WOODWARD

Product Manual 26223
(Revision NEW)
Original Instructions

Stepper Motor Controller

Version 1.01

Module User's Guide

IMPORTANT

DEFINITIONS

This is the safety alert symbol. It is used to alert you to potential personal injury hazards. Obey all safety messages that follow this symbol to avoid possible injury or death.

- **DANGER**—Indicates a hazardous situation which, if not avoided, will result in death or serious injury.
- **WARNING**—Indicates a hazardous situation which, if not avoided, could result in death or serious injury.
- **CAUTION**—Indicates a hazardous situation which, if not avoided, could result in minor or moderate injury.
- **NOTICE**—Indicates a hazard that could result in property damage only (including damage to the control).
- **IMPORTANT**—Designates an operating tip or maintenance suggestion.

WARNING

The engine, turbine, or other type of prime mover should be equipped with an overspeed shutdown device to protect against runaway or damage to the prime mover with possible personal injury, loss of life, or property damage.

The overspeed shutdown device must be totally independent of the prime mover control system. An overtemperature or overpressure shutdown device may also be needed for safety, as appropriate.

Read this entire manual and all other publications pertaining to the work to be performed before installing, operating, or servicing this equipment. Practice all plant and safety instructions and precautions. Failure to follow instructions can cause personal injury and/or property damage.

This publication may have been revised or updated since this copy was produced. To verify that you have the latest revision, be sure to check the *publications page* on the Woodward website:
www.woodward.com/publications
The current revision and distribution restriction of all publications are shown in manual *26311*.

The latest version of most publications is available on the *publications page*. If your publication is not there, please contact your customer service representative to get the latest copy.

Any unauthorized modifications to or use of this equipment outside its specified mechanical, electrical, or other operating limits may cause personal injury and/or property damage, including damage to the equipment. Any such unauthorized modifications: (i) constitute "misuse" and/or "negligence" within the meaning of the product warranty thereby excluding warranty coverage for any resulting damage, and (ii) invalidate product certifications or listings.

NOTICE

To prevent damage to a control system that uses an alternator or battery-charging device, make sure the charging device is turned off before disconnecting the battery from the system.

NOTICE

To prevent damage to electronic components caused by improper handling, read and observe the precautions in Woodward manual *82715*, *Guide for Handling and Protection of Electronic Controls, Printed Circuit Boards, and Modules*.

Contents

Illustrations and Tables

Important Notice

Woodward Governor Company and its subsidiaries (WGC) reserve the right to make changes to their products or to discontinue any product or service without notice, and advise customers to obtain the latest version of relevant information to verify, before placing orders, that information being relied on is current and complete. All products are sold subject to the terms and conditions of sale supplied at the time of order acknowledgement, including those pertaining to warranty, patent infringement, and limitation of liability.

Customers are responsible for their applications using WGC products. 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 or procedural hazards.

WGC assumes no liability for applications assistance or customer product design. WGC 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 WGC covering or relating to any combination, machine, or process in which such products or services might be or are used. WGC's publication of information regarding any third party's products or services does not constitute WGC's approval, license, warranty or endorsement thereof.

Reproduction of information in WGC manuals or product specifications is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations and notices. Representation or reproduction of this information with alteration voids all warranties provided for an associated WGC product or service, is an unfair and deceptive business practice, and WGC is not responsible or liable for any such use.

Resale of WGC's products or services with statements different from or beyond the parameters stated by WGC for that products or service voids all express and any implied warranties for the associated WGC product or service, is an unfair and deceptive business practice, and WGC is not responsible nor liable for any such use.

IN NO EVENT SHALL WGC BE LIABLE FOR ANY COSTS, EXPENSES, LOSSES, DAMAGES (INCLUDING, BUT NOT LIMITED TO, DIRECT, INDIRECT, CONSEQUENTIAL, INCIDENTAL, SPECIAL OR PUNITIVE DAMAGES) OR INJURIES INCURRED BY USER OR ANY THIRD PARTY AS A RESULT OF THIS AGREEMENT OR ANY USE OF THE THESE MATERIALS.

Chapter 1.
General Information

Introduction

This User's Guide documents the Woodward Stepper Motor Controller software and provides an overview of the recommended hardware to interface with the Woodward Swift™ or Swift-Plus valve. The hardware design recommendation is limited to the output drive circuitry to facilitate application with the Swift valve family.

This software has been optimized for use with and is only intended to be used in conjunction with the Woodward Swift valve product family. At the time of this printing, the software object code is compiled for, and thus will only run on, the Texas Instruments TMS320C28xx (C28x) DSP.

This Stepper Motor Controller software is available through licensing agreements to customers wishing to use the Swift valve family in applications without the Woodward Swift Driver. This software is WGC intellectual property, and as such, WGC owns all proprietary rights, including patent, copyright, trade secret, trademark and other proprietary rights.

Audience

The intended audience is software programmers and hardware engineers to facilitate application of this software work product.

Scope

This guide includes:
- Documentation of Stepper Controller Software
- Stepper Controller Software Object Code
- Example C28x Application Software

This guide assumes that the reader is familiar with the Woodward Swift Metering Valve, TI Code Composer Studio IDE, TI C28x DSP family, stepper motor control, DSP and C-language programming. Information pertaining to these items is not be provided within this guide.

Acronyms

PWM	pulse width modulated
TI	Texas Instruments
C28x	Texas Instruments TMS320C28x™ Fixed-Point DSP
IQmath	High Accuracy Mathematical Functions (32-bit implementation)
QMATH	Fixed Point Mathematical computation
Swift™	Woodward Swift or Swift-Plus valves
WGC	Woodward Governor Company and its subsidiaries

Application Software Requirements

Texas Instruments Code Composer Studio
Texas Instruments IQmath Library

Application Hardware Requirements

Texas Instruments C28x DSP

References

- Woodward Manual 26202, Swift and Swift-Plus Gas Metering Systems
- Spectrum Digital 506265-0001, eZdsp™ F2812 Technical Reference
- Texas Instruments C28x product documentation including: SPRS174F, SPRU078, SPRU065, SPRU060, SPRU059, SPRU051, SPRU095, SPRU067 (found on Texas Instruments' web site, www.ti.com)
- Texas Instruments IQmath Library: Module User's Guide Version 1.4.1

NOTICE	**Follow Recommended Validation Tests provided in this guide prior to attempting to run an engine or fuel system.**

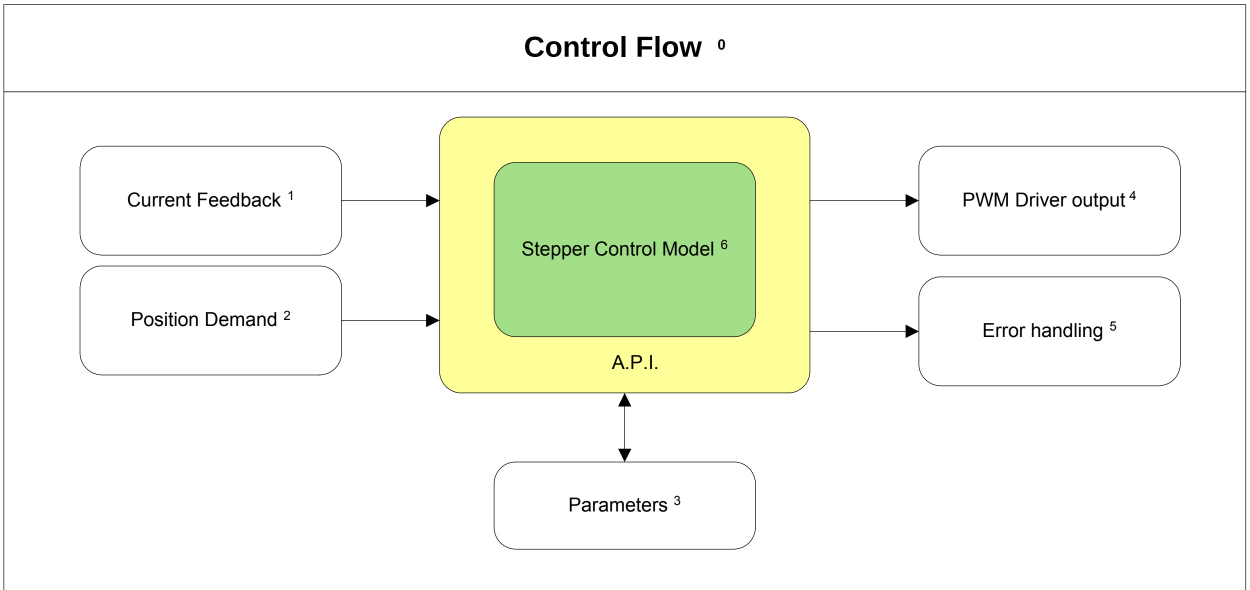
Installation

System Requirements
None listed

Software Installation
The following files are installed:
- C28xStepper1.lib
- Mlm2f.h
- Twmtypes.h
- Simstruct_types.h
- Rtlibsrc.h

The following TI support files are required:
- IQmath.lib
- IQmathLib.h

Application Architecture



NOTES:
0) This is an example of the flow diagram that will handle the startup and error conditions for the control model.

1) Read ADC current feedback and scale's to appropriate value for control model. Possible diagnostics on input signal.

2) Generates the demand signal and scales to the appropriate value for control model.

3) Parameters management. Possible just constants or eeprom based parameter store and retrieval system. Parameter are for customizing the control model to different values.

4) Setup and scaling of PWM signal to timer values.

5) Handle error flags coming back from control model as needed.

6) The control model that will run the algorithm for the stepper motor. The stepper control model is the part of the software that is licensed. (*.lib and *.h's). The API is the interface to the control model. (*.doc)

Figure 1-1. Control Model Application

Chapter 2.
Software Description

Overview

The control is implemented in fixed point math, so all I/O signals must be converted to/from 16.16 format before use in the control model.

The control model has the following characteristics:
- Model-based controller
- Re-Zero calibration
- Offset (zero) compensation
- Current Fault detection
- Parameterization

Function Calls
The control model has two function calls, an initialize and a run.
- mlm2f_initialize
- mlm2f_step

The initialization must be called once during startup and the 'run' must be called every 100 microsecond execution loop.

Control Model I/O
The following diagram shows the model inputs and outputs. The model input and outputs are intended to be as parametric as possible, this allows growth for the driver with different valve types.

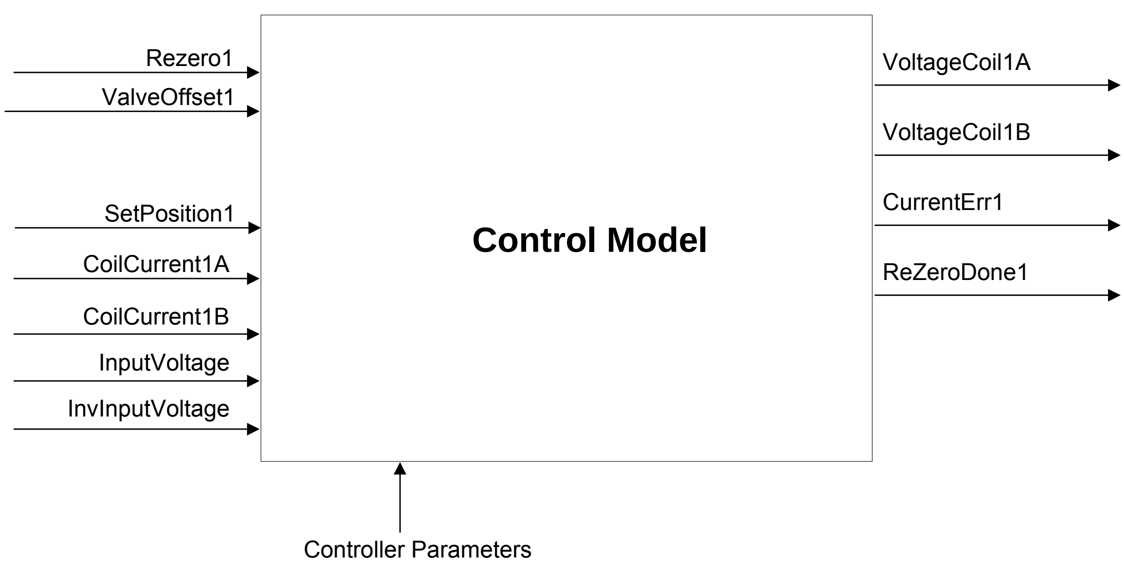


Figure 2-1. Control Model I/O Overview

Control model software requirements

The MatLab model should be run every 100 μs (10 kHz). The format for the numbers used in the model is 16.16 fixed-point. The current to each stepper coil (quantity 2) must be monitored and faults annunciated. These signals should be calibrated. The controller parameters and Valve Offset must be initialized in the model prior to running the model.

Re-Zero

Since the model uses open-loop positioning, a re-zero must be performed to ensure correct valve position. A valve re-zero must be done on power-up and whenever a fault occurs that might cause the system to lose its positioning orientation. A re-zero is initiated by setting 'Rezero1'—the model will command the stepper motor position towards min. When complete, the 'ReZeroDone1' is set by the model, indicating normal operation can begin.

Calibration (Valve Offset)

The Valve Offset will be determined by Woodward during the valve calibration procedure and provided to Capstone for each valve. This Offset value is unique to each valve and represents the valve's closed position—the mechanical and electrical "0". This Valve Offset value must be stored in non-volatile memory within the controller and passed to the Model on initialization.

Control Model I/O Definition

Parameter	Input / Output (TYPE)	Units	Min resolution	Update Rate (min)	Description
SetPosition1	Input 16.16	% / 100	16 bits	1 ms	Commanded position to the control model
InputVoltage	Input 16.16	Volts	16 bits	10 ms	Input Voltage
InvInputVoltage	Input 16.16	Volts	16 bits	10 ms	Inverse of the Input Voltage (i.e. 1/InputVoltage)
CoilCurrent1A	Input 16.16	Amps	16 bits	10 ms	Coil A current feedback. Simultaneously sampled with 1B.
CoilCurrent1B	Input 16.16	Amps	16 bits	10 ms	Coil B current feedback. Simultaneously sampled with 1A.
Rezero1	Input (BOOL) uint16	'1' = Initiate ReZero sequence	1 bit	Only req'd on rezero	Command to begin re-zero algorithm
VoltageCoil1A	Output Int32	counts	16 bits	0.1 ms	Drive voltage to COIL A This output must be scaled based on the hardware configuration such that a VoltageCoil setting of 0 corresponds to zero current (50% PWM duty).
VoltageCoil1B	Output Int32	counts	16 bits	0.1 ms	Drive voltage to COIL B This output must be scaled based on the hardware configuration such that a VoltageCoil setting of 0 corresponds to zero current (50% PWM duty).
CurrentErr1	Output (BOOL) uint16	'1' = Fault	1 bit	10 ms	Model detected current errors Detection of a CurrentErr shall shut the driver down. Since the zero position may be effected, a re-zero shall be performed when the condition is cleared.
ReZeroDone1	Output (BOOL) uint16	'1' = ReZero complete	1 bit	Only req'd on rezero	Feedback boolean to indicate re-zeroing algorithm is complete.
ValveOffset1	Input 16.16	counts	16 bits	Init param	Calibration offset to be added to position command in model.
K01 thru K27 (controller parameters)	Inputs 16.16		16 bits	Init param	These 27 parameters set up the model for various applications.

Table 2-1. Control Model I/O Definition

Model Fault Detection

The Model software will detect if the coil or wiring is open, as indicated by a CurrentErr. Since zero current is a valid operating level to a coil, there are static situations where the detection will not activate a failure. At those positions, the motor torque is valid, however attempting to move the output from this position will activate the fault if one of the coil wirings has a fault.

The Model software will NOT detect if the coil has been connected in the wrong direction. In this case, the motor will just rotate in the wrong direction.

The software will detect if the coils are cross wired to other coils, however each coil will appear to have the correct current during start-up. This detection will occur when the total current is at inappropriate levels.

Recommended System Faults

Detection of a coil fault, as indicated by the Model, should shut down the driver. This diagnostic indication should be filtered before action is taken. Since the zero position may be effected, a re-zero should be performed when the condition is cleared.

The software should detect if any of the coils are OPEN during start-up. This could be checked by during start-up by setting a fixed current to each coil and comparing this to the current readback.

Stepper Current Driver

The driver provides stepper driver outputs with an update rate of 0.1 ms to position the stepper motor. The PWM freq should be 20 kHz.

The model provides a value between –1 and 1 as the fractional part of a 16.16 number which is passed as an int32 out of the model. The user's application software must scale this signed integer into a uint16 for the PWM drive output—where 50% duty cycle corresponds the zero current (see scaling below). It is also recommended that these counts be limited between 5 and 95% duty cycle to protect the drive circuitry.

Signals	Disabled State	Enabled State
PWM signal	Low	High
Brake signal	High	Low
Direction Signal	Zero Current/Voltage	Commanded Model Current/Voltage

Table 2-2. Driver Output Signals

Scaling

The duty cycle to current relation is:

- 100% duty = Max positive current
- 50% duty is zero current
- 0% duty is max negative current

Description of API Functions

The control model has two function calls, an initialize and a run.

mlm2f_initialize

void mlm2f_initialize(boolean firstTime)

This function is run once during initialization to set all timers and model parameters into a known state.

mlm2f_step

void mlm2f_step(void)

This function runs the control model (stepper algorithm). This function must be called every 100 microseconds.

Model parameters

Parameters are for customizing the control model to different values. The following parameters must be sent to the model prior to running the stepper algorithm. These parameters can be stored as just constants or as EEPROM-based parameters using a store and retrieval system.

```
SwiftCons_K01 = -589824;
SwiftCons_K02 = 589824;
SwiftCons_K03 = 23593;
SwiftCons_K04 = 21474836;
SwiftCons_K05 = 109;
SwiftCons_K06 = -109;
SwiftCons_K07 = 2185;
SwiftCons_K08 = 458752000;
SwiftCons_K09 = -458752000;
SwiftCons_K10 = 36030382;
SwiftCons_K11 = 51118;
SwiftCons_K12 = 1192;
SwiftCons_K13 = 147456;
SwiftCons_K14 = 390306;
SwiftCons_K15 = 2185;
SwiftCons_K16 = 514720;
SwiftCons_K17 = 4902;
SwiftCons_K18 = -1979;
SwiftCons_K19 = 1359143;
SwiftCons_K20 = 3160;
SwiftCons_K21 = 15401;
SwiftCons_K22 = 95683;
SwiftCons_K23 = 6553600;
SwiftCons_K24 = 51471974;
SwiftCons_K25 = -131072;
SwiftCons_K26 = -514720;
SwiftCons_K27 = 393216;
```

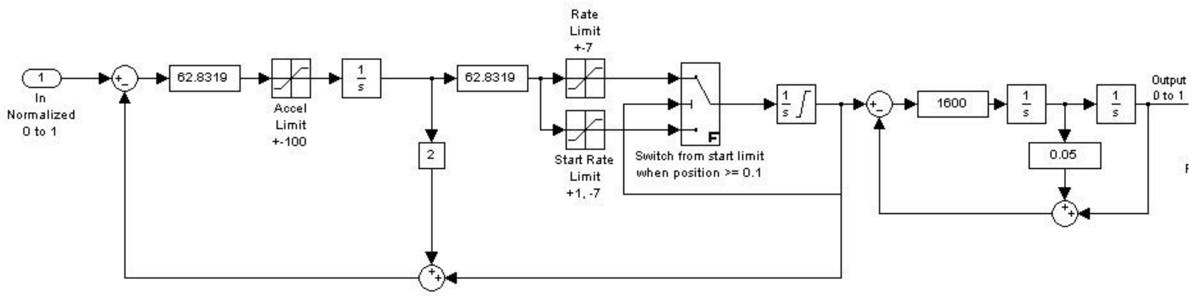
Timing

Minimum update rates for the control model are provided in Table 2-1.

NOTICE	**Reading control model's output voltage (VoltageCoil1A/B) and updating the PWM output registers must be done on a strict periodic basis. Failure to comply with this can result in jitter or a beat frequency.**

Control Model Properties

Transfer Function



Performance
Execution Time (DSP running at 150 MHz):
- Running out of RAM 20 µs
- Running out of FLASH with 4 wait states 31.5 µs

Code Size
Library Code Size by section (in hex):
- Text – 8DF
- Bss – 11E
- Const – 401

Total: 0xDFE (3582)

Functionality
The control model consists of the following functional units:
- Position Controller
- Rate Limiter
- Current Controller
- ReZero Algorithm
- Resistance Estimator

Position Controller
The position loop is a state estimator for the motor/load. Based on position command, it provides a position estimator to the current controller. The control gains are calculated based on motor characteristics.

Current Controller
The purpose of the current controller is to convert the model position into electrical position, estimate the current in the coils, and calculate the A and B coil drive voltages which are output from the model to the PWM timers in the CPU.

The input to the model is the requested electrical position, which is compared to the position estimator output. The signal is then converted to the sine/cosine drive and scaled to the maximum current. The required coil voltages are then generated by the current loop while accounting for the back EMF, coil inductance, I*R drop across the coil and the available battery voltage. The resistance estimator handles the estimate of the coil resistance.

Rate Limiter
The rate limiter shapes the command signal to limit acceleration and velocity. In addition, the motor rate has a second limit when it is less than 10% position. It is set to approximately 1/5 of the max rate. This is done to gradually bring the motor of the stop in a region where transient time is not critical.

Resistance Estimator with current error detection

The resistance estimator compares the measured current in coils to the state estimated current in the current controller and adjusts the estimated resistance to drive the error to zero. This resistance estimate is used in the current controller. The estimator also looks for coil/wiring errors based on the sum of the squares of the coil currents is constant.

A limitation of this method is that depending on the position of the motor, it is possible that the current in one coil is 0 A. In this case, if the coil is opened, the sum of the squares of the currents will not change. However, in this motor position, the maximum torque is being delivered to the motor, and as soon as the model tries to move the motor, the error will be detected.

The current error detection window is set from 3.5 to 7.1 A so that if the sum of the squares of the two coil currents goes outside this window, a current error is detected. This window is equal to $(2.25)^2$ ±40%. Due to the way the H-bridge IC (integrated circuit) works, an external overcurrent error will not be detected because the H-bridge will self-limit the average current through the coil. This is done through cycle-by-cycle limiting internal to the IC at a 10 A threshold.

The Resistance estimator integrator also has a minimum limit on it. This output is set to about 1.9 Ω so that at low temperatures more current is driven to the coil, increasing force margin at low temperature where motor output decreases.

The Resistance estimation and error detection are severely limited by the current-sampling method, so the two functions are disabled during high velocity position changes.

Re-Zero Algorithm

The rezero algorithm drives the valve output closed to the zero position, to ensure that the open-loop valve positioner is correctly oriented. A re-zero is initiated by with a re-zero command, which runs the motor at a slow rate (ReZeroVel) towards the min stop. This velocity is 1.5 rad/s so will rotate the motor 9 radians after 6 seconds. The full stroke of the motor is $2\pi*1.25 = 7.85$ radians, allowing the motor to stroke from full open into the stop, with margin. When complete, the 'ReZeroDone1' is set by the model indicating normal operation can begin.

Chapter 3.
Example Program

General

The example program is the source code of a simple implementation of the control model on a 2812 DSP. This is not part of the license and is delivered 'as is'. This code is only intended to be used as clarification on the implementation details.

The Overview and State Diagram figures below provide a high-level overview of the example program's functionality.

Hardware

The example program runs on a Spectrum Digital eZdsp TMS320F2812 evaluation board running at 150 MHz. The code can run out of FLASH (default) or RAM/ROM, depending on settings in the F2812_lnk.cmd file. The stepper output drive circuitry can be found in Chapter 4 of this guide and schematically in Appendix A.

DSP to Hardware Connections:

- PWM1 and PWM3 outputs is connected to DriverDirPwmA (coil A) and DriverDirPwmB (coil B).
- GPIO A3, A4, and A5 is connected to DriverPwmLL, DriverBrakeLL, and DriverTempFlagLL, respectively.
- CurrentFeedbackA and B are connected to ADC inputs A0 and B0.
- An analog input for position demand is connected to ADC input A1.
- A shutdown (close valve command) digital input is connected to GPIO input A7.

Software Installation

In addition to the licensed software, the following files are included in the example software:

• StepperC28x.pjt	Code Composer project
• StepperMain.c	Main
• IQmath.lib & IQmathLib.h	Math library
• F2812_lnk.cmd	Linker command file
• PosState.c & .h	State machine for stepper output states
• GlobalVars.c & .h	Declarations of global variables
• ModelVars.c & .h	Model input and output function calls
• DSP28_Adc.c & .h	ADC initialization and support functions
• DSP28_CpuTimers.c & .h	CPU Timer initialization and support functions
• DSP28_DefaultIsr.c & .h	ISR initialization and support functions
• DSP28_Ev.c & .h	EV initialization and support functions
• DSP28_GlobalVariableDefs.c & .h	Global Variables and Data section pragmas
• DSP28_Gpio.c & .h	GPIO initialization and support functions
• DSP28_PieCtrl.c & .h	PIE Control Regs initialization
• DSP28_PieVect.c & .h	PIE Vector table initialization
• DSP28_Spi.c & .h	SPI initialization and support functions
• DSP28_SysCtrl.c & .h	System control initialization and support functions
• DSP28_usDelay.asm	delay function
• DSP28_CodeStartBranch.asm	branch to code start

IMPORTANT **The DSP28 files have been provided by TI for evaluation purposes and modified by WGC for this example. Additional header files have been included, although not required for this example.**

Example Program Software Overview

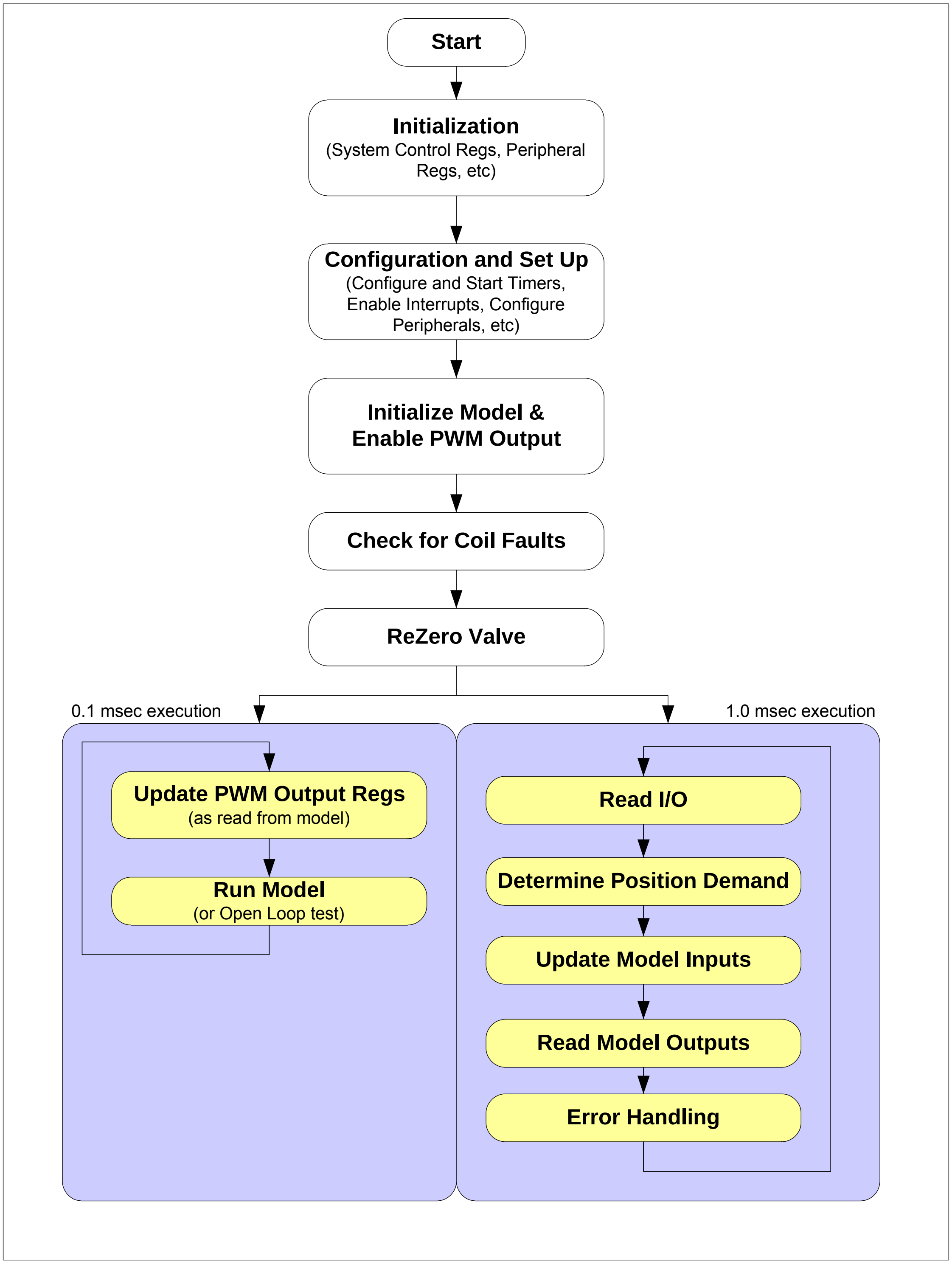


Figure 3-1. Example Program Overview

Software Description

Modes

Included in the example are several operational options. The stepper motor can be controlled through the licensed model software (RunMode = 3) or full-step continuous rotation (RunMode = 1 for CCW and RunMode = 2 for CW). In addition, when in the normal operational mode, the Position Demand setpoint can be set either internally (RunInternalPosDmd = 1) or through an analog input (RunInternalPosDmd = 0).

NOTICE

RunMode settings of a '1' or a '2' are intended for software development and integration testing. In these modes, the output should be connected to an unmounted stepper motor and not to the Swift Valve.

Variables

The following identifies and defines a select list of variables available in the example program.

RunMode	Determines mode of operation for stepper controller
RunMode = 0	OFF – no current
RunMode = 1	Full-step continuous rotation in CCW direction
RunMode = 2	Full-step continuous rotation in CW direction
RunMode = 3	Runs the control model provided by the library
Speed	Speed adjustment for open-loop stepper rotation (RunMode = 1 or 2)
PositionDemand	Position Demand setting. Range 0 to 1 (where 1=100%)
RunInternalPosDmd = 0	Position Demand is set by the analog input (ADC input A1)
RunInternalPosDmd = 1	Position Demand is set by the internal variable PositionDemand
UseShutdownInput = 0	Shutdown discrete input is not used
UseShutdownInput = 1	Shutdown discrete input will close valve when opened

Important Setup Items

All hardware and software modules must be initialized and configured properly before use (SysCtrl, ADC, GPIO, SPC, SPI, EV, etc). Besides these obvious items, a few additional items are noted here.

PWM_FREQ_HALF (in GlobalVars.h) must be set to match the Event Manager's settings for zero current to the pwm drive circuitry.

Additional Notes

The software runs out of FLASH memory—changes must be made in the linker command file to run out of RAM.

Error handling has not been implemented, only noted.

The coil fault detection during the startup sequence has not been implemented, only noted.

GPIO A8, A9 and A10 have been configured and used for timing analysis.

NOTICE

MAX_PWM_COUNT and MIN_PWM_COUNT parameters are available to limit the counts to the hardware to prevent damage. During code development it is recommended these be set to a tighter range to limit the current to the stepper driver circuitry.

Example Program State Diagram

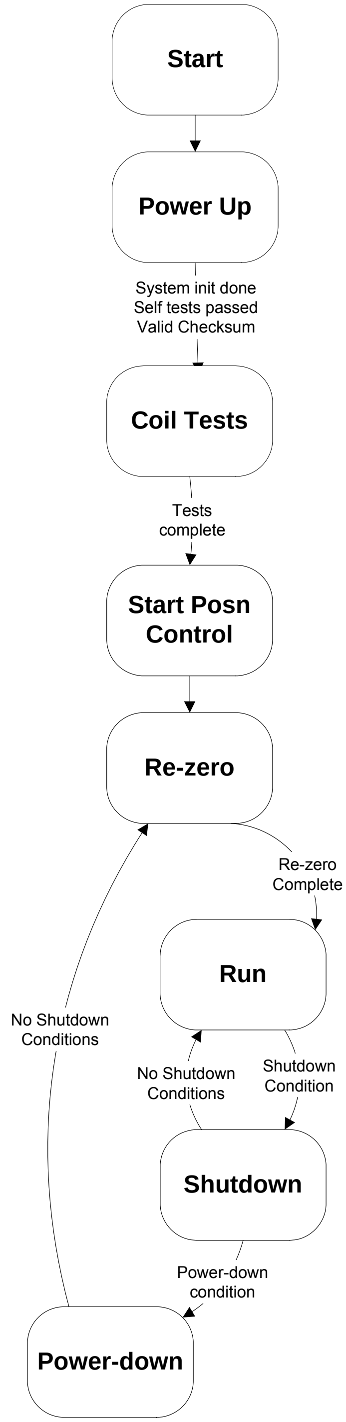


Figure 3-2. Example Program States

Timing

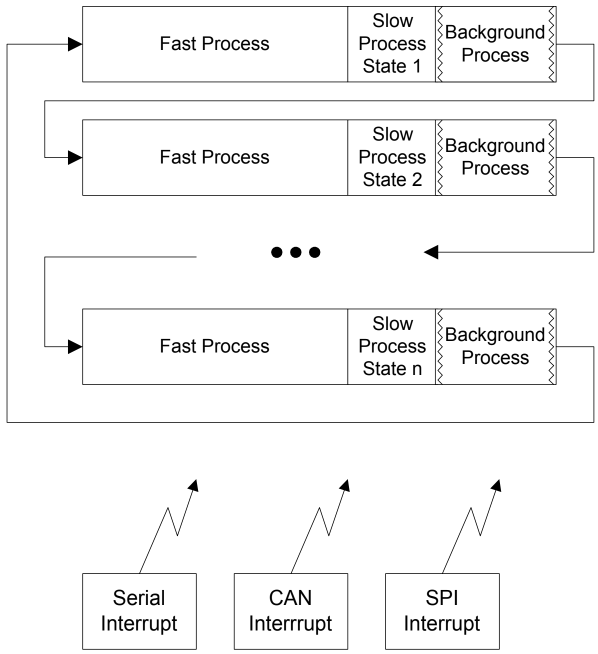


Figure 3-3. Example Program Timing Overview

The Fast Process
The fast process runs every 0.1 ms and is triggered by a periodic interrupt generated by CPU timer 0. It consists of the positioning loop tasks.

The Slow Process
The slow process is used to if there is not enough horsepower to complete the fast process. Many of the diagnostic functions need not execute every 0.1 ms, for example. The slow process is intended to execute in discrete chunks as shown.

The Background Process
Application layer access to the EEPROM and other functions are carried out in the background process.

Interrupts
Serial, CAN, SPI interrupts can all provide simple buffering of characters/messages/data for use by various application and device (like EEPROM or ADC) tasks.

Swift Valve Parameters

A basic understanding of the Swift Valve is presented to aid in product applications. The following picture represents the actuation system.

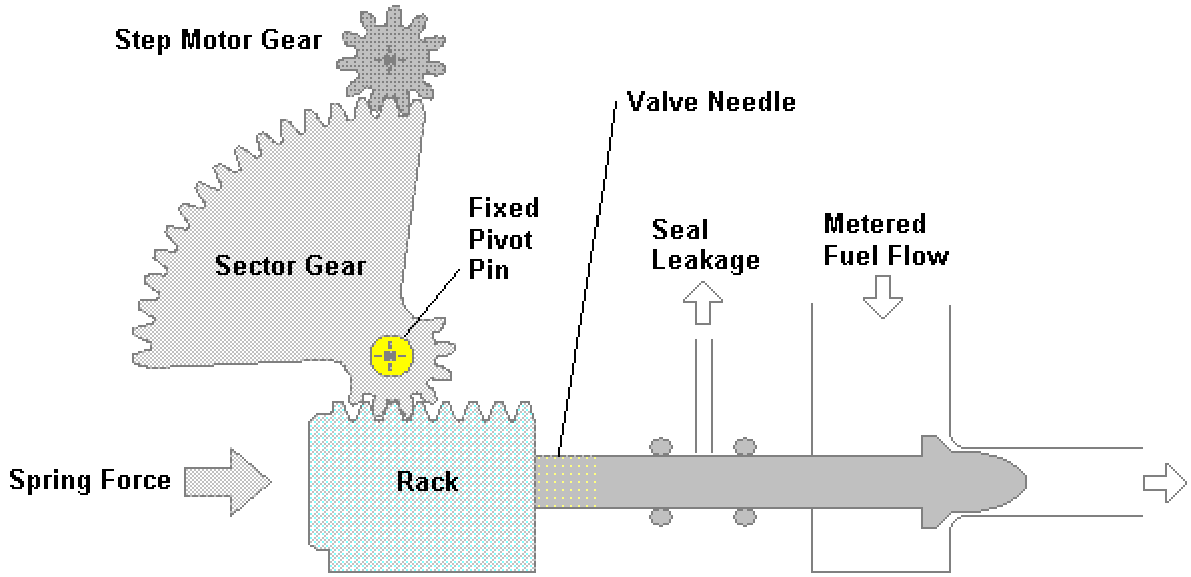


Figure 3-4. Swift Single Valve Schematic

Stepper Motor rotation: 1.25 rotations provide full needle valve travel

Needle Travel: 0.325"

Gear ratio: 0.324"/(1.25Turns*2*pi) = 0.04127 in/rad. = .00104 m/rad. This is the needle travel per radian of motor rotation.

The motor is 1.8 degree/step, so 1.25 rotations is 250 steps. 1 electrical rotation (full rotation of sine/cosine) is 4 steps, therefore there are 150/4 = 62.5 electrical rotations in 1.25 rotations of the motor.

Chapter 4.
Example Hardware Application Guide

Description of Operation

The stepper motor control rotates a stepper motor with a sinusoidal current waveform in each coil, where the phase offset in coil B is 90 degrees from coil A. The sinusoidal current waveform minimizes torque ripple.

The waveforms below show one electrical revolution of the stepper motor. The actual motor has 4*1.8 degrees of rotation per electrical revolution or 7.2 degrees. Because of the gear ratio, the full stroke motor rotation is 1.25 turns, or (1.25*360)/7.2 = 62.5 electrical revolutions.

The maximum electrical rotational speed is approximately 70 rad/s. Although many stepper motors are run full step or half step (half-step sequence shown in figure), micro-stepping is achieved by modulating the drive current to each coil sinusoidally that approximates the square envelope, as shown in the figure. This improves accuracy, and dynamic force margins.

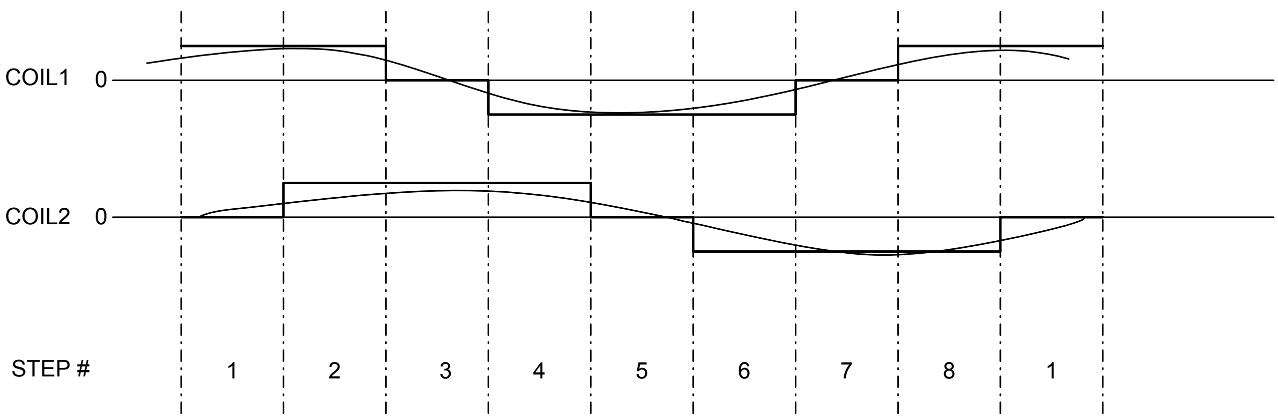


Figure 4-1. Half Step Motor Modulation

The stepper motor power stage drives a 2-coil, 4-wire stepper motor. Maximum current is set by software and is 2.25 A max for both steady state and transient operation.

The drive signals for each motor phase are form a microcontroller/DSP. For the mode that the drive IC (LMD 18201) is running, the PWM and BRAKE signals are tied to a fixed level and the PWM duty cycle (5% to 95%) is from the DIR input.

Current feedback for the controller is through a current sense resistor and instrumentation amplifier, one per coil.

Stepper Motor Driver Power Stage Specifications

Parameter	Value
Frequency	20kHz
Duty Cycle range	0.05 – 0.95
Supply Voltage	21.5V to
Motor resistance range	0.6 ---0.8 at 25C
Motor Inductance	.0022H
Current limits	2.25A steady state and transient,
Short Circuit Current Detection and limit	Driver H-bridge (LMD 18201) is limited to 10A current transients through the upper transistors of the H-bridge. The drive output turns off at this point. This will limit the average current through he current sense resistors to < 1Amp typically. Also, Driver IC is self limiting (outputs turn off) at 170 degrees C junction temperature.
Wire Distance limits	Total distance from power supply to driver plus driver to valve shall not exceed 30ft.
Wire gauge	18AWG
PWM Resolution	10.5 bits, min
Current Feedback Accuracy	+/- 3% over temperature (-20C to 85C)
Current Feedback Bandwidth	1kHz min

Table 4-1. Driver Power Stage Specifications

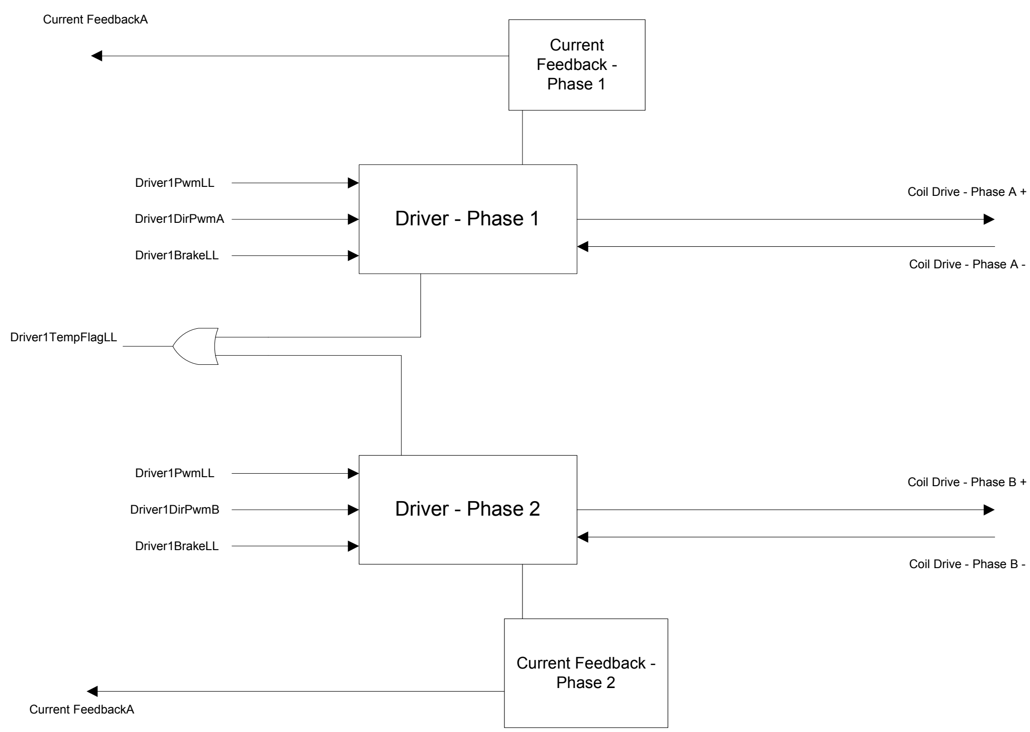


Figure 4-2. Driver Block Diagram

Signal	Description	Definition
DriverPWMLL	h-bridge PWM	See "switching sequence" and "State Table" Held at 5V.
DriverBrakeLL	h-bridge BRAKE	See "switching sequence" and "State Table" Held at 0V.
DriveDirPwmA/B	Duty cycle command	50 % is zero current dc through coil
DriverTempFlagLL	Overtemp alarm from driver IC	Overtemp alarm set at 145C die temp (logic "1"). Driver IC shutdown occurs at 170C
CurrentFeedbackA/B	h-bridge current feedback	Transfer Function: 2.496V + 0.499*Icoil (scaled to ADC input, current range is +/- 4Amps)
Coil Drive PhaseA/B	Stepper Output driver	2.25A peak sin/cos waveform

Table 4-2. Signal Specifications

Switching Sequence

This sequence alternately drives the coil voltage by turning on source1, sink2 for one half switching cycle, then source2 , sink1 for the other half of the cycle. 50% duty cycle represents 0 average current through the coil.

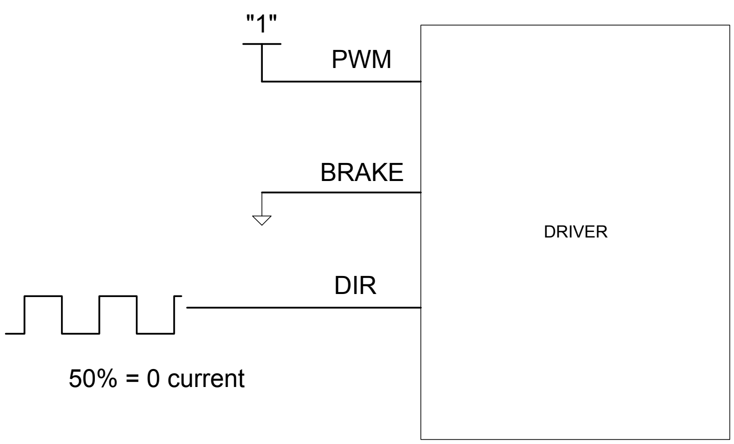


Figure 4-3. Driver Signals

PWM	Dir	Brake	Active Output Driver
H	H	L	SOURCE 1, SINK 2
H	L	L	SINK 1, SOURCE 2
L	X	L	SOURCE 1, SOURCE 2
H	H	H	SOURCE 1, SOURCE 2
H	L	H	SINK 1, SINK 2
L	X	H	NONE

Table 4-3. Driver States

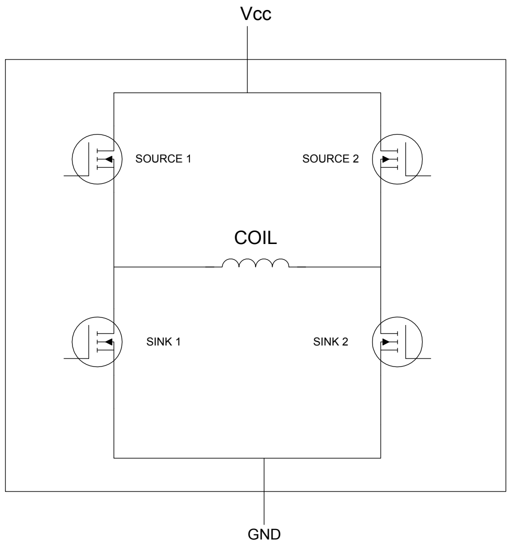


Figure 4-4. Driver H-Bridge Representation

Thermal Design

Internal ambient temperature for package design is targeted for 85 °C. Rjc for the LMD18201 is 1 °C/W. Assuming

Max power dissipation in device is:
Pd = (Tjmax – Ta)/Rja
Pd = 2A * 0.6 Ω = 1 W

Current Feedback

Current feedback is used on each winding of the motor for independent current limiting and fault detection. The current feedback voltage is fed to the microcontroller and needs to be scaled to 3.3 V max and read ±4 A. The current feedback output signal must have an anti-aliasing and high-pass filter of 10 µs.

Thermal Warning

The thermal warning flag in the driver IC is set at 145 °C. A logic low signal is asserted in this condition. The driver IC will shut down at 170 °C.

Valve Calibration Offset Procedure

The valve supplied by Woodward will have the calibration offset number written on a label attached to the valve. This number is intended to be entered into the embedded software and passed to the controller via the ValveOffset1 variable. See Control Model I/O definition and Calibration Offset in the next chapter.

Chapter 5.
Recommended Validation Tests

Single Step Model

The following tests are described using the functionality and variables provided in the example software program.

Stepper Motor Driver Circuitry and PWM Software Settings
1. Connect driver to bare motor (not Swift Valve).
2. Set MAX_PWM_COUNT and MIN_PWM_COUNT parameters to a tighter range, say ±100 counts, to prevent hardware damage or driver IC overheating.
3. Set the RunMode to a '1' for CCW or a '2' for CW. The Speed variable adjusts the velocity of rotation—if set too high or too low the motor will not rotate.
4. Verify functionality.
5. Set RunMode=3 and RunInternalPosDmd=1.
6. Verify functionality of position controller using PositionDemand to set the position. Note the PositionDemand is scaled 0...1.

Model Implementation

The following tests are intended to verify the proper interface and scaling of the control model. Descriptions are independent of the example software program.

Position demand to output scaling
1. Connect driver to motor.
2. Set Input Voltage to 24V, ±0.1 V.
3. Set ValveOffset1=0.
4. Set SetPosition1 to 0.5.
5. Monitor VoltageCoil1A, Monitor VoltageCoil1B.
6. Take data from VoltageCoil1A, VoltageCoil1B as SetPosition1 is moved from 0.500 to 0.516 in increments of 0.002 or smaller.
7. Plot VoltageCoil1A, VoltageCoil1B, vs. SetPosition1. They should trace out a sine/cosine waveform relationship, with one full period equal to 0.016 change in SetPosition1.

Calibration Offset to output scaling
1. Connect driver to motor.
2. Set Input Voltage to 24V, ±0.1 V.
3. Set SetPosition1=0.
4. Set ValveOffset1=0.
5. Monitor VoltageCoil1A, VoltageCoil1B.
6. Take data from VoltageCoil1A, VoltageCoil1B as ValveOffset1 is moved from 0 to 1.00 in increments of 0.1 or smaller.
7. Plot VoltageCoil1A, VoltageCoil1B, vs. ValveOffset1. They should trace out a sine/cosine waveform relationship, with one full period equal to 1.0 change in ValveOffset1. Note: The units of the Offset parameter is electrical revolutions.

System Validation

The following tests are recommended to validate the proper implementation of the valve/motor controller at the integrated driver/valve level.

ReZero

Upon power-up, ensure that the motor rotates in the closed direction for approximately 6 seconds. If the valve is fully open at the beginning of the ReZero (this may be difficult due to the return spring), the ReZero operation will close the valve fully.

Test:
1. Upon power-up, monitor motor shaft position.
2. Motor should rotate in closed valve direction for approximately 6 seconds and at least 1.5 revolutions.

Current Waveforms

1. Apply a 50%/sec (of full scale) ramp input to the demand input (i.e. 0–100% amplitude, 0.25 Hz triangle wave). Use a current probe to monitor coil A current and coil B current.
2. Measured during the ramp of the triangle wave, Coil A and Coil B currents should be 90 degrees out of phase , frequency=31.25 Hz, Amplitude=2.25 Apk, ±0.25 A.

Transient Dynamics

Apply 10%–90% step to the input command. Measure the output (valve or motor position) transient time. The time between 10% to 90% points of the corresponding output transient should be less than 130 ms.

Bandwidth

1. Set valve position to 50%.
2. Apply a sine wave on top of the input signal of 1% of full scale amplitude. Sweep from 0.1 to 10 Hz and measure motor or valve output. Control bandwidth should be < 5Hz (measured at the –6 dB point). Repeat this test with a 5% of full scale amplitude sine wave.

Calibration Offset

Check the scaling of the calibration offset function. Changing the calibration offset value from 0 to 1 in 0.1 increments should result in a motor position variation of 7.2 degrees, ±0.5%.

Full Stroke Accuracy

Motor rotation should be 450 degrees, ±0.25%, for a 0 to 100% input command.

Input Voltage Sensitivity

Record Motor or valve position at 21.5 V, 24 V, and 28 V. Position should be constant, within 0.1%.

Failure Mode Response

1. Set Current in Coil 1A to 0 ±0.1 A by moving the input command a small amount.
2. Verify current in Coil 1B = 2.25 A ±10%.
3. Open Coil 1B—Should detect coil error.
4. Set Current in Coil 1B to 0 ± 0.1 A.
5. Verify current in coil 1A = 2.25 A ±10%.
6. Open Coil 1A—Should detect coil error.

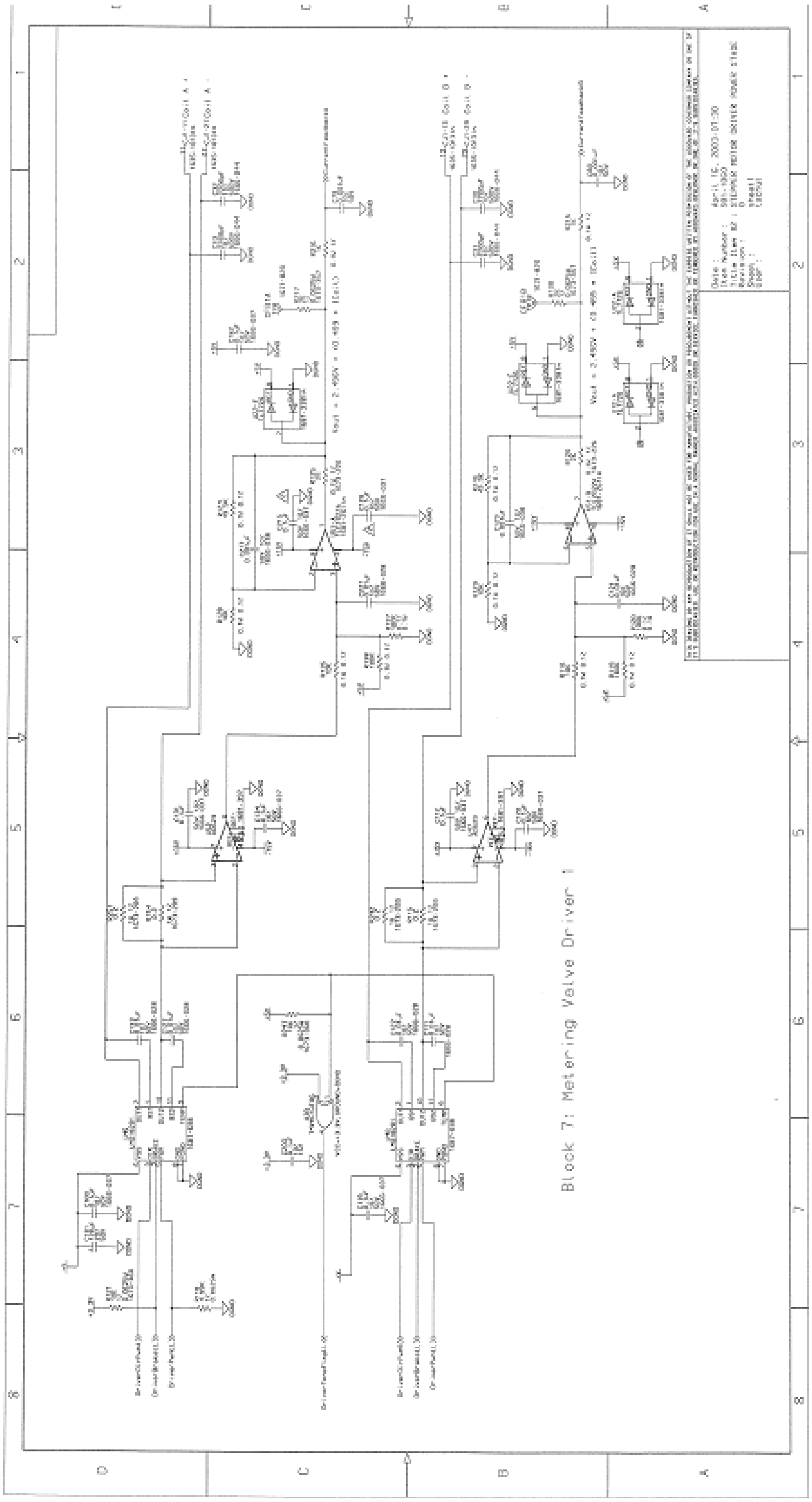
Force Margins Test

1. Connect valve at 2X intended engine system pressure.
2. Run 10%–90% and 90%–10% transients and measure flow rate at 10% and 90% points to verify correct flow.
3. Run 0%–100% and 100%–0% transients and measure flow rate at 10% and 90% points to verify correct flow.

Troubleshooting

See the Swift valve manual for troubleshooting help (Woodward manual 26202).

Appendix A.
Reference Schematic



Appendix B.
Reference Bill of Materials

STEPPER CONTROLLER HW PARTS LIST
Ref Dwg: 901-1060
Ref BOM: 601-1060

Woodward Part No.	Component Reference Designator	Description	Quan.	Manufacturer	Manufacturer Part No.
1662-083	C101	CAPACITOR-120 µF, 50 V	1	NICHICON CORPORATION PANASONIC	UPL-1H121MPH6 ECA1-HFQ121
1687-698	U40, U45	IC-LMD18201, H-BRIDGE	2	NATIONAL SEMICONDUCTOR	LMD18201T
1666-028	C111, C121, C122, C127, C133, C221	CAPACITOR - .01 µF 50 V 10% X7R 0805	6	AVX AVX	08055C103KAT2A 08055C103KATMA
1666-037	C100, C112, C119, C120, C128, C131, C134, C182, C215	CAPACITOR - .1 µF 50 V 10% X7R 0805	9	AVX KEMET ELECTRONIC CORP.	08055C104KATMA C0805C104K5RAC
1666-038	C79, C80, C129, C217	CAPACITOR - .001 µF 50 V 10% NPO 0805	4	AVX AVX	08055A102KAT2A 08055A102KATMA
1666-044	C30, C31, C32, C33	CAPACITOR – 2200 pF 500 V 10% X7R 1206	4	AVX AVX	12067C222KAT2A 12067C222KATMA
1666-086	C203	CAPACITOR - .1 µF, 16 V, 0603 CERAMIC	1	AVX KEMET ELECTRONIC CORP.	0603YC104KATMA C0603C104K4RAC
1671-820	TP8, TP18	TERMINAL - SMT TEST POINT	2	COMPONENTS CORP. KEYSTONE	TP-108-02 5015
1673-141	R116, R128, R129, R135	RESISTOR – 10 kΩ .1 W .1% 0805	4	DALE ELECTRONICS INC. IRC	TNPW08051002BT-9 W0805R021002B
1673-144	R119, R120, R137, R138	RESISTOR – 100 kΩ .1 W .1% 0805	4	DALE ELECTRONICS INC. IRC	TNPW08051003BT-9 W0805R021003B
1673-172	R215, R216	RESISTOR – 1 kΩ .1 W 1% 0805	2	DALE ELECTRONICS INC. KOA SPEER ELECTRONICS	CRCW08051001FRT1 RK73H2AT1001F
1673-209	R115, R134, R242, R251	RESISTOR - .2 Ω 1 W 1% 2512	4	DALE ELECTRONICS INC.	WSL2512-.2-1%
1673-226	R125, R126	RESISTOR – 2 kΩ .1 W 1% 0805	2	DALE ELECTRONICS INC. IRC	CRCW08052001FRT1 WCR08052001FPLT
1673-279	R124, R246	RESISTOR - 49.9 kΩ .1 W .1% 0805	2	DALE ELECTRONICS INC. IRC	TNPW08054992BT-9 W0805R024992B
1673-357	R130, R217	RESISTOR – 2 kΩ, 1% 0603	2	KOA SPEER ELECTRONICS SEI ELECTRONICS, INC	RK73H1JT2001F RMC1/16-2K-1%
1673-358	R118	RESISTOR – 4.99 kΩ, 1% 0603	1	KOA SPEER ELECTRONICS SEI ELECTRONICS, INC	RK73H1JT4991K RMC1/16-4.99K-1%
1673-360	R121, R247	RESISTOR – 10 kΩ, 1% 0603	2	KOA SPEER ELECTRONICS SEI ELECTRONICS, INC	RK73H1JT1002F RMC1/16-10K-1%
1681-261	U51	IC - TLE2022AMD OP AMP SMT	1	TEXAS INSTRUMENTS TEXAS INSTRUMENTS	TLE2022AID TLE2022AMD
1681-338	U12, U22	IC - 7726 HEX CLAMPING CIRCUIT	2	TEXAS INSTRUMENTS	TL7726QD
1681-397	U47, U52	IC - AD629 HIGH CMRR INSTRUMENTATION AMP	2	ANALOG DEVICES ANALOG DEVICES	AD629AR AD629BR
1681-415	U30	IC - 74VHC1GT86 SINGLE GATE 2 INPUT XOR	1	ON SEMICONDUCTOR	MC74VHC1GT86DF

We appreciate your comments about the content of our publications.

Send comments to: icinfo@woodward.com

Please reference publication 26223.

WOODWARD

PO Box 1519, Fort Collins CO 80522-1519, USA
1000 East Drake Road, Fort Collins CO 80525, USA
Phone +1 (970) 482-5811 • Fax +1 (970) 498-3058

Email and Website—www.woodward.com

Woodward has company-owned plants, subsidiaries, and branches, as well as authorized distributors and other authorized service and sales facilities throughout the world.

Complete address / phone / fax / email information for all locations is available on our website.

2012/9/Colorado