostInterrupt and FaultOut signal on the MC7x112 ICs:

Signal	64-p[in TQFP Pin #	56-Pin VQFN Pin #	Description
HostInterrupt	46	43	HostInterrupt output signal. Whether this pin is used or not this pin must be connected to a 10 Kohm pullup resistor.
FaultOut	52	40	FaulOut output signal. If this signal is used an external 10 Kohm pull down resistor is recommended to prevent glitches during reset.

11.7 Trace

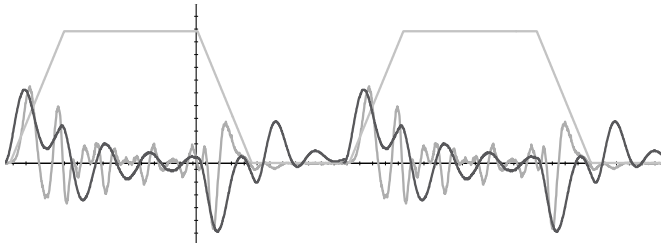


Figure 11-1: Example Motion Trace Capture

Trace is a powerful feature of the MC7x112 ICs that allows various parameters and registers to be continuously captured and stored to an on-chip RAM buffer. The captured data may later be downloaded and displayed by the host. Traces are useful for optimizing servo performance, verifying trajectory behavior, capturing sensor data, or to assist with any type of monitoring where a precise time-based record of the system's behavior is required.

Trace data capture (by the MC7x112 IC) and trace data retrieval by the host are executed as two separate processes. The host specifies which parameters will be captured, and how the trace will be executed. The IC then performs the trace. Finally, the host retrieves the data after the trace is complete. It is also possible to perform continuous data retrieval, even as the MC7x112 is continuing to collect additional trace data.

To start a trace, the host specifies several parameters. They are:

Parameter	Description
Trace period	Specifies whether trace variables are captured at every single cycle, every other cycle, or at any specified period.
Trace variables	There are more than 50 separate variables and registers within the MC7x112 IC that may be traced; for example, actual position, Event Status register, position error, etc. The user specifies up to four variables to be traced.
Trace mode	Traces can occur in one of two modes: one-time, or rolling-buffer mode. This determines how the data is stored, and whether the trace will stop automatically or whether it must be stopped by the host.
Trace start/stop conditions	To allow precise synchronization of data collection, it is possible to define the start and stop conditions for a given trace. The specified conditions and start or stop the trace automatically without host intervention.

11.7.1 Trace Period

The tracing system supports a configurable period register that defines the frequency at which data is stored to the trace buffer. The tracing frequency is specified in units of cycle times.

11.7.2 Trace Variables

When traces are running, one to four data variables may be stored to the trace buffer at the same time. The four trace variable registers are used to define which parameters are stored.

More than 50 variables are available for trace on MC7x112 ICs. For a complete list of traceable variables for MC7x112 ICs see Section A.1, "Traceable Variables."

11.7.3 Trace Modes

As trace data is collected it is written to sequential locations in the trace buffer. When the end of the buffer is reached the trace mechanism will behave in one of two ways depending on the selected mode. If one-time mode is selected then the trace mechanism will stop collecting data when the buffer is full. If rolling-buffer is selected then the trace mechanism will wrap around to the beginning of the trace buffer and continue storing data. Data from previous cycles will be overwritten by data from subsequent cycles. In this mode the diagnostic trace will not end until the trace stop conditions are met.

While not technically a mode, a useful alternate way to use the trace facility that doesn't require buffers and is not affected by trace period, start or stop condition settings is to directly request the current value of a traceable variable. This is done with the **GetTraceValue** command.

11.7.4 Trace Start/Stop Conditions

MC7x112s allow one of several trigger types to be programmed as indicated in the following table:

ID	Name	Description
0	Immediate	This trigger type indicates that the trace starts or stops immediately when the set trace start or stop command is issued. If this trigger type is specified, the trigger axis, bit number, and bit state value are not used.
2	Event Status	The specified bit in the Event Status register will be constantly monitored. When that bit enters the defined state (0 or 1), then the trace will start (stop).

ID	Name	Description
3	Activity Status	The specified bit in the Activity Status register will be constantly monitored. When that bit enters the defined state (0 or 1), then the trace will start (stop).
4	Signal Status	The specified bit in the Signal Status register will be constantly monitored. When that bit enters the defined state (0 or 1), then the trace will start (stop).
5	Drive Status	The specified bit in the Drive Status register will be constantly monitored. When that bit enters the defined state (0 or 1), then the trace will start (stop).

11.7.5 Downloading Trace Data

Once a trace has executed and the trace buffer is full (or partially full) of data, the captured data may be downloaded by the host using commands to read from the MC7x112 internal RAM memory. See Section 11.8.1, "Related Host Commands" for a complete description of memory buffer commands.

The command **GetTraceCount** is used to get the number of 32-bit words of data stored in the trace buffer. This value may be used to determine the number of **ReadBuffer** commands that must be issued to download the entire contents of the trace buffer.

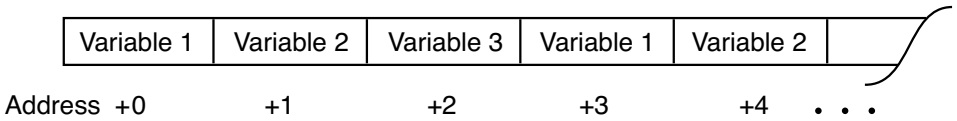


Figure 11-2: Trace Data Format

During each trace period, each of the trace variables is used in turn to store a 32-bit value to the trace buffer. Therefore, when data is read from the buffer, the first value read would be the value corresponding to trace variable 1, the second value will correspond to trace variable 2, up to the number of trace variables used. Figure 11-2 show this, illustrating an example trace buffer when 3 variables were specified for trace.

11.7.6 Related Host Commands

The table below lists host commands used with MC7x112's trace function.

Parameter	Host Command Mnemonic	Description
Trace mode	SetTraceMode GetTraceMode	Specified value is 0 for one-time capture mode, 1 for rolling buffer capture mode.
Trace period	SetTracePeriod GetTracePeriod	Specified value is the trace period in cycle times. The default MCx112 cycle time is 102.4 µSec.
Trace start condition	SetTraceStart GetTraceStart	Specified values are the condition for starting the trace, and depending on the condition set the status register monitored and the associated register bit trigger state.
Trace stop condition	SetTraceStop GetTraceStop	Specified values are the condition for stopping the trace, and depending on the condition set the status register monitored and the associated register bit trigger state.
Trace variable selection	SetTraceVariable GetTraceVariable	Specifies one of four trace variables to be simultaneously traced.
Directly request trace variable value	GetTraceValue	This command allows the value of any traceable variable to be immediately requested.

For more information on MC7x112 host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

11.8 RAM & NVRAM Memory Buffers

MC7x112 ICs are capable of accessing memory for the retrieval of trace data stored in RAM, in connection with the NVRAM, or for other user-defined purposes.

These memory operations utilize buffers, which are contiguous blocks of memory with a defined start location, size, and access ID #. A total of eight buffers are supported, including two pre-defined buffers; a RAM trace buffer which has a start address of 0x0000 0000 a size of 12, 288 bytes and an ID # of 0, and an NVRAM buffer which has a start address of 0x2000 0000 a size of 16,384 bytes and an ID# of 1.

Central to accessing buffers are the read index and the write index. The read index may be assigned a value between 0 and L-1 (where L is the buffer length). It defines the location from which the next value will be read. When a value is read from the memory buffer, the read index is automatically incremented, thus selecting the next value for reading.

Similarly, the write index ranges from 0 to L-1 and defines the location at which the next value will be written. The write index is incremented whenever a value is written to a buffer. The default value for both the read and write indexes are 0. If either the read or the write index reaches the end of the buffer, it is automatically reset to 0 on the next read/write operation.

The trace buffer (BufferID 0) may be written to and read from using the **SetBuffer** and **GetBuffer** commands listed below in Section 11.8.1, "Related Host Commands." The NVRAM buffer (BufferID1) may be read from using the **GetBuffer16**, but to write to the NVRAM buffer a special writing process must be used described in Section 14.2.1, "Writing to NVRAM."

11.8.1 Related Host Commands

The table below details host commands that access and monitor buffers.

Command	Argument	Description
GetBufferStart	bufferID	Returns the base address of the specified buffer.
GetBufferLength	bufferID	Returns the length of the specified buffer.
SetBufferReadIndex	bufferID, index	Sets the read index for the specified buffer. index is a 32-bit integer in the range 0 to length−1, where length is the current buffer length.
GetBufferReadIndex	bufferID	Returns the value of the read index for the specified buffer.
SetBufferWriteIndex	bufferID, index	Sets the write index for the specified buffer. index is a 32-bit integer in the range 0 to length−2, where length is the current buffer length.
GetBufferWriteIndex	bufferID	Returns the value of the write index for the specified buffer.
ReadBuffer	bufferID	Returns a 32-bit value from the specified RAM buffer. The location from which the value is read is determined by adding the base address to the read index. After the value has been read, the read index is incremented. If the result is equal to the current buffer length, the read index is set to zero (0).
ReadBuffer16	bufferID	Returns a 16-bit value from the specified NVRAM buffer. This command is otherwise identical to the ReadBuffer command.
WriteBuffer	bufferID, value	Writes a 32-bit value to the specified RAM buffer. The location to which the value is written is determined by adding the base address to the write index. After the value has been written, the write index is incremented. If the result is equal to the current buffer length, the write index is set to zero (0).

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

12. Brushless DC Motor Control

12

In This Chapter

- Hall-Based Commutation
- Encoder-Based Commutation
- Encoder Feedback
- Third Leg Floating Control
- Related Host Commands
- Signal Processing

MC7x112 ICs provide a number of special features for support of Brushless DC motors. These include input of Hall sensors, multi-phase signal generation, and a number of other capabilities related to phasing control and commutation.

To drive a Brushless DC motor the motor's rotor angle must be known as it continually changes. This is accomplished using one of two methods. The first is by using Hall sensors, and the second is by using the quadrature position encoder. In both cases these sensors must be directly connected to the motor shaft. Generally speaking, if an encoder is available it should be used for commutation as it will provide smoother motion and higher overall performance than Hall sensors.

Selection of whether the phasing of the motor will be Hall-based or encoder-based, is user programmable. Note that frequently *both* Hall sensors and encoder feedback signals are used. The Hall sensors are used during phase initialization, and the encoder is used thereafter to achieve the smoothest commutation during regular motor operation. See Section 12.2.1.1, "Hall-Based Phase Initialization" for more information.

12.1 Hall-Based Commutation

The Hall sensor signals are input directly through the signals *HallA, HallB, and HallC*. To accommodate varying types of Hall sensors, or sensors containing inverter circuitry, the signal level/logic interpretation of the Hall sensor input signals may be user programmed via the set signal sense register.

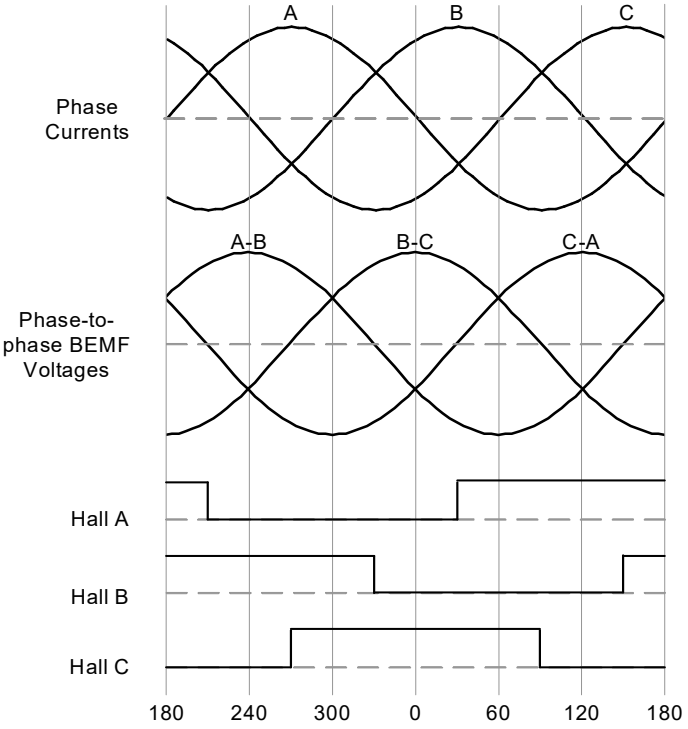
12.2 Encoder-Based Commutation

If motor commutation occurs via the MC7x112 IC encoder the number of encoder counts per electrical cycle must be user specified. This parameter indicates the number of encoder counts required to complete a single full electrical cycle. The number of electrical cycles can usually be determined from the motor manufacturer's specification and is exactly half the number of poles, or is equal to the number of pole pairs.

The number of encoder counts per electrical cycle is not required to be an exact integer.

12.2.1 Phase Initialization

Figure 12-1: Hall-based Phase Initialization



When commutating via the encoder, in addition to specifying the counts per electrical cycle MC7x112s must determine the proper initial phase angle of the motor relative to the encoder position. This information is determined using a procedure called phase initialization. Note that a phase initialization procedure is not necessary if Hall-based commutation is selected.

MC7x112 ICs provide four methods to perform phase initialization: Hall-based, algorithmic phase, pulse phase, and direct-set. Each are described in the subsequent sections.

12.2.1.1 Hall-Based Phase Initialization

The most common and the simplest method of encoder phase initialization is Hall-based. In this mode three Hall sensor signals are used to determine the motor phasing. Encoder-based commutation begins automatically after the motor has moved through at least one Hall state transition.

Figure 12-1 illustrates the relationship between the state of the three Hall sensor inputs, the sinusoidally commutated phase current commands, and the motor phase-to-phase BEMF (Back-EMF) waveforms during forward motion of the motor. A Hall to BEMF phasing diagram is a common way to specify the required alignment and as such a diagram is often provided by the motor supplier.

MC7x112s expect 120-degree separation between Hall signal transitions. To commutate using Hall sensors located 60 degrees apart, swap and invert the appropriate Hall signals and motor phases to generate the expected Hall states. Note that the signal sense register can be used to accomplish hall signal inversion.

12.2.1.2 Algorithmic Phase Initialization

MC7x112s provide the ability to initialize encoder-based phasing of Brushless DC motors without the need for explicit phasing sensors such as Hall sensors using one of two methods, the first is called algorithmic phase initialization (and is described in this section) and the second is called pulse phase initialization (and is described in Section 12.2.1.3, "Pulse Phase Initialization").

Algorithmic initialization executes by energizing the motor coils in a particular sequence using a user-specified parameter for the applied motor command (called the ramp command) and for the motor command application duration (called the ramp time).

The following table shows the parameters that are set in association with algorithmic phase initialization:

Parameter	Range	Description
Ramp time	0 – 32,767	Is the amount of time, in cycle times, that the ramp command is applied during the initialization procedure.
Ramp command	1 – 32,767	Is the motor command that will be applied during the initialization procedure. If applied with the current loop disabled the specified value represents a % HV voltage command from 0 to 100%. If current loop is enabled this value represents a current in amps scaled according to the amplifier sense circuitry.

Once these parameters are loaded an **InitializePhase** command can be sent. The specified ramp time is generally determined by the motor size, with typical values for NEMA 23 motors being 0.5 to 1.0 second. A flag in the Activity Status register indicates whether phase initialization has completed, and a flag in the Event Status register indicates whether an error was detected during phase initialization.

Algorithmic initialization can be executed with the current loop enabled or disabled. If enabled the specified ramp command will represent a current in amps scaled according to amplifier sensing circuitry See Section 8.5.3, "Current Signal Scaling" for an example. If the current loop is not enabled the ramp command will represent a percentage of the available HV voltage. See Section 9.1.1, "Motor Output Module Command Scaling" for an example. In either case this value should be carefully chosen to not result in overly aggressive moves that may result in oscillation or overheating of the motor.

Algorithmic phase initialization can only function properly if motor movement is free and unencumbered.

12.2.1.3 Pulse Phase Initialization

Pulse phase initialization is an alternate procedure for phase initialization by exciting the motor coils and observing the resultant motion. It executes by applying a rapid burst of positive/negative command pulse pairs followed by a longer ramp pulse.

The following table shows the parameters that are set in association with pulse phase initialization:

Parameter	Range	Description
PositivePulseTime	0-32,767	Is the time, in cycle times, that the positive portion of the burst pulse is applied.
NegativePulseTime	0-32,767	Is the time, in cycle times, that the negative portion of the burst pulse is applied.
PulseMotorCommand	1-32,767	Is the motor command that will be applied during the burst mode of the initialization procedure. See "Ramp command" in the previous table for details.
RampMotorCommand	1-32,767	Is the motor command that is applied during the ramp portion of the initialization procedure. See "Ramp command" in the previous table for details.
RampTime	0-32,767 cycles	Is the time that the slow ramp and hold is applied, in cycle times.

Once these parameters are loaded an **InitializePhase** command can be sent. While varying somewhat with motor size, typical total durations of the pulse phase sequence is 250-500 mSec. As for algorithmic initialization, a flag in the Activity Status register indicates whether phase initialization has completed, and a flag in the Event Status register indicates whether an error was detected during phase initialization.

While the control parameter settings are simpler for algorithmic initialization, the advantage of pulse phase initialization is that the motor moves less while executing the procedure, typically 1/20th to 1/50th of an electrical cycle whereas algorithmic phase init may move up to 1/2 an electrical cycle.

Pulse phase initialization can only function properly if motor movement is free and unencumbered.

12.2.1.4 Direct-Set Phase Initialization

If, after power-up, the location of the motor phasing is known, the phase angle can be directly specified by the user using the **SetCommutationParameter** commad. This typically occurs when sensors such as resolvers are used where the returned motor position information is absolute in nature. Note that this approach typically only applies to microprocessor controlled, rather than standalone NVRAM-based operation.

When direct-set phase initialization is used the InitializePhase host command should not be executed.

12.2.2 Automatic Phase Correction

To enhance commutation reliability the MC7x112 ICs provide the ability to automatically correct the commutation phase during encoder-based commutation. Note that if Hall-based commutation is used, this feature is not necessary. Either an index signal or Hall signals can be used for this automatic correction function.

12.2.2.1 Index-Based Phase Correction

To utilize automatic phase correction the motor encoder provides an index pulse signal once per rotation. Index phase referencing is recommended for all rotary brushless motors with quadrature encoders directly mounted on the motor shaft. For linear brushless motors, it is generally not used. However, it may be used as long as the index pulses are arranged so that each pulse occurs at the same phase angle within the commutation cycle.

With an index signal properly installed, MC7x112 will automatically adjust the commutation angle to correct for any small losses of encoder counts that may occur. If the loss of encoder counts becomes excessive, or if the index pulse does not arrive at the expected location within the commutation cycle, a commutation error occurs, which is indicated via bit 11 in the Event Status register. This bit is set if the required correction is greater than (PhaseCounts / 128) + 4.

To recover from a commutation error this bit should be cleared by the host. Depending on the cause of the error there may be no further errors or errors may continue to occur.

A commutation error may indicate a serious problem with the motion system, potentially resulting in unsafe motion. It is the responsibility of the host to determine and correct the cause of commutation errors.

12.2.2.2 Hall-Based Phase Correction

Hall signals may also be used for automatic commutation phase adjust when an index pulse is not available.

Hall-based phase error detection functions similarly to index-based phase correction. Bit 11 of the Event Status register signals a commutation error and recovery occurs in the same manner. The difference is that the absolute amount of error allowed before a commutation error occurs is PhaseCount/32 + 4 with Hall-based phase correction.

12.2.3 Adjusting the Phase Angle

MC7x112 ICs support the ability to change the motor's phase angle directly, both when the motor is stationary and when it is in motion. Although this is not generally required, it can be useful during testing, or when direct-set phase initialization methods are used. Note that the phase angle can not be changed if Hall-based commutation is used.

12.3 Encoder Feedback

The MC7x112 torque control ICs support quadrature position encoder input for the purpose of commutation or general purpose position feedback. To provide encoder tracking the *QuadA* and *QuadB* feedback signals are continually accumulated in a 32-bit position value called the actual position. At power-up, the default actual position is zero. The full range of trackable positions is -2,147,483,647 to +2,147,483,647.

In the event that a spinning axis exceeds either of these position limits the actual position wraps around, with the largest positive position wrapping around to become the smallest negative position, and the smallest negative position wrapping around to become the largest positive number. When such a wraparound occurs a corresponding bit in the Event Status register is set.

Position wraparound, should it occur during operation, is generally not a consequential event. A position wraparound will have no impact on the behavior of the axis, nor is there a limit to the number of such wraparound events that may occur. For more information on the Event Status register see Section 11.1.1, "Event Status Register."

12.3.1 Position Capture

MC7x112s support a high-speed position capture function that allows the current axis location (as determined by the attached encoder) to be captured when triggered by the *Index* signal. When a capture is triggered, the content of the actual position register is transferred to a position capture register, and the capture-received indicator of the Event Status register is set.

The capture register can be read using host commands. Reading the position capture register causes the trigger to be re-armed, allowing for more captures to occur.

12.4 Third Leg Floating Control

For Brushless DC motors that do not have an encoder but have Hall sensors, a different control scheme is often used called third leg floating control. Third leg floating drives only two of three legs (motor windings) at any moment with the third, non-driven leg, floating. Hall sensors are required to use this control approach.

Although the majority of systems will use the third-leg floating scheme in combination with current control, it is in fact possible to operate the third-leg floating scheme with current control disabled. The overall control scheme is the same, with two phases energized and one left floating at any particular moment.

12.4.1 Third Leg Floating in Voltage Mode

Some motors will actually achieve their highest no-load speed (with or without a current loop active) operating with third-leg control in volage mode. This is particularly true of motors with very low inductances. Applications that operate motors in this way include surgical drills and some types of pump and fan controls.

Whenever motors are run in voltage mode special care should be taken to avoid excessive levels of current flow in the motor coils. This is especially true when operating low inductance motors from a standstill. A common approach to limiting current flow in this case is to use the MC7x112's ramp generator to gradually increasing the motor command.

12.5 Related Host Commands

The table below provides a summary of the settable and readable parameters and commands described in this chapter:

Parameter	Host Command Mnemonic	Range & Description
Operating mode register	SetOperatingMode GetOperatingMode	Specified value is a bit-oriented mask determining the state of the command source, current loop, and motor output enable/disable flags.
Commutation mode	SetCommutationMode GetCommutationMode	Specified value is a fixed code selecting Hall-based or encoder-based commutation.
Phase counts	SetCommutationParameter GetCommutationParameter	Two parameter command, the first must be 0. The second has a range of 0 to 134,217,727 and specifies the number of encoder counts per mechanical rotation of the motor.
Phase counts denominator	SetCommutationParameter GetCommutationParameter	Two parameter command, the first must be 3. The second has a range of 0 to 32,767 and specifies the number of electrical cycles per mechanical rotation of the motor.
Phase angle	SetCommutationParameter GetCommutationParameter	Two parameter command, the first must be 1. The second has a range of 0 to the value of the phase counts register. To convert phase angle in counts to degrees divide the phase angle by phase counts and multiply by 360.
Phase initialization mode	SetPhaseInitializeMode GetPhaseInitializeMode	Specified value is a fixed code selecting Hall-based phase initialization or pulse phase initialization.
Positive pulse time	SetPhaseParameter GetPhaseParameter	Two parameter command, the first must be 1. The second has a range of 0 to 65,534 and specifies the positive pulse time in units of profile cycles.
Negative pulse time	SetPhaseParameter GetPhaseParameter	Two parameter command, the first must be 2. The second has a range of 0 to 65,534 and specifies the negative pulse time in units of profile cycles.
Pulse motor command	SetPhaseParameter GetPhaseParameter	Two parameter command, the first must be 3. The second has a range of 0 to 32,767 and specifies the motor command in units of % / 327.67 during burst mode of pulse phase initialization.
Ramp motor command	SetPhaseParameter GetPhaseParameter	Two parameter command, the first must be 5. The second has a range of 0 to 32,767 and specifies the motor command in units of % / 327.67 during ramp mode of pulse phase initialization.
Ramp time	SetPhaseParameter GetPhaseParameter	Two parameter command, the first must be 0. The second has a range of 0 to 65,534 and specifies the pulse phase initialization ramp time in units of profile cycles.
Initialize phase	InitializePhase	This host command has no parameters associated with it and initiates phase initialization using the phase initialization mode previously specified by the user.
Commutation phase correction mode	SetPhaseCorrectionMode GetPhaseCorrectionMode	Specified value is a fixed code selecting either no phase correction, phase correction via index signal, or phase correction via Hall signals.
Actual encoder position	GetActualPosition	Retrieves the current 32-bit position of the quadrature encoder.
Position capture register	GetCaptureValue	Retrieves the 32-bit capture register value. Reading the capture register causes the capture trigger to be re-armed, allowing further captures to occur.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

12.6 Signal Processing

12.6.1 Hall Sensors

Although not required, Hall signals are commonly used when driving Brushless DC motors. All of the MC7x112 Hall signal inputs are digital TTL level signals, so for typical applications external signal processing circuitry is used.

The table below provides a summary of the MC7x112 Hall-based signals.

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
HallA	35	23	HallA input
HallB	31	42	HallB input
HallC	64	34	HallC input

There is some variation in how typical motors are wired, so for convenience MC7x112 ICs provide the ability to change the signal sense interpretation of these incoming signals.

12.6.1.1 Typical Hall Processing Circuitry

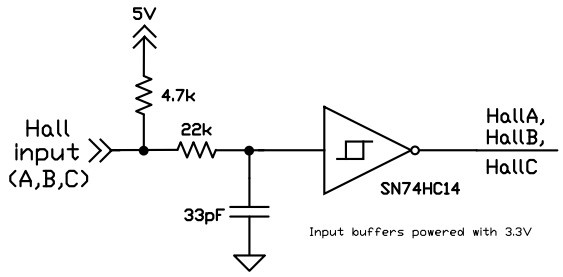


Figure 12-2: Typical Hall Processing Circuitry

Figure 12-2 shows a typical circuit for processing Hall sensor signals. The 4.7k resistor pull-up resistor is required when the Hall signals are open-collector. The 22k resistor and 33pF capacitor provide a low-pass filter with a bandwidth of 219 kHz. The selection of the low-pass filter bandwidth depends on noise and Hall signal frequency. This buffer stage has an inversion on its output.

12.6.2 Quadrature Encoder Signal Interfacing

The table below provides a summary of the encoder related signals:.

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
QuadA	62	55	Quad A input
QuadB	63	56	Quad B input
Index	4	2	Index input

By default the *Index* capture signal is active low, however this interpretation is user programmable.

All of the encoder related signals are digital TTL level signals, so for typical cable-based connection to a motor encoder differential transceiver chips are used.

QuadA and *QuadB* are expected to be offset from each other by 90 degrees. When the motor moves in the position direction, *QuadA* should lead *QuadB*. When the motor moves in the negative direction *QuadB* should lead *QuadA*. Four resolved quadrature counts occur for one full phase of each A and B channel.

The *Index* signal provides a capture trigger for the instantaneous up/down quadrature position. This input signal is most often tied to the encoder's index output, but this is not required.

See Section 3.2.1, "Quadrature Encoder Input" and Section 4.1, "Quadrature Encoder Input" for timing information related to the quadrature input signals.

12.6.2.1 Typical Quadrature Encoder Processing Circuitry

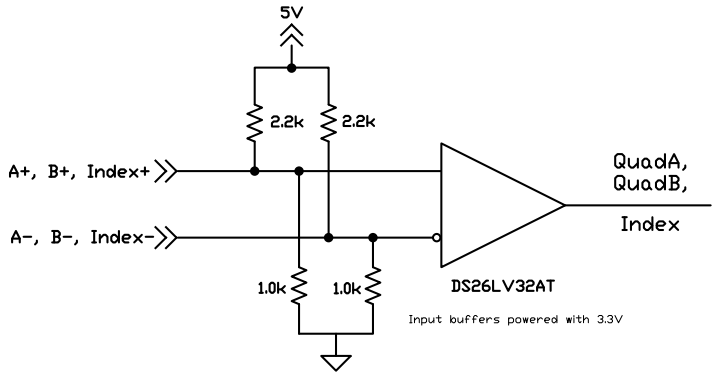


Figure 12-3: Quadrature Encoder Processing Circuitry

Figure 12-3 shows a typical circuit for processing differential quadrature and index signals. The pull-up and pull-down resistors provide both termination and a bias voltage. When single ended encoder signals are used connect to the positive input and leave the negative input unconnected.

13.Drive & DC Bus Safety

13

In This Chapter

- Drive & DC Bus Safety
- Current Foldback
- Shunt Control
- Related Host Commands
- Signal Processing

13.1 Drive & DC Bus Safety

The MC7x112 torque control ICs provide sophisticated drive and DC Bus safety features. These features include overtemperature monitoring, over and under voltage monitoring, overcurrent monitoring, and current foldback.

The following sections provide detailed information for these drive and DC Bus safety related features.

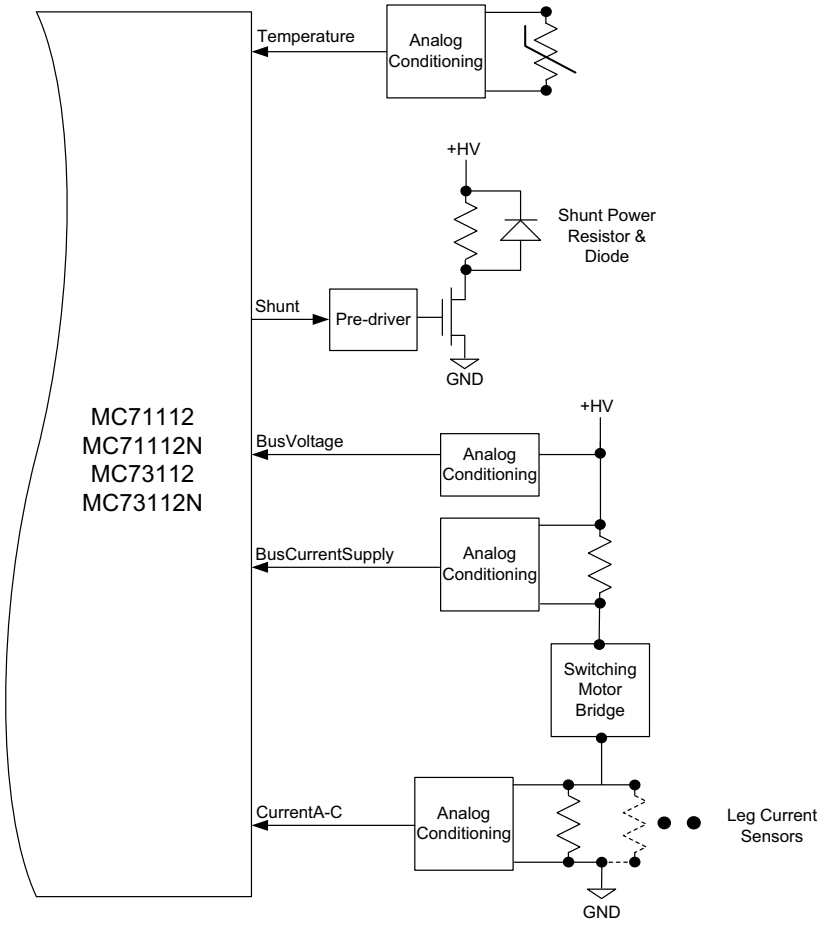


Figure 13-1: Drive & DC Bus Safety Feature Circuitry Overview

13.1.1 Overtemperature Protection

MCx112 ICs support a *Temperature* sensor input to continuously monitor the temperature of the power electronics or another part of the motion system. Although various temperature sensors may be used, the most common type of sensor is a thermistor.

The value of the temperature sensor and downstream analog conditioning circuitry determine the overall temperature range that can be measured. The overall temperature sense range should be 15% to 25% above the highest expected temperature.

The *Temperature* signal input expects a voltage in the range of 0V to 3.3V representing the sensed temperature. Both temperature-voltage-increasing (voltage increases with increasing temperature) and temperature-voltage-decreasing (voltage decreases with increasing temperature) thermistors are supported. For voltage increasing thermistors 0.0V represents the lowest possible temperature, and for voltage decreasing thermistors 3.3V represents the lowest possible temperature.

The overtemperature threshold is user programmable. The sign of the overtemperature threshold selects whether the *Temperature* input increases or decreases with rising temperature; positive thresholds indicate voltage increase while negative thresholds indicate voltage decrease. The actual limit threshold utilized for the comparison is the absolute value of the specified limit value. The magnitude of the numerical threshold value has a range of 0 to 32,767 with 0 being the lowest readable temperature and 32,767 the highest.

In addition to the settable overtemperature threshold MC7x112 ICs support a settable temperature hysteresis. This function is used to avoid spurious re-triggering of an overtemperature event.

Example

A temperature-voltage-increasing thermistor and associated analog processing circuitry generate a voltage of 2.9V when the amplifier is at the hottest safely operable temperature. The overtemperature limit specified should thus be set to 32,768 * 2.9V/3.3V = 28,796. Later the design is changed so that a voltage decreasing thermistor is used. With a voltage decreasing thermistor an overtemperature limit of -28,796 would be programmed.

The overtemperature detect functions continuously once programmed. To disable the overtemperature check a threshold value of 32,767 is set.

Processing and recovering from overtemperature events is described in Section 11.3, “Event Action Processing.”

13.1.2 Overcurrent Monitoring

Figure 13-1 shows the DC bus monitoring scheme used with the MC7x112 torque control ICs. Both the supply-side and return-side DC bus current are monitored to detect overcurrent conditions. Supply side current measurement detects shorts of the motor windings to ground and shorts of the windings to each other. Return current measurement can not measure shorts to ground, but can measure winding shorts to each other.

The *BusCurrentSupply* signal directly encodes the total current flowing through the motor amplifier bridge(s) from the +HV supply, with 0.0V encoding no current flow and 3.3V encoding the maximum measurable amount of current flow. The value of the sense resistor and downstream analog conditioning circuitry determine the overall current range that can be measured. For the DC bus supply current input this range should be at least 150% of the maximum expected DC bus peak current flow. The measured bus supply current is an unsigned number with range of 0 to 65,535.

The DC Bus supply overcurrent threshold function operates continuously once programmed. To disable a DC bus supply overcurrent check a threshold value of 65,535 is set.

Return current flow from the DC bus is measured via the leg current sensors as part of the current control mechanism. The bus return current threshold has different scaling than the supply side. It is an unsigned number with range of 0 to 32,767 and has the same scaling as the leg current scaling. A DC bus return overcurrent threshold is set to compare against the measured leg current reading. See Section 8.5.3, “Current Signal Scaling” for more information on leg current scaling.

Bus return and overcurrent detection functions continuously once programmed. To disable a bus return overcurrent check a threshold value of 32,767 is set.

Example

An isolating op-amp and sense resistor generate 3.3V at a DC bus supply current flow of 15 amps. The numerical scaling of the current threshold is therefore 15.0A/65,536 = .228 mA/count. The overcurrent threshold is set at 9.5 amps, or 9,500 mA/.228 mA/count = 41,667.

Using the scaling defined in the example in Section 8.5.3, "Current Signal Scaling" to set the same overcurrent threshold for the DC bus return current of 9.5A, a value of 9.50 A * 1.25 * 32,767 / 10.0 A = 31,128 is used.

Return current measurement is only available when the motor output mode is set to PWM High/Low.

If current control is not implemented then it is not possible for the MC7x112 IC to measure DC bus return current. DC bus supply current measurement is not affected however and can function with or without current control active.

Processing and recovering from overcurrent events is described in Section 11.3, "Event Action Processing."

13.1.3 Over/Under Voltage Monitoring

The MC7x112 ICs can monitor the DC bus voltage for overvoltage and undervoltage conditions utilizing the *BusVoltage* analog input signal. Overvoltage and undervoltage detection is accomplished by checking the measured voltage of the DC bus and comparing with user-provided thresholds.

The value of the sense resistor and downstream analog conditioning circuitry determines the overall voltage measurement range. This overall voltage range should be 15% to 50% above the maximum expected DC bus voltage.

The *BusVoltage* input range is 0.0V to 3.3V, with 0.0V representing a DC bus voltage of 0V, and +3.3V representing the largest measurable DC bus voltage. Both an over and undervoltage threshold are user programmable, and have a range of 0 to 65,535.

Example

In an application that will have a motor voltage of 48 volts, external circuitry has been selected to present 3.3V at the *BusVoltage* input when the DC bus voltage is 65 volts. The scaling is 65V/65,536 or .992 mV/count. To set an undervoltage threshold of 45V a value of 45,000 mV/.992 mV/count = 45,362 is specified. To set an overvoltage threshold of 52V a value of 52,000 mV/.992 mV/count = 52,419 is specified.

The under and overvoltage thresholds function continuously. To disable the under voltage or over voltage check, threshold values are set to 0 or 65,535 respectively.

Processing and recovering from over or undervoltage events is described in Section 11.3, "Event Action Processing."

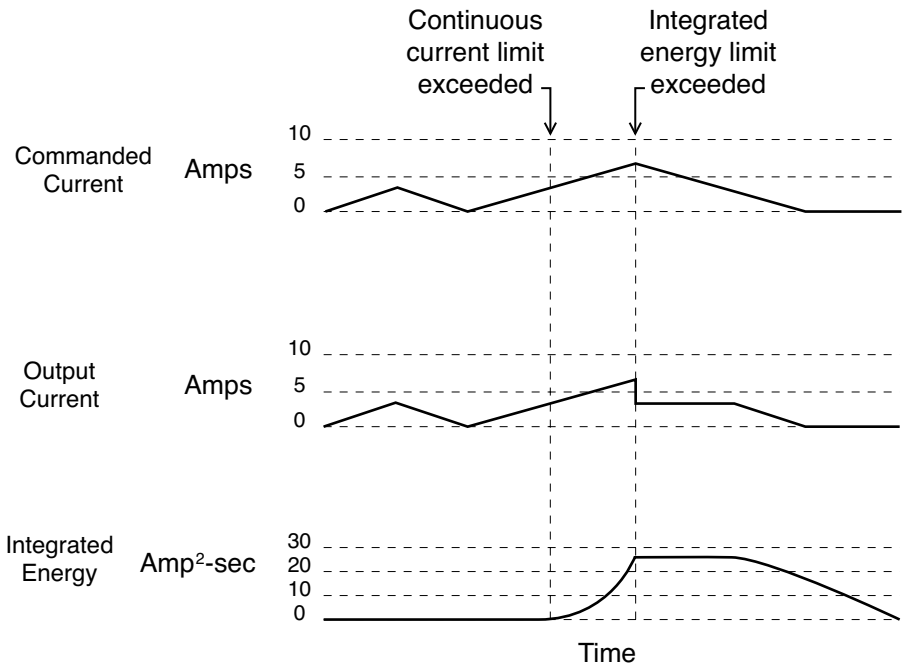
13.2 Current Foldback

MC7x112 ICs support a current foldback feature, sometimes referred to as I^2t foldback, which can be used to protect the drive output stage or motor windings from excessive current. I^2t current foldback works by integrating, over time, the difference of the square of the actual motor current and the square of the user-settable continuous current limit.

When the integrated value reaches a user-settable energy limit, the IC goes into current foldback. The default response to this event is to cause the current loop and motor output modules to be disabled. However it is also possible to program the IC to attempt to clamp the maximum current to the continuous current limit value. Note that the ability to do so depends on a properly functioning current loop.

The MC7x112 will stay in foldback until the integrator returns to zero. This is shown in Figure 13-2.

Figure 13-2: Current Foldback Processing Example



Setting continuous current limit and energy limit to less than the maximum available from the amplifier circuitry is useful if the required current limit is due to the motor, rather than to the drive electronics.

The instantaneous state of current foldback (whether the foldback limit is active or not) is available in the Drive Status register. In addition, if a foldback event has occurred, this event is recorded in the Event Status register.

Example

A particular motor has an allowed continuous current rating of 3 amps. In addition, this motor can sustain a temporary current of 5 amps for 2 seconds.

In this example the *continuous current limit* would be set to 3 amps, and the energy limit would be set to:

Energy Limit = (peak current^2 - continuous current limit^2) * time

Energy Limit = ($5^2A^2 - 3^2A^2$) * 2 Sec

Energy Limit = $32A^2$Sec

Following the current scaling example in Section 8.5.3, "Current Signal Scaling" the programmed continuous current limit would be 3.0 A * .625 * 32,767 / 10.0A = 6,144.

The programmed energy limit value must convert seconds to MC7x112 current loop cycles (19,531 cycles/sec) and must factor in the scaling of the MC7x112's programmed energy limit value which is $1/2^{31}$. This gives:

Programmed Energy Limit = 32 A^2Sec * (1.25 * 32,768 / 10A)^2 * 19,531 cycles/Sec * $1/2^{31}$

Programmed Energy Limit = 4,883

Current foldback, when it occurs, may indicate a serious condition affecting motion stability, smoothness, and performance. It is the responsibility of the user to determine the appropriate response to a current foldback event.

Processing and recovering from current foldback events is described in Section 11.3, "Event Action Processing."

13.3 Shunt Control

As shown in Figure 13-1 MC7x112s can control a shunt PWM output, which in turn typically drives a MOSFET or IGBT switch which connects the HV bus voltage to the DC bus ground via a power resistor, thereby lowering the DC bus voltage. Shunt is useful for controlling excessive DC bus voltage which is most commonly caused by the motor decelerating rapidly resulting in back-EMF generation.

The shunt functions by continually comparing the DC bus voltage, as presented at the *BusVoltage* signal, to a user programmable threshold. If the DC bus voltage exceeds the comparison threshold the *Shunt* signal outputs a PWM waveform at a user programmable duty cycle. This PWM frequency is equal to the motor drive PWM frequency. Once active, shunt PWM output will stop when the DC bus drops to 2.5% below the threshold comparison value.

Once programmed, the shunt comparison function operates continuously. To disable it, a value of 65,535 should be programmed. The shunt function is not active when motor output is not enabled (the active operating mode output bit is not set).

The scaling and units of the shunt comparison threshold is the same as for the over and undervoltage functions—an unsigned 16-bit number having a range of 0 to 65,534. The shunt PWM duty cycle is an unsigned 16-bit numbers with a range of 0 to 32,767.

Example

In the system from the example in Section 13.1.3, "Over/Under Voltage Monitoring" the shunt will be activated when the DC bus voltage climbs to 51 volts with a duty cycle of 95%. The shunt comparison threshold is set to 51,000 mV/.992 mV/count = 51,411 and the duty cycle value is .95 * 32,768 = 31,130.

Whether the shunt output is active at any given moment can be determined using the Drive Status register. For more information refer to Section 11.1.3, "Drive Status Register."

13.4 Related Host Commands

There are a number of parameter settings related to drive safety features. In the table below both 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences.

Parameter	Host Command Mnemonic	Range & Description
Overtemperature threshold	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 4, the second has a range of -32,768 to 32,767 and specifies the thermistor sense (voltage increasing or decreasing) via the sign of the specified value and the threshold limit via the magnitude of the specified value.
Overtemperature hysteresis	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 5, the second has a range of 0 to 6,400 and specifies the hysteresis.
DC bus overcurrent supply threshold	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 10, the second has a range of 0 to 65,535 and specifies the DC bus supply overcurrent threshold.
DC bus overcurrent return threshold	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 11, the second has a range of 0 to 32,767 and specifies the DC bus return overcurrent threshold.
DC bus overvoltage threshold	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 0, the second has a range of 0 to 65,535 and specifies the DC bus overvoltage threshold.
DC bus undervoltage threshold	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command. The first must be 1, the second has a range of 0 to 65,535 and specifies the DC bus undervoltage threshold.

Parameter	Host Command Mnemonic	Range & Description
Current foldback continuous current limit	SetCurrentFoldback GetCurrentFoldback	Two parameter command. The first must be 0, the second has a range of 0 to 32,767 and specifies the foldback continuous current limit.
Current foldback energy limit	SetCurrentFoldback GetCurrentFoldback	Two parameter command. The first must be 1, the second has a range of 0 to 32,767 and specifies the foldback energy limit.
Shunt comparison threshold	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command, the first must be 8. The second has a range of 0 to 65,535 and specifies the shunt comparison threshold.
Shunt duty cycle	SetDriveFaultParameter GetDriveFaultParameter	Two parameter command, the first must be 9. The second has a range of 0 to 32,767 and specifies the shunt PWM output duty cycle.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

13.5 Signal Processing

The following table shows the signals that are used in connection with the drive and DC Bus safety features.

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
Temperature	9	7	Measured temperature
BusVoltage	13	11	Measured bus voltage
BusCurrentSupply	10	8	Measured DC bus current

When using a thermistor the *Temperature* input signal should be filtered to minimize noise. The *Temperature* input is sampled at a rate of 1 kHz or higher, and therefore a low pass filter with a rolloff at 500 Hz or lower is recommended.

13.5.1 Typical Overtemperature Processing Circuitry

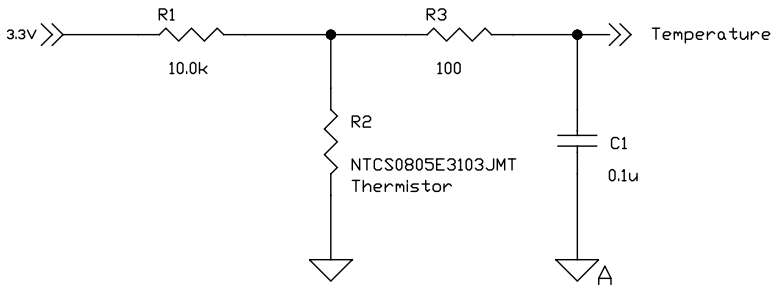


Figure 13-3: Over-temperature Processing Circuitry

Figure 13-3 shows a typical signal processing circuit for use with the *Temperature* input. The thermistor is a 10k NTC temperature-voltage-decreasing type. C1 is referenced to analog ground and should be placed close to the *Temperature* pin of the MC7x112 IC. R3 is optional. It can provide additional filtering to improve noise immunity if needed.

See Section B.7, "Drive-Related Safety and Monitoring Features" for a complete example schematic of temperature input using MC7x112 ICs.

13.5.2 Typical Overcurrent Processing Circuitry

The DC bus current supply sensor typically consists of a sense resistor, as shown in Figure 13-1, or a linear Hall sensor. The analog processing circuitry required for each is somewhat different. If a dropping resistor is used an isolating operational amplifier, current mirror, or similar circuit should be used. Linear Hall sensors typically use just a ground-referenced operational amplifier.

The *BusCurrentSupply* input range is 0.0V to 3.3V with 0.0V representing no (zero) current flowing and 3.3 volts representing the maximum measurable amount of current flowing. The signal should be filtered to minimize noise,

and the source impedance of the signal conditioning circuit should be less that 100 ohms. Current inputs are sampled by a dedicated high speed circuit internal to the MC7x112 IC. To minimize false positives a low pass filter with a roll off value of 350 kHz is recommended.

13.5.2.1 Typical Overcurrent-Processing Circuitry

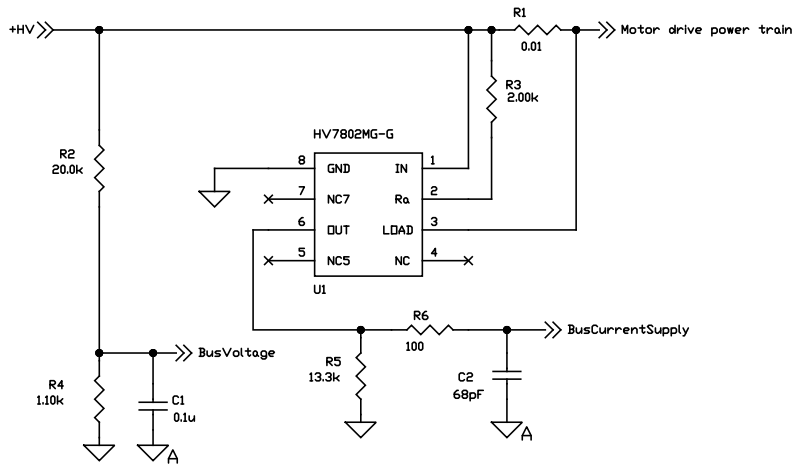


Figure 13-4: DC Bus Monitoring Circuitry

Figure 13-4 shows a typical processing circuit for DC Bus voltage and *BusCurrentSupply* over current sensing. The bus current sensing includes R1, U1, U2A and related passive parts. U1 is a high-side bus current sensing IC, and its output on R7 represents the bus current.

During motor deceleration or other motion conditions it may be possible for the DC bus supply current flow to be negative. Care should be taken to insure that negative currents do not generate a negative voltage at the MC7x112's BusCurrentSupply analog input pin. This is generally accomplished via a diode. See Section B.7, "Drive-Related Safety and Monitoring Features" for examples of DC bus safety-related schematics.

13.5.3 Typical Over/Under Voltage Processing Circuitry

The DC Bus voltage sensor typically consists of a voltage divider, which may be created from an isolating operational amplifier, current mirror, or similar circuit.

The *BusVoltage* signal should be filtered to minimize noise. The DC bus voltage input is sampled at a rate of 20 kHz, and therefore a low pass filter with a rolloff of 10 kHz or less is recommended.

Figure 13-4 shows a typical processing circuit for DC Bus voltage monitoring. The bus voltage sensing consists of R2, R4 and C1. R2 and R4 scale the bus voltage, between 0 and 3.3V. C1 is referenced to analog ground and should be placed close to the *BusVoltage* pin.

See Section B.7, "Drive-Related Safety and Monitoring Features" for complete example schematics of DC bus management designs.

13.5.4 Shunt Related Circuitry

The shunt resistor connected should have a resistance such that the current flow through the shunt switch, diode and resistor do not exceed the ratings of those components at the expected DC bus voltage.

For example if the MOSFET switch being used has a current limit of 10 amps, and with an HV supply of 48 Volts and a PWM duty cycle programmed of 75 % (generating an effective voltage of 48V x .75 = 36V) the resistance should be no less than 3.6 ohms.

Another important consideration for selection of the resistor is the amount, and rate, of energy flow that it is expected to be 'shunted' away. As shunt is engaged and current flows through the resistor it heats up. Care should

be taken to ensure that the resistor can dissipate this heat safely in worst case shunting events. Along these lines for some applications resistors which include heat sink attachments may be considered.

For more information on shunt-related circuitry see Section B.8, “Shunt Regulation” for a reference schematic.

14.Power-up, Configuration Storage & NVRAM

In This Chapter

14.1 Power-up

After receiving stable power at the Vcc pins MC7x112 ICs begin their initialization sequence. In a power-up where no user-provided configuration settings have been stored this takes approximately 250 mSec. At the end of this sequence all parameters are at their default values, and both the current loop & commutation module and the motor output module are disabled. At this point the IC is ready to receive serial host commands by microprocessor command.

MC7x112s also supports the ability to store configuration settings that are applied during the power up sequence. For this purpose, a 1,024 word non-volatile (NVRAM) memory is provided, meaning the data stored will be available even after power to the MC7x112 IC is removed.

The power-up initialization information stored in the NVRAM takes the form of host command packets, however rather than being sent via the serial port, these packet words are stored in memory. If the non-volatile memory has been loaded with configuration information the power-up sequence detects this and begins executing these commands. Note that processing stored commands may increase the overall initialization time depending on the command sequence stored.

14.2 NVRAM

MC7x112 torque control ICs provide a 1,024 16-bit word NVRAM. The primary purpose of the NVRAM is to allow configuration information to be stored, so that upon power up the IC can be configured and initialized automatically rather than requiring serial host commands to perform this function.

All data stored in the NVRAM utilizes a data format known as PMD Structured data Storage Format (PSF). Users who rely only on PMD's Pro-Motion software package need not concern themselves with the details of PSF. Users who want to address the NVRAM from their own software, or who want to create their own user-defined storage should refer to the *Juno Velocity & Torque Control IC Programming Reference* for detailed information on PSF.

14.2.1 Writing to NVRAM

There are significant restrictions to writing to MC7x112 IC's NVRAM area. In particular it is not possible to erase and rewrite selected sections of the memory space. Only the sequence described in this section can be used to write memory into the user NVRAM space.

Data stored into the NVRAM area must follow PSF (PMD Structured data Format). Failure to do so may result in unexpected behavior of the MC7x112 IC.

The following sequence is used to store command initialization data or other data to the non-volatile memory area:

1. Send a **NVRAM** command with an argument of *NVRAMMode*. Sending this command places the MC7x112 in a special mode allowing it to store memory into the NVRAM. Before proceeding the host controller should delay 1 second or more.
2. Send a **NVRAM** command with an argument of *EraseNVRAM*. This command will erase the entire NVRAM memory area. Before proceeding the host controller should delay four seconds or more.
3. For each 16-bit word of data that is to be written into the NVRAM area the command **NVRAM** with an argument of *Write* is sent, along with the data word to be written. After each word is written the MC7x112 increments an internal pointer so that subsequent data words are automatically stored in the correct location.
4. Once all data is successfully written the host controller should send a **Reset** command, which will cause the MC7x112 to reboot and execute a power up sequence. Note that this power-up sequence will include processing the stored data sent using the above sequence.

If an error occurs when processing NVRAM the instruction error event bit will be set and the **GetInstructionError** command may be used to read the error code.

It is not possible to write to the NVRAM area using the buffer commands. The procedure outlined in Section 14.2.1, "Writing to NVRAM" must be used to write data to the NVRAM area.

14.2.2 Reading from NVRAM

If desired, it is possible to directly read the NVRAM memory area using buffer commands. See Section 11.8, "RAM & NVRAM Memory Buffers" for more information on buffer host commands.

For convenience the NVRAM buffer is pre-defined with an ID# of 1. The **ReadBuffer16** command can be used to read all or part of the NVRAM space.

14.3 Initialization Control

To make the initialization sequence as flexible as possible MC7x112s provide a facility to control execution of the initialization commands stored in NVRAM. Command execution can be suspended for a specific period of time, or until various internal or external conditions are satisfied. This is useful for coordinating IC startup with external processes on the user's controller board, or to execute simple motion sequences prior to normal operation.

The execution conditions that can be used to control initialization are: delay a specified amount of time, compare against the Event Status register, compare against the Activity Status register, compare against the Drive Status register, or compare against the Signal Status register. These conditions allow initialization command sequences such as "Rotate the motor after the *Index* signal goes high," and "Change the profile torque once the profile velocity has reached zero."

Initialization control is provided by a host command called **ExecutionControl**.

14.4 Related Host Commands

The table below lists the commands discussed in this chapter.

Function	Host Command Mnemonic	Range & Description
Initialization control settings	ExecutionControl	Two parameter command. The first is a bit-encoded word that specifies the execution condition as well as the scaling of the timeout delay. The second is either the timeout delay value or selection & sense masks, depending on the execution condition selected in the first parameter.

Function	Host Command Mnemonic	Range & Description
Write to NVRAM	NVRAM	This command provides multiple functions including how to enter NVRAM write/erase mode, how to define beginning and ending NVRAM block addresses, and more.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

14.5 Output Signal Status During Power-up

The following table summarizes the MC7x112 IC output signal states during power up and after power-up when no initialization data is stored in the NVRAM.

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	State during power-up	State after power up
PWMHighA/PWMMagA	56	49	Tri-stated	Tri-stated
PWMLowA/PWMSignA	55	48	Tri-stated	Tri-stated
PWMHighB	54	47	Tri-stated	Tri-stated
PWMLowB	53	46	Tri-stated	Tri-stated
PWMHighC	51	45	Tri-stated	Tri-stated
PWMLowC	50	44	Tri-stated	Tri-stated
AmplifierEnable	3	22	Pulled high	Driven low
FaultOut	52	40	Tri-stated	Driven low
SrlXmt	27	24	Pulled high	Pulled high
SPIXmt	34	30	Pulled high	Driven low
~HostInterrupt	46	43	Pulled high	Driven high

If configuration data has been stored in the NVRAM then the final power-up condition of various outputs signals may be affected. See the detailed description of the specific commands that are stored into the NVRAM for details.

14.6 Signal Processing

14.6.1 Reset

MC7x112 ICs require various conditions to be present on the *Reset* pin for proper reset and power-up.

The table below shows the *Reset* pin location:

Pin Name	64-Pin TQFP Pin#	56-Pin VQFN Pin#	Description
Reset	7	5	Reset input

14.6.2 Typical Reset Processing Circuitry

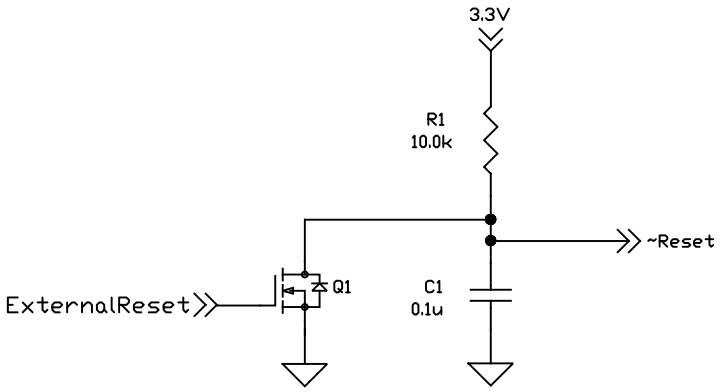


Figure 14-1: Typical Reset Processing Circuitry

Figure 14-1 shows a typical reset circuit for the MC7x112 torque control ICs. Although included in this circuit, external control of the *Reset* signal is not required since the IC will trigger an internal reset upon power up. The *~Reset* pin is driven low by the MC7x112 IC under a power-on or reset condition. If external reset is implemented an open-drain device is used. If no external reset is implemented, Q1 from the above circuit is eliminated.

15. Host Communication

In This Chapter

Communication via the MC7x112's serial port is used to specify the torque command value, to monitor the status of the MC7x112 IC and the motor, or both. MC7x112 ICs provide point-to-point asynchronous serial communications.

Serial communications to and from the MC7x112 IC are in the form of packets. A packet is a sequence of transfers to and from the host, resulting in an action or data transfer. Packets may consist of a command with no data (dataless command), a command with associated data that are written to the chipset (write command), or a command with associated data that are read from the chipset (read command).

Every command sent by the host has a structured format that does not change, even if the amount of data and nature of the command vary. Each command has an instruction word (16 bits) that identifies the command. There may be zero or more words of data associated with the command that the host writes to the IC. This is followed by zero or more words of data that the host reads from the IC. The type of command determines whether there are data written to the MC7x112 and to the host.

Most commands with associated data have one, two, or three words of data. See the *Juno Velocity & Torque Control IC Programming Reference* for more information on the length of specific commands. If a read or a write command has two words of associated data (a 32-bit quantity), the high word is loaded/read first, and the low word is loaded/read second.

Design Note: While some users will develop their own low-level libraries for host command interfacing to a MC7x112 IC, PMD's C-Motion libraries provide convenient C-language APIs (Application Programmmer Interface) for all MC7x112 commands.

15.1 Serial

The serial port may be configured to operate at baud rates ranging from 1,200 baud to 460,800 baud.

The following table shows the MC7x112 default serial settings:

Communications Mode	Default Value
point-to-point serial	57.6K, 1 start bit, 1 stop bit, no parity.

The basic unit of serial data transfer (both transmit and receive) is the asynchronous frame. Each frame of data consists of the following components.

- One start bit.
- Eight data bits.
- An optional even/odd parity bit.
- One or two stop bits.This data frame format is shown in the following figure.

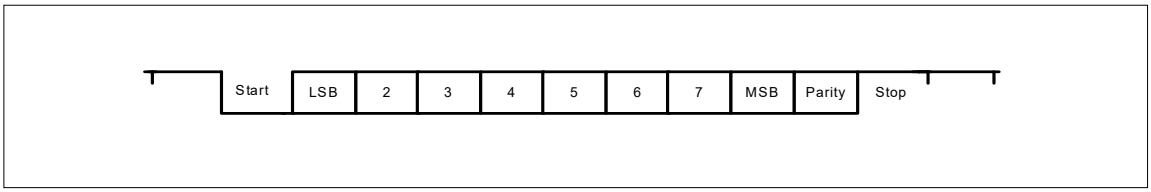


Figure 15-1: Typical Data Frame Format

15.1.1 Command Format

The command format that is used to communicate between the host and MC7x112 IC consists of a command packet sent by the host processor, followed by a response packet sent by the MC7x112. The host must wait for, receive, and decode the response packet.

Command packets sent by the host contain the following fields.

Field	Byte#	Description
Address	1	This field should always be zero.
Checksum	2	One byte value used to validate packet data. See Section 15.1.3, "Checksums" for detailed information.
Instruction code	3–4	Two byte instruction, the upper byte of which (sent first) is zero, followed by the lower byte which contains the command code. Command codes are detailed in the *Juno Velocity & Torque Control IC Programming Reference*.
Data	5 +	Zero to six bytes of data, sent most significant byte (MSB) first. See the individual command descriptions for details on data required for each command.

In response to the command packet, the MC7x112 will respond with a packet in the following format.

Field	Byte#	Description
Status	1	Zero if the command was completed correctly; otherwise, an error code specifying the nature of the error. See Section 15.1.2, "Instruction Errors" for more information.
Checksum	2	A one-byte checksum value used to validate the packet's integrity. See Section 15.1.3, "Checksums" for details.
Data	3 +	Zero to six bytes of data. No data will be sent if an error occurred in the command (i.e. the status byte was non-zero). If no error occurred, then the number of bytes of data returned would depend on the command to which the MC7x112 was responding. Data are always sent MSB first.

15.1.2 Instruction Errors

There are a number of checks made by the MC7x112 IC on commands it receives. These checks improve safety of the motion system by eliminating some obviously incorrect command data values. All such checks associated with host commands are referred to as instruction errors. The status byte in the response packet will contain one of the error codes. For more information see Section 11.1.1.1, "Instruction Errors."

15.1.3 Checksums

Both command and response packets contain a checksum byte. The checksum is used to detect transmission errors, and allows the MC7x112 to identify and reject packets that have been corrupted during transmission or were not properly formed.

Checksums are mandatory when using serial communications. Any command packets containing invalid checksums will not be processed and will result in a data packet being returned containing an error status code.

The serial checksum is calculated by summing all bytes in the packet (not including the checksum) and negating (i.e., taking the two's complement of) the result. The lower eight bits of this value are used as the checksum. To check for a valid checksum, all bytes of a packet should be summed (including the checksum byte), and if the lower eight bits of the result are zero, then the checksum is valid.

Example

If a command packet is sent containing command 077h (**SetMotorCommand**) with the one-word data value 1234h, then the checksum will be calculated by summing all bytes of the command packet (00h + 77h + 12h + 34h = BDh) and negating this to find the checksum value (43h). On receipt, the MC7x112 will sum all bytes of the packet, and if the lower eight bits of the result are zero, then it will accept the packet (00h + 43h + 77h + 12h + 34h = 100h).

15.2 Serial Host/IC Synchronization

There is no direct way for MC7x112s to determine the beginning of a new serial command packet, except by context. Therefore, it is important for the host to remain synchronized with the MC7x112 when sending and receiving data. To ensure that the processors remain synchronized, it is recommended that the host implement a time limit when waiting for data packets to be sent by the MC7x112. The suggested minimum timeout period is the amount of time required to send one byte at the selected baud rate plus one millisecond. For example, at 9600 baud each bit takes 1/9600 seconds to transfer, and a typical byte consists of 8 data bits, 1 start bit, and 1 stop bit. Therefore, one byte takes just over 1 millisecond, and the recommended minimum timeout is 2 milliseconds.

If the timeout period elapses between bytes of received data while the host is waiting on a data packet, then the host should assume that it is out of synchronization with the MC7x112. To resynchronize, the host should send a byte containing a value of zero and wait for a data packet to be received. This process should be repeated until a data packet is received from the MC7x112; at which point the two processors will be synchronized.

15.3 Related Host Commands

In the table below both the 'Set' and 'Get' commands may be used with host commands, while only Set commands are used in NVRAM startup command sequences:

Parameter	Host Command Mnemonic	Description
Serial port settings	SetSerialPortMode GetSerialPortMode	Specifies all serial port parameters including baud rate, parity, and # stop bits. The MC7x112's default serial port settings are 57.6 Kbaud, 1 stop bit, no parity.

For more information on host commands refer to the *Juno Velocity & Torque Control IC Programming Reference*.

15.4 Signal Processing

All of the MC7x112's serial-related signals are digital TTL level signals, so for typical cable-based use of serial communication external transceiver chips are used.

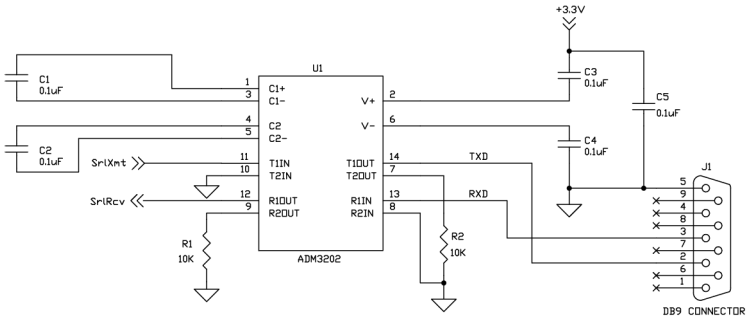
The table below provides a summary of the signals used with serial communications:

Pin Name	64-Pin TQFP Pin #	56-pin VQFN Pin #	Description
SrlRcv	32	28	Serial receive inputs the serial data
SrlXmt	27	24	Serial transmit outputs the serial data

15.4.1 Typical Processing Circuitry

Figure 15-2 shows a typical circuit for interfacing the MC7x112's serial pins to an RS232 transceiver. RS232 is typically used for off-board point to point serial communications. For serial communications between on-board devices such as a microcontroller connected to an MC7x112 IC RS232 transceivers are generally not needed.

Figure 15-2: Example RS232 Transceiver Interface Schematic



A. Reference

A

In This Appendix

- Traceable Variables
- MC7x112 Default Values
- Setting Biquad Filter Values

A.1 Traceable Variables

The table below lists all of the MC7x112 variables that can be traced via the Trace facility detailed in Section 11.7, "Trace."

Trace Variable Name	Variable ID Code
Ramp Generator	
Commanded Current	3
Commanded Ramp Rate	4
Encoder	
Actual Position	5
Position Capture Register	9
Phase Angle	15
Phase Offset	16
Status Registers	
Event Status Register	12
Activity Status Register	13
Signal Status Register	14
Drive Status Register	56
Drive Fault Status Register	79
Commutation/Phasing	
Active Motor Command	7
Phase A Command	17
Phase B Command	18
Phase C Command	19
Phase Angle Scaled	29
A/B Current Control	
Phase A Reference	66
Phase A Error	30
Phase A Actual Current	31
Phase A Integrator Sum	32
Phase A Integrator Contribution	33
Current Loop A Output	34
Phase B Reference	67
Phase B Error	35
Phase B Actual Current	36
Phase B Integrator Sum	37
Phase B Integrator Contribution	38
Current Loop B Output	39
D Feedback	42
Q Feedback	48
Leg A Current	69
Leg B Current	70
Leg C Current	71
Field Oriented Control	
D Reference	40
D Error	41
D Feedback	42
D Integrator Sum	43
D Integrator Contribution	44

Trace Variable Name	Variable ID Code
D Output	45
Q Reference	46
Q Error	47
Q Feedback	48
Q Integrator Sum	49
Q Integrator Contribution	50
Q Output	51
FOC Alpha Output	52
FOC Beta Output	53
Phase Alpha Actual Current	73
Phase Beta Actual Current	74
Drive	
Bus Voltage	54
Temperature	55
Foldback Energy	68
Bus Current Supply	86
Bus Current Return	87
PWM Output A	75
PWM Output B	76
PWM Output C	77
Analog Inputs	
CurrentA	20
CurrentB	21
CurrentC	22
Temperature	24
Bus Current Supply	25
Bus Voltage	26
AnalogCmd	27
Miscellaneous	
None (disable variable)	0
Motion Control IC Time	8
Analog Command Biquad Input	101
AnalogCmd Signal	103
SPI Direct Input	105

A.2 MC7x112 Default Values

The table below shows the default settings of MC7x112 registers and parameters. Note that if an NVRAM initialization script has been loaded which re-programs these settings then those programmed values will become the default settings after power-up or reset.

	Default Value
Interrupts	
Interrupt Mask	0
Commutation	
Commutation Mode	Encoder-based
Phase Angle	0
Phase Counts	1
Phase Denominator	1
Phase Offset	–1
Phase Initialize Mode	Algorithmic
Phase Initialize Ramp Time	0
Phase Initialize Negative Pulse Time	0
Phase Initialize Positive Pulse Time	0
Phase Initialize Ramp Command	0
Phase Initialize Pulse Command	0
Phase Correction Mode	Index signal
Current Control	
Current Control Mode	FOC
FOC Kp (Both D & Q Loops)	0
FOC Ki (Both D & Q Loops)	0
FOC Integrator Sum Limit	0
A/B Kp (Both A & B Loops)	0
A/B Ki (Both A & B Loops)	0
A/B Integrator sum	0
Current Limit	16383
Encoder	
Actual Position	0
Encoder Source	Quadrature incremental
Motor Output	
Operating Mode	0001h
Output Mode	None
Motor Type, MC71112	DC Brush
Motor Type, MC73112	3-phase Brushless DC
PWM Frequency	20 kHz
PWM Limit	16384
PWM Dead Time	16879 *must be changed*
PWM Signal Sense	80FFh
PWM Refresh Period	1
PWM Refresh Time	32767 *must be changed*
PWM Current Sense Time	32767 *must be changed*
Position Servo Loop Control	
Sample Time	102

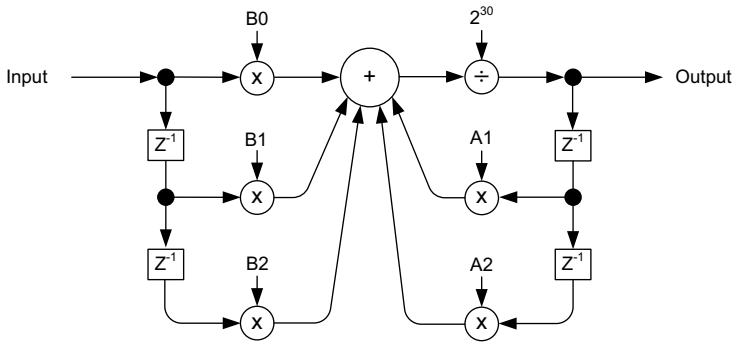
	Default Value
Ramp Generation	
Target Current	0
Current Increase Rate	0
Current Decrease Rate	0
RAM Buffer	
Buffer Length	buffer 0 12,288 bytes buffer 1 16,384 bytes
Buffer Read Index	0
Buffer Start	buffer 0 00000000h buffer 1 20000000h others 0
Buffer Write Index	0
Safety	
Current Foldback Event Action	Disable Motor Output
Encoder Position Capture	No Action
Overtemperature Event Action	Disable Motor Output
Disabled Event Action	Disable Motor Output
Overcurrent Event Action	Disable Motor Output
Overvoltage Event action	Disable Motor Output
Undervoltage Event Action	Disable Motor Output
Watchdog Timeout Event Action	No Action
Brake Signal Event Action	Passive braking
OvervoltageThreshold	65535
Undervoltage Threshold	0
OvertemperatureThreshold	32767
i^2T Continuous Current Limit	32768
i^2T Energy Limit	32768
Status & Mask Registers	
Signal Sense Register	0000h
FaultOut Mask	0600h
Trace	
Trace Mode	one-time
Trace Period	1
Trace Start	Immediate
Trace Stop	Immediate
Trace Variables	all are 0
Miscellaneous	
Serial Port Settings	57600 baud, 8 data bits, 1 stop bit

A.3 Setting Biquad Filter Values

A biquad is a generic digital filter structure. With proper coefficients it can be programmed to be a low-pass filter, high-pass filter, band-pass filter, notch filter, or custom filter. Programs such as Octave (www.octave.org) may be used to find the coefficients. In addition, Pro-Motion, PMD's motion application development software package can perform similar calculations for various programmable filter types.

The MC7x112 ICs support a biquad filter on the *AnalogCmd* signal input. The form of the MC7x112 biquad filter is shown below:

Figure A-1: Biquad Calculation Flow



The output of the filter at time *n* is determined with the following equation:

$$Y_n = \frac{1}{2^{30}} \times (B_0 \times X_n + B_1 \times X_{n-1} + B_2 \times X_{n-2} + A_1 \times Y_{n-1} + A_2 \times Y_{n-2})$$

where:

Yn = output of the filter at time n

Xn = input to the filter at time n

B0 = programmable biquad coefficient

B1 = programmable biquad coefficient

B2 = programmable biquad coefficient

A1 = programmable biquad coefficient

A2 = programmable biquad coefficient

The following table shows biquad filter coefficients that are set using the command **SetLoop**. They can be read back using the command **GetLoop**. The following table summarizes the representation and range for the settable biquad parameters:

Term	Name	Representation & Range
A1	Coefficient A1	signed 32 bits (–2,147,483,648 to +2,147,483,647)
A2	Coefficient A2	signed 32 bits (–2,147,483,648 to +2,147,483,647)
B0	Coefficient B0	signed 32 bits (–2,147,483,648 to +2,147,483,647)
B1	Coefficient B1	signed 32 bits (–2,147,483,648 to +2,147,483,647)
B2	Coefficient B2	signed 32 bits (–2,147,483,648 to +2,147,483,647)

A.3.1 Determining & Programming Biquad Coefficients

In the example below we use Octave to generate biquad coefficients for a specific filter type and then convert to chip units.

Example: What are the motion IC biquad coefficient settings for a 2nd-order lowpass Butterworth filter with a cut-off frequency of 300 Hz?

In Octave this filter function is calculated using a command of form:

[b,a] = butter(FO, CF)

Where FO (filter order) = 2, and CF (cutoff frequency) is calculated using:

CF = Filter cutoff in Hz/ (MC7x112 cycle rate / 2) = 300 Hz / (10,000 Hz / 2) = .06

Prompting Octave with [b,a] = butter(2, 0.06) generates the following values for the biquad coeficients:

B0 = .0081756

B1 = .016351

B2 = .0081756

A1 = -1.7284

A2 = 0.7611

If the motion control IC's filter equation (shown at the top of this page) is compared to the filter equation used by Octave (type help filter in Octave), there is a difference in that the Ax components are subtracted in Octave as opposed to being added in the motion control IC. The result is that the A1 and A2 coefficients from Octave (or Matlab) need to be multiplied by –1 before being sent to the motion control IC.

The table below shows the negated coefficient values as well as the corresponding values that are programmed into the MC7x112. These programmed values are determined by multiplying the floating point values by 2^{30}.

Coefficient	Floating point value	MC7x112 Programmed Value
B0	.0081756	8,778,484
B1	.016351	17,556,753
B2	.0081756	8,778,484
A1	1.7284	1,855,855,369
A2	-0.7611	-817,224,902

This page intentionally left blank.

B. Application Notes

In This Chapter

B.1 General Design Notes

In the subsequent sections please note that unless otherwise noted MC71112 refers to both the MC71112 and MC71112N ICs, and MC73112 refers to both the MC73112 and MC73112N ICs.

Logic functions presented in the example schematics are implemented by standard logic gates. In cases where specific parameters are of significance (propagation delay, voltage levels, etc.) a recommended part number is given.

In the schematics, pins with multiple functions are referenced by the name corresponding to the specified functionality. For example, pin 56 on the MC71112 is named “PWMHighA/PWMMagA” but will be referenced by the name “PWMHighA” in the PWM High/Low motor drive schematic example and “PWMMagA” in the sign/magnitude motor drive schematic example.

The schematic designs presented in this chapter are accurate to the best of PMD’s knowledge. They are intended for reference only and have not all been tested in hardware implementations.

B.1.1 Interfacing to Other Logic Families

When integrating different logic families, consideration should be given to timing, logic level compatibility, and output drive capabilities. The MC71112 and MC73112 are 3.3V CMOS input/output compatible and cannot be directly interfaced to 5V CMOS components. In order to drive a 5V CMOS device, level shifters from the 5V CMOS AHCT (or the slower HCT) families can be used. When using a 5V CMOS component to drive the CP, a voltage divider may be used or a member from the CMOS 3.3V LVT family may serve as a level shifter.

B.2 Design Tips

B.2.1 Controlling PWM Output During Reset

When the MC71112 or MC73112 are in a reset state (when the reset line is held low), or immediately after a power on, PWM output will be in a high impedance state, which will provide design flexibility to prevent undesirable motor movement at system level. For example, when the power train is active high in PWM High/Low mode, pull-down resistors can be used to keep the power train off during reset and power up. For an active low power stage, pull-up resistors can be used.

B.2.2 Thermal Considerations

The recommended operating junction temperature range for the MC71112 and MC73112 is between -40°C and 150°C. Proper thermal design will ensure the system reliability. Based on a simplified resistor model for heat transfer, following thermal matrices under different conditions are provided for thermal design.

AIR FLOW (64-Pin TQFP Package)

Parameter	0 lfm	150 lfm	250 lfm	500 lfm
θ_{JA}[°C/W] High k PCB	56.5	44.7	42.9	40.3
ψ_{JT} [°C/W]	0.15	0.42	0.51	0.67

AIR FLOW (56-Pin VQFN Package)

Parameter	0 lfm	150 lfm	250 lfm	500 lfm
θ_{JA}[°C/W] High k PCB	34.8	23.6	22.3	20.5
ψ_{JT} [°C/W]	0.24	0.36	0.43	0.56

θ_{JA} is the junction-to-ambient thermal resistance. Although it is an important design reference, this thermal metric highly depends on the board design and system configurations. Directly using it for junction temperature estimation could result in misleading results because the environmental factors are different from design to design.

ψ_{jT} (junction to top of package) is provides as an useful thermal metric for estimating the in-situ junctional temperature. The environmental factors do not affect this metric as much, and it can be easily measured. Also, because ψ_{jT} is small, if a user chooses to, the top of package temperature might be approximated as the junctional temperature for design estimation when enough thermal design margin is included.

This page intentionally left blank.

B.3 Power Supplies

In the schematic shown in Figure B-1 the design is powered by an external +5VCC power source. The MC71112 and MC73112 require a 3.3V supply input. +3.3Vs, the 3.3V digital supply, is generated by the TPS76733QPWPRG4, a 1.0 Amp fixed 3.3V low-dropout voltage regulator.

If the MC71112 or MC73112 analog inputs are used it should be supplied with a filtered +3.3Vs supply.

The following is the list of supplies which are referenced in the example schematics within this application notes section of the manual:

- +3.3Vs and +3.3V_Analog: +3.3Vs is the main digital supply for the MC71112 and MC73112 devices. +3.3V_Analog is the filtered version of the +3.3Vs supply for ADC and its related conditioning circuitry. The extra filtering is used to provide additional decoupling of the analog elements from the digital elements in the circuitry.

Notes:

- The power supplies schematic provided in Figure B-1 is for reference only, and is designed only to meet the requirements of the example schematics used in the application notes section of the manual. The actual supplies used should be designed according to the stability and precision requirements of the application.
- Power supplies for the motor drive amplifiers/switchers are not shown. Care should be taken when designing these power supplies, as they should be capable of sinking high switching currents.

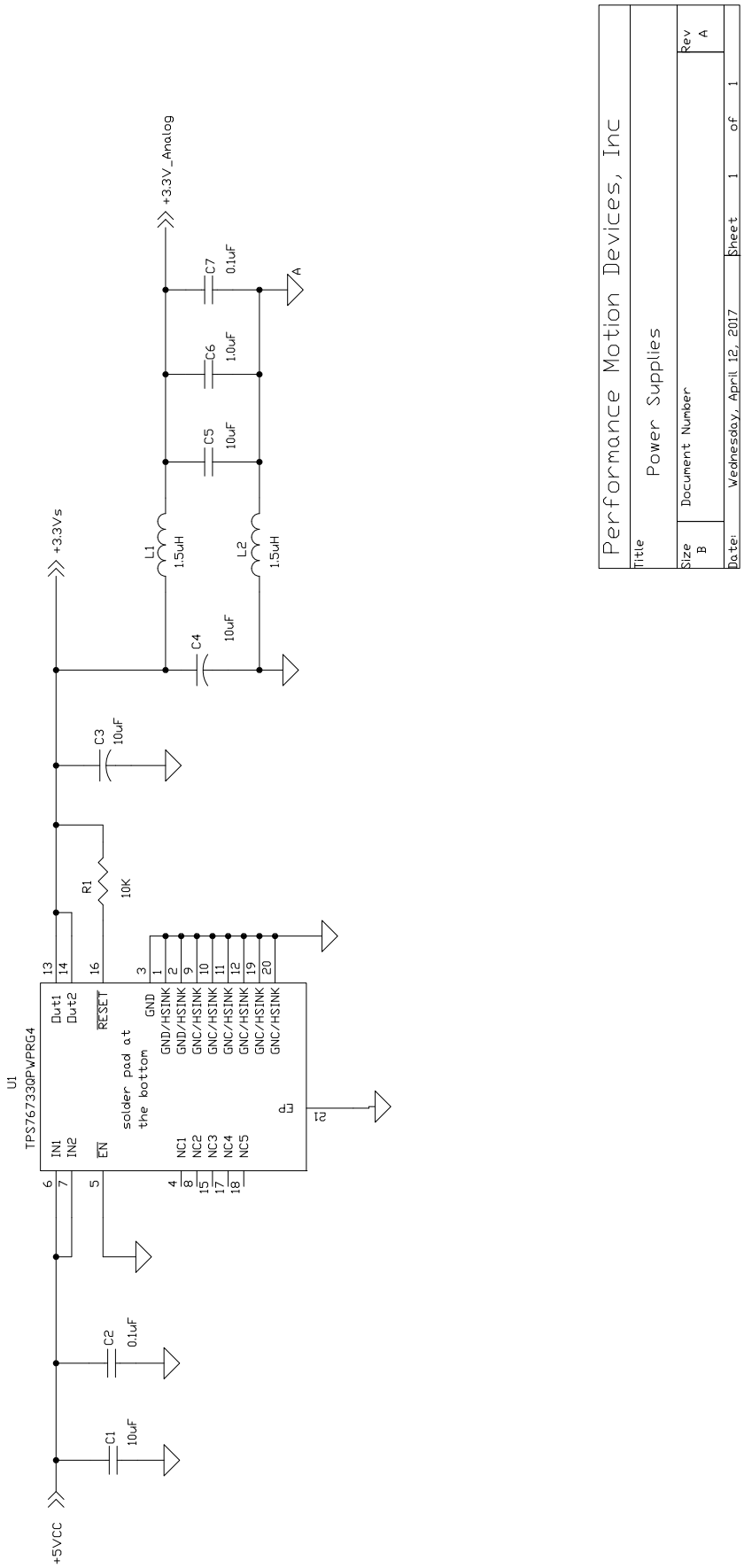


Figure B-1: Basics, Power Supplies, MC71112 and MC73112

B.4 Clock Generator, Grounding and Decoupling

B.4.1 Grounding and Decoupling

As shown in Figure B-2, each of the digital supply voltage pins should be connected to the +3.3 Vcc. A minimum of 1.2μF capacitor should be used to decouple each Vcc pin. A 2.2μF ceramic capacitor is recommended. If the +3.3 Vcc source is noisy, additional ferrite bead can be placed in series with the decoupling cap to form a LC filtering network on the power pin.

Each of the "FltCap" pins should be connected to a minimum of 1.2μF filtering capacitor which in turn connects to ground. A 2.2μF ceramic capacitor is used in the schematic. The filtering capacitors must be placed as close as possible to each one of the "FltCap" pins. This general rule applies to all analog and digital components, although in some of the schematics that follow these capacitors are not shown for reasons of brevity. In some cases, especially for analog processing circuitry, it may be beneficial to run a separate power line from the power supply to the component in order to prevent power supply fluctuations from impacting low-level signal components.

The same points should be considered when designing the ground. A good board layout practice should have a star connection at one point in the power supply.

Additional filtering, such as ferrite beads, may be inserted between the analog and digital grounds to suppress high frequency ground noise. Some components, such as motor drivers, require special grounding. The system designer should refer to the component data sheets of selected components in order to ensure correct usage of the grounding methods.

B.4.2 Decoupling of the On-chip ADC

The voltage supply to the ADC should be decoupled with a 2.2μF ceramic capacitor (typical) on the pin. It should be placed as close as possible to the ADC power supply input pins.

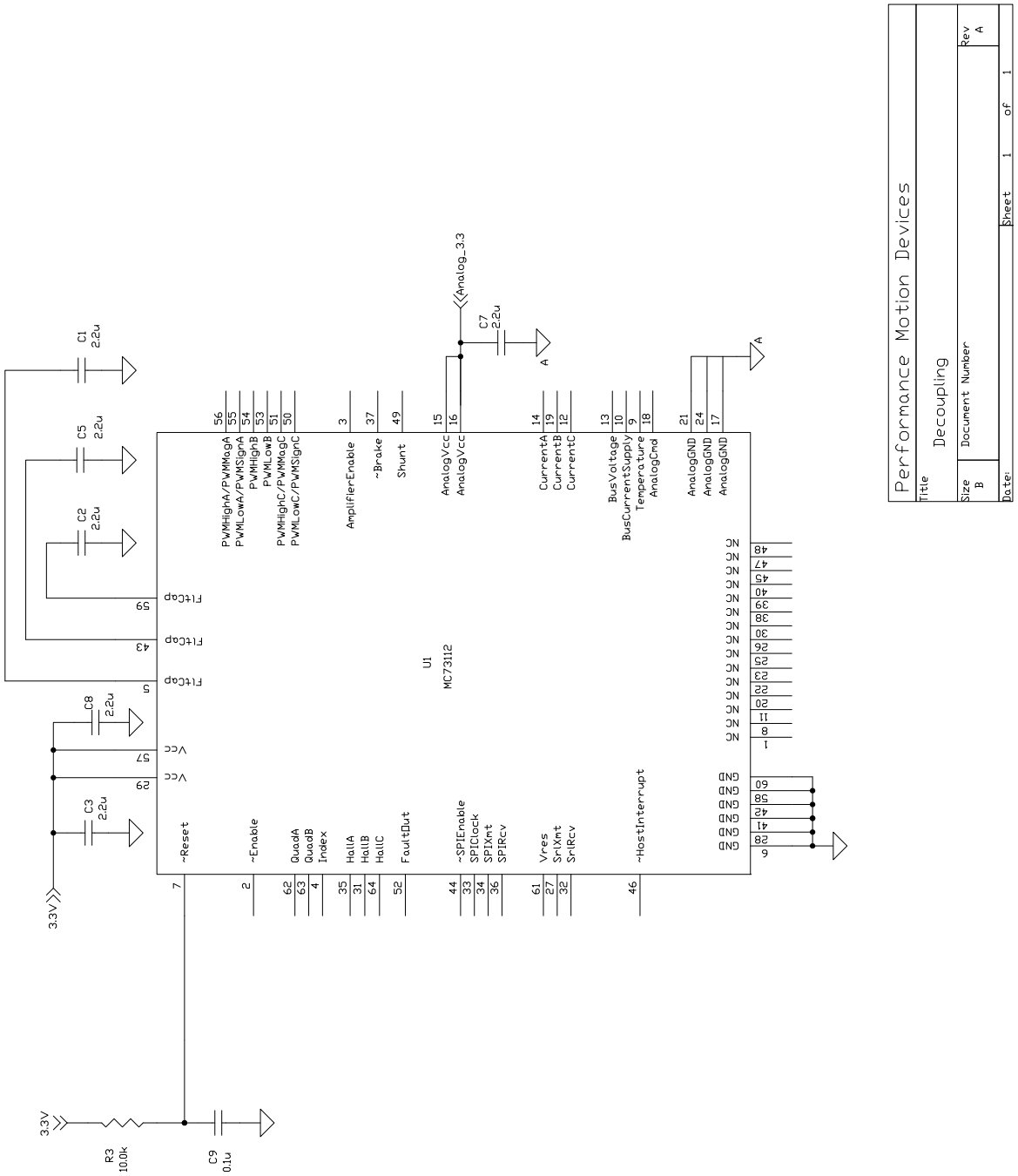


Figure B-2: Basics, Clock and Bypass Caps, MC71112 and MC73112

B.5 Reset Signal

The MC71112 and MC73112 chips have a built-in power supervisory circuitry that generates an internal reset signal when a power-on or brown-out condition occurs. As such, no external circuitry is needed to generate a reset input pulse. An R-C circuit must be connected to this pin for noise immunity reasons.

The supervisor reset release delay time is typically 600 μsec after the power-on or brown-out condition event is removed. If need be, an external circuit may also drive this pin to assert a device reset. In this case, it is recommended that this pin be driven by an open-drain device.

All digital output signals have an internal pullup except for *FaultOut* and the *PWM* signals. See Section 14.5, "Output Signal Status During Power-up" for more information. If the *AmplifierEnable* and *FaultOut* signals are used they should have an external pull down resistor to prevent any glitches during reset.

During the initialization period, the MC71112 and MC73112 are configured with the default initialization parameters for communication, analog offsets, control loop gains etc. If necessary, the on chip non-volatile storage can be used to store user-programmed initialization parameters. For more information see Section 6.1, "Loading the NVRAM."

B.6 Analog Input

MC7x112s support an analog command signal. The following example schematic in Figure B-3 illustrates the analog conditioning circuit.

The MC7x112 is equipped with a 12-bit ADC. The ADC converts from 0 to 3.3V fixed full scale range. The ADC power supply should be decoupled with a low-ESR capacitor. For example, place a 2.2uF ceramic capacitor or a 2.2-6.8 μF tantalum capacitor in parallel with a 0.01-0.1 μF ceramic capacitor as closely as possible to the power supply and ground pins.

The digital value for analog command signal is determined using the following formula.

Digital value = 0 when input = Vref

Digital value = 32,768 x (input voltage- Vref)/3.3V when 0V < = input < = 3.3V

where Vref is 1.65V typical, and the user can set offset value to compensate the offset in the circuit.

Figure B-3 shows a differential signal conditioning circuit for +/-10V analog command signal. The analog command signal is assumed to be differential, AnalogCmd+ and AnalogCmd-. For single-ended input signal, connect the input signal to AnalogCmd+ and the AnalogCmd- end to ground. The same circuit can be used for tachometer analog feedback signal conditioning.

The differential amplifier stage scales the input signal to the range of 0 and 3.3V as

$AnalogCmd = G(AnalgogCmd^{+} - AnalogCmd^{-}) + 1.65V$

For +/-10V input and Vref as 1.65V, the gain is G = 0.165.

When R1 = R5, and R2 = R4, its DC gain is

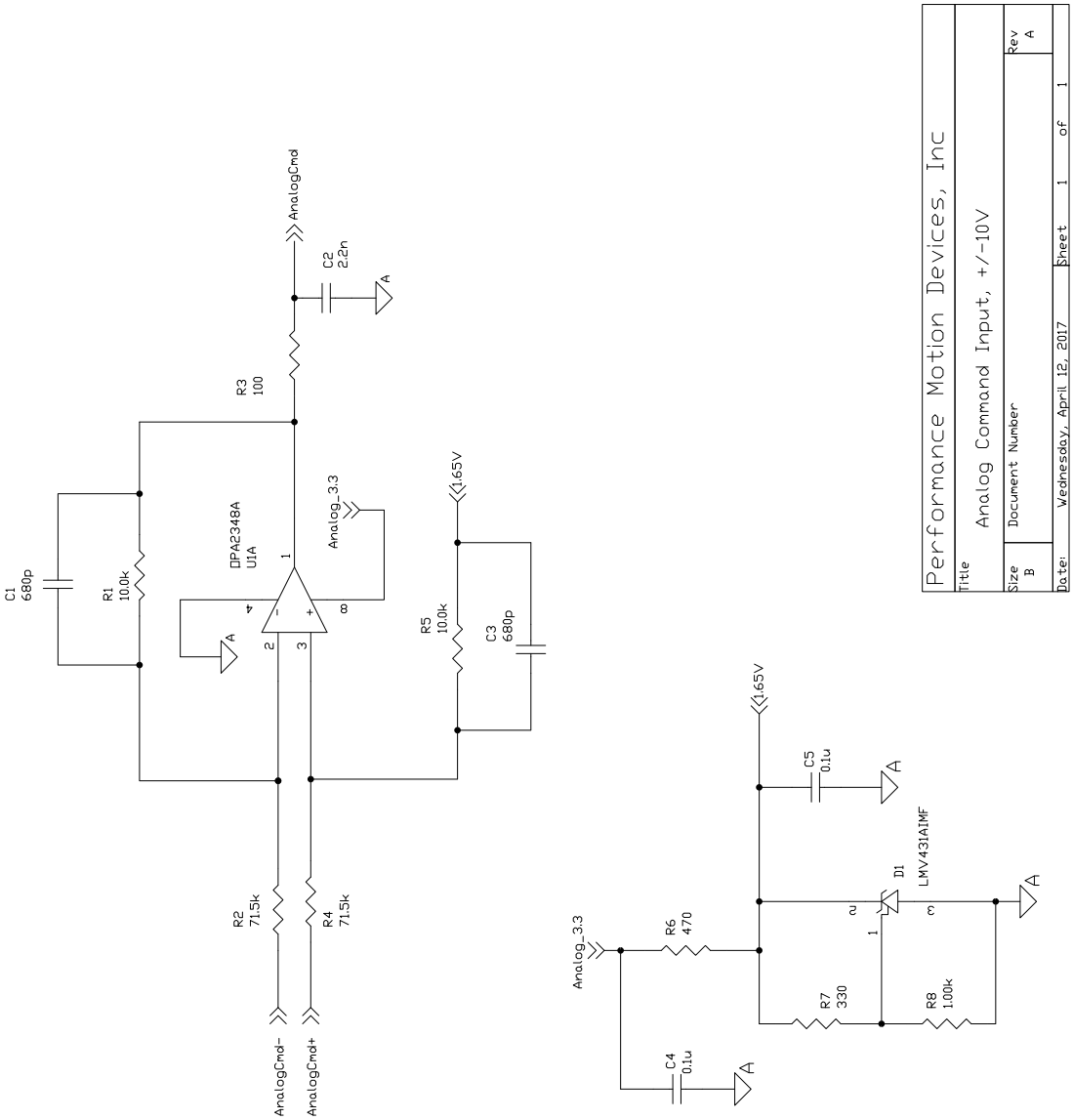
$AnalogCmd = (R1/R2)(AnalgogCmd^{+} - AnalogCmd^{-}) + 1.65V$

When R1 = R5 = 10.0 kohms and R2 = R4 = 71.5kohms, the gain is 0.14, which is slightly lower than 0.165 to accommodate offsets and tolerances.Please use 1% or better grade resistors for better common mode performance

C1 and C3 set the bandwidth of the filter. With C1 = C3, the bandwidth is 1/(2*pi*R1*C1). For R1 = 10.0kohms and C1 = 680pF, the bandwidth is 23.4 kHz.

The low-pass RC filter of R3 and C2 limits the load on the opamp. C2 should be placed close to the ADC input pin, as it partially drives the sample capacitor of the ADC.

Figure B-3: Analog Input Interface



B.7 Drive-Related Safety and Monitoring Features

This example shows the motor drive-related analog monitoring features. Please refer to Section B.9, "PWM High/ Low Motor Drive with Leg Current Sensing/Control" for leg current sensing functions.

The block of R9, R10, R13 and C5 is for temperature sensing. R13 is a thermistor, and its resistance depends on the temperature. MC71112 and MC73112 will sense the scaled voltage and convert it into temperature reading. C5 need to be tied close to the Temperature pin to improve noise immunity.

In this example, it is assumed that the thermistor is away from the MC71112 and MC73112 and close to the power train, which usually has the highest temperature. Accordingly, C5 is referred to the analog ground, and R13 to digital ground. R10 is optional to improve the noise immunity. If R13 is close to the analog portion, Vcc can be AnalogVcc instead and R13 be tied to the analog ground.

The block of R2, R3, R5 and C3 is for input voltage sensing. This voltage divider will scale the +HV into the range between ground and 3.3V. In this example, it will scale 63V to 3.3V. C3 need to be tied close to BusVoltage pin. The voltage divider is referenced to digital ground while C3 is to analog ground. An optional resistor can be put between R5 and C3 to improve noise immunity as R10 does for temperature sensing function. This block is also a low-pass filter with bandwidth of 1.5 kHz. This bandwidth should be selected to respond to real voltage fault event while attenuating bus noise.

U1 is the high side bus current sensing IC. With current sensing resistor R1 at 10mOhm, U1 has a current scaling factor of (13.3/2*0.01) = 66.5mV/A.

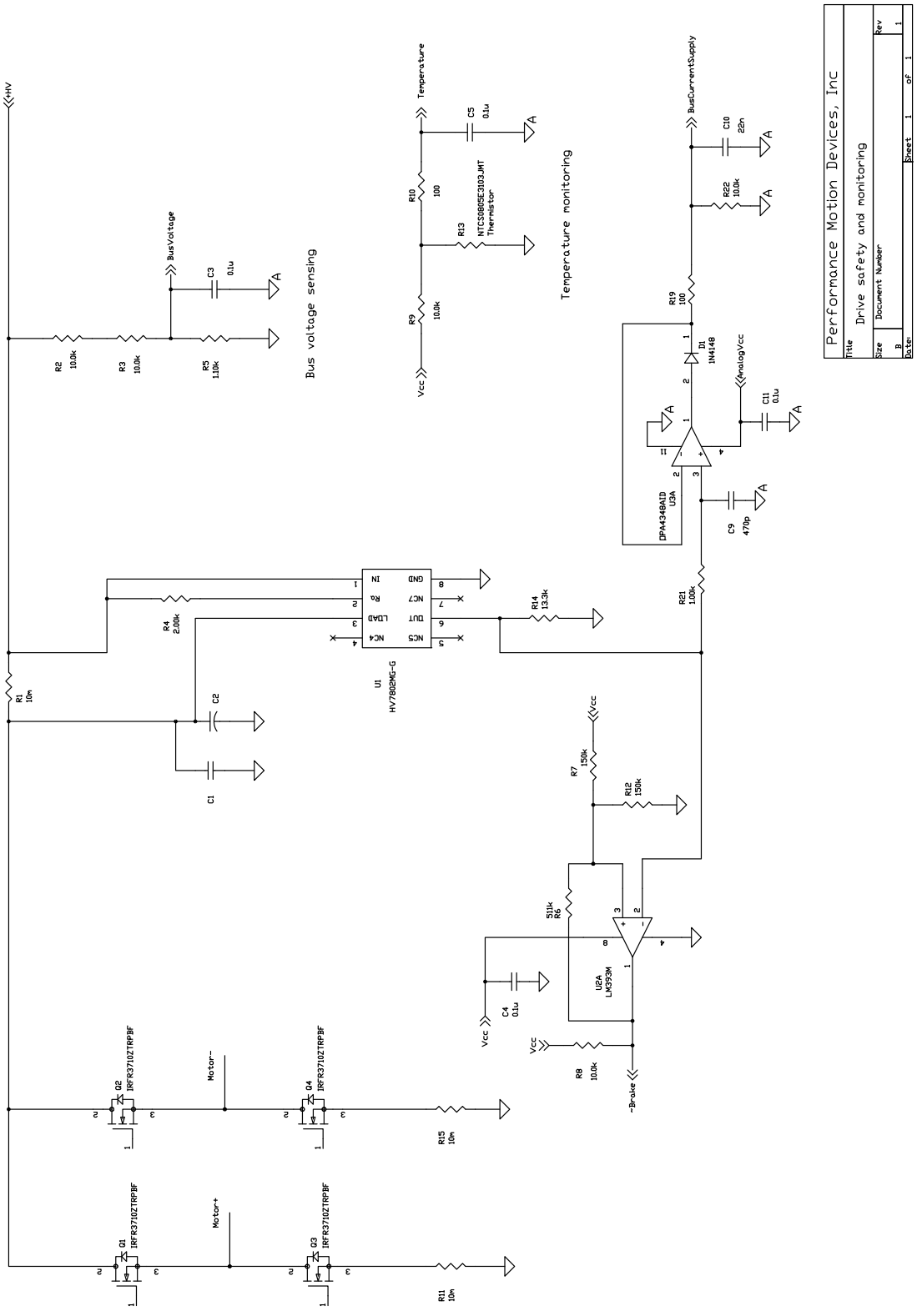
U2A is for short circuit protection. R12 and R7 set the protection trigger point, and R6 provides a hysteresis. When Vcc is 3.3V, the trigger point is 1.65V with hysteresis. Brake will go to low and trigger protection when U1 output is over 1.65V, which is (1.65/66.5mv/A) = 25A.

The output of U1 also goes to U3A, which is a peak-detection circuit. The analog input is sampled at 20 kHz. The peak-detection circuit will hold the maximum peak current reading between the sampling points so MC7x112 can detect the maximum current. The necessity of this peak-detection circuit depends on the power train design. For example, if C1 and C2 have big enough capacitance so that the current in R1 will be close to DC, U3A can be a buffer instead.

This peak detection circuit (U3A) is optional. If not used, the BusCurrentSupply pin has an internal comparator, and this comparator can also detect overcurrent.

MC71112 and MC73112 supports leg current sensing with current loop control, and R11 and R15 are leg current sensing resistors.

Figure B-4: Drive Safety and Monitoring



B.8 Shunt Regulation

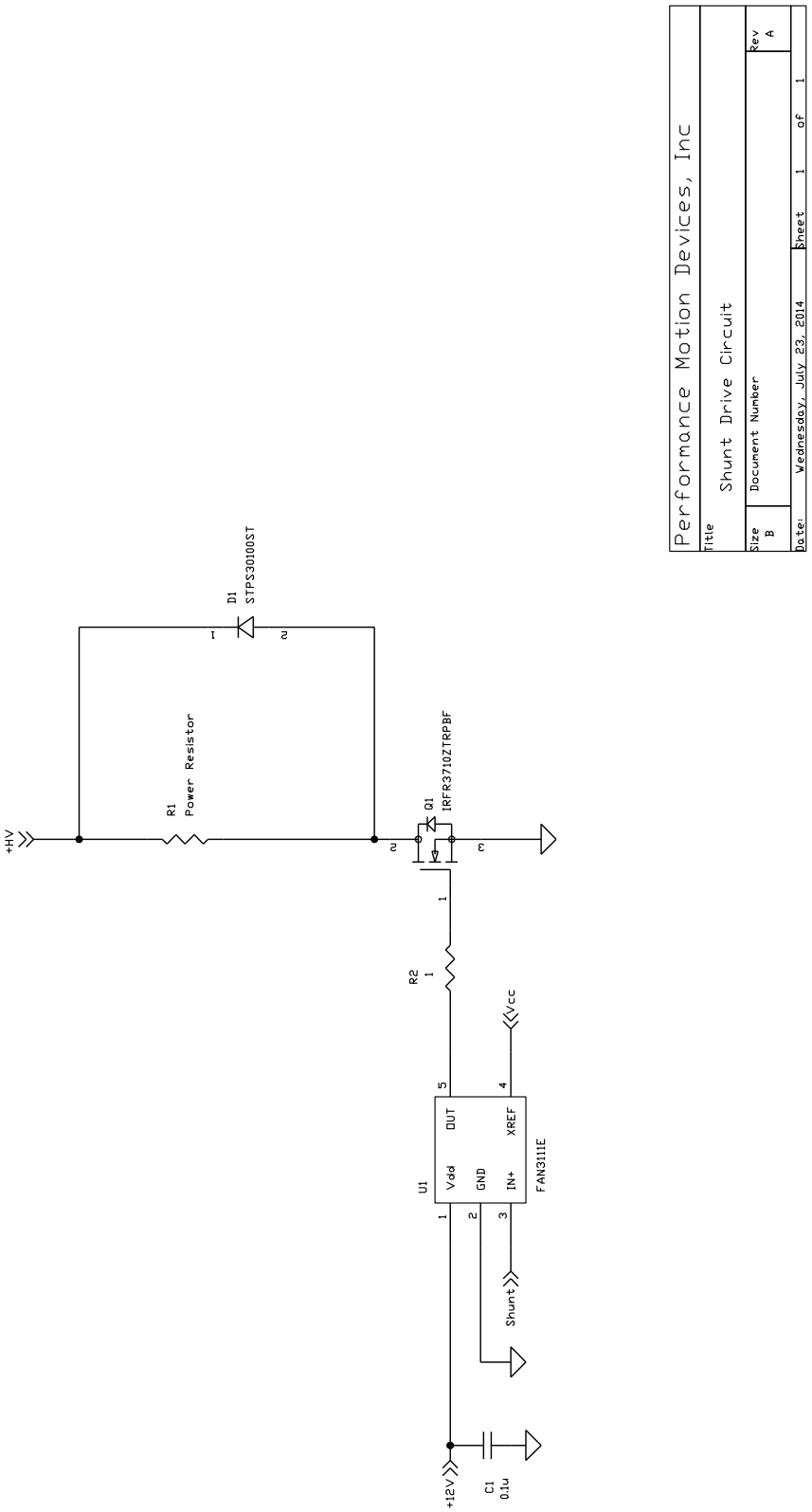
This example schematic shows a basic shunt regulation function.

The shunt signal is output by the MC7x112 IC and when so-programmed will become active with a specified PWM duty cycle when the DC Bus voltage exceeds a specified voltage threshold.

In this schematic the shunt output drives MOSFET Q1 through driver U1. When Q1 is turned on, current will flow through R1. When Q1 is turned off, D1 will provide a free-wheeling path for the current in R1 decaying to zero. Q1 and D1 should be sized to handle the current. Also, R1, Q1 and D1 should be sized to handle both the instantaneous power and average power when the shunt signal is active.

Upon power up or during reset, Shunt pin is high impedance. In this example, U1 has internal pull-down resistor to ensure Q1 being off. If a different driver is used, a pull-down or pull-up resistor might be necessary to ensure the shunt function be off for safe power up and reset.

Figure B-5: Shunt Drive Circuit



B.9 PWM High/Low Motor Drive with Leg Current Sensing/ Control

This section presents several design examples PWM high/low motor drive with leg current sensing. The examples focus on different priorities including power rating, cost, and noise immunity.

B.9.1 Leg Current Sensing

Figure B-6 shows an example for leg current sensing. Only phase A is shown here while the design for other legs are the same.

This example has two functional sections. The first is the current sensing sensor, and the second is the analog signal conditioning circuit.

In this example, the leg current sensor is a resistor, R2. Q1 and Q2 are the half-bridge power train for motor winding phaseA. Current sensing resistor R2 senses the leg current in Q2, which equals the motor winding current when Q2 conducts.

Q2 is switching at the PWM frequency, and the voltage drop on R2 is proportional to the motor winding current when Q2 is on. Therefore, the voltage signal is also a chopping signal. The signal is always sampled when Q2 is on to ensure an accurate reading. Also, the voltage drop can be positive, negative or zero depending on the winding current direction.

U1 and the passive parts are the analog conditioning circuit. It scales and filters the voltage signal on R2 and input to the analog input pin CurrentA.

U1A is configured as a differential amplifier with R3 = R5 and R1 = R6. It amplifies the voltage drop across R2, which is the differential voltage. It also attenuates the common mode noise including the noise on the power train.

D1 provide a 1.65V voltage bias source as half the 3.3V ADC range. This bias can be shared with current sensing stages of other phases. With this voltage bias, R2 current is sensed in either direction.

By default, the MC71112 and MC73112 take 1.65V reading as zero current. Host commands can be sent to compensate the error introduced by the offsets and tolerances of the current sensing circuit.

R4 and C1 is a low pass filter to reduce output noise. It also alleviates the signal glitch due to ADC sampling. R4 and C1 have to be placed close to CurrentA pin. Please note, because the signal on R2 is a chopping signal at the PWM frequency, the bandwidth of R4 and C1 should be much higher than the PWM frequency preventing signal distortion/delay.

The gain of the current sensing circuit is

$V_{CurrentA} = I\_leg * R2 * R1/R3 + 1.65$

For this example, it is

$V_{CurrentA} = I\_leg * 0.02 * 54.9/10.0 + 3.3/2 = 0.1098 * I\_leg + 1.65$

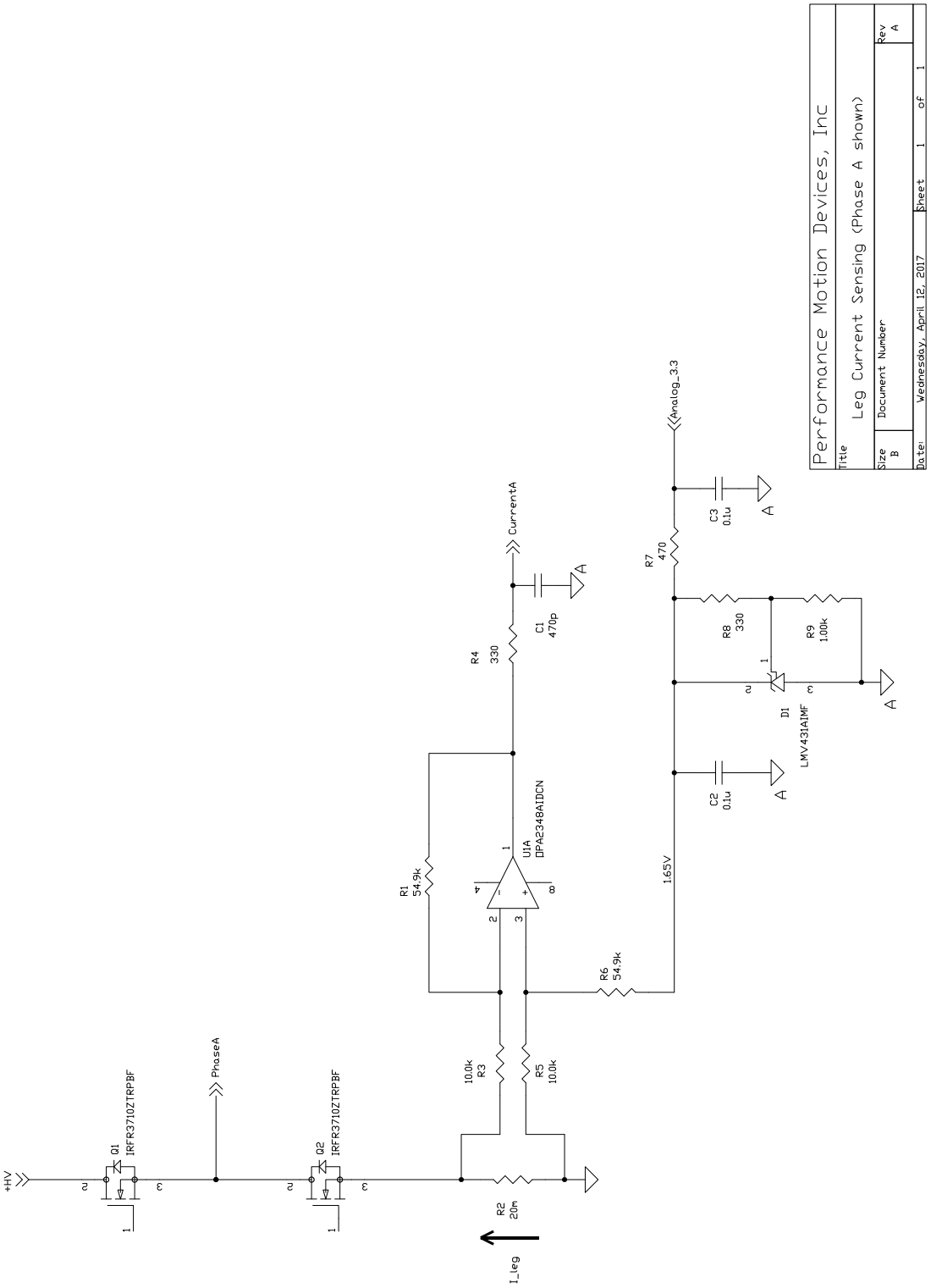
I_leg is defined as positive flowing from the ground to Q2 as shown in the schematic. It is because the current out of the half bridge and into motor winding is defined as positive.

Therefore, the current sensing range is ±15A. This current range should include the peak current during dynamic regulation. For example, during acceleration, the instantaneous current could be twice of the continuous current or even more.

With the continuous current and peak current known, the power rating of the current resistor can be determined. The conservative design rule is to assume that the current will go through R2 continuously, and the power dissipation is $I^{\wedge 2} * R_2$. However, the current will only flow through the resistor when Q2 is on; that is, if the duty cycle is known for this leg, the power dissipation can be approximated as $I^{\wedge 2} * R_2 * (T_{Q2ON}/T_{PWM})$ and resistor can be sized accordingly.

The board layout is critical for an optimal current sensing signal. The current sensing traces (to R3/R5) should be separated from the power path through R2, and these two traces should be routed in pair to improve its common-mode noise immunity. Also, a motor power train has multiple current sensing resistors, and these resistors are referred to ground. During layout, please treat those ground traces (e.g. trace to R5) as separated traces for each leg.

Figure B-6: Leg Current Sensing



B.9.2 Low Cost DC Brush Motor Drive

This example presents a low-cost, high performance DC motor drive.

The power train shown is an H-bridge (two half-bridge) for DC motor winding Motor + and Motor-. The input voltage can be up to 24V. It is capable of driving 1A continuous current with peak current of 2A.

The design for the two half-bridges and their current sensing circuits are the same. Using Motor + as an example, the half bridge has a P-channel MOSFET as the upper switch and an N-channel MOSFET as the lower switch, which is driven by PWMHighA and PWMLowA through buffer U1.

During normal operation, PWMLowA is active low. A logic "0" will generate 5V output at U1B, which turn on the lower N-channel MOSFET. R15, R7 and D5 provide an unsymmetrical turn-on and turn-off capability.

PWMHighA is active high. A logic "1" will generate 0V output at U1A, which will pull C2 to ground. +HV will charge C2 through D1 and R1 with voltage clamped by D1 (5.1V typical). This clamped voltage will turn on the P-channel MOSFET. When PWMHighA is "0", U1A outputs 5V, and C2 will discharge and turn off the P-channel MOSFET. R3, D3 and R5 provide an unsymmetrical turn-on and turn-off capability.

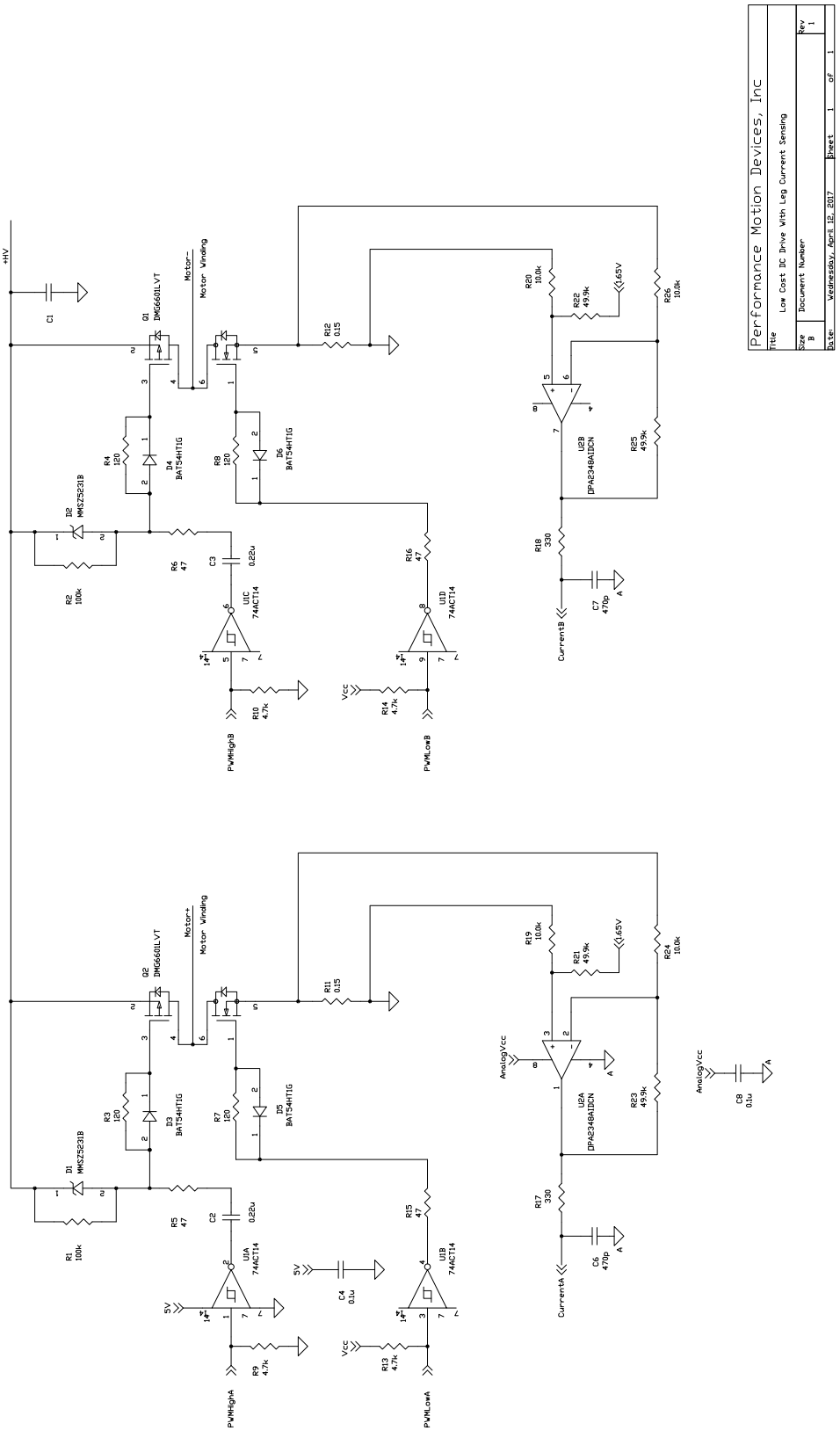
MC7x112 allows the user to configure the active polarity of all PWM output signals.

Upon power up or during reset, PWMHighA and PWMLowA outputs are high impedance. Therefore, pull-up resistor R13 and pull-down resistor R9 are used to ensure that the upper and lower switches are all off. Also, when a hard fault is triggered, PWMHighA and PWMLowA will go into high impedance, and R13 and R19 will turn off the MOSFET and put the output of the half bridge into high impedance.

R11 is the current sensing resistor, and U2A is the differential amplifier for signal conditioning. Please see Section B.9.1, "Leg Current Sensing" for more design considerations on leg current sensing.

This design has a low BOM cost and a small board footprint suitable for cost-sensitive or size-sensitive applications. However, its driving capability is limited by U1, and current capability is limited by the P-channel MOSFET. Therefore, this design is a good candidate for low voltage and low current applications.

Figure B-7: Low Cost DC Drive with Leg Current Sensing



B.9.3 Brushless DC Motor Drive

This example shows a brushless DC motor drive with leg current sensing. The power train has three half-bridge for BLDC motor's three winding terminals. The input voltage in this example can be up to 56V. It is capable of driving 15A continuous current with peak current of more than 25A.

The design considerations for each half-bridge and their current sensing circuits are the same. Using PhaseA + as example, the half bridge uses N-channel MOSFETs for both the higher and the lower switch to achieve high efficiency. The half bridge is driven by PWMHighA and PWMLowA through MOSFET driver U4, which is powered by 15V.

During normal operation, PWMHighA and PWMLowA are active high. For PWMLowA, a logic "1" turns on the MOSFET Q3. R25, R26 and D3 provide an unsymmetrical turn-on and turn-off capability.

A logic "1" PWMHighA will turn on Q1. C11 is the bootstrapping capacitor, and it is charged through D1 when Q3 is turned on. C11 provides the power to turn on Q1, and C11 needs to be a low-ESR capacitor such as a ceramic capacitor. D1 should be a fast switching diode with low leakage current, and its voltage rate should be chosen based on +HV and the +15V. R23 is optional; it can limit the charging current, especially during power up when C11 is zero voltage. R20, R24 and D2 provide an unsymmetrical turn-on and turn-off capability.

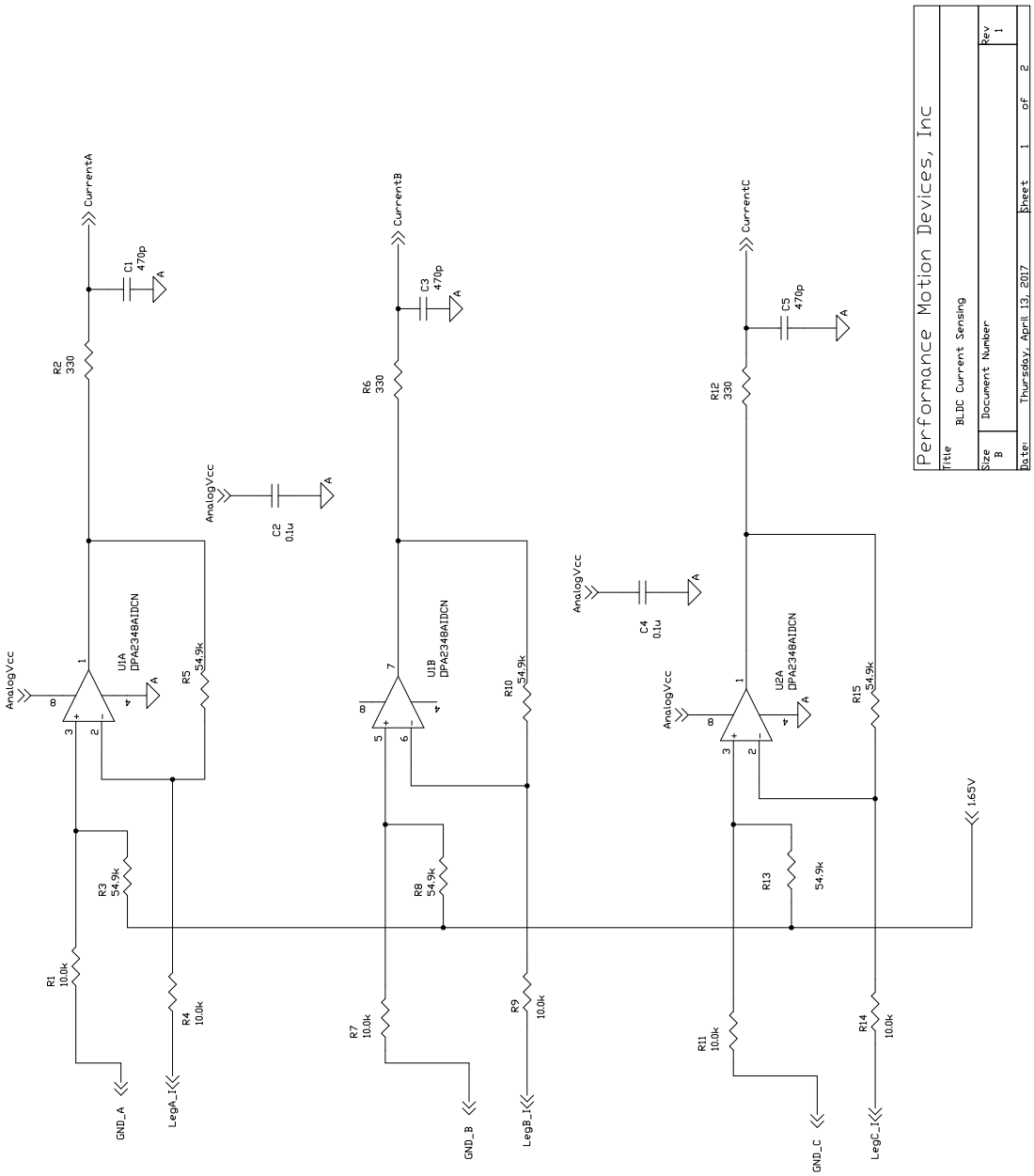
Upon power up or during reset, PWMHighA and PWMLowA output are high impedance. Therefore, pull-down resistors R36 and R37 ensure that the upper and lower switches are all off so that the half bridge output is high impedance. Usually the MOSFET driver has internal pull-up or pull-down resistors, and the user needs to check the driver's datasheet and decide if the resistors are necessary.

MC7x112 has an AmplifierEnable output, and it is used to shut down the MOSFET driver through the SD pin. Upon power up or during reset, AmplifierEnable is high impedance, and pull-down resistor R19 will ensure all motor output high impedance. When AmplifierEnable pin is used, the pull-down resistors R36/R37 are options.

R29 is the current sensing resistor, and U1A is the differential amplifier for signal conditioning. Please see Section B.9.1, "Leg Current Sensing" for more design considerations on leg current sensing.

This design uses dedicated MOSFET drivers and N-channel MOSFETs to achieve high efficiency and high performance. This example is suitable for applications in high noise environments. The MOSFET driver U4 has two ground references: Vss for the digital side and COM for the power side. The MOSFET turn-on and turn-off current out of pin LO will return from dedicated traces to pin COM instead of ground plane, which makes the board layout easier for noise immunity. The COM connection scheme shown also applies for H-bridge and 3-phase MOSFET driver with dedicated COM pin. For this example with independent half-bridge MOSFET drivers, COM pins can also be connected to respective MOSFET source pins to further improve the driving performance. Please refer to the low cost DC drive example if cost is critical and to the Stepper drive example for general-purpose applications.

Figure B-8: BLDC Motor Drive - Leg Current Sensing



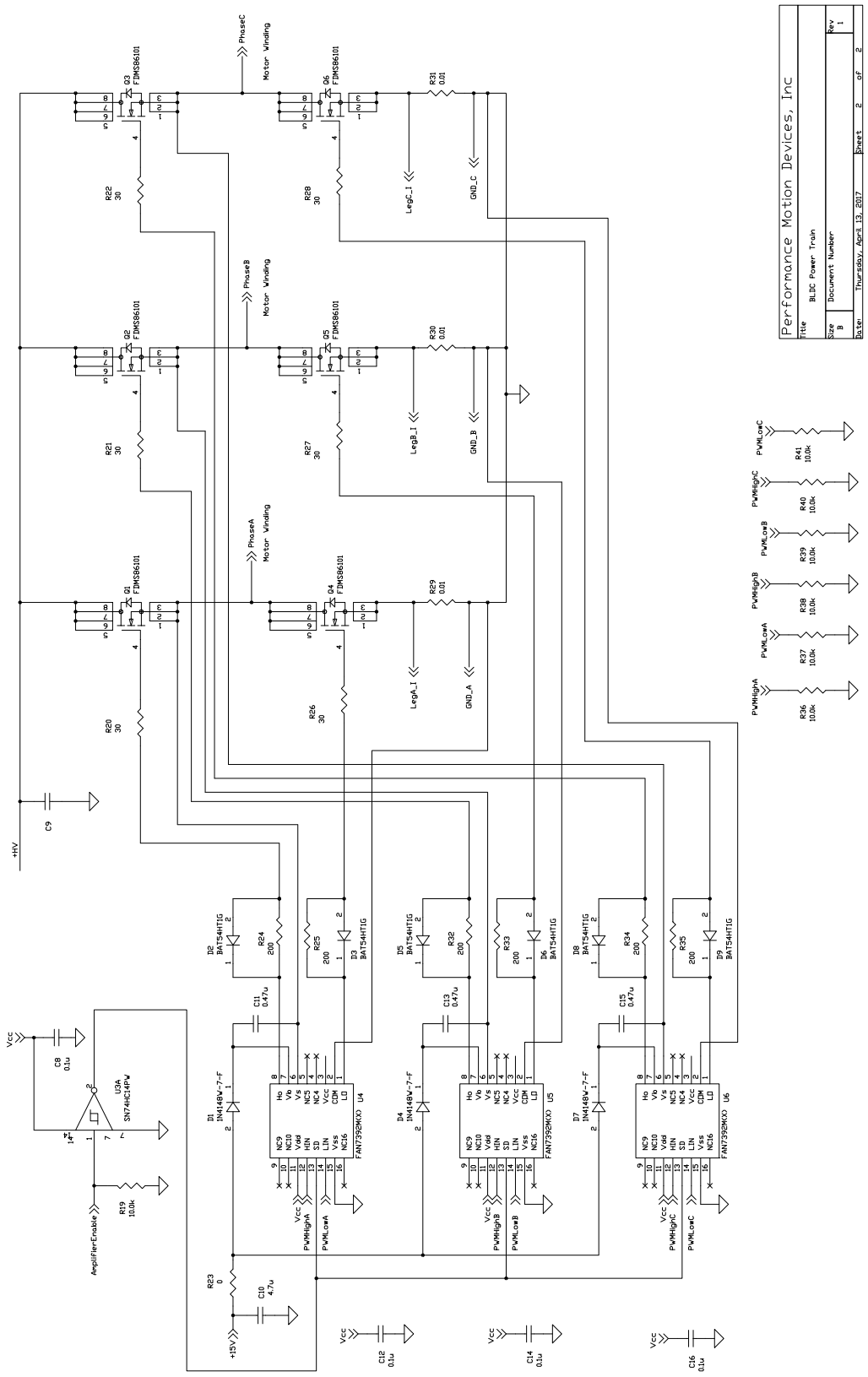


Figure B-9: BLDC Motor Drive - Power Train

B.9.4 Brushless DC Motor Drive with Intelligent Power Module

Intelligent power modules integrate a power train, gate driver and protection circuitry into a single package. They have advantages of simply design, improved system reliability and fast time-to-market. This schematics show an example of a BLDC drive with an intelligent power module with current capability of 1A.

The module operates in the same way as the discrete solution as shown in previous sections. Please refer to the section on BLDC motor power train and step motor drive power train for details.

Because intelligent power modules have control signals and a noisy power train in one package some special filters might be required. Please refer to the specific power module's user guide for details.

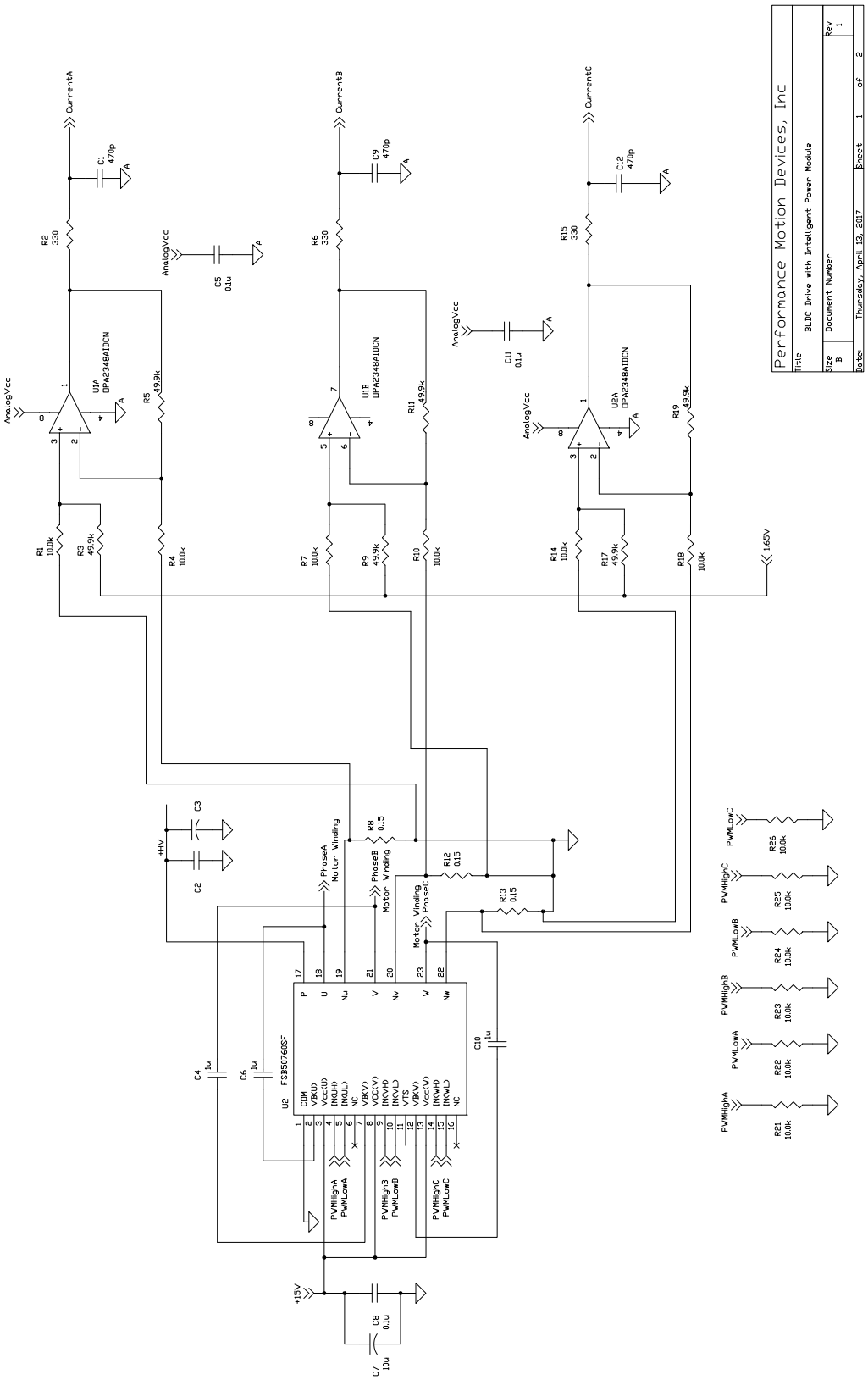


Figure B-10: BLDC Drive with Intelligent Power Module

B.10 Using the TI LMD18200 to Drive DC Brush Motors

In the following schematic, a magnitude and direction *PWM* signal is used to drive a DC brush motor with a nominal 24V, 2A drive. The H-bridge driver selected for this task is the LMD18200, which can be driven directly from a 3.3V CMOS logic output and as such can be directly interfaced to MC7x112 ICs.

Using the *sign/magnitude control* mode, sign and magnitude *PWM* signals are applied to both the PWM and DIR inputs of the LMD18200. In this mode, the resultant current ripple is reduced resulting in smoother operation of the motor.

The LMD18200 is equipped with an internal overcurrent circuit, which is tuned to a 10A threshold. External over-current circuitry may be added for currents with a lower threshold by using the sense output. In order to detect malfunctions, the *Vsense* signal may be used to sense the amount of current flowing through the motor windings. The sense output of the LMD18200 samples only a fraction of the drive current, with a typical 377 μA sensing per 1A driving current. .

Note that the circuitry provided in this section does not provide active current control. See Section B.9, "PWM High/Low Motor Drive with Leg Current Sensing/Control" for PWM-based application examples using the MC7x112 IC that provide active current control.

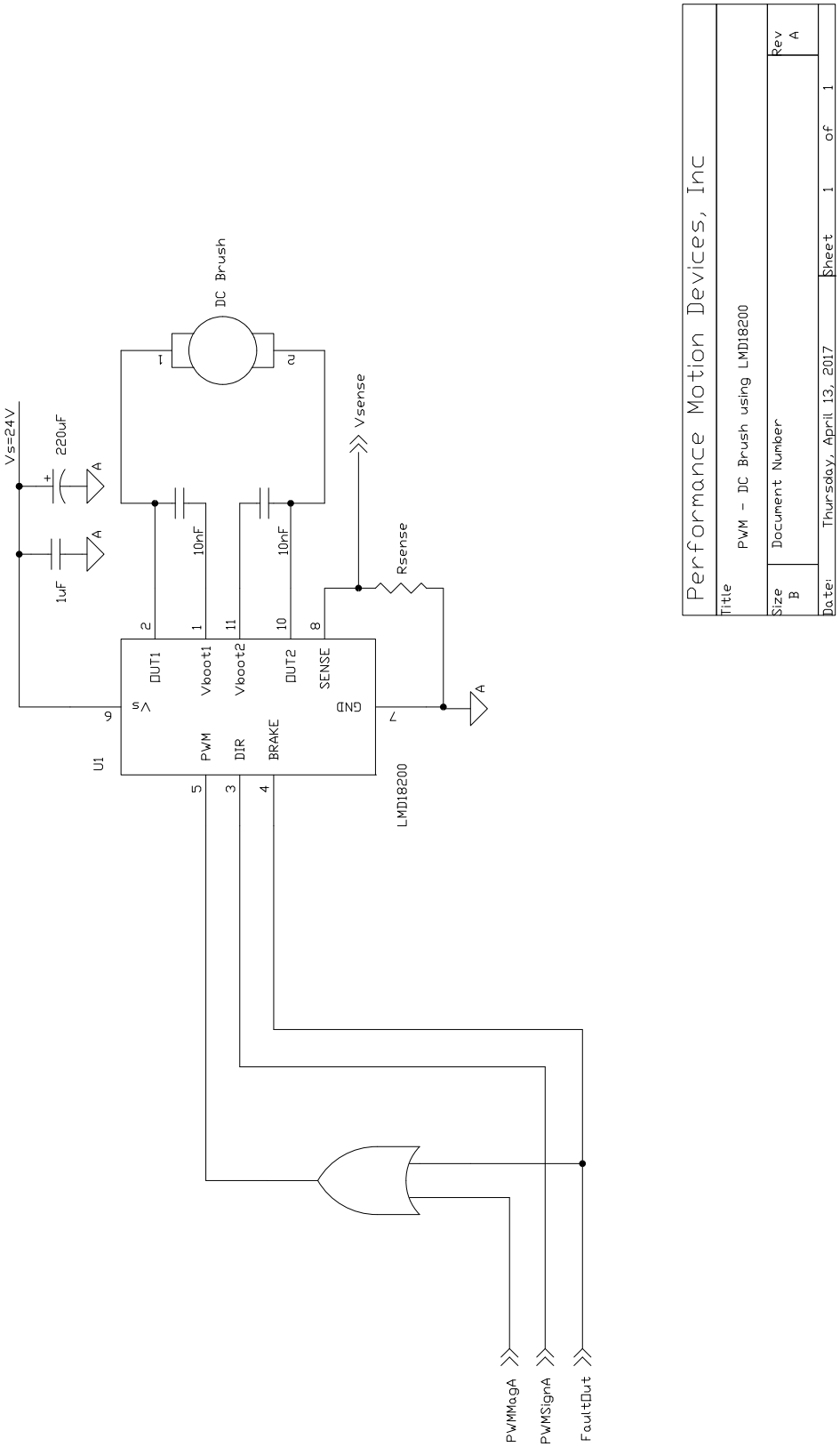


Figure B-11: DC Brush Amplifier Using LMD 18200

B.11 MC58420 Multi-Axis Motion Control IC Driving Two MC73112-Based Motor Amplifiers

The following schematic shows a two-axis application with two MC7x112-based BLDC motor amplifiers with a PMD MC58420 multi-axis motion control IC.

B.11.1 SPI Communication

MC7x112 receives control commands through an SPI interface and functions as an SPI slave. The MC58420 will output a continuous stream of desired torques to each MC7x112 via SPI. SPI communication is enabled when ~SPIEnable is pulled down. Only one MC7x112 can be enabled at any given time.

In this schematic the SPI master is a four-axis MC58420. Only the connections with MC7x112 are shown. For complete MC58420 wiring, please refer to the *MC58000 Electrical Specifications*. In this example, axis 1 and axis 2 are under control. The MC58420 sends torque commands to MC7x112s by pulling *SPIEnable1* and *SPIEnable2* low, respectively.

B.11.2 MC73112N Axis 1

For Axis 1, the MC73112N provides torque control with encoder-based commutation. It receives torque command from the MC58420 via an SPI bus and controls the motor with PWM high/low and leg current sensing. Please refer to Section B.9.3, “Brushless DC Motor Drive ” for more details

The encoder signals go to both the MC58420 and MC73112N. The MC58420 uses the encoder information for position control while the MC73112N uses it for commutation.

B.11.3 MC73112N Axis 2

For Axis 2, the MC73112N provides torque control with Hall-based commutation. It receives torque command from the MC58420 via an SPI bus and controls the motor with PWM high/low and leg current sensing. Please refer to Section B.9.3, “Brushless DC Motor Drive ” for more details.

The MC58420 uses the encoder information for position control, and the MC73112N uses Hall signals for commutation.

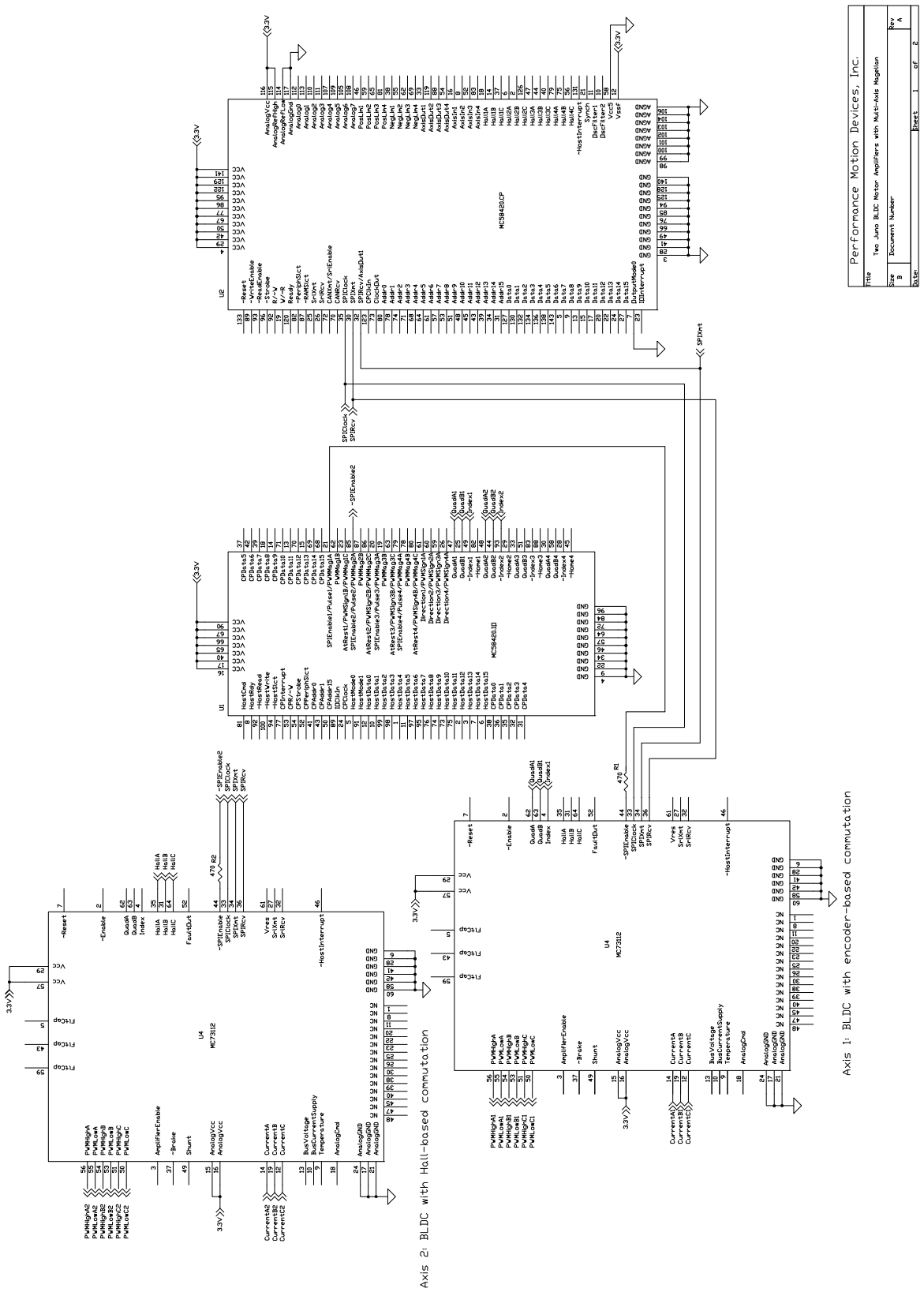


Figure B-12: Two-Axis MC7x112 BLDC Motor Drive with Multi-Axis MC58420 Motion Control IC

This page intentionally left blank.