TI Designs
Closed Loop Stepper Motor Design With Encoder for Stall-Detection Reference Design

Description

The TIDA-01370 reference design demonstrates the stall-detection in a stepper motor driven by the DRV8880 in the presence of a high-resolution closed loop feedback obtained with an optical rotary incremental encoder looping back to the microcontroller MSP430F2617 the rotor actual position.

As a result, this TI Design can increase the precision and accuracy of the position and vibration estimation in an application using a stepper motor.

Resources

TIDA-01370	Design Folder
DRV8880	Product Folder
MSP430F2617	Product Folder
SN74LV4T125	Product Folder
TPS62175	Product Folder
TPS73501	Product Folder

TI E2E™ Community

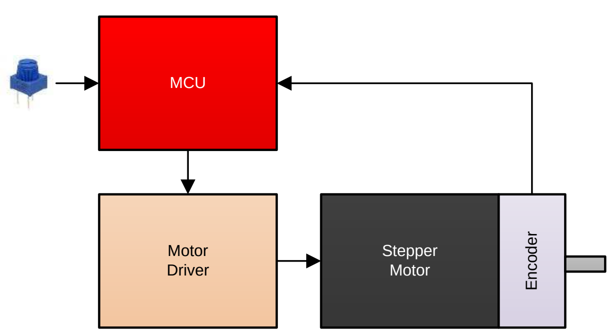
ASK Our E2E Experts

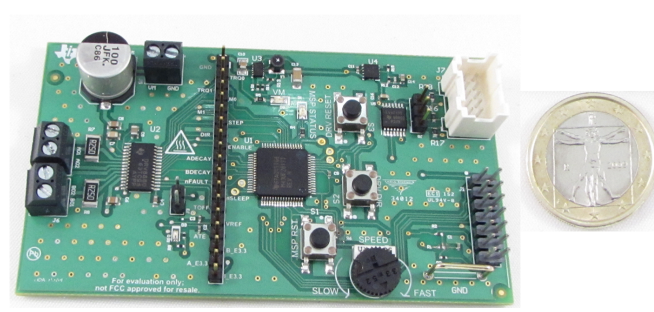
Features

- 6.5 to 45 V (Limited to 28 V Due to Buck V_{IN} Limitations) Operation With up to 2.0-A Full-Scale Current for Driver
- 5-V Optical Rotary Incremental Encoder Connected to Board With 12-Pin, 2-mm Shrouded Header
- High-Resolution Stall Detection
- Motor Resonance Frequency Detection
- Motor Vibration Width Detection
- Closed Loop Algorithm

Applications

- Robotics Control
- General Purpose Position Encoders or Conveyor Belts
- Manufacturing Robots, Textiles, and Sewing Machines
- Banking Automation
- Multi-Axis Printers





An IMPORTANT NOTICE at the end of this TI reference design addresses authorized use, intellectual property matters and other important disclaimers and information.

1 System Overview

1.1 System Description

Stepper motors move in distinct steps, known as fixed angular increments, depending on the motor step angle. Therefore, the motor moves a known amount each time a step is issued and then the position of the rotor can be expected after every full step. This ability makes stepper motors very useful for positioning systems to be able to monitor the rotor position. However, this is valid only under normal operating conditions.

In case of overload, loss of steps, stall condition, and other unexpected events, the stepper motor can lose its ability to predict the exact rotor position because there is no synchronization between the actual rotor movements and the driving commands anymore. These stepper motors indeed work in open loop configurations without the possibility to check the rotor position. Due to this fact, many stepper motor designers adopt an overload torque design, a longer and more expensive motor testing and tuning, to avoid malfunctions, resonant, and noise issues occurring in the stepper motor. This TI Design uses an optical rotary incremental encoder attached to the rear part of the motor shaft in order to detect every instants the rotor position revealing the stall-condition and step-missing events.

This reference design highlights the eventual loss of synchronization of the motor movements with its driving inputs. By using an encoder, the actual output of the system constantly loops back into the controller that determines the rotor conditions and can eventually determine the future and new outputs by compensating the errors. However, this TI Design has the target to only detect the stall condition of the motor using the closed loop and flag an LED in this case.

There are many benefits achieved by using a closed loop stepper motor; the TIDA-01370 has a greater efficiency during load fluctuations, resonance free system, stall detection, intelligent current regulation, rapid commissioning without expensive tuning, and precise positioning due to monitoring and correction, thus allowing a better exploitation of the motor performance.

1.1.1 Encoder

The optical rotary incremental encoder used in this TI Design has two code tracks with sectors positioned 90 degrees out of phase (A and B channels) plus an additional track (I channel) known as the indexer signal.

This encoder has two different patterns of dark and light lines that alternate on the disk's surface. When a light source (an LED) enlightens the disk, the light transmitted is captured by light photo detectors, which generate the pulses in the forms of sinusoidal waveforms. After that, an analog-to-digital system converts these signals into square waves.

Therefore, these encoder outputs are then read by the microcontroller and compared to the driving inputs in order to check whether the synchronization between the driving inputs and rotor movement is still achieved.

By monitoring both the number of pulses and the relative phase of signals A and B, it is possible to track both the position and direction of rotation. In addition, this encoder uses a third output channel called indexer (I), which points out the reference signal and supplies a single pulse per revolution. This signal is used for precise determination of a reference position. Figure 1 briefly shows the functioning of the used encoder:

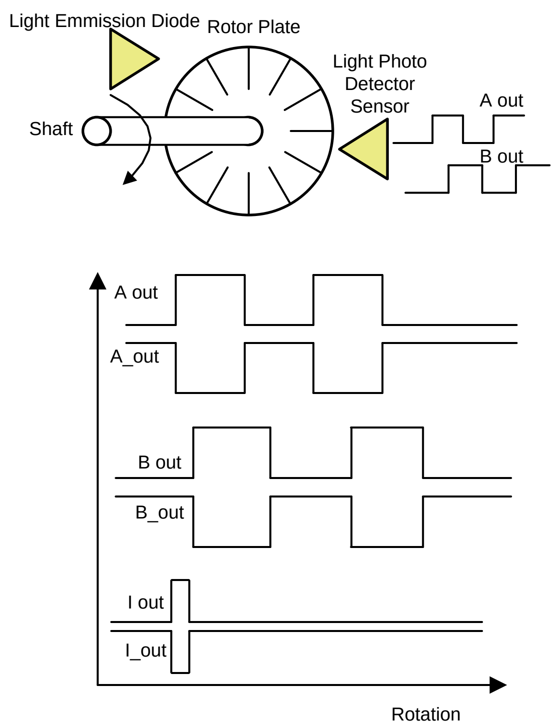


Figure 1. Encoder Structure (Above) and Encoder Outputs (Below)

1.1.2 Closed Loop Control Algorithm

This section describes the control algorithm implemented for the closed loop stepper motor to detect a stall condition.

In the absence of the feedback loop, the microcontroller determines the motion of a stepper motor just by generating two signals: STEP and DIR. However, there is not any control or double check on motor spinning with respect to the microcontroller's commands. This problem is therefore solved by using a feedback loop as can be seen in Figure 2.

In this case, the microcontroller receives the feedback signals A and B from the encoder, and at the same time it generates the STEP and DIR commands to be sent to the driver. Under normal operating conditions, the encoder outputs should match with the STEP commands sent by the microcontroller. Then nothing happens and the motor keeps spinning.

Otherwise, if the encoder outputs do not match with the STEP inputs, there is an unexpected condition that can reveal into a stall event or loss of steps or resonance situation. The encoder has a resolution of 4000 increments per revolution. This means that according to the different step modes chosen, either full step or microstepping, the user should be able to see into a scope a certain number of encoder outputs per each step interval. These numbers should match under normal conditions; otherwise, if the encoder counts are less than expected, this might be due to a stall event as described in Section 4.

This reference design can detect the unpredicted conditions and detect and flag the stall condition with an LED.

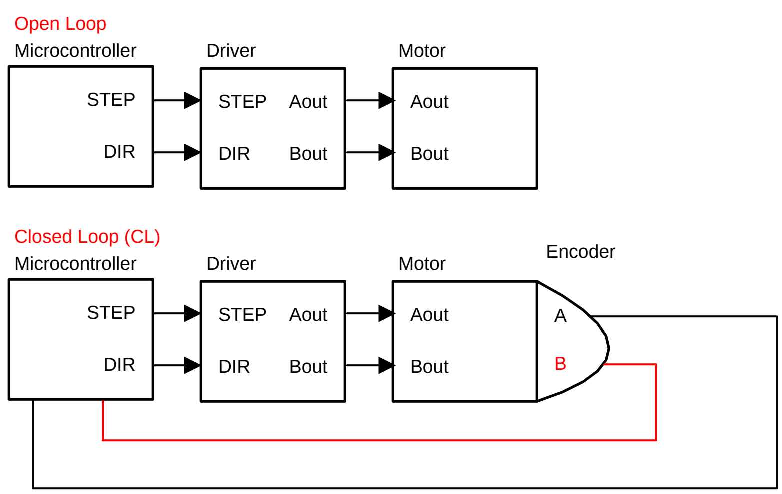


Figure 2. Open Loop versus Closed Loop Diagram

1.2 Key System Specifications

Table 1. Key System Specifications

PARAMETER	DESCRIPTION	VALUE
TI SYSTEM LEVEL DESCRIPTION (DRIVER, LDO, BUCK, VOLTAGE LEVEL TRANSLATOR)		
Motor voltage (VM)	Supply voltage for the motor driver	6.5 to 45 V
Driver current (I)	Full-scale driver current	2:00 AM
Buck input voltage (Vin_BUCK)	Input voltage for the buck converter	4.7 to 28 V
Buck output voltage (Vout_BUCK)	Output voltage for the buck converter	5 V
LDO input voltage (VIN_LDO)	Input voltage for the LDO converter	2.7 to 6.5 V
LDO output voltage (Vout_LDO)	Fixed output voltage for the LDO	3.3 V
Buck current	Buck maximum output current	500 mA
LDO current	LDO maximum output current	500 mA
Supply voltage translator	Single supply voltage for the voltage level translator	3.3 V
Step frequency range	Step frequency interval	50 to 1000 Hz
OPTICAL ENCODER (NANOTEC NOE2 SERIES)		
Encoder voltage	Supply voltage for the encoder	5 V
Encoder resolution (increments per revolution)	Resolution of the encoder	4000
MOTOR SPECIFICATION (NANOTEC SC4118L1804-EN005K)		
Motor current	Maximum current in the motor phases	1.8 A
Motor resistance (Ω) at 25°C	Resistance motor or phase	1.75 ± 15%
Motor inductance (mH) at 1 kHz	Inductance motor or phase	3.3 ± 20%
Motor speed range	Motor speed range used in this design	15 to 300 rpm
Motor step angle (°)	Motor step angle	1.8
CLEAN DAMPER (ORIENTAL MOTOR D4CL-5.0F)		
Damper	Damper inertia of the clean damper	0.186 oz-in^2

1.3 Block Diagram

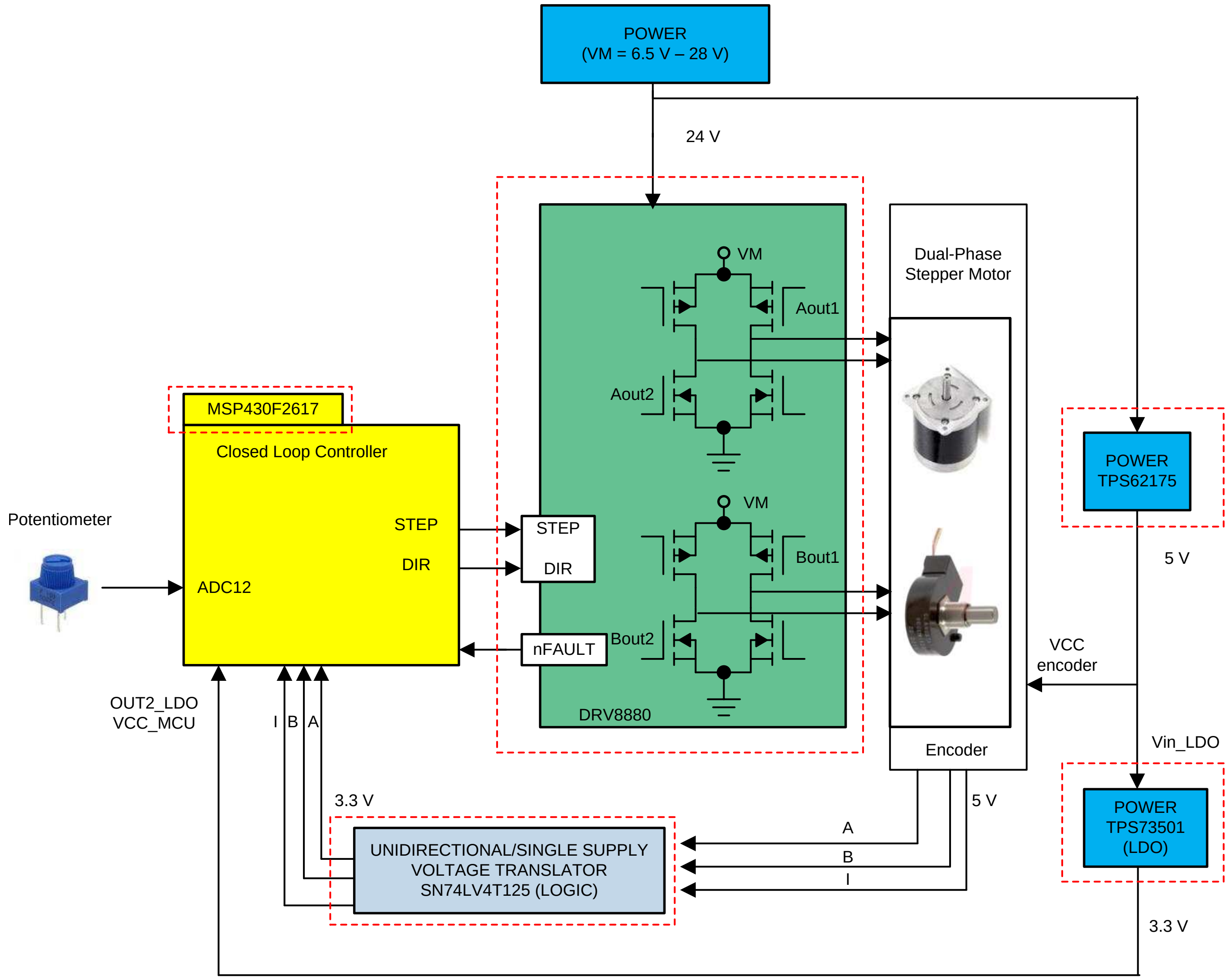


Figure 3. TIDA-01370 Block Diagram

1.4 Highlighted Products

1.4.1 DRV8880

The DRV8880 is a bipolar stepper motor driver for industrial applications. The device has two N-channel power MOSFET H-bridge drivers and a microstepping indexer. The DRV8880 is capable of driving 2.0 A full-scale current or 1.4-A rms current (with proper PCB ground plane for thermal dissipation and at 24 V and TA = 25°C).

Auto-Tune™ automatically tunes stepper motors for optimal current regulation performance and compensates for motor variation and aging effects. Additionally, slow, fast, and mixed decay modes are available.

The STEP/DIR pins provide a simple control interface. The device can be configured in full-step up to 1/16-microstep modes. A low-power sleep mode is provided for very low quiescent current standby using a dedicated nSLEEP pin.

Internal protection functions are provided for undervoltage, charge pump faults, overcurrent, short circuits, and over-temperature. Fault conditions are indicated by a nFAULT pin. This driver is used to drive a dual-phase stepper motor.

Features:

- Microstepping stepper motor driver:
  - STEP/DIR interface
  - Up to 1/16 microstepping indexer
  - Non-circular and standard ½ step modes
- 6.5- to 45-V operating supply voltage range
- Multiple decay modes to support any motor (Auto-Tune):
  - Mixed decay
  - Slow decay
  - Fast decay
- Adaptive blanking time for smooth stepping
- Configurable off-time PWM Chopping
  - 10-, 20-, or 30-μs off-time
- 3.3-V, 10-mA LDO regulator
- Low-current sleep mode (28 μA)
- Small package and footprint 28 HTSSOP (PowerPAD™)
- 28 WQFN (PowerPAD)
- Protection features:
  - VM undervoltage lockout (UVLO2)
  - Logic undervoltage (UVLO1)
  - Charge pump undervoltage (CPUV)
  - Overcurrent protection (OCP):
    - Latched OCP mode
    - Retry OCP mode
  - Thermal shutdown (TSD)
  - Fault condition indication pin (nFAULT)

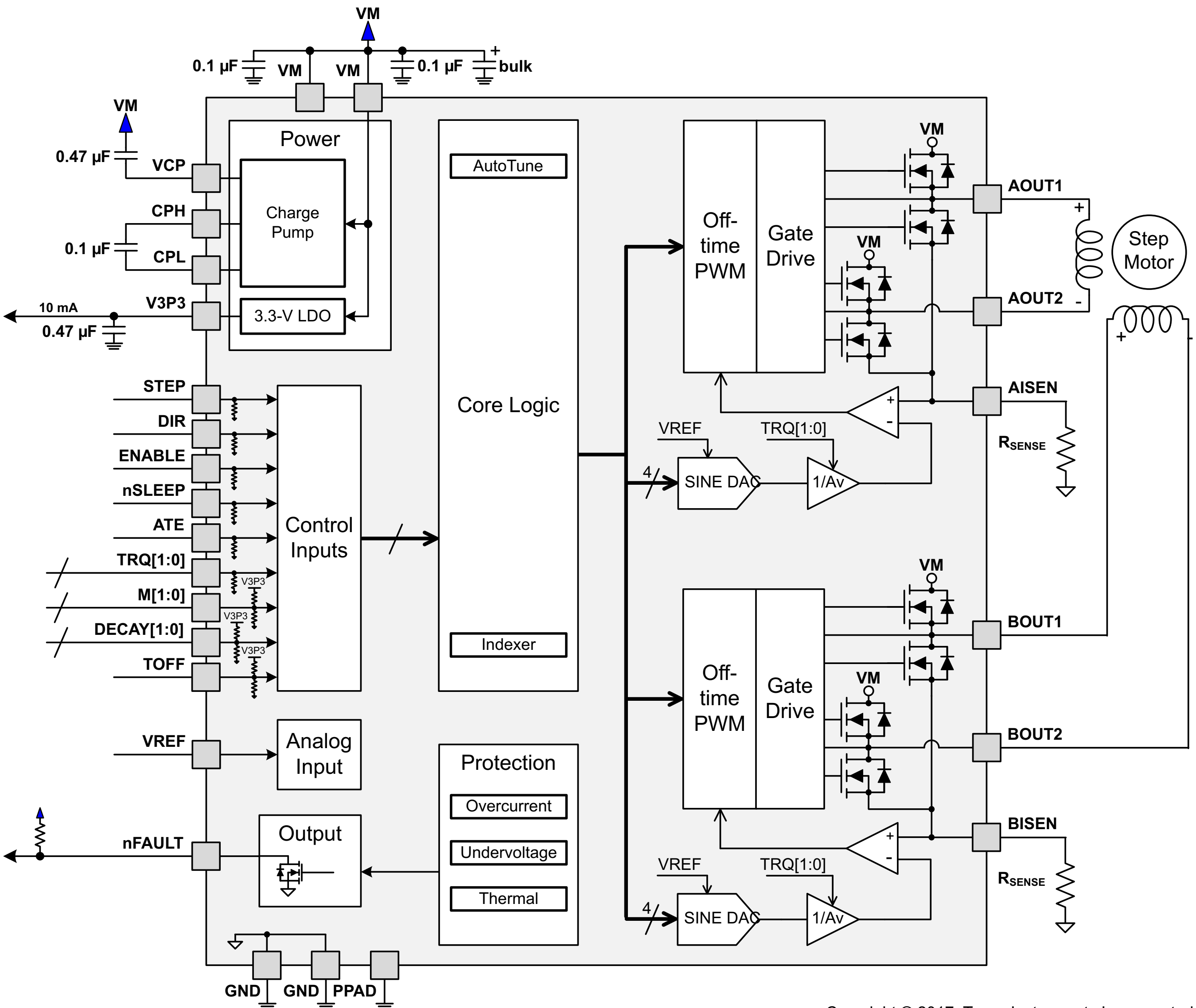


Figure 4. DRV8880 Functional Block Diagram

1.4.2 MSP430F2617

The Texas Instruments MSP430™ family of ultra-low-power microcontrollers consists of several devices featuring different sets of peripherals targeted for various applications. The architecture, combined with five low-power modes, is optimized to achieve extended battery life in portable measurement applications. The device features a powerful 16-bit RISC CPU, 16-bit registers, and constant generators that contribute to maximum code efficiency. The calibrated digitally controlled oscillator (DCO) allows wake-up from low-power modes to active mode in less than 1 µs.

The MSP430F261x series are microcontroller configurations with two built-in 16-bit timers, a fast 12-bit analog-to-digital converter (ADC), a comparator, dual 12-bit digital-to-analog converters (DACs), four universal serial communication interface (USCI) modules, DMA, and up to 64 I/O pins.

Typical applications include sensor systems, industrial control applications, and hand-held meters. The 12-mm×12-mm LQFP-64 package is also available as a non-magnetic package for medical imaging applications.

Features:

- Low supply voltage range 1.8 to 3.6 V
- Ultra-low-power consumption:
  - Active mode: 365 µA at 1 MHz, 2.2 V
  - Standby mode (VLO): 0.5 µA
  - Off mode (RAM retention): 0.1 µA
- Wake-up from standby mode in less than 1 µs
- 16-bit RISC architecture, 62.5-ns instruction cycle time
- Three-channel internal DMA
- 12-bit ADC with internal reference, sample-and-hold, and autoscan feature
- Dual 12-bit DACs with synchronization
- 16-bit Timer_A with three capture/compare registers
- 16-bit Timer_B with seven capture/compare-with-shadow registers
- On-chip comparator
- Four USCIs:
  - USCI_A0 and USCI_A1: Enhanced UART supporting auto-baudrate detection, IrDA encoder and decoder, synchronous SPI
  - USCI_B0 and USCI_B1: I^2C, synchronous SPI
- Supply voltage supervisor and monitor with programmable level detection
- Brownout detector
- Bootstrap loader
- Serial onboard programming, no external programming voltage needed, programmable code protection by security fuse
- MSP430F2617: 92KB + 256B Flash Memory, 8KB RAM
- Available in 80-pin quad flat pack (LQFP), 64-pin LQFP, and 113-pin ball grid array (BGA)

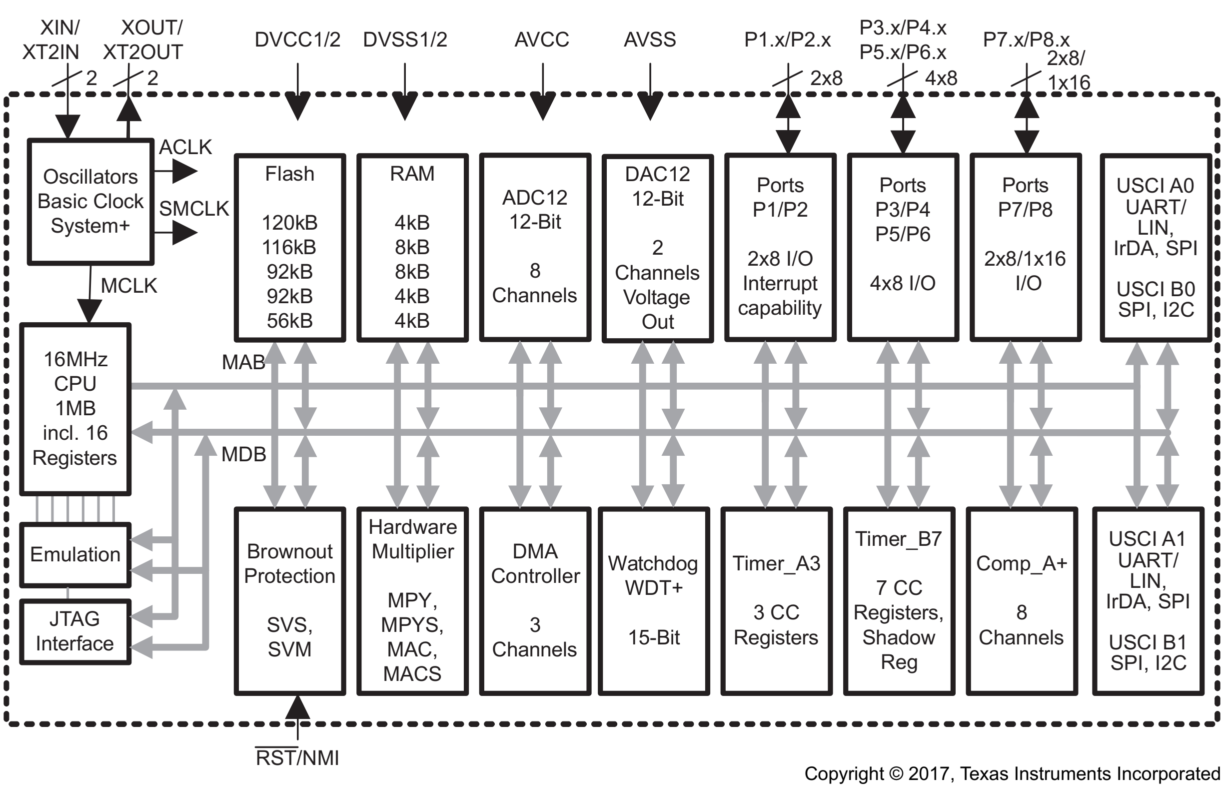


Figure 5. MSP430F2617 Functional Block Diagram

1.4.3 TPS62175

The TPS6217x is a high-efficiency synchronous step-down DC/DC converter, based on the DCS-Control™ topology.

With a wide operating input voltage range of 4.75 to 28 V, this device is ideally suited for systems powered from multi-cell Li-Ion as well as 12 V and even higher intermediate supply rails, providing up to a 500-mA output current.

The TPS6217x automatically enters power save mode at light loads to maintain high efficiency across the whole load range. It also features a sleep mode to supply applications with advanced power save modes like ultra-low-power microcontrollers. The power good output may be used for power sequencing or power-on reset.

The device features a typical quiescent current of 22 μA in normal mode and 4.8 μA in sleep mode. In sleep mode, the efficiency at very low load currents can be increased by as much as 20%. In shutdown mode, the shutdown current is less than 2 μA and the output is actively discharged. The TPS6217x, available in an adjustable and a fixed output voltage version, is packaged in a small 2-mm×3-mm 10-pin WSON package.

This buck steps down the V_M to 5 V so to allow the encoder to operate.

Features:

- DCS-Control topology
- Input voltage range: 4.75 to 28 V
- Quiescent current: typically 4.8 μA (sleep mode)
- 100% duty cycle mode
- Active output discharge
- Power Good output
- Output current: 500 mA
- Output voltage range: 1-V DC to 6 V
- Switching frequency: typically 1 MHz
- Seamless power save mode transition
- Undervoltage lockout
- Short-circuit protection
- Over-temperature protection
- Available in 2-mm×3-mm 10-pin WSON package

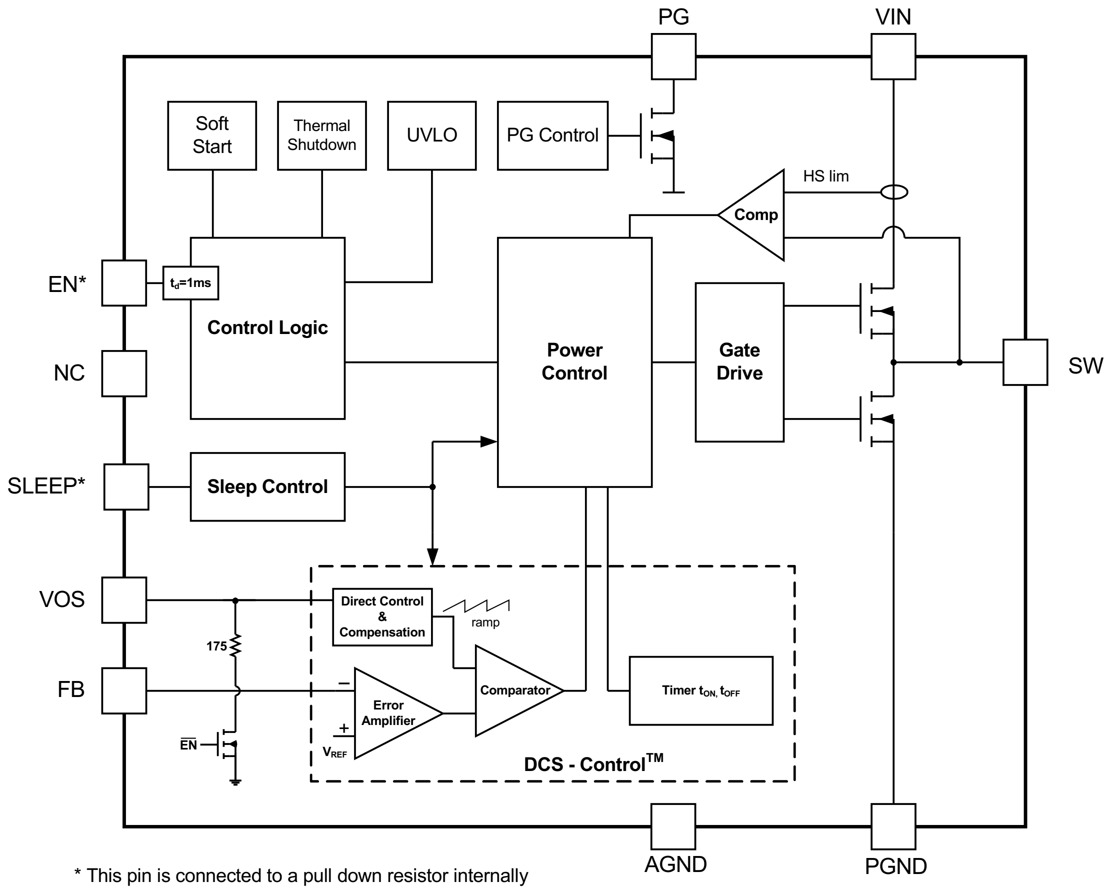


Figure 6. TPS62175 Functional Block Diagram

1.4.4 TPS73501

The TPS735 family of low-dropout (LDO), low-power linear regulators offers excellent AC performance with very low ground current. High power-supply rejection ratio (PSRR), low noise, fast start-up, and excellent line and load transient responses are provided while consuming a very low 46-µA (typical) ground current.

The TPS735 family of devices is stable with ceramic capacitors and uses an advanced BiCMOS fabrication process to yield a typical dropout voltage of 280 mV at 500-mA output. The TPS735 family of devices uses a precision voltage reference and feedback loop to achieve overall accuracy of 2% (VOUT > 2.2 V) over all load, line, process, and temperature variations. This family of devices is fully specified from T_A = –40°C to 125°C and is offered in a low-profile, 3-mm×3-mm SON-8 package and a 2-mm×2-mm SON-6 package.

This LDO steps down the buck V_{OUT} of 5 V to 3.3 V to power the MSP430 V_{CC} and the voltage translator logic single supply rail.

Features:

- Input voltage: 2.7 to 6.5 V
- 500-mA LDO with EN
- Low I_Q: 46 µA
- Multiple output voltage versions available:
  - Fixed outputs of 1.2 to 4.3 V
  - Adjustable outputs from 1.25 to 6 V

- High PSRR: 68 dB at 1 kHz
- Ultra-low noise: 13.2 µV_{RMS}
- Fast start-up time: 45 µs
- Stable with standard, 2.2-µF, low-ESR output capacitor
- Excellent load and line transient response
- 2% overall accuracy (load, line, and temperature, V_{OUT} > 2.2 V)
- Very LDO: 280 mV at 500 mA
- 2-mm × 2-mm SON-6 Package
- 3-mm × 3-mm SON-8 Package

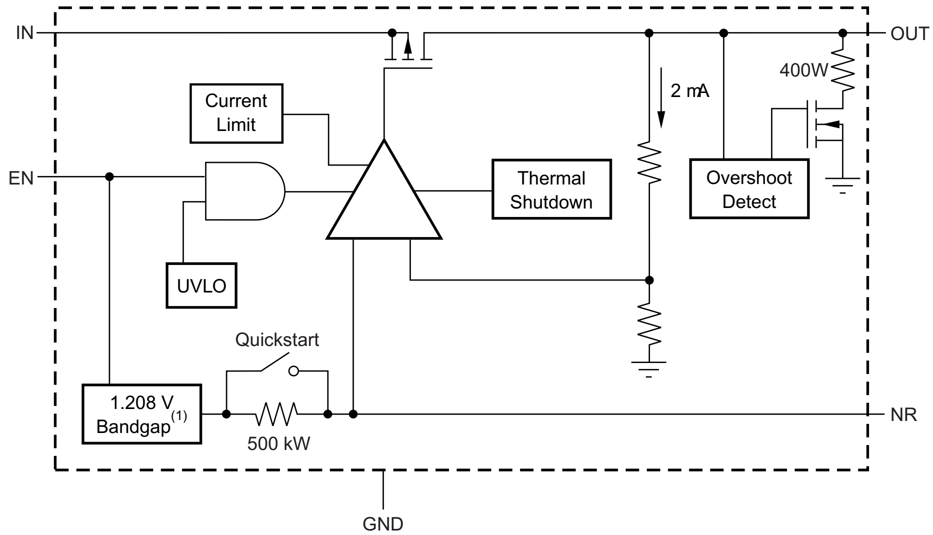


Figure 7. TPS73501 Functional Block Diagram (Fixed Output Voltage)

1.4.5 SN74LV4T125

The SN74LV4T125 is a low-voltage CMOS buffer gate that operates at a wider voltage range for portable, telecom, industrial, and automotive applications. The output level is referenced to the supply voltage and is able to support 1.8-V, 2.5-V, 3.3-V, and 5-V CMOS levels.

The input is designed with a lower threshold circuit to match 1.8-V input logic at V_{CC} = 3.3 V and can be used in 1.8- to 3.3-V level-up translation. In addition, the 5-V tolerant input pins enable down translation (for example, a 3.3-V to 2.5-V output at V_{CC} = 2.5 V). The wide VCC range of 1.8 to 5.5 V allows the generation of desired output levels to connect to controllers or processors.

The SN74LV4T125 device is designed with current-drive capability of 8 mA to reduce line reflections, overshoot, and undershoot caused by high-drive outputs.

Features:

- Single-supply voltage translator at 5.0-V, 3.3-V, 2.5-V, and 1.8-V V_{CC}
- Operating range: 1.8 to 5.5 V
- Up translation:
  - 1.2 to 1.8 V at 1.8-V V_{CC}
  - 1.5 to 2.5 V at 2.5-V V_{CC}
  - 1.8 to 3.3 V at 3.3-V V_{CC}
  - 3.3 V to 5.0 V at 5.0-V VCC
- Down translation:
  - 3.3 to 1.8 V at 1.8-V V_{CC}

  - 3.3 to 2.5 V at 2.5-V V_{CC}
  - 5.0 to 3.3 V at 3.3-V V_{CC}
- Logic output is referenced to V_{CC}
- Characterized up to 50 MHz at 3.3-V V_{CC}
- 5.5-V tolerance on input pins
- –40°C to 125°C operating temperature range
- Pb-free packages available: SC-70 (RGY) 3.5 × 3.5 × 1 mm
- Latch-Up performance exceeds 250 mA per JESD 17
- ESD performance tested per JESD 22:
  - 2000-V Human-Body Model (A114-B, Class II)
  - 200-V Machine Model (A115-A)
  - 1000-V Charged-Device Model (C101)
- Supports standard logic pinouts
- CMOS Output B compatible with AUP125, LVC125; refer to the VIH/VIL and output drive for lower V_{CC} condition

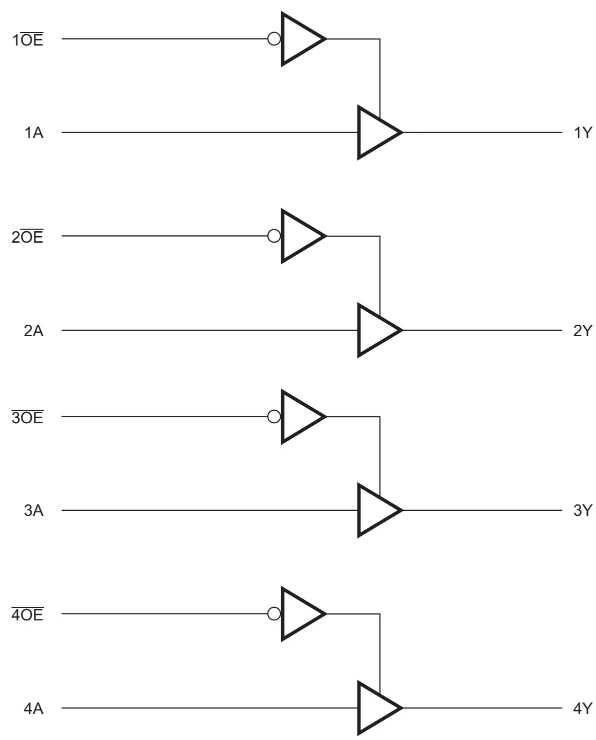


Figure 8. SN74LV4T125 Functional Block Diagram

2 System Level Description

Figure 9 shows the stepper driver DRV8880 driving a dual-phase stepper motor. The driver is controlled through the interface STEP/DIR by the microcontroller MSP430F2617 (as shown in Figure 10), which sends the pulses to the step input of the driver so to control the motion of the motor.

Moreover, a potentiometer communicates with the ADC inside the microcontroller, which varies the speed of the motor from a minimum speed (20-ms step interval equal to a 15-rpm motor speed) to a maximum speed (2-ms step interval equal to a 150-rpm motor speed).

Furthermore, the motor has an encoder attached to the rear part of its shaft. The encoder is used to close the control loop in order to send back to the microcontroller the effective steps issued by the motor, thus verifying at each cycle the synchronism between issued and actual rotor position. The encoder used has two output signals A and B in binary form, which are 90 degrees out of phase. By comparing on an oscilloscope the pattern of these encoder signals with the step frequency's pattern (STEP pulses sent from the MCU to the driver), it is then possible to detect the stall condition, the direction of movement of the rotor (clockwise or counterclockwise), its speed, and its position.

This encoder works at 5 V and is powered by the Texas Instruments buck converter TPS62175. However, the encoder outputs must be read and processed by the microcontroller working at 3.3 V; that is why this TI Design needs a voltage level translator (SN74LV4T125) to shift these 5-V pulses to 3.3-V signals. The encoder resolution is 4000 increments per revolution, while the motor step angle is 1.8 degree. Using the full step mode means the rotor needs to do 200 steps to complete one full mechanical revolution.

By comparing the encoder resolution with the motor step angle, at least 20 encoder outputs each step are issued to the driver in order to match the synchronization between the actual rotor movement and the encoder outputs. In case of a stall condition, the encoder outputs should be zero, but notice that stalling the motor, rather by using an electromagnetic brake or a normal pulley, still gets around five to nine encoder outputs due to the vibrations of the motor, even if it is stalled. That is the reason why this TI Design detects and flags a stall condition when the encoder counts are less than 10 as it is done in the software. The TIDA-01370 design has the target to detect and flag the stall condition of the stepper motor by using the encoder.

Figure 9 shows that the STEP input is the pin where the step pulses are sent from the MCU to the driver. The ENABLE and DIR pins are also controlled by the MCU and communicate with the driver.

The outputs of the driver connecting the driver to the motor are called Aout1/2 and Bout1/2. M0 and M1 are pins driven by the MCU to set the full step mode (M0 and M1 both low). The ADECAY, BDECAY, and AUTOTUNE pins are driven by the MCU to have an Auto-Tune decay mode for the phase current used for the phase current regulation. TRQ0, TRQ1, and VREF are driven as well by the MCU and set the full scale current limit threshold. The nFAULT pin driven by the MCU and pulled up to 3.3 V through a pullup resistor should detect a fault of the driver itself either due to VM undervoltage (UVLO2), logic undervoltage (UVLO1), charge-pump undervoltage (CPUV), overcurrent (OCP), or thermal shutdown (TSD). In case of a fault, the nFAULT LED stays on (red). AISEN and BISEN could be used for current sensing purposes.

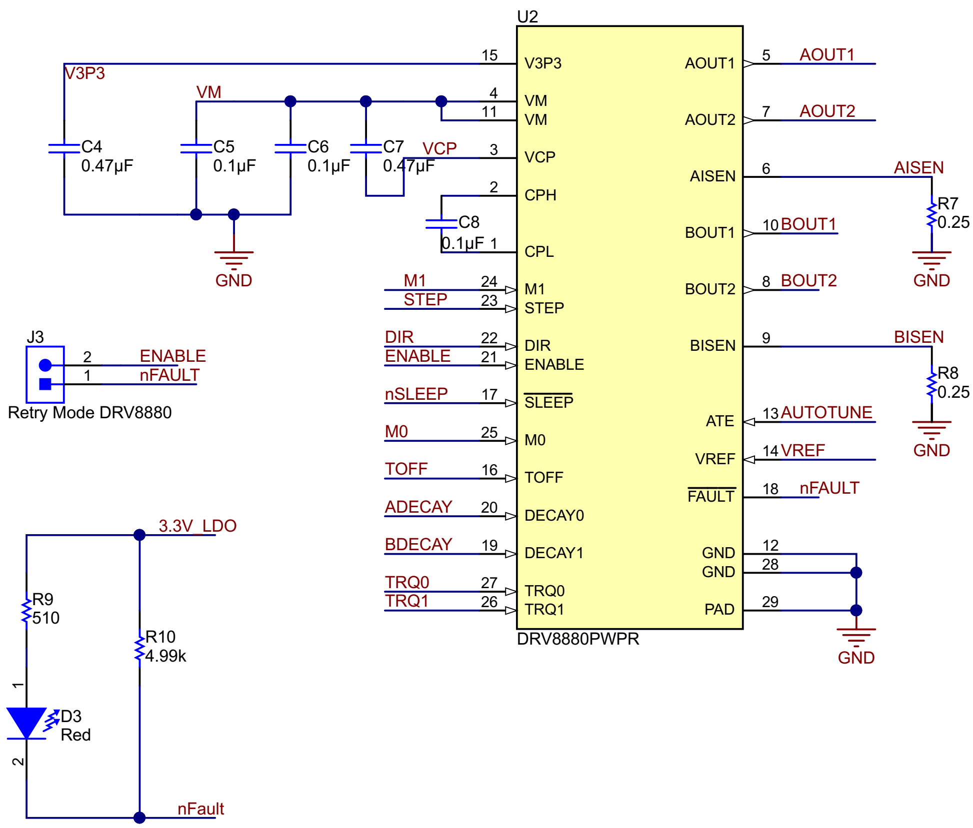


Figure 9. DRV8880 Schematic

Figure 10 depicts the MCU MSP430F2617. The MCU controls three buttons: DRV_DIR, used to change direction of the rotor movement; DRV_RESET, used to reset the driver; and MSP_RST, used to reset the MCU.

The POT input receives a varying voltage obtained by manually moving the potentiometer. Then the 12-bit ADC inside the MCU should be able to convert this voltage into a frequency which then is sent to the driver through the STEP pin.

A/B/I_Encoder 3.3 V are the three encoder outputs in binary form sent back from the encoder and passed through the voltage level translator, which close the loop between the stepper motor and the MCU.

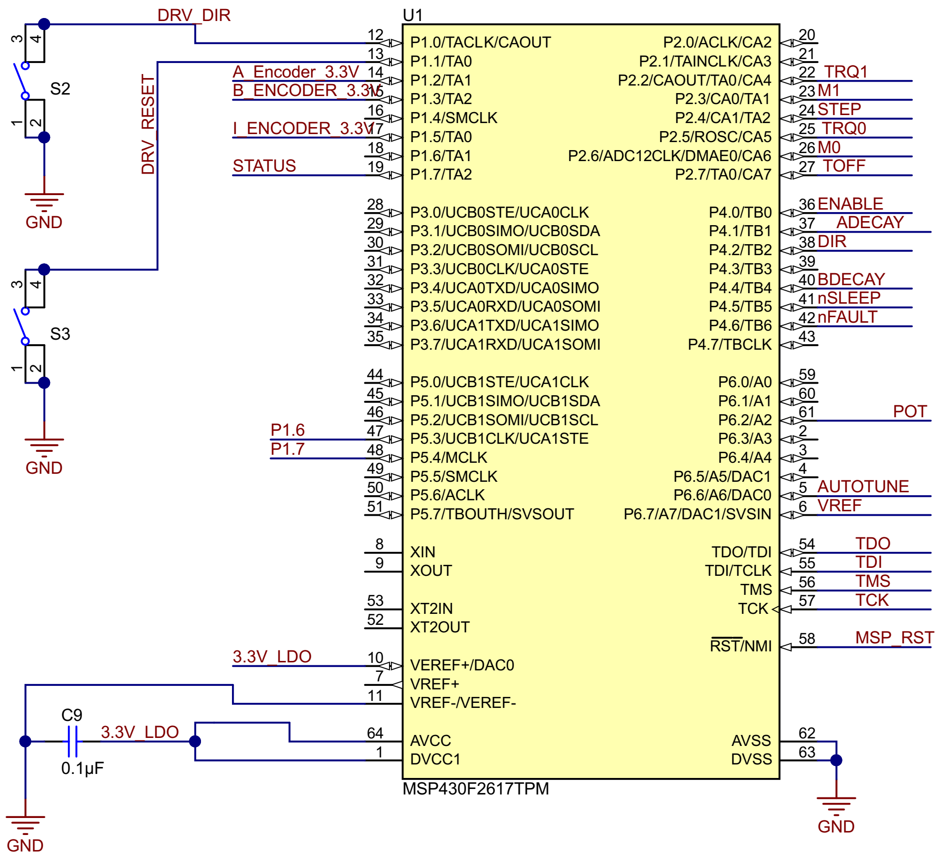


Figure 10. MSP430F2617 Schematic

Figure 11 describes the encoder header J7 where the encoder pins will be connected to the board and the voltage level translator SN74LV4T125 from Texas Instruments is used to shift the encoder output signals from 5 to 3.3 V to make them processed by the MCU working at 3.3 V.

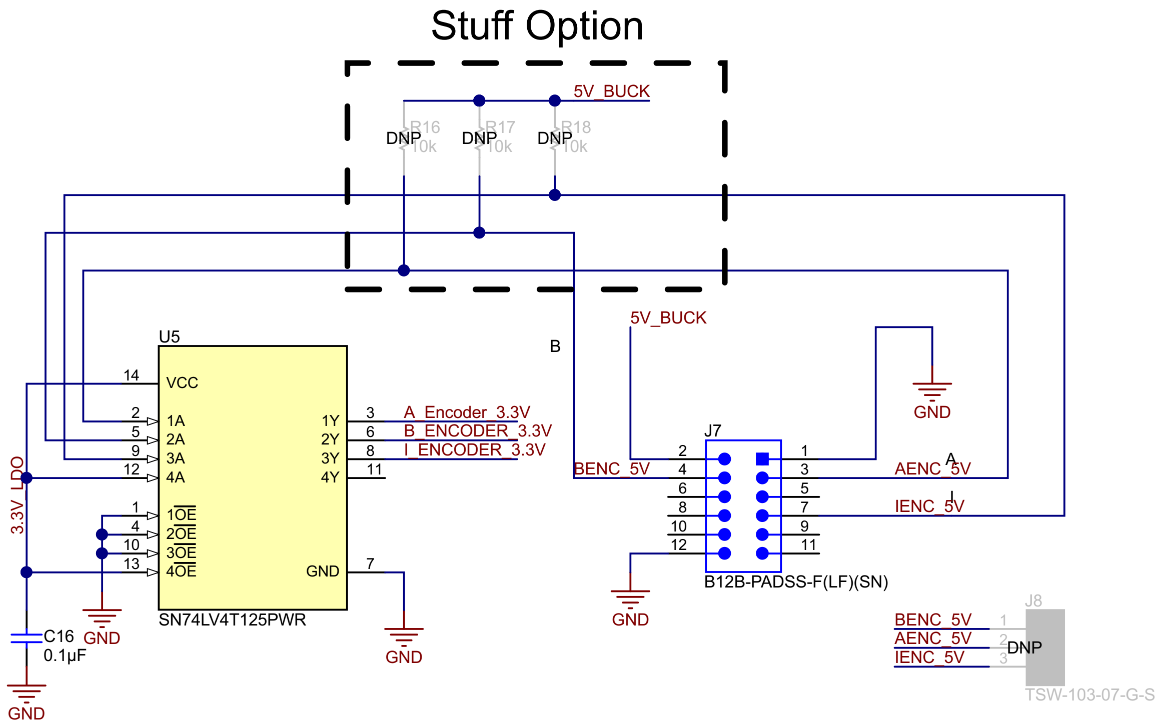


Figure 11. Encoder Header and Voltage Level Translator Schematic

Figure 12 represents the potentiometer (POT), which varies from 0 to 3.3 V in order to change the motor speed from 15 to 150 rpm. The MSP_RST button can reset the whole MCU. Instead, the STATUS LED is set to solid on (green) while the motor is spinning under normal conditions, and blinks repeatedly under a stall condition. The VM LED just indicates that the board is powered up and in this case is solid on (green).

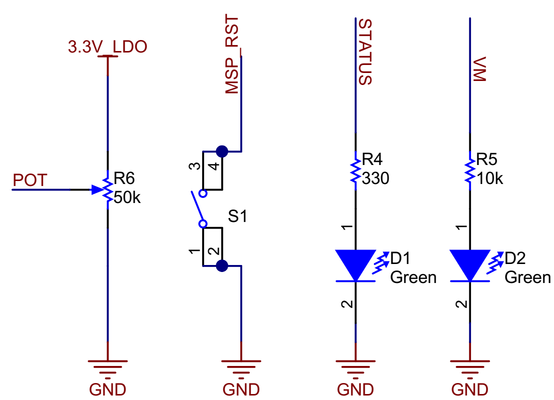


Figure 12. POT, MSP_RST Button, STATUS, and VM LED Schematic

Figure 13 represents the power management of the board. On the left, there is the step-down converter TPS62175, which steps the VM down to 5 V to power the encoder. The device on the right is the LDO TPS73501, which steps the 5 V from the buck down to 3.3 V to power the MCU and the voltage level translator.

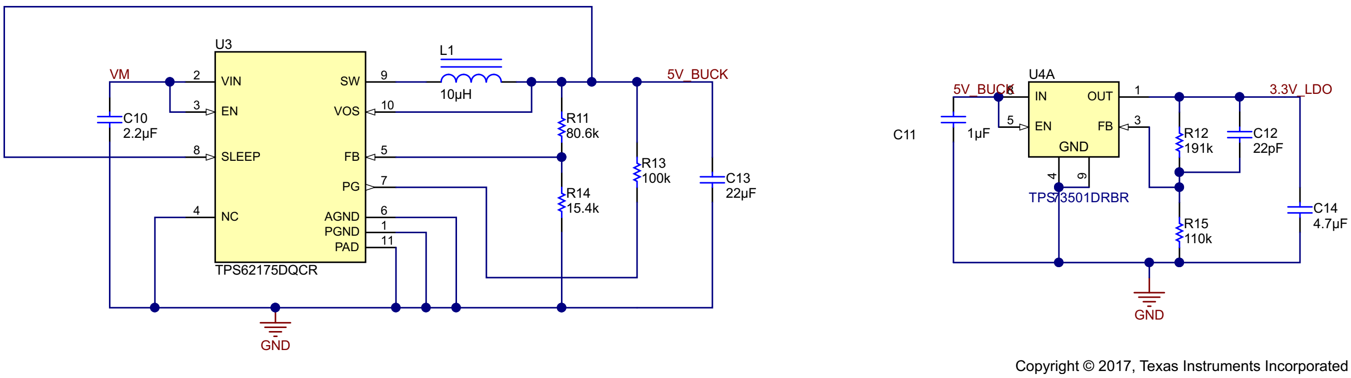


Figure 13. Board Power Management Schematic

3 Getting Started Hardware

The TIDA-01370 requires a 6.5- to 28-V power generator, a dual-phase stepper motor, and a 5-V encoder.

3.1 Procedure

1. Connect the motor terminals to the AOUT1/AOUT2 and BOUT1/BOUT2 headers on the board.
2. Connect the encoder pins to the appropriate header on the board to power the encoder at 5 V and to give access to the encoder outputs on the board.
3. Connect the power source to the VM and GND pads on the board.
   - If using a power supply for testing, set the output voltage from 6.5 to 28 V and maximum current to 2.0 A. Be careful not to use voltages that are higher than 28 V because they might cause the buck converter to malfunction.
4. Press and hold the DRV_DIR button to change the direction of the motor.
5. Move the potentiometer on the board to speed up or slow down the motor.
6. Press the MSP_RST button in case a reset of the MCU is needed.

3.2 Hardware Descriptions

Figure 14 shows the entire board layout, and the red circles highlight all the input and output connections to the board. VM and GND represent the input voltage supply for the board. The motor connections stand for the output of the board where the driver pumps the current through the two phases A and B of the stepper motor.

The encoder connection to the board is the 100-mil header where the encoder pins must be plugged in to power the encoder at 5 V and to send the encoder outputs to the MCU. The JTAG connection is the header where the code is flashed from the Code Composer Studio™ platform to the MCU.

Figure 14 also highlights in red the 18-pin, 100-mil header connector used to take measurements on the driver and MCU pins, in particular used for STEP (driver pin), A_E3.3, and B_3.3 (encoder output pins).

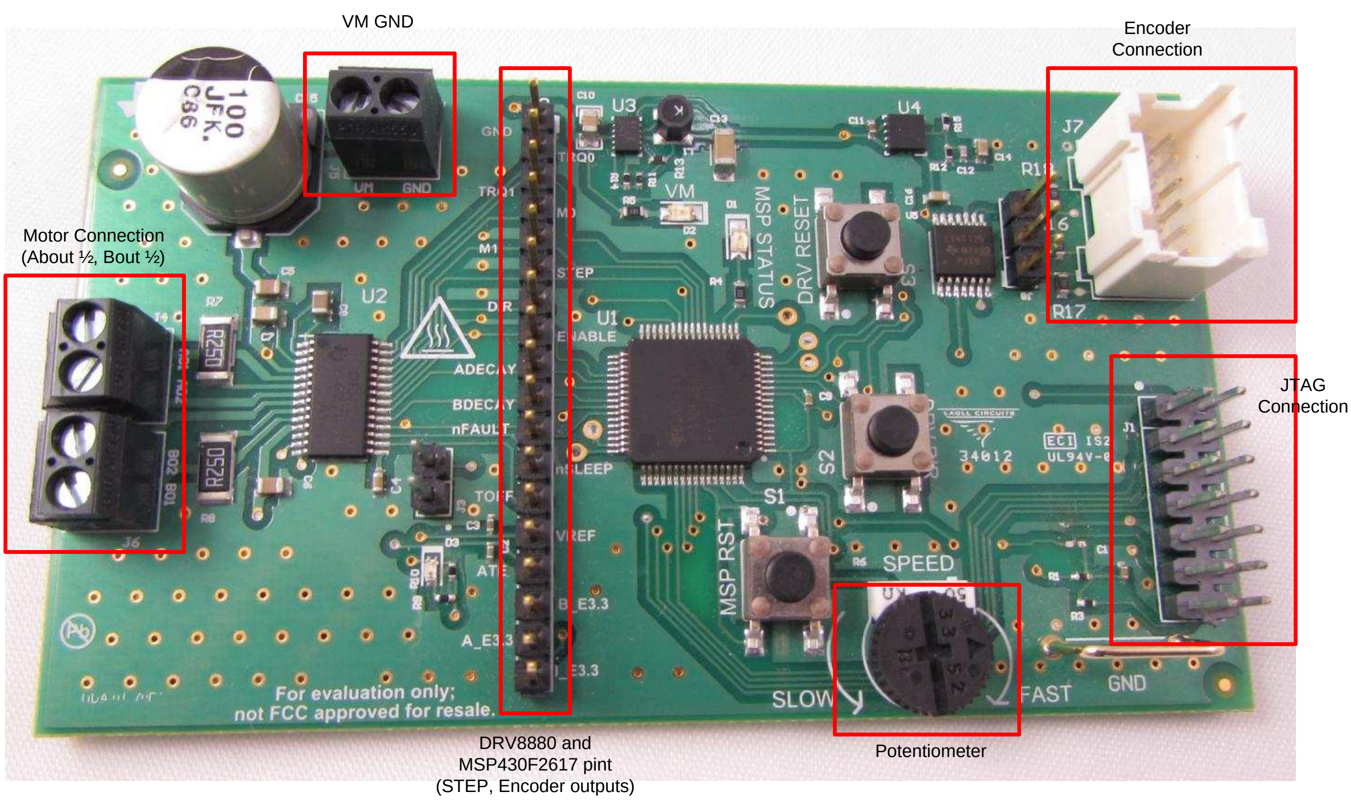


Figure 14. PCB Input/Output Connections

Figure 15 has red circles around the stepper driver DRV8880, the MCU MSP430F2617, and the potentiometer.

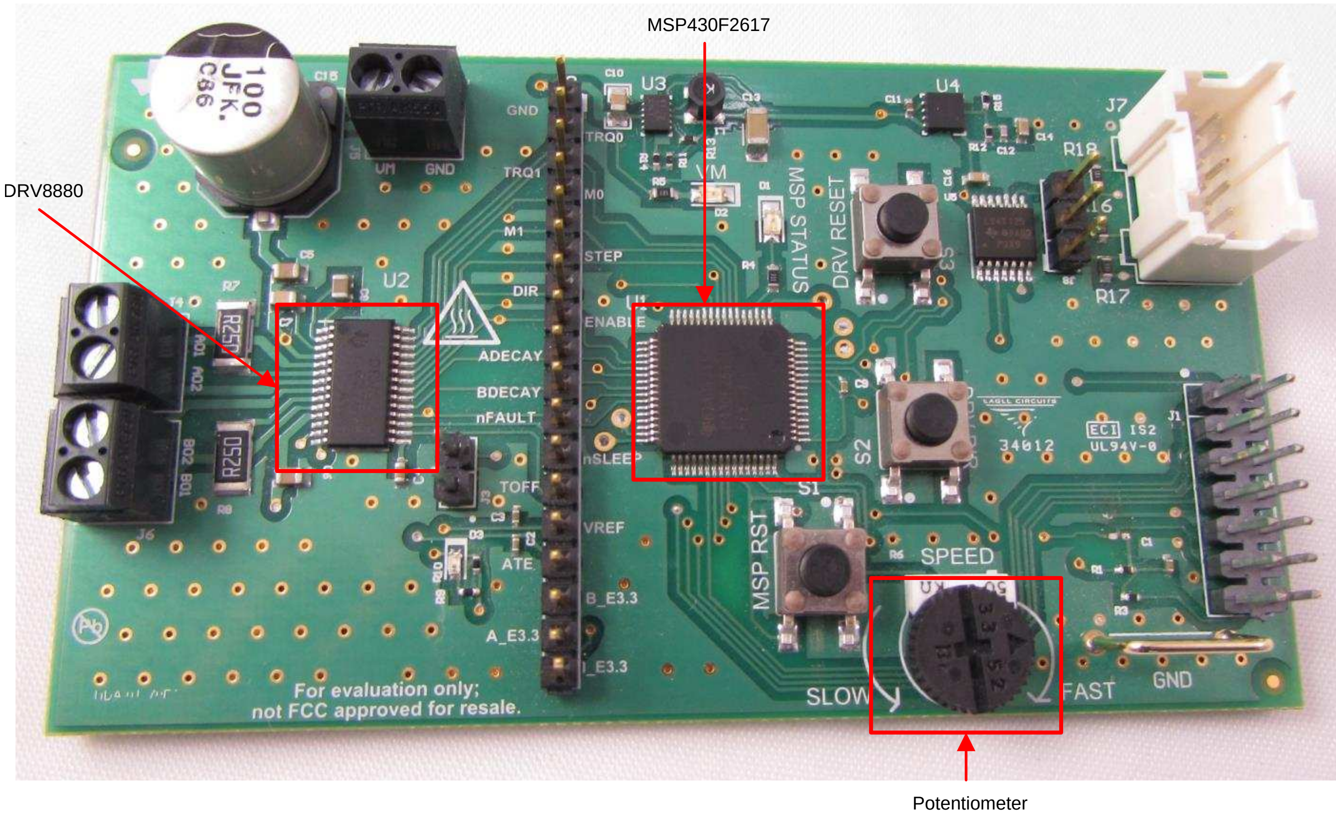


Figure 15. PCB Driver, MCU, and POT

The board has three LEDs: VM, MSP_STATUS, and nFault (see Section 2). The board also has three push-buttons: DRV_RESET, MSP_RST, and DRV_DIR (see Section 6) as shown in Figure 16.

To change the direction of the motor, the DRV_DIR must be pressed and held until the motor slows down to 0 rpm before changing the direction completely.

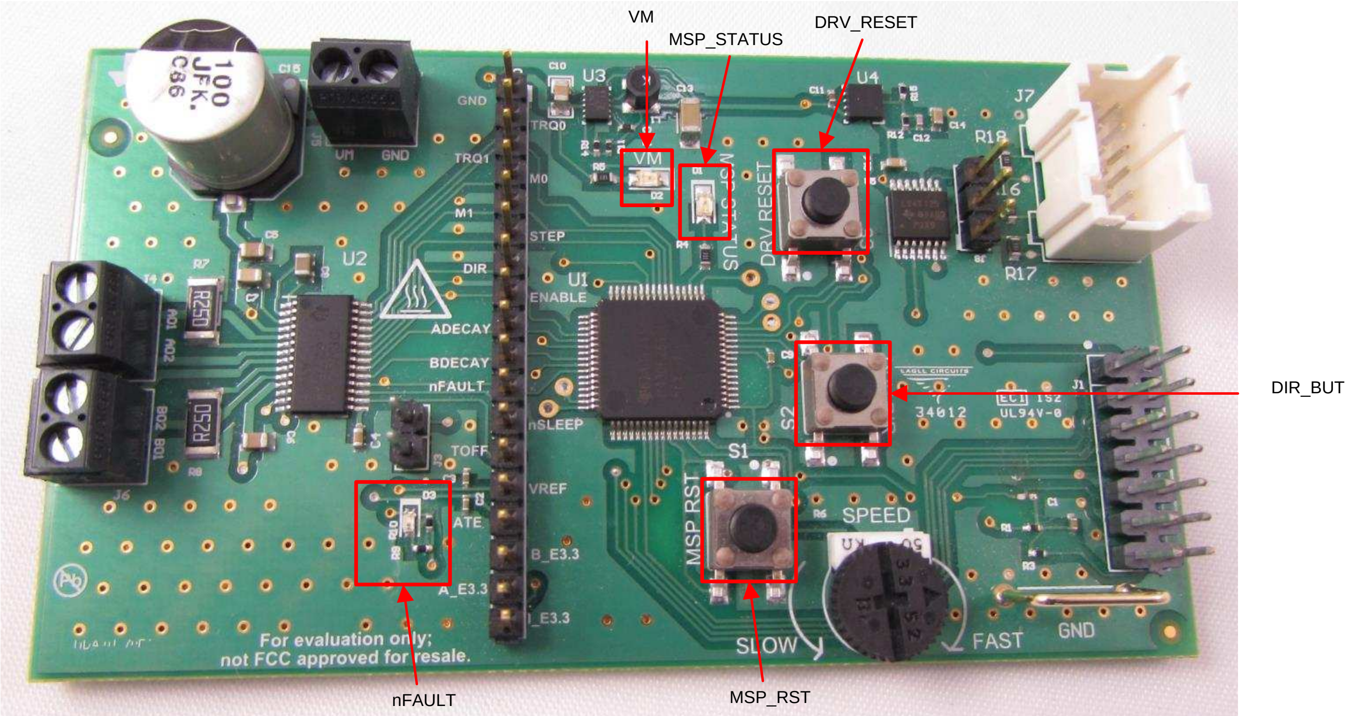


Figure 16. PCB LEDs and Push-Buttons

3.3 Software

The software of this TI Design was written in TI Code Composer Studio firmware version 6.1.3. A Spy-Bi-Wire protocol, which is a two-wire joint test action group (JTAG), is used to flash the code into the MCU used, the MSP430F2617.

This firmware can be downloaded from the TIDA-01370 design folder (see Section 6). The following describes at a high level what the software requires to perform:

1. Define the pins, constants, and variables used in the MCU.
2. Define a look-up table for the different speeds through which the stepper motor should spin.
3. Create a look-up table to activate the gray scale decoder.
4. The watchdog timer is disabled in code because there is no reset procedure needed.
5. The ports 1.7, 2.2, 2.3, 2.4, 2.5, 2.6, 2.7, 4.0, 4.1, 4.2, 4.4, 4.5, 4.6, 6.6, and 6.7 directions must be set as outputs.
6. Process and sample the Potentiometer’s voltage change through the ADC12 in order to get a varying and relative STEP frequency.

7. Configure the clock, the timers and the interrupts:
   - TACCR0 controls the gray scale quadrature decoder interrupt (occurring every 20 µs)
     (a) Enable the Gray Scale Quadrature Decoder to be able to read a net number of bit changes (called as EncCount) from the encoder outputs each step issued to the driver from the MCU.
   - TACCR1 generates the STEP pulses to be sent to the driver after that the ADC12 had sampled the POT voltage, therefore TACCR1 sets the motor speed (0 rpm; min speed= 15 rpm (STEP f = 20 ms); max speed = 150 rpm (STEP f = 2 ms).
   - TAIFG is used for stall detection, speed control, direction change of the motor movement (clockwise and counter-clockwise), and Potentiometer reading values.
     (a) Activate the stall detection such that when the encoder counts are less than 10, there is a stall situation and the stall_led (MSP_STATUS) blinks. Otherwise, it remains solid (green).
     (b) Change the direction of the motor by pushing and holding the DRV_DIR button. If this button is released, the direction is clockwise (forward). Otherwise, the rotation is counter-clockwise.
     (c) Vary the speed of the motor through a linear acceleration or deceleration defined at Step 2.

Figure 17 shows the software flow chart.

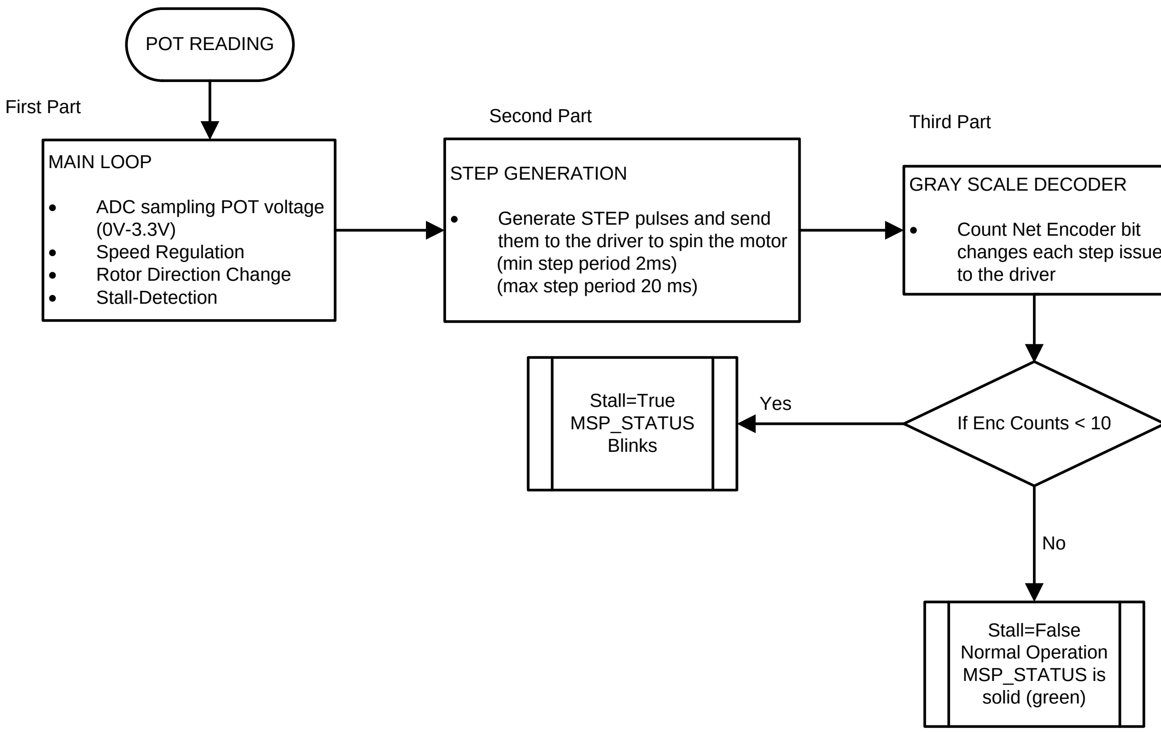


Figure 17. TIDA-01370 Software Flow Chart

4 Testing and Results

4.1 Test Equipments

- Oscilloscope: Tektronix TDS5054B
- Power Generator: Chroma Programmable DC Power Supply model 62012P-100-50
- Current Probe: Tektronix TCP202
- Current Probe Stepper Motor: Nanotec-SC4118L1804
- 600 Resolution Encoder: Ruland H38S600B

4.2 Test Data

Figure 18 shows the board setup with all the input/output connections as described in Section 3.1. The device on the left is the dual-phase stepper motor with the encoder attached to the shaft on the rear part, while the yellow disk is the clean damper.

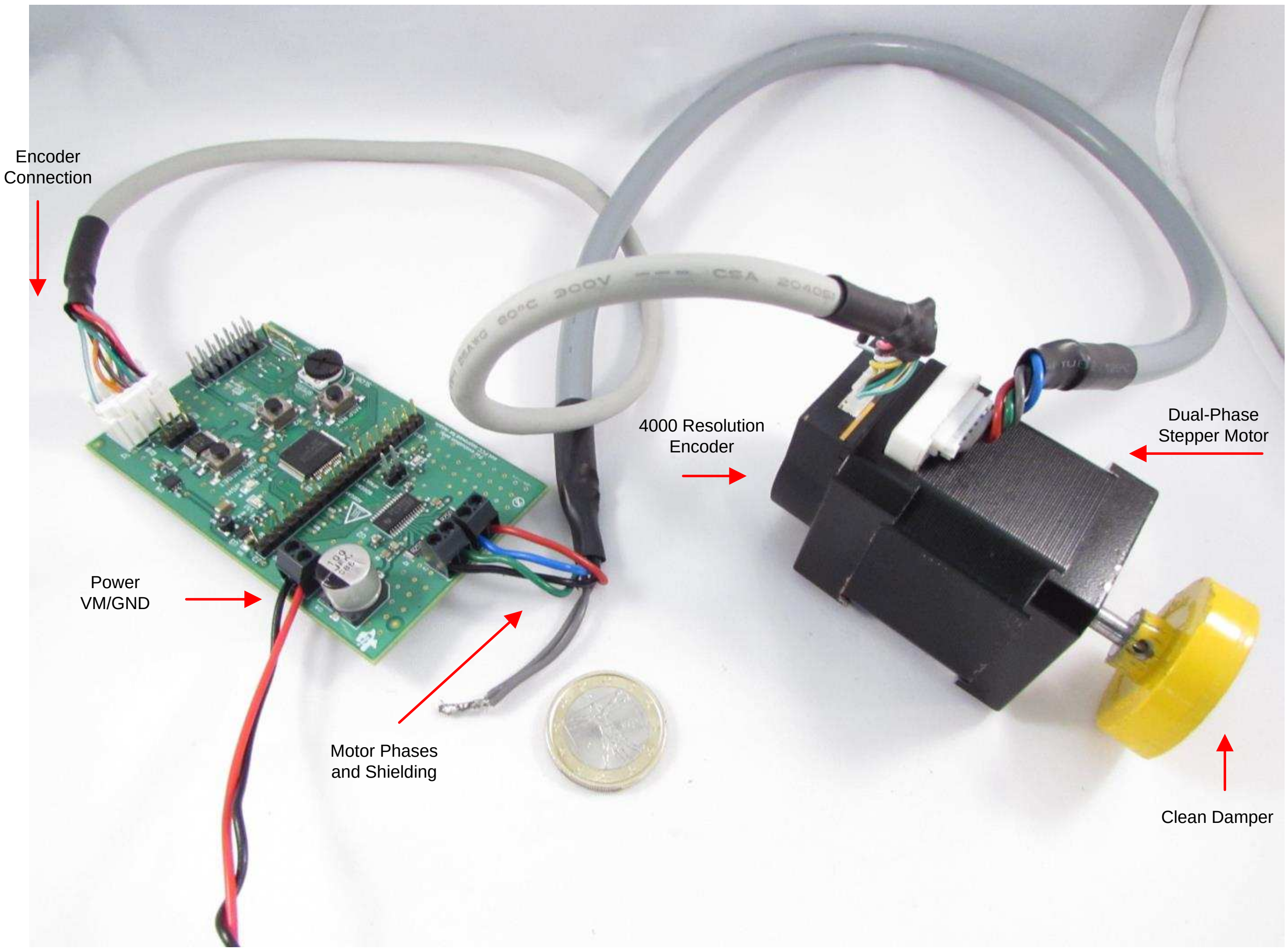


Figure 18. TIDA-01370 Board Setup With Motor

This TI Design has the goal to detect a stall condition of the motor. To do this, the tests performed in the lab had to compare the STEP pulses provided by the MCU to the driver with the encoder outputs, which are the signals A and B in binary form. The STEP pulses determine the speed of the motor, which then depends on the STEP interval. The STEP interval is regulated by the potentiometer; by varying its voltage from 0 to 3.3 V (the potentiometer can be changed manually), the device varies the STEP interval from 20 ms to 2 ms. The minimum speed of the motor (15 rpm) corresponds to the STEP interval of 20 ms, while the maximum speed of the motor (150 rpm) chosen for this TI Design corresponds to the STEP interval of 2 ms. The board is powered at 24 V and the current is limited at 2 A.

Figure 19 shows the motor speed-torque curve, which confirms that the motor speed chosen meet the actual motor speed requirements:

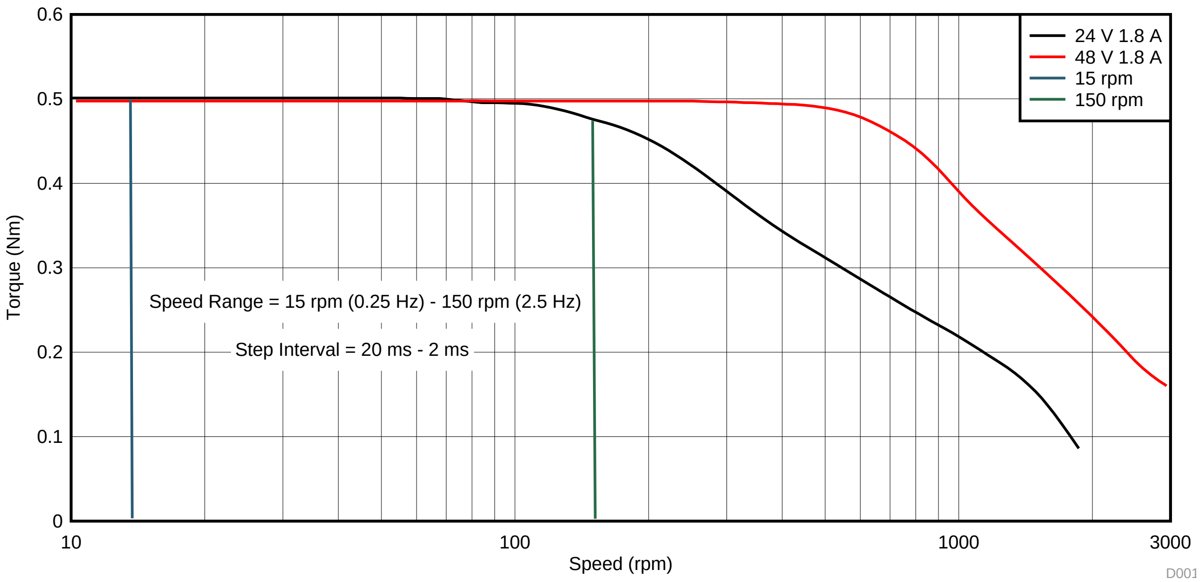


Figure 19. Motor Speed-Torque Curve

The motor step angle (see Section 1.2) is 1.8°; therefore, in full step mode as it is spun in this TI Design, the motor needs to perform 200 steps in order to accomplish one full mechanical revolution of 360°.

Figure 20 shows the current (I_phase_A_in violet) in the motor phases which has the typical behavior of the full step mode.

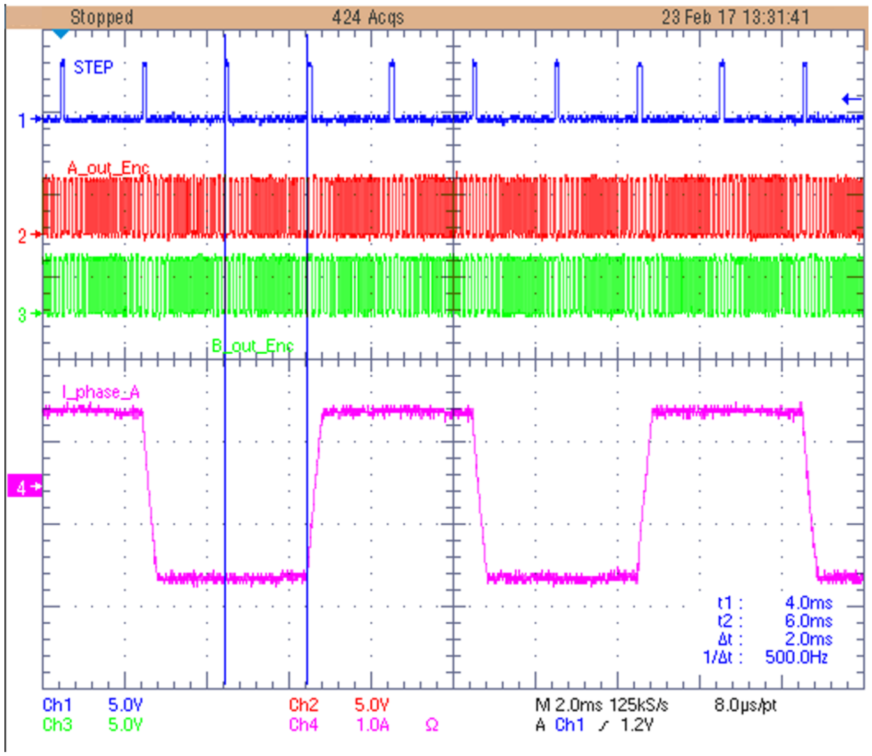


Figure 20. Current Waveform in Motor Phases Driven in Full Step Mode

From these considerations, in order to make the motor spin at 15 rpm (0.25 Hz), spin the motor at 50 steps per second (0.25 Hz × 200 steps per revolution), which corresponds to issue a step to the driver with a period of 20 ms (1/50 steps per second).

In order to detect a stall condition, first define that the stall event occurs when the rotor stops spinning while issuing steps to the driver. This TI Design uses an optical incremental encoder to determine when the stall has occurred through the algorithm described in Section 1.1 and Section 3.2.

The encoder has a resolution of 4000 increments per revolution. This results in observing 20 net increments (outputs from the encoder) for each STEP issued from the MCU to the driver under normal conditions. Because the encoder has two outputs A and B shifted 90 degrees out of phase among each other, the increments are considered net when the direction is unique (clockwise or either counter-clockwise). On the other hand, when A and B lead and lag each other, the motion of the motor is not forward in one direction but it might oscillate back and forth.

The gray scale decoder is used to count the net counts from the encoder, meaning it decodes the phase shifts between A and B and its output is the net count of the encoder increments per STEP.

Under normal conditions, the net counts should approach 20 for each STEP issued, while for a stall event the net increments should tend to zero because there is no moving part. However, due to the high resolution encoder (4000 increments per resolution) being able to detect up to 0.09 degrees of rotation and the noise and vibration of the motor, it is possible to detect up to 5 to 9 net increments of the encoder outputs during a stall event.

That is the reason why the software implemented checks at every STEP issued to the driver whether the number of net encoder increments is larger than 10 (no stall) or smaller than 10 (stall detected). During the normal condition, the stall LED (MSP_STATUS) blinks; otherwise, under normal conditions the MSP_STATUS LED stays on (green).

Figure 21 through Figure 24 show the regular working conditions of the stepper motor at the following speeds:

- 15 rpm (minimum speed) with a STEP interval of 20 ms
- 30 rpm with a STEP interval of 10 ms
- 75 rpm with a STEP interval of 4 ms
- 150 rpm (maximum speed) with a STEP interval of 2 ms

Figure 21. 15 rpm, No Stall

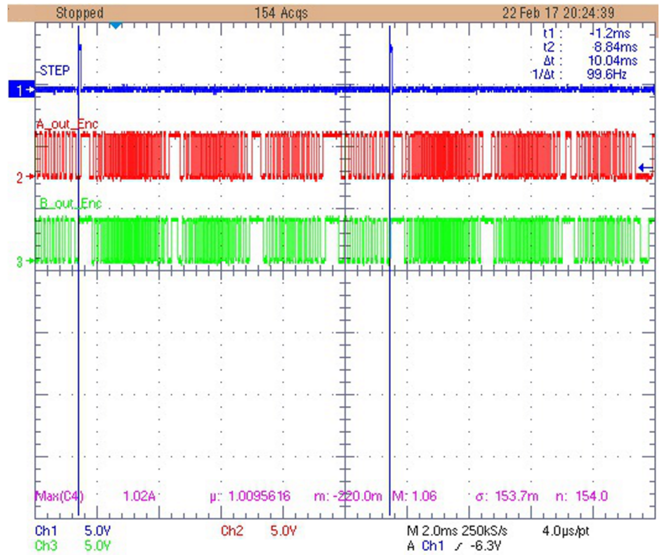


Figure 22. 30 rpm, No Stall

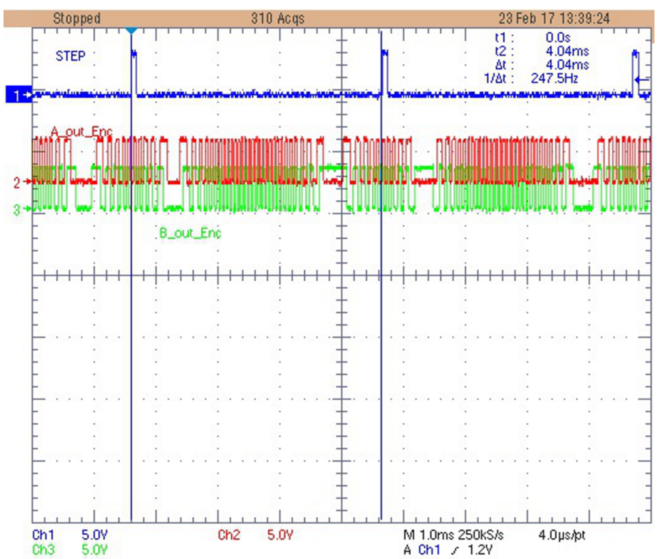


Figure 23. 75 rpm, No Stall

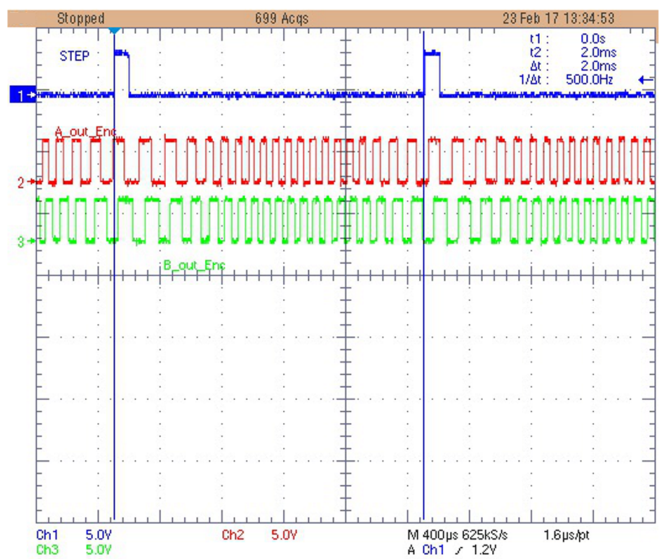


Figure 24. 150 rpm, No Stall

These waveforms represent the no stall condition. As shown in Figure 24, it is possible to count 20 net increment pulses from the encoder outputs on both channel A (A_out_Enc) and B (B_out_Enc) between each STEP.

In addition, at the fastest speed the encoder outputs A and B always have the same phase shift as can be seen in Figure 25 where the patterns are zoomed in. This means it is always the same phase (A in this case), which is leading the other and the rotor movement is straight forward with no oscillations.

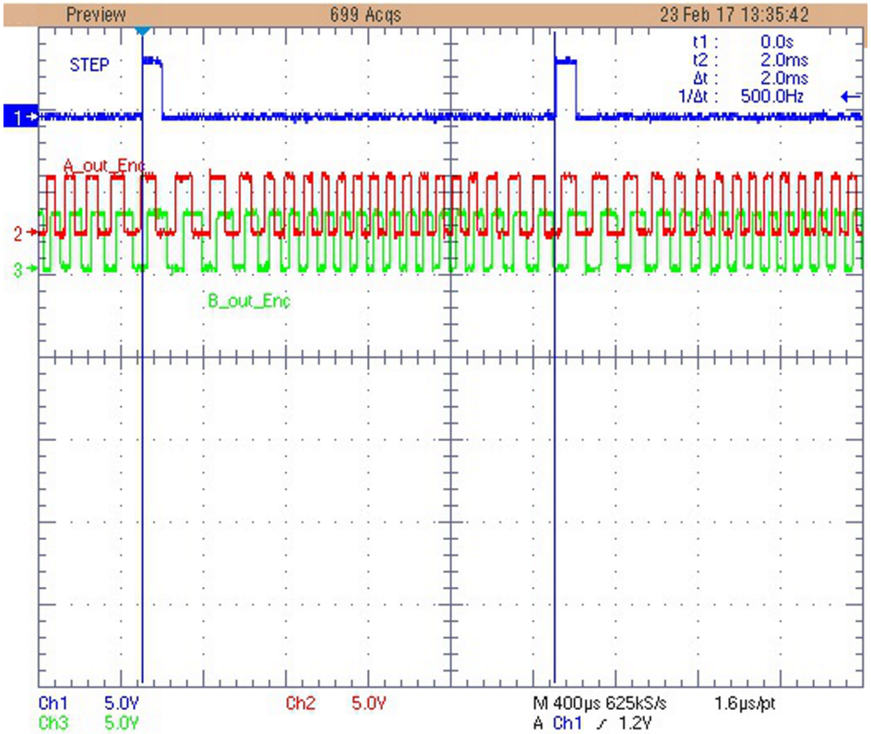


Figure 25. Zoom on A and B Patterns at Maximum Speed

On the other hand, in Figure 21 through Figure 23, the A and B outputs lead and lag each other. This is due to the fact that at slower speeds, the motor inertia decreases, the Back-EMF reduces, and then the vibrations affecting the motor movement become more relevant and audible. However, the net motor movement is always forward and it does not change direction. In all the cases presented, the gray scale decoder outputs net encoder increments approaching 20 counts each step issued, resulting in a non-stall event.

Figure 26 represents the stall condition where the motor was stalled on purpose while spinning at different speeds. (For simplicity, it is here reported as one figure because the other figures representing the stall event have all the same features.)

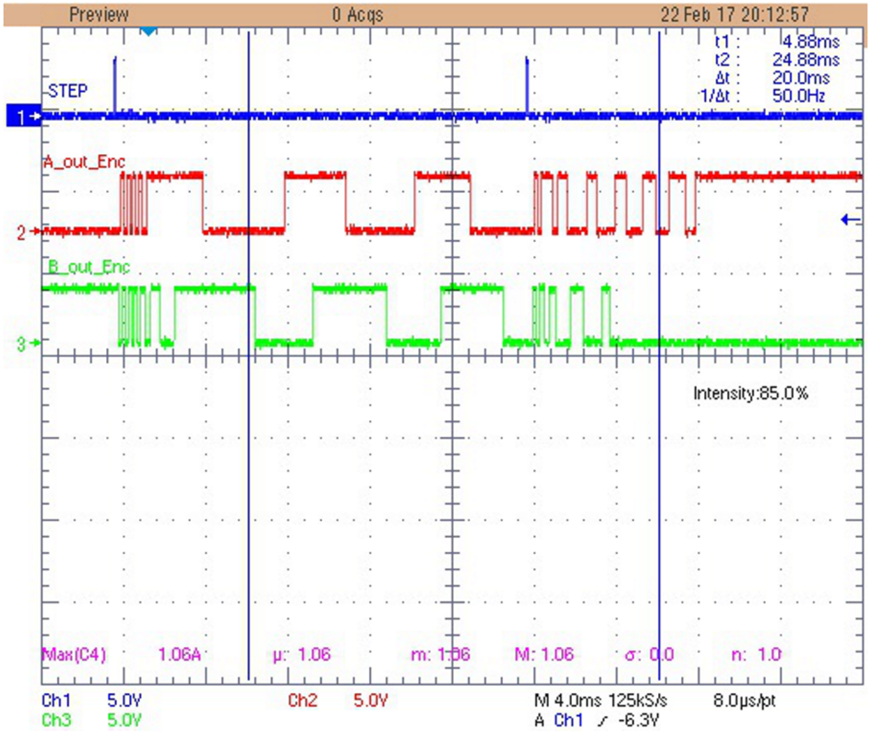


Figure 26. Stall Condition

Figure 26 shows that in the event of a stall, the encoder outputs reveal a non-regular behavior. In an ideal case, the encoder pulses should be zero because the motor stopped to spin; however, due to high resolution encoder and noise, the net encoder counts are still present but their number is below 10 (stall condition). In this case, the software detects the stall and the respective LED on the board (MSP_STATUS) changes its status from solid to blinking.

In addition, Figure 27 shows the encoder outputs obtained with a 600-resolution encoder (Ruland H38S600B). In this case, using the same considerations performed earlier should result in 3 net encoder counts for each STEP issued under normal conditions as can be seen in Figure 27:

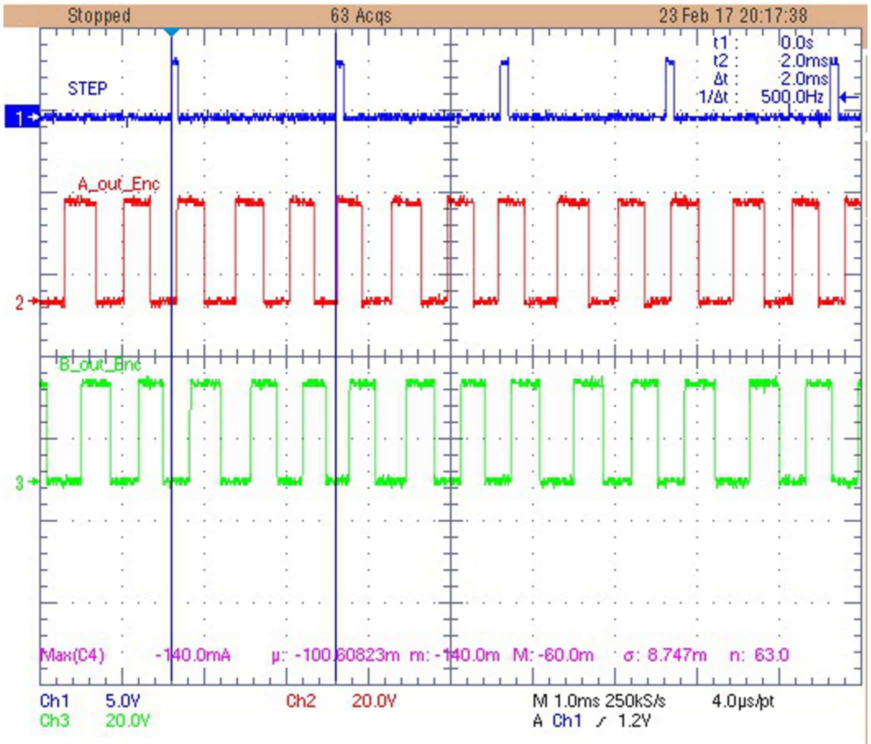


Figure 27. Encoder Outputs Using a 600-Resolution Encoder

When there is a stall event, the encoder outputs from the 600-resolution encoder are zero. This shows the difference between this low-resolution encoder and the 4000-resolution encoder used in this TI Design. The 600-resolution encoder does not detect any vibration detected by the higher resolution encoder; thus, when the motor is stalled, the encoder counts are exactly zero as shown in Figure 28. This does not mean the motor does not vibrate anymore, just that the low-resolution encoder cannot detect any small vibrations. While the 4000-resolution encoder can detect up to 0.09 degrees of movement, the 600-resolution encoder can detect up to 0.6 degrees of movement. This means that during a stall event for this TI Design, the vibrations produce a rotation smaller than 0.6 degrees.

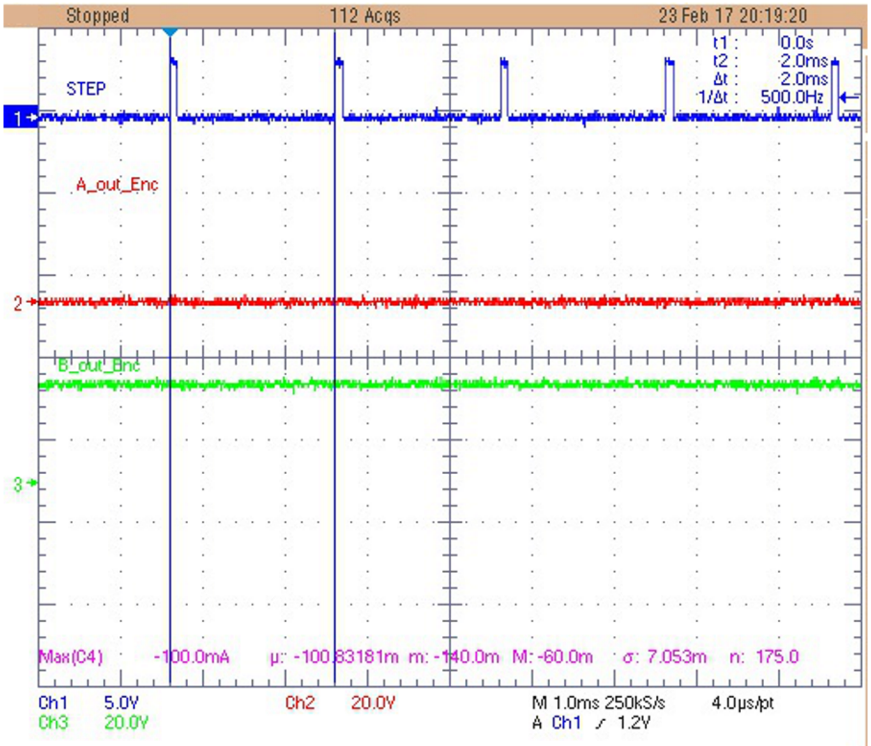


Figure 28. Encoder Outputs With 600-Resolution Encoder During Stall Condition

While running the software, it was possible to find out the natural resonant frequency of the stepper motor. When the STEP period matches with a particular frequency range, the motor began to oscillate back and forth while emitting noise and vibrating (resonance condition). However, as soon as the STEP interval is amended in order to speed up or slow down the motor, the resonant condition disappears.

The resonant condition is an undesired event because in this case the motor oscillates back and forth and might stall or miss some steps. That is the reason why the speed table through which the stepper motor should spin through avoids to meet that resonant speed range defined for the STEP interval between 4.5 and 6 ms (between 49.8 rpm and 66 rpm for the motor speed). It is possible to get rid of the resonant condition either by changing the STEP interval or by microstepping the motor.

Figure 29 shows the resonant condition occurring at 5 ms for the STEP interval. In this case, the stall condition is verified because the net encoder pulses are less than 10. This means that the motor is oscillating back and forth repeatedly without a net forward or reverse movement, and in this case the stall detection reveals the resonant condition as well.

Figure 29. Resonance Condition at 5 ms

Therefore, it is now possible to build a table of the speed range the motor is spinning through and declare when the motor is spinning smoothly and when it is in resonant condition as shown in Table 2:

Table 2. Motor Speed Table

STEP PERIOD (ms)	MOTOR SPEED (RPM)	MOTOR MOVEMENT
20.0	15.00	Smooth (No stall, LED blinking)
10.0	30.00	Smooth (No stall, LED blinking)
8.0	37.50	Smooth (No stall, LED blinking)
6.0	50.00	Non-regular back and forth oscillations (Stall, LED blinking)
5.5	54.54	Non-regular back and forth oscillations (Stall, LED blinking)
5.0	60.00	Non-regular back and forth oscillations (Stall, LED blinking)
4.5	66.67	Non-regular back and forth oscillations (Stall, LED blinking)
4.0	75.00	Smooth (No stall, LED blinking)
3.0	100.00	Smooth (No stall, LED blinking)
2.0	150.00	Smooth (No Stall, LED blinking)

The TIDA-01370 design shows how to detect a stall condition in software using the gray-scale decoder applied to an optical incremental encoder. This TI Design also reveals the resonant frequency of the stepper motor.

5 Design Files

5.1 Schematics

To download the schematics, see the design files at TIDA-01370.

5.2 Bill of Materials

To download the bill of materials (BOM), see the design files at TIDA-01370.

5.3 PCB Layout Recommendations

Figure 30 represents some guidelines for the layout of the TIDA-01370 design.

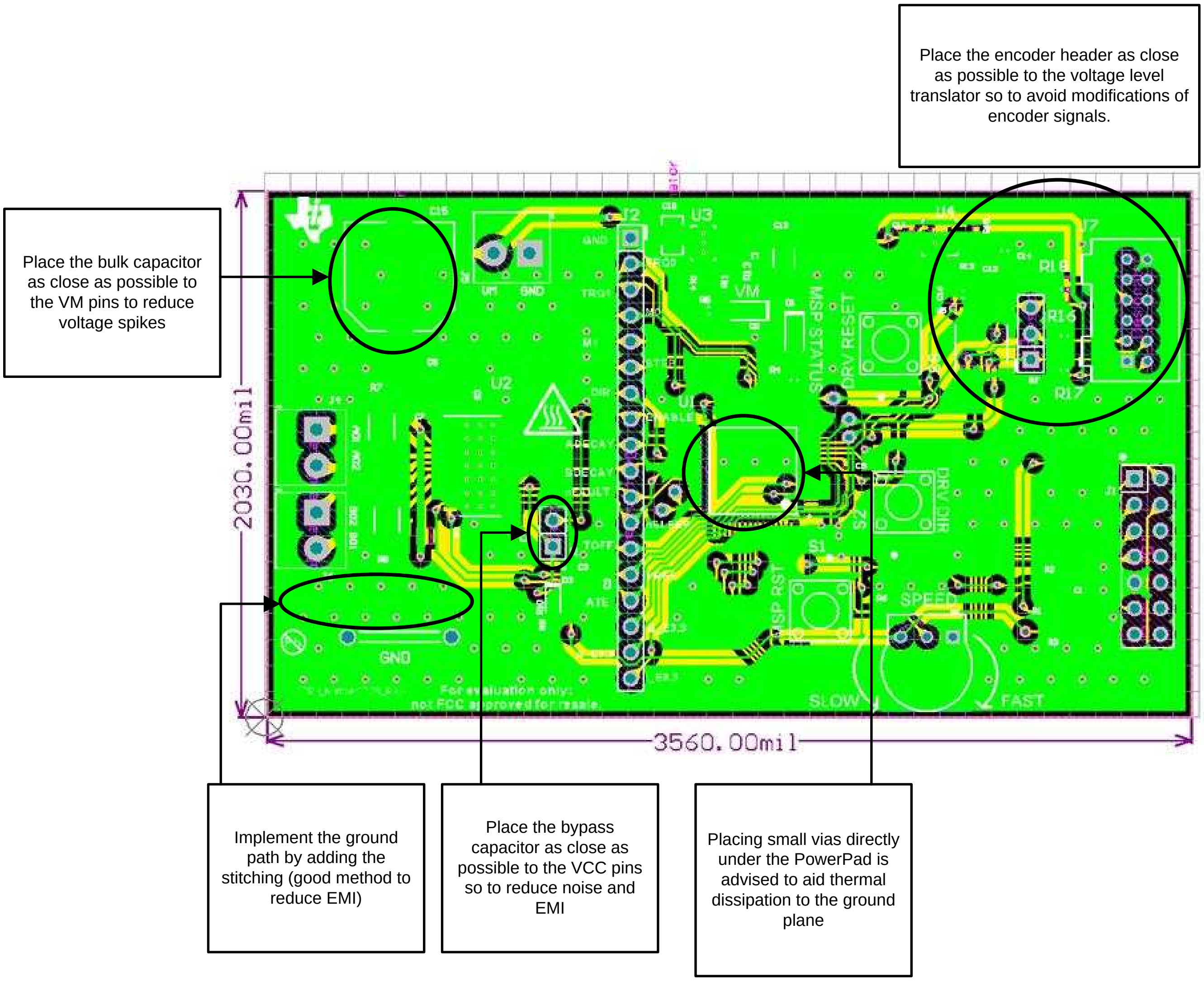


Figure 30. TIDA-01370 Layout Recommendations

5.3.1 Layout Prints

To download the layer plots, see the design files at TIDA-01370.

5.4 Altium Project

To download the Altium project files, see the design files at TIDA-01370.

5.5 Gerber Files

To download the Gerber files, see the design files at TIDA-01370.

5.6 Assembly Drawings

To download the assembly drawings, see the design files at TIDA-01370.

6 Software Files

To download the software files, see the design files at TIDA-01370.

7 Related Documentation

1. Texas Instruments, *DRV8880 Evaluation Module (EVM) User's Guide* (SLVUAI1)

7.1 Trademarks

PowerPAD, MSP430, DCS-Control, Code Composer Studio are trademarks of Texas Instruments.
Auto-Tune is a trademark of Antares Audio Technologies.
All other trademarks are the property of their respective owners.

8 About the Author

ANTONIO FAGGIO is an applications engineer at Texas Instruments where he is currently part of the Applications Rotational Program. He supports stepper motor drivers. Antonio got his master's degree in electrical engineering-power system from Politecnico di Milano in Milan (Italy) in 2015.

IMPORTANT NOTICE FOR TI DESIGN INFORMATION AND RESOURCES

Texas Instruments Incorporated ('TI") technical, application or other design advice, services or information, including, but not limited to, reference designs and materials relating to evaluation modules, (collectively, "TI Resources") are intended to assist designers who are developing applications that incorporate TI products; by downloading, accessing or using any particular TI Resource in any way, you (individually or, if you are acting on behalf of a company, your company) agree to use it solely for this purpose and subject to the terms of this Notice.

TI's provision of TI Resources does not expand or otherwise alter TI's applicable published warranties or warranty disclaimers for TI products, and no additional obligations or liabilities arise from TI providing such TI Resources. TI reserves the right to make corrections, enhancements, improvements and other changes to its TI Resources.

You understand and agree that you remain responsible for using your independent analysis, evaluation and judgment in designing your applications and that you have full and exclusive responsibility to assure the safety of your applications and compliance of your applications (and of all TI products used in or for your applications) with all applicable regulations, laws and other applicable requirements. You represent that, with respect to your applications, you have all the necessary expertise to create and implement safeguards that (1) anticipate dangerous consequences of failures, (2) monitor failures and their consequences, and (3) lessen the likelihood of failures that might cause harm and take appropriate actions. You agree that prior to using or distributing any applications that include TI products, you will thoroughly test such applications and the functionality of such TI products as used in such applications. TI has not conducted any testing other than that specifically described in the published documentation for a particular TI Resource.

You are authorized to use, copy and modify any individual TI Resource only in connection with the development of applications that include the TI product(s) identified in such TI Resource. NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT OF TI OR ANY THIRD PARTY IS GRANTED HEREIN, including but not limited to any patent right, copyright, mask work right, or other intellectual property right relating to any combination, machine, or process in which TI products or services are used. Information regarding or referencing third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of TI Resources may require a license from a third party under the patents or other intellectual property of the third party, or a license from TI under the patents or other intellectual property of TI.

TI RESOURCES ARE PROVIDED "AS IS" AND WITH ALL FAULTS. TI DISCLAIMS ALL OTHER WARRANTIES OR REPRESENTATIONS, EXPRESS OR IMPLIED, REGARDING TI RESOURCES OR USE THEREOF, INCLUDING BUT NOT LIMITED TO ACCURACY OR COMPLETENESS, TITLE, ANY EPIDEMIC FAILURE WARRANTY AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY YOU AGAINST ANY CLAIM, INCLUDING BUT NOT LIMITED TO ANY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON ANY COMBINATION OF PRODUCTS EVEN IF DESCRIBED IN TI RESOURCES OR OTHERWISE. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, DIRECT, SPECIAL, COLLATERAL, INDIRECT, PUNITIVE, INCIDENTAL, CONSEQUENTIAL OR EXEMPLARY DAMAGES IN CONNECTION WITH OR ARISING OUT OF TI RESOURCES OR USE THEREOF, AND REGARDLESS OF WHETHER TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

You agree to fully indemnify TI and its representatives against any damages, costs, losses, and/or liabilities arising out of your non-compliance with the terms and provisions of this Notice.

This Notice applies to TI Resources. Additional terms apply to the use and purchase of certain types of materials, TI products and services. These include; without limitation, TI's standard terms for semiconductor products http://www.ti.com/sc/docs/stdterms.htm), evaluation modules, and samples (http://www.ti.com/sc/docs/sampterms.htm).

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265