

# TI Designs Auto-Torque Implementation Reference Design



## TI Designs

TI Designs provide the foundation that you need including methodology, testing and design files to quickly evaluate and customize the system. TI Designs help you accelerate your time to market.

## Design Resources

<a href="#">TIDA-00740</a>	Tool Folder Containing Design Files
<a href="#">DRV8711 BoosterPack</a>	Product Folder
<a href="#">DRV8711</a>	Product Folder
<a href="#">MSP430G2 LaunchPad</a>	Product Folder
<a href="#">CSD88537ND</a>	Product Folder



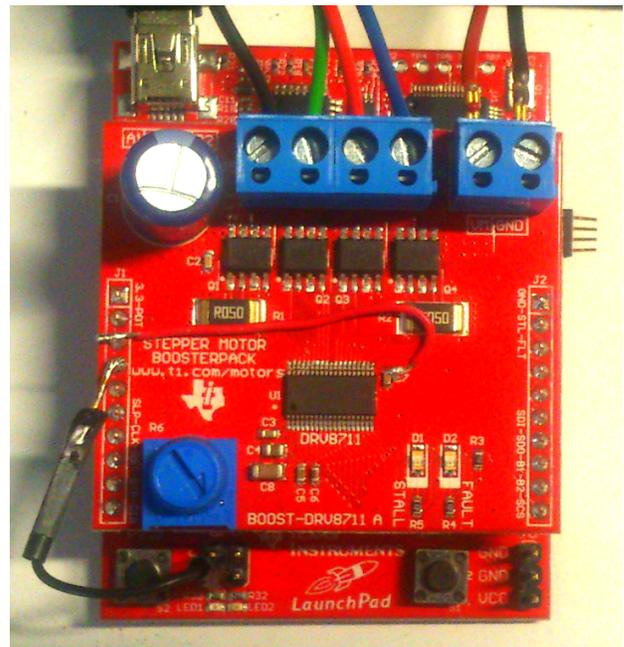
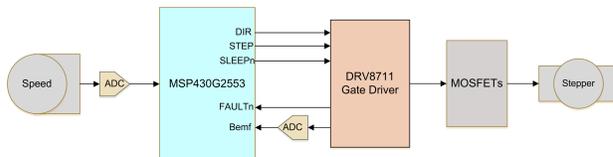
[ASK Our E2E Experts](#)  
[WEBENCH® Calculator Tools](#)

## Design Features

- Reduces Energy Consumption of a Stepper Motor by 66%
- Avoids Stalling at Variable Loads and at Variable Speeds
- Offers Simple Control

## Featured Applications

- Robotics
- Printers
- Stage Lighting
- Security Cameras
- Medical
- Industrial Machines



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

LaunchPad, BoosterPack, NexFET, MSP430, Code Composer Studio are trademarks of Texas Instruments. All other trademarks are the property of their respective owners.

## 1 Design Overview

This TI Design uses a TI stepper motor driver booster pack and ultralow-power MSP microcontroller LaunchPad™ to demonstrate the implementation of a new stepper motor control strategy. TI intends this strategy to reduce energy consumption and prevent stalling of a stepper motor

## 2 System Description

The setup consists of the TI DRV8711 BoosterPack™ that forms the power stage for the stepper motor and the MSP430G2 LaunchPad, which performs the desired control functions. The stepper motor connects to the BoosterPack and implements dynamic torque control is implemented through the DRV8711 and MSP430.

### 2.1 DRV8711 BoosterPack

Boost-DRV8711 is a stepper motor BoosterPack based on the DRV8711 stepper motor controller and CSD88537ND Dual 60-V N-Channel NexFET™ Power MOSFETs. This BoosterPack provides a complete stepper motor drive stage to evaluate motor applications. See [Figure 1](#) for more details.

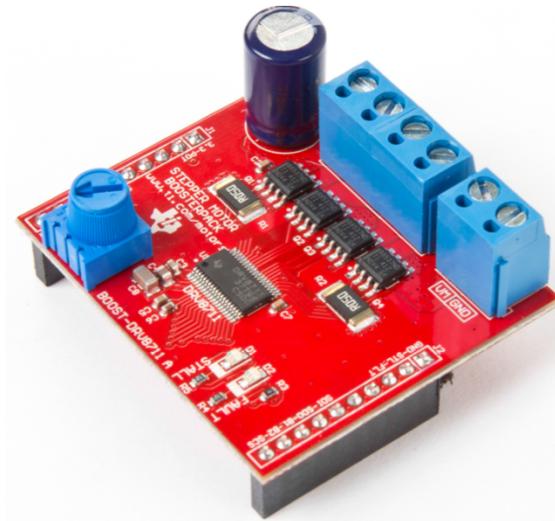


Figure 1. DRV8711 BoosterPack

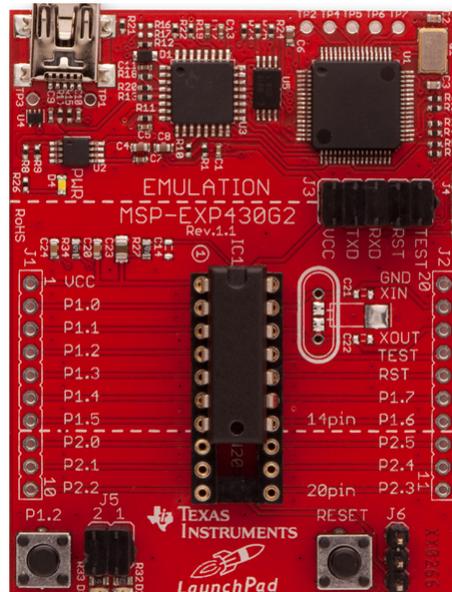
#### Features

- Offers a complete stepper motor drive stage in a small form factor (1.75 inch × 2.00 inch)
- Supports 8.2 V to 52 V and up to 4.5 A continuous for each H-bridge
- Offers 4× CSD88537ND Dual (8 FETs) 60-V N-Channel NexFET Power MOSFETs (12.5 mΩ)
- Offers motor stall and device fault LED indicators
- Offers a fully protected drive stage including overcurrent, overtemperature, undervoltage, and motor stall detect
- Can be combined with TI LaunchPad kits to create a complete stepper motor drive and control platform

For further details, see the *DRV8711 Stepper Motor Controller IC datasheet* ([SLVSC40E](#)), the *CSD88537ND datasheet* ([SLPS455A](#)), and the *BOOST-DRV8711 User's Guide* ([SLVU967](#)).

## 2.2 MSP430G2 LaunchPad

The MSP-EXP430G2 LaunchPad is an inexpensive and simple evaluation kit for the MSP430G2xx Value Line series of microcontrollers. The LaunchPad features an integrated DIP target socket that supports up to 20 pins, allowing MSP430™ Value Line devices to be plugged into the LaunchPad board. The MSP-EXP430G2 LaunchPad comes with an MSP430G2553 device by default. The MSP430G2553 has the most memory available of the compatible value line devices. The MSP430G2553 16-bit MCU has 16-KB flash, 512 bytes of RAM, a CPU speed up to 16-MHz, 10-bit ADC, capacitive touch enabled I/Os, universal serial communication interface, and more. See [Figure 2](#) for more details.



**Figure 2. MSP430G2 LaunchPad**

### Features

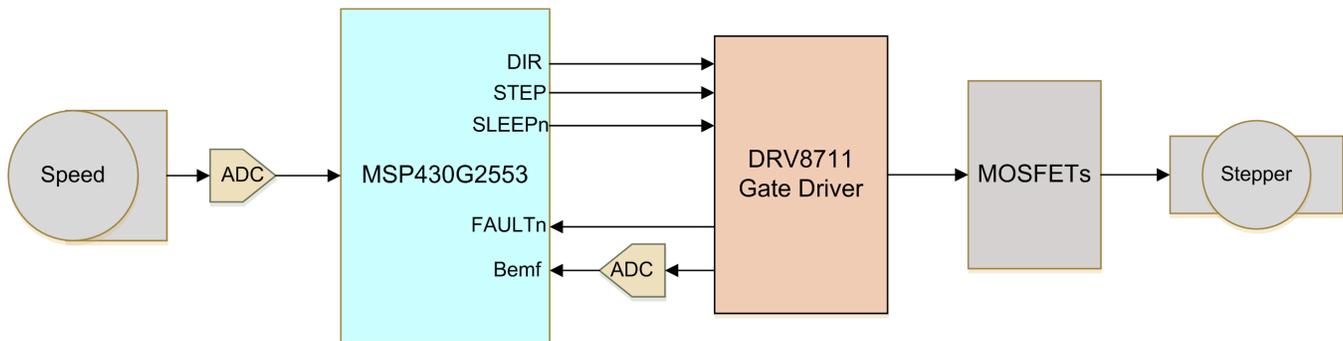
- Offers a USB debugging and programming interface featuring a driverless installation and application UART serial communication with up to 9600 baud
- Supports MSP430G2xx2, MSP430G2xx3, and MSP430F20xx devices in PDIP14 or PDIP20 packages
- Offers two general-purpose digital I/O pins connected to green and red LEDs for visual feedback
- Offers a two-push button for user feedback and device reset
- Offers easily accessible device pins for debugging purposes or as a socket for adding customized extension boards
- Offers a high-quality 20-pin DIP socket for an easy plug-in or removal of the target device

See *MSP-EXP430G2 LaunchPad Evaluation Kit User's Guide* ([SLAU318F](#)) for further details.

## 2.3 Stepper Motor

A stepper motor is an electromechanical device which converts electrical power into mechanical power. The motor is a brushless, synchronous electric motor that can divide a full rotation into an expansive number of steps. The position of the motor can be controlled accurately without any feedback mechanism, as long as the motor is carefully sized to the application. Stepper motors are similar to switched reluctance motors. Any application that uses a catalog stepper motor subjected to variable load is likely to benefit from this control strategy.

### 3 Block Diagram



### 4 Highlighted Products

#### 4.1 DRV8711 Driver IC

The DRV8711 device is a stepper motor that uses external N-channel MOSFETs to drive a driver bipolar stepper motor or two brushed DC motors. A microstepping indexer is integrated and capable of step modes from full step up to 1/256-step. An ultrasmooth motion profile can be achieved using adaptive blanking time and various current decay modes, including an auto-mixed decay mode. Motor stall is reported with an optional BEMF output. A simple STEP/DIR or PWM interface allows easy interfacing to controller circuits. An SPI serial interface programs the device operation. Output current (torque), step mode, decay mode, and stall detection functions are all programmable through the SPI serial interface.

#### 4.2 MSP430G2553 Microcontroller

MSP430G2553 is a ultralow-power mixed signal microcontroller with built-in 16-bit timers, up to 24-I/O capacitive-touch enabled pins, a versatile analog comparator, and built-in communication capability using the universal serial communication interface. In addition, the microcontroller has a 10-bit analog-to-digital converter (ADC). Using the digitally-controlled oscillator (DCO), the MSP430G2553 can wake up from low-power modes in less than 1  $\mu$ s.

#### 4.3 CSD88537ND Power MOSFET

The CSD88537ND is a Dual SO-8, 60-V, 12.5-m $\Omega$  N-Channel NexFET power MOSFET designed to serve as a half bridge in low-current motor control applications.

### 5 System Design Theory

When connected to a variable load, a stepper motor must be configured to drive the maximum possible torque to prevent motor stall conditions. This continuous maximum torque is driven even when the motor load is very small, which leads to a high-power dissipation. Missing a few steps might be unacceptable in certain position control applications like robotics, medical equipment, and other end equipment. In these applications, the output current must be set at a constant maximum value to prevent missed steps. Alternatively, a closed-loop control can be implemented to adjust the output current automatically to prevent step loss and stall. The feedback signal for this control can be anything related to position, speed, or torque, which can be obtained by adding sensors such as hall sensors, encoders, and so forth. Adding sensors is counterproductive as stepper motors are used primarily because they provide easy, sensorless position control.

When measured, back emf (BEMF) of the motor is an indication of the rotational speed of the motor. When the motor gets loaded down, the speed of the motor is momentarily reduced. This reduction can be captured using the BEMF signal and missed steps or stall conditions can be prevented if the stepper motor current is increased. DRV8711 stepper motor is capable of sampling the BEMF of the stepper motor during operation and provides an analog voltage output for the BEMF. This BEMF can be a feedback signal to close the loop and incorporate the torque control.

## 5.1 Control Strategy

After making the hardware changes in the DRV8711 BoosterPack and MSP430G2 LaunchPad, a PI controller implements the control algorithm. The potentiometer on the BoosterPack is used to the speed of the motor commanded by the MSP430.

## 5.2 Working Algorithm

The BEMF waveform is read into the ADC, which is averaged over several cycles to smooth the waveform. When the analog value is obtained, the current motor speed is determined from a programmed look-up table. The reference speed from the potentiometer is used to set the operation state of the motor. If the reference speed is too low, the motor halts. If the reference speed is very low, the motor spins but the control is omitted because the BEMF waveform may be unpredictable or have a signal that is too small. When the motor speed is acceptable, the torque control activates and the PI controller manipulates the stepper motor current based on the measured speed error. Speed error has been relaxed to certain extent to avoid rapid fluctuation in the torque which translates to current fluctuation (this fluctuation translates to noise). The following describes the algorithm flow:

- ADC obtains the digital value of the potentiometer reading which is used to obtain the reference speed of the motor from the look-up table.
- Based on the reference speed, the motor spins under full control, spins without control, or stops.
- ADC obtains the digital value of the BEMF reading, which is used to obtain the present speed of the motor from the look-up table. Speed error is calculated from reference and present speeds of the motor.
- Depending on the range of speed error, average the BEMF readings to eliminate any unwanted transient behavior
- When Integral function is performed as part of PI control, the integral variable is obtained. The PI controller output is directly manipulated to reduce the output torque if the motor load is constant or decreasing. Choose the range to avoid stalling.
- The final output of the PI controller is again modified to accommodate extreme conditions.
- The MSP430 writes to the DRV8711 TORQUE register through a SPI transaction to set the chopping current and change the output current in the stepper motor.

Figure 3 shows the algorithm:

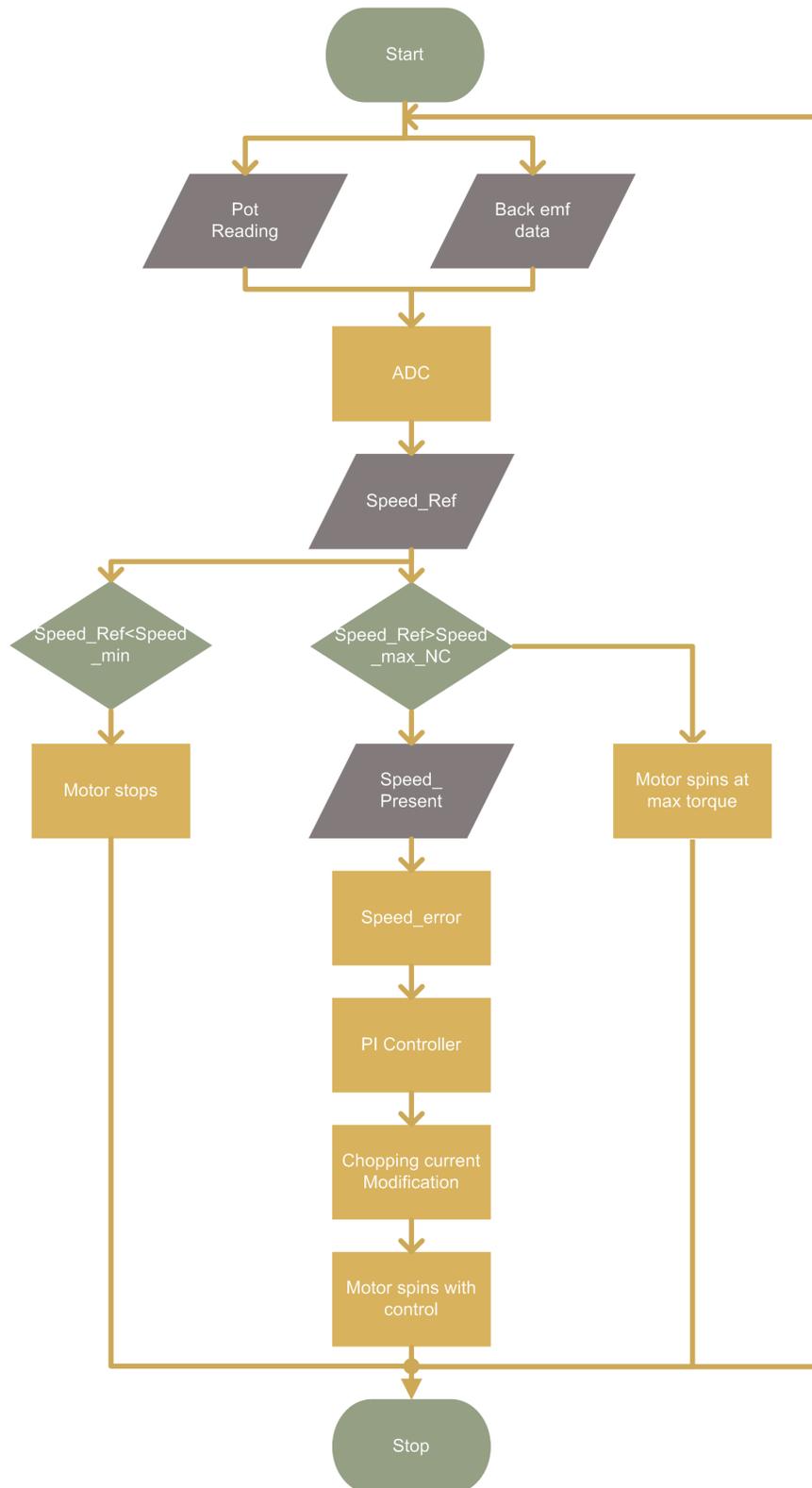


Figure 3. Flow Chart

### 5.3 PI Controller Tuning

The PI controller plays a key role in increasing the torque level when the motor is suddenly loaded. Because the algorithm is based on the principle that present speed of the motor can never exceed the reference speed, the PI controller is unable to reduce the torque level of the motor. An additional control is placed on top of the PI controller to reduce the torque level periodically and settle down at the minimum required torque level. This control ensures maximally efficient operation. The values of  $K_p$  and  $K_i$  are pivotal in transient and steady state performances of the motor. While the value of  $K_p$  increases the response speed, the value of  $K_i$  determines the steady state error of the system.

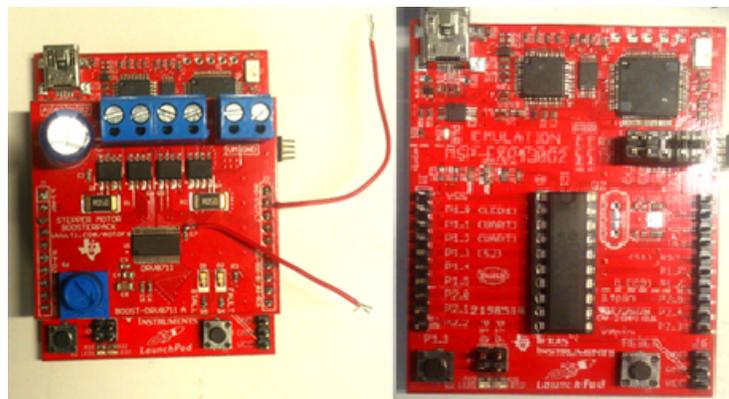
Do not increase the value of  $K_p$  too much as this increases the time for the motor current transition from loaded to unloaded condition, which wastes power. Increasing  $K_i$  value reduces the transition period, which might cause the motor to stall at certain speeds. Choose an optimum value to satisfy the previously mentioned recommendations.

## 6 Getting Started

### 6.1 Motor Characterization

The relationship between (BEMF) and speed forms the heart of the algorithm because this is how motor load changes are determined. For any motor, BEMF and speed are directly proportional and their relation can be obtained by using the following procedure:

1. Plug the stepper motor BoosterPack onto the LaunchPad as shown in [Figure 4](#).
  - (a) Align the TXD and RXD jumpers on the MSP-EXP430G2 Pad horizontally (the black rectangle in [Figure 4](#)) to operate the hardware UART with a software application.
  - (b) Solder a wire to the BEMF pin (pin 20) of the DRV8711 driver IC (yellow rectangle shown in [Figure 4](#)) and another to pin 2 that corresponds to nSTALL pin (blue rectangle shown in [Figure 4](#)) of jumper J2 on the BoosterPack.
  - (c) Connect the outputs of BEMF and nSTALL to the oscilloscope.



**Figure 4. BoosterPack and MSP430G2 LaunchPad**

2. Connect your stepper motor to the terminal block header J5. (The motor should have two windings, each with a + and a – termination.)
  - (a) Connect one winding to A1/A2.
  - (b) Connect the other winding to B1/B2. (See *DRV8711 Stepper Motor Controller IC datasheet* [SLVSC40E] for further details.)
3. Connect the power supply that powers the DRV8711 of the stepper motor BoosterPack and drive stage to the terminal block header J6.

---

**NOTE:** The connections have been labeled VM and GND. The stepper motor BoosterPack has an operating range from 8.2 V to 52 V up to 4.5 A continuous for each H-bridge. When using the MSP430G2 LaunchPad and an MSP430G2553, BOOST-DRV8711 GUI controls the motor. To get started with the GUI, see sections 4.3 and 4.4 of the *BOOST-DRV8711 User's Guide* (SLVU967).

---

4. Use the following settings for motor characterization using the BOOSTXL-DRV8711 GUI.

**Table 1. Recommended Values**

Setting Names	Recommended Values
EXSTALL	Turned on
MODE	1/4 /Half-Step
ISGAIN	20
SMPLTH	300 $\mu$ s
VDIV	Divided by 32/16/8/4
Target Speed (PPS)	$(\text{RPM} \times 40) \div 3$ – Quad step-or- $(\text{RPM} \times 20) \div 3$ – Half step

5. Connect the outputs of BEMF and nSTALL to the oscilloscope.

---

**NOTE:** By changing the Target Speed (PPS), the speed of the motor can be changed. Perform the following steps.

---

6. Gradually increase the target speed to find the maximum speed at which the motor can rotate.
7. Note this maximum speed as Speed\_max.
8. For speeds less than this Speed\_max, measure the BEMF pin and nSTALL pins on the EVM board to get the BEMF characteristics.

---

**NOTE:** The DRV8711 begins sampling BEMF data when there is a low-to-high transition in the nSTALL pin. The voltage on the BEMF pin of the DRV8711 is high immediately after a low-to-high transition and the beginning of the DRV8711 sampling. After the motor phase current has decreased to zero, the BEMF sample is stable. This BEMF sample can be read by the microcontroller on a high-to-low transition from the STALL pin. The BEMF can be recorded then the nSTALL pin transitions from high to low. The BEMF and nSTALL pins operate as shown in [Figure 5](#) and [Figure 6](#).

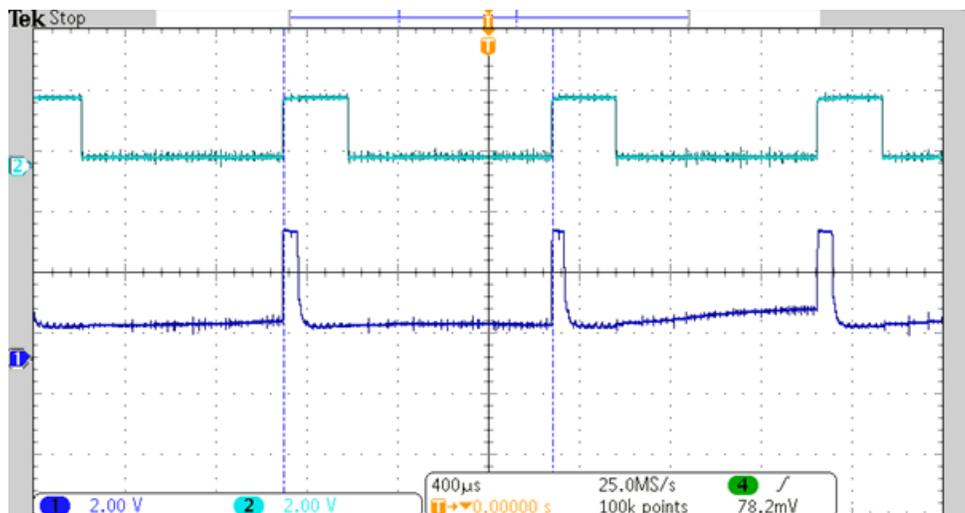
Large motors may exhibit an unstable BEMF at lower speeds. This lower speed, less than which the BEMF waveforms are unredictable, is considered as Speed\_min.

---

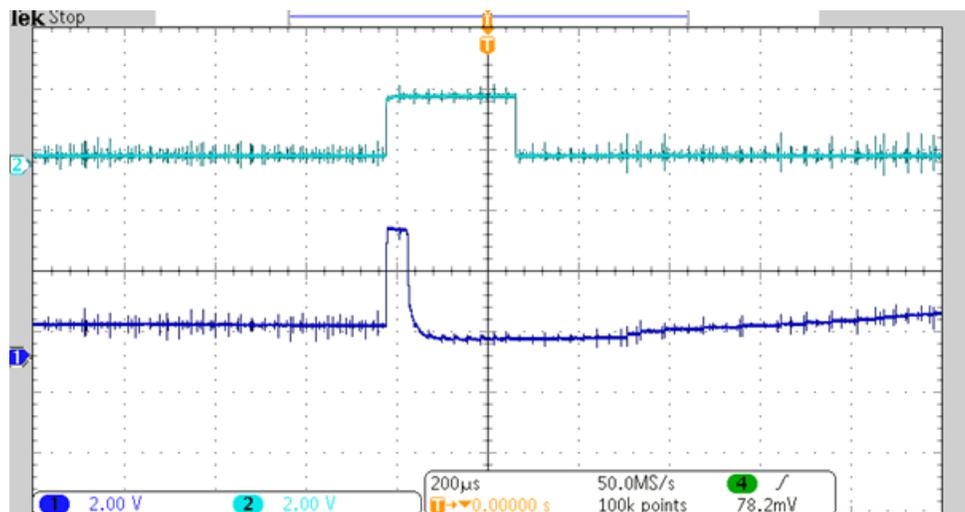
9. Slowly change the speed from zero to Speed\_max.
10. Observe the BEMF waveform. (If the BEMF value goes higher than 2.5 V, change the DRV8711 to a higher VDIV setting.)

**NOTE:** The BEMF waveform can lose its periodicity at higher speeds (less than maximum speed). An unstable BEMF at high speed can be easily identified through the fact that the BEMF waveform lacks a step pulse. Find the maximum speed where the BEMF waveform is periodic and unsaturated. Consider this value the new Speed\_max.

The BEMF vs speed (rpm) data can be obtained at different speeds less than Speed\_max to use in the header file look-up table.



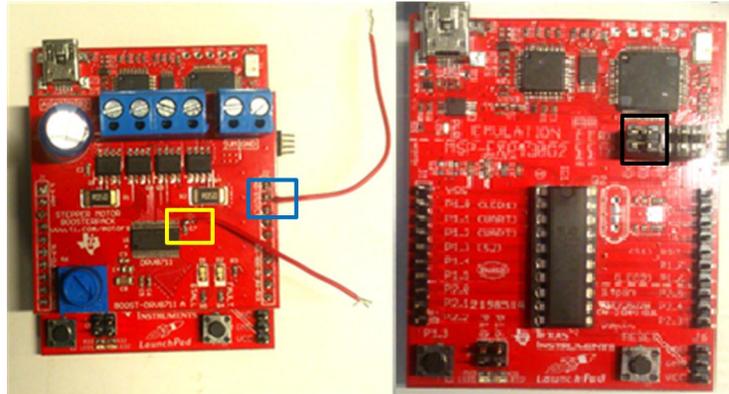
**Figure 5. Several Periods of nSTALL and BEMF**



**Figure 6. One nSTALL Event**

## 6.2 Modifying Hardware

The BEMF pin in the DRV8711 BoosterPack is not routed to the MSP430G2 LaunchPad by default. The BEMF pin must be routed to an MSP430 pin with an available ADC peripheral. The pinout diagram omit show that pins p1.1 and p1.2 of header J1 are reserved for UART communication and also have ADC as one of the peripherals. Because UART is unneeded for this application, one of these pins can be used to read the BEMF presented by the DRV8711 on the BEMF pin. A small wire is used to tie the BEMF pin to p1.1 of the MSP430.



**Figure 7. Modified BoosterPack and MSP430G2 LaunchPad**

In the MSP-EXP430G2 LaunchPad, the jumper connections pertaining to UART connection must be disconnected to be able to use the pins p1.1 and p1.2. [Figure 7](#) shows the location of jumpers on the launch pad (in yellow rectangles). The jumper corresponding to the LED should also be used, which indicates when the motor is not operated under control (in black rectangles in [Figure 7](#)).

## 7 Getting Started Firmware

The MSP430™ firmware is provided. This code was developed and modified in Code Composer Studio™ (CCS).

### 7.1 Header File Changes

Open the header file in CCS and make the following changes.

- **Look-up table for speed variation from potentiometer:**
  1. Calibrate the potentiometer to 258 rpm.
  2. Set breakpoints to monitor the present speed.
  3. Note the maximum speed of the motor.
  4. Recalibrate the potentiometer for the maximum value.

---

**NOTE:** This increases the scaling of the speed and is optional.

$$\text{Reference Speed} = [\text{Speed\_max} \times (\text{ADC output})] \div 31$$

The excel sheet includes the formula for this conversion.

---

5. Modify the header file lookup-table for Speed\_funcn[].

- **Look-up table for timer count variation:**

- Based on the look-up table for speed variation from potentiometer, obtain the look-up table for timer count by applying the following formula in the excel sheet:  

$$\text{timer count} = (1000000 \times 60) \div (\text{reference speed} \times \text{step size} \times 200)$$

- **Look-up table for speed from BEMF:** When the BEMF vs speed data is obtained using either of the previously mentioned methods, do the following.

1. Prepare a look up table to include the motor characteristics for the algorithm.
2. Enter 128 discrete values relating the BEMF ADC measurement to speed.

---

**NOTE:** The digital values corresponding to the BEMF values are directly used to point at the speed of the motor in the look-up table.

---

3. Enter the next highest speed at which to operate the Automatic Torque control.
4. Enter the maximum speed value into the values next to the maximum speed

---

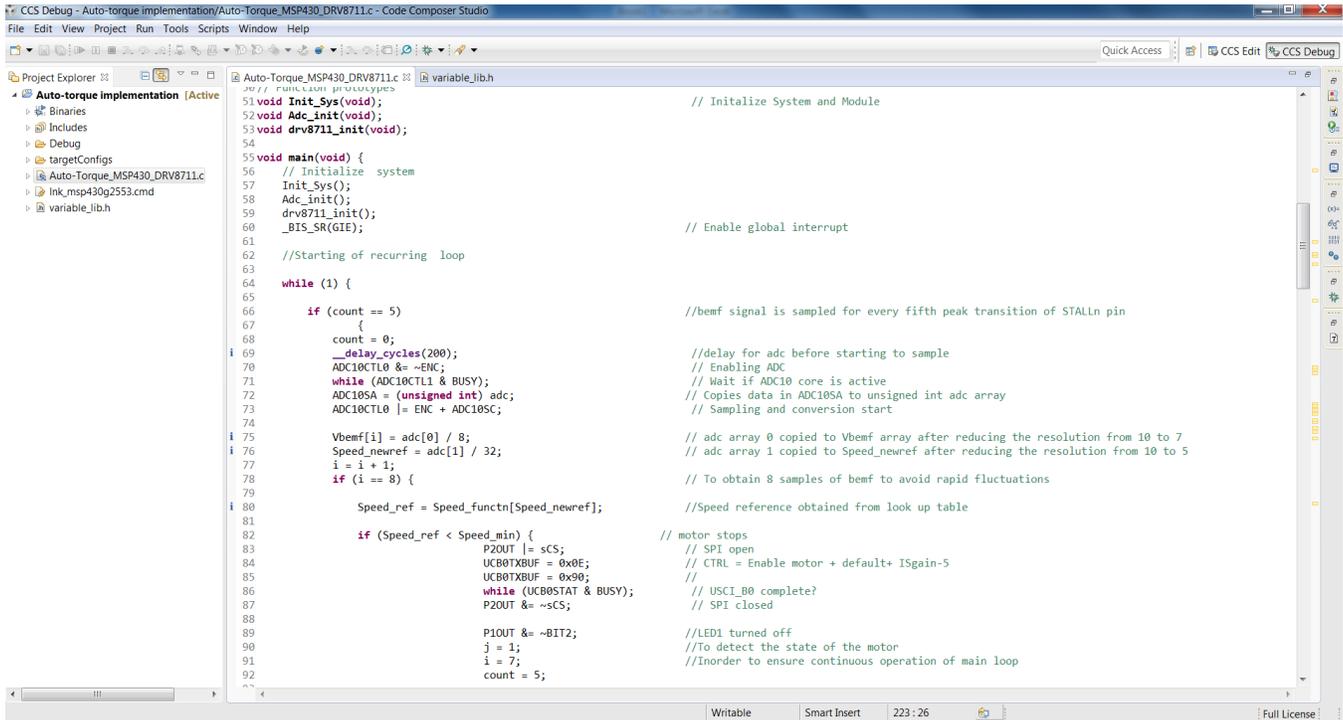
**NOTE:** The excel sheet provides instructions to get the digital values of BEMF readings.

- **Speed details:** The following includes details on speed min, speed max\_NC, speed max, and torque min values.
    - The motor remains at rest if the speed is less than Speed min.
    - The algorithm is disabled if the motor speed is less than max\_NC. Select max\_NC based on the BEMF waveform consistency at low-speed operation. Motors with low inductance and high-current ratings have poor torque handling capabilities at lower speeds.
  - **Control parameters:** Kp and Ki values of the PI controller might require tuning according the performance of the motor after the control implementation. There is a trade off between transient performance and steady state performance. To obtain optimum values, use trial and error.
-

## 7.2 Main Setup

The following steps explain how to program the hardware.

1. Download the Auto-Torque Implementation software project from the software section.
2. Open the project in the CCS (see [Figure 8](#)).
3. Make the changes in the header file if necessary (variable\_lib.h).
4. Debug the project to run the motor.



```

// Auto-Torque Implementation/Auto-Torque_MSP430_DRV8711c - Code Composer Studio
// variable_lib.h
51 void Init_Sys(void); // Initialize System and Module
52 void Adc_init(void);
53 void drv8711_init(void);
54
55 void main(void) {
56 // Initialize system
57 Init_Sys();
58 Adc_init();
59 drv8711_init();
60 _BIS_SR(GIE); // Enable global interrupt
61
62 //Starting of recurring loop
63
64 while (1) {
65
66     if (count == 5) //bemf signal is sampled for every fifth peak transition of STALLn pin
67     {
68         count = 0;
69         __delay_cycles(200); //delay for adc before starting to sample
70         ADC10CTL0 &= ~ENC; // Enabling ADC
71         while (ADC10CTL1 & BUSY); // Wait if ADC10 core is active
72         ADC10SA = (unsigned int) adc; // Copies data in ADC10SA to unsigned int adc array
73         ADC10CTL0 |= ENC + ADC10SC; // Sampling and conversion start
74
75         Vbemf[i] = adc[0] / 8; // adc array 0 copied to Vbemf array after reducing the resolution from 10 to 7
76         Speed_newref = adc[1] / 32; // adc array 1 copied to Speed_newref after reducing the resolution from 10 to 5
77         i = i + 1;
78         if (i == 8) { // To obtain 8 samples of bemf to avoid rapid fluctuations
79
80             Speed_ref = Speed_funcn[Speed_newref]; //Speed reference obtained from look up table
81
82             if (Speed_ref < Speed_min) { // motor stops
83                 P2OUT |= sCS; // SPI open
84                 UCB0TXBUF = 0x0E; // CTRL = Enable motor + default+ ISgain-5
85                 UCB0TXBUF = 0x90; //
86                 while (UCB0STAT & BUSY); // USCI_B0 complete?
87                 P2OUT &= ~sCS; // SPI closed
88
89                 P1OUT &= ~BIT2; //LED1 turned off
90                 j = 1; //To detect the state of the motor
91                 i = 7; //Inorder to ensure continuous operation of main loop
92                 count = 5;
93             }
94         }
95     }
96 }

```

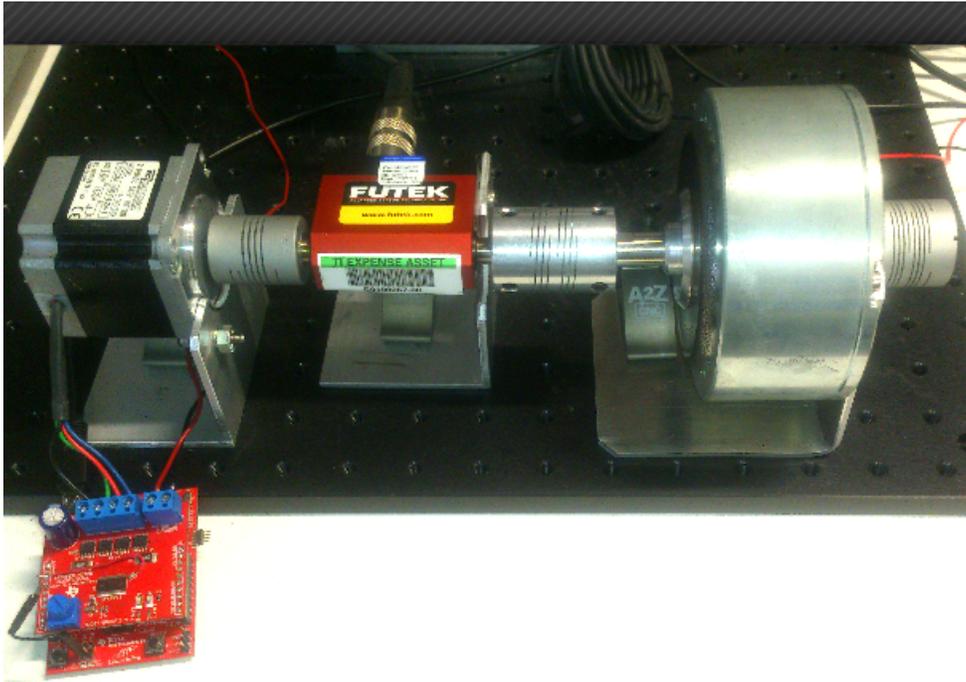
Figure 8. Screenshot of CCS With Auto-Torque Implementation Project Loaded

## 8 Test Setup

The testing setup consists of the DRV8711 BoosterPack with MSP430 LaunchPad, stepper motor, torque sensor, and dynamometer. See [Figure 9](#).

The specifications of the motor tested are as follows:

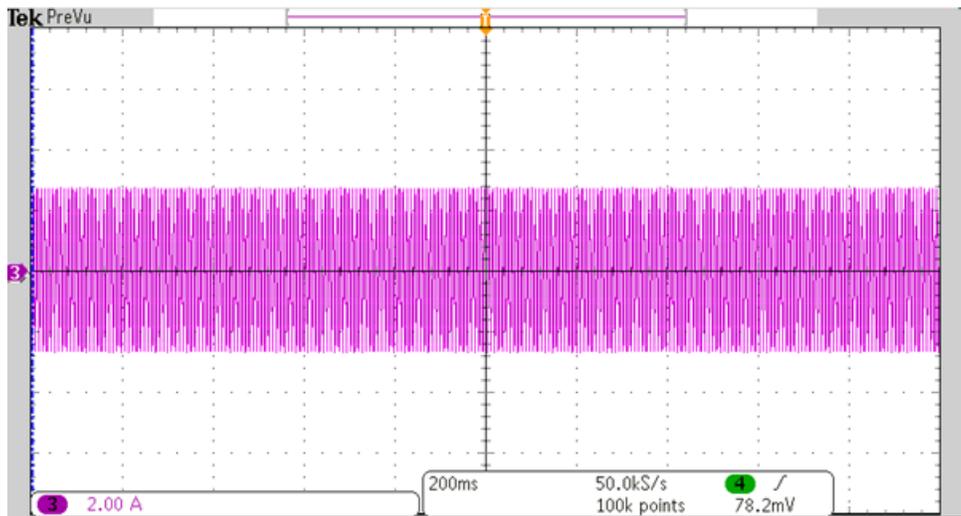
- Step angle: 1.8 degrees
- Rated Current: 1.4 A



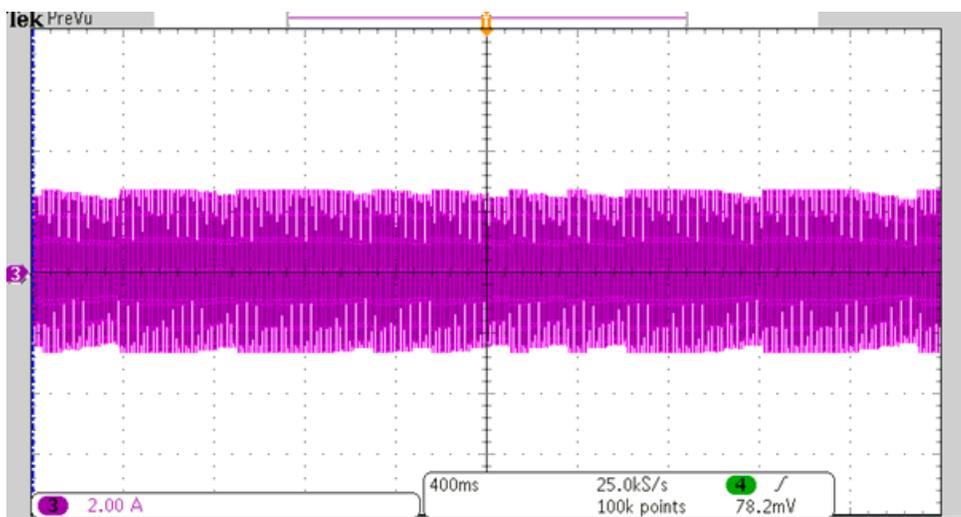
**Figure 9. BoosterPack + LaunchPad, Stepper Motor, Torque Sensor, and Dynamometer**

## 9 Test Data

### 9.1 Current Waveforms at 140 rpm



**Figure 10. Phase Current at Full Load of 34N-cm**



**Figure 11. Phase Current at Load of 29N-cm**

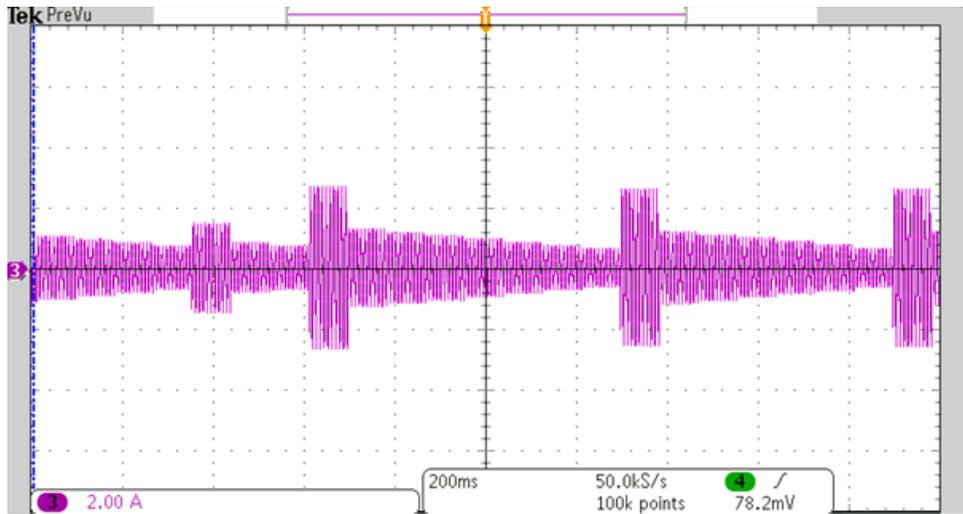


Figure 12. Phase Current at a Load of 21N-cm

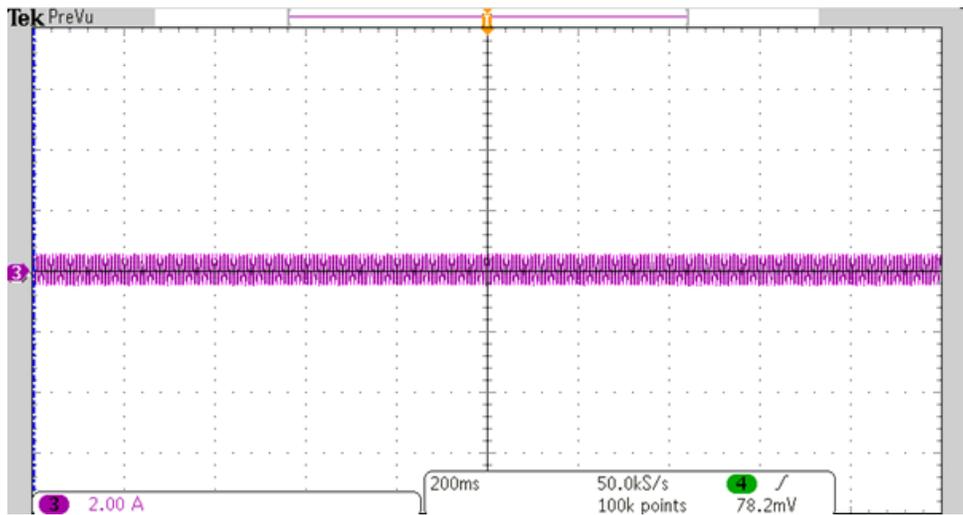


Figure 13. Phase Current at No Load

## 9.2 Current Waveforms at Extreme Conditions

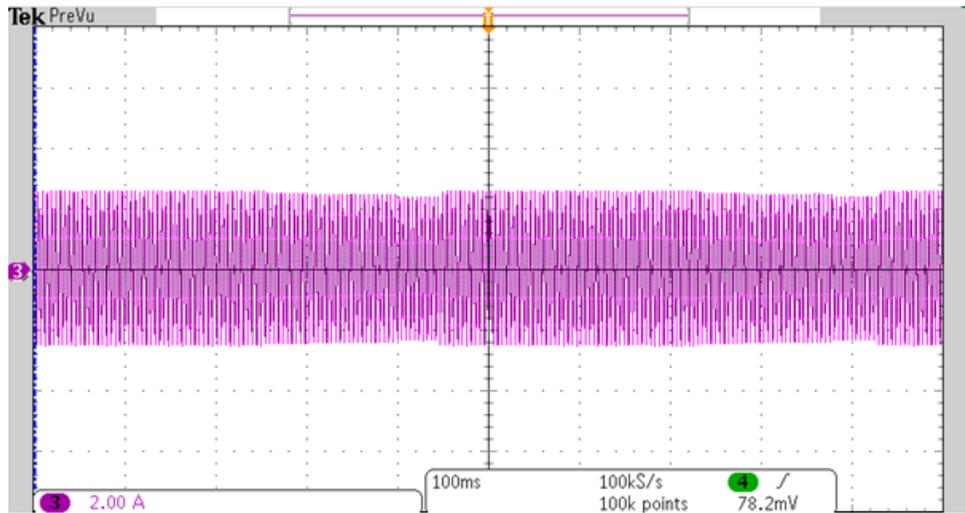


Figure 14. Current at Maximum Speed of 258 rpm and Full Load of 34N-cm

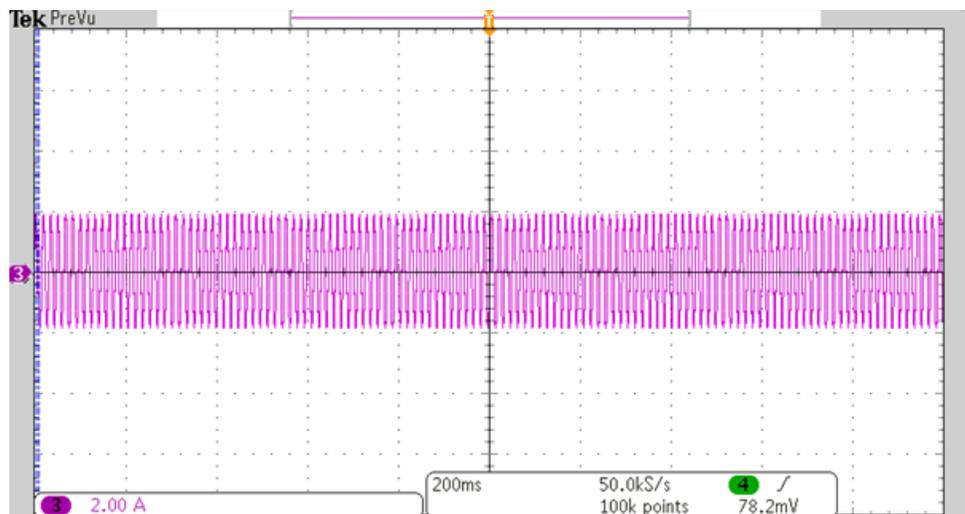


Figure 15. Current at 80 rpm When Motor is Not in Control

### 9.3 Current Transition From One Load State to Another Load State

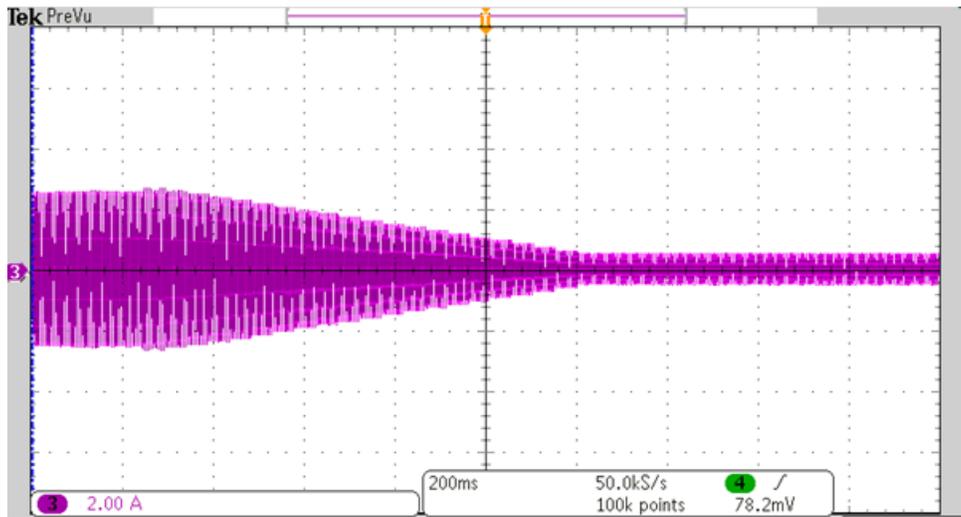


Figure 16. Current Transition From Full Load of 34N-cm to No Load at Maximum Speed

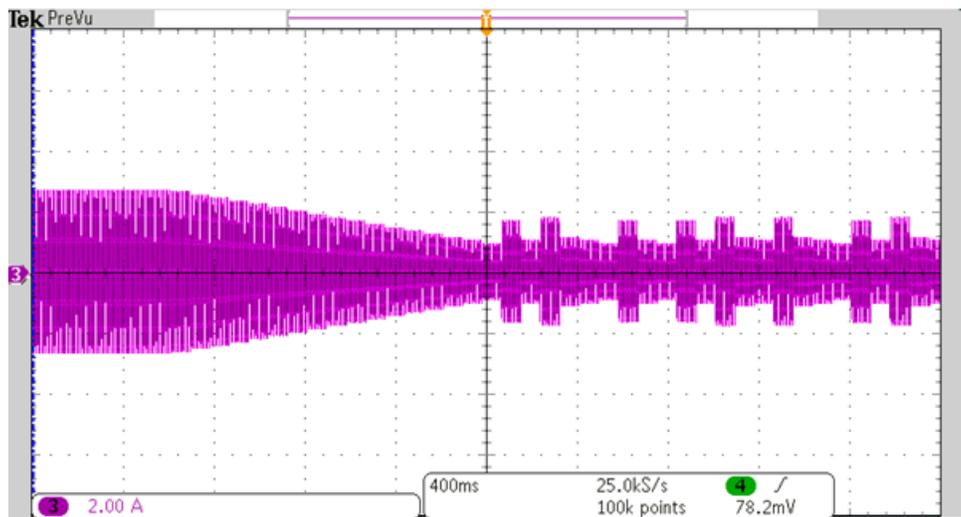


Figure 17. Current Transition From Full Load to Load of 21N-cm at Max Speed

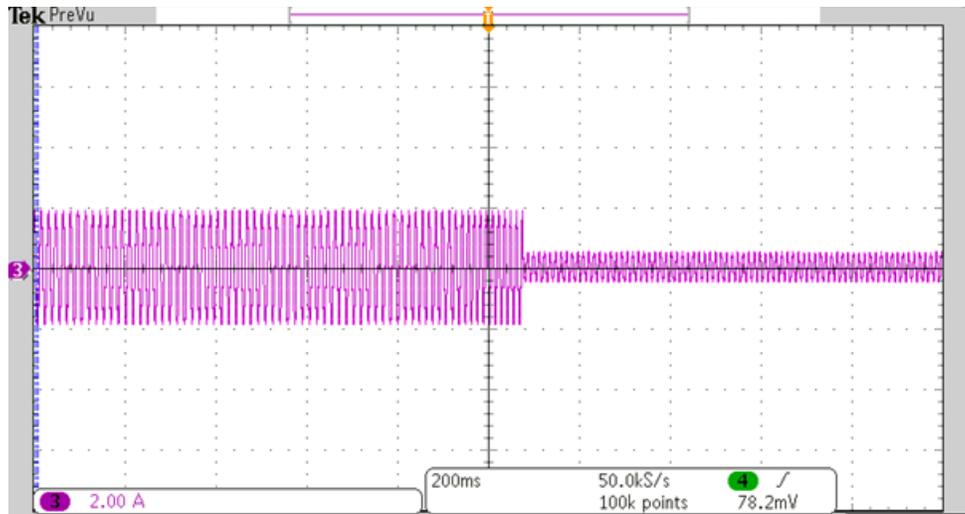


Figure 18. Current Transition From No Control to Control

#### 9.4 Energy Consumption

The energy consumed by the motor is measured using at different speeds and at different loads with and without Auto-Torque control. There is a significant reduction in energy consumption, especially at lower loads. Figure 19 shows an increase in the energy consumption at 225 rpm when the motor is in control because of an observed inconsistency in the BEMF waveform.

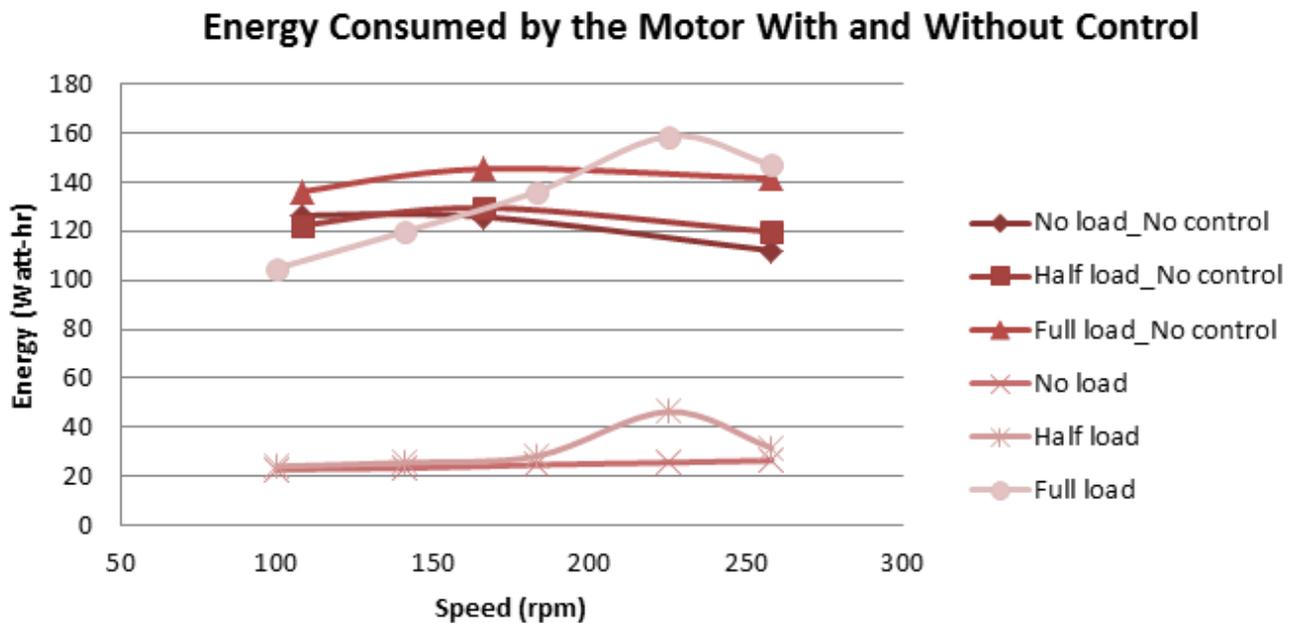


Figure 19. Energy Consumption of the Stepper Motor

## 10 Software Files

To download the software files for this reference design, see the link at <http://www.ti.com/tool/TIDA-00740>.

## 11 References

1. TI WEBENCH Design Center, <http://www.ti.com/webench>
2. TI E2E Community, <http://e2e.ti.com/>
3. DRV8711 Stepper Motor Controller IC datasheet ([SLVSC40E](#))
4. CSD88537ND Datasheet ([SLPS455A](#))
5. BOOST-DRV8711 User's Guide ([SLVU967](#))

## 12 About the Author

**Preethi Peddagorla** is a systems intern at TI, where she develops control strategies for various motor technologies including stepper drivers. Preethi is currently pursuing her Masters in Electrical Engineering (MSEE) at Texas A&M University at College Station, Texas. She is specializing in power electronics and control systems.

## IMPORTANT NOTICE FOR TI REFERENCE DESIGNS

Texas Instruments Incorporated ("TI") reference designs are solely intended to assist designers ("Buyers") who are developing systems that incorporate TI semiconductor products (also referred to herein as "components"). Buyer understands and agrees that Buyer remains responsible for using its independent analysis, evaluation and judgment in designing Buyer's systems and products.

TI reference designs have been created using standard laboratory conditions and engineering practices. **TI has not conducted any testing other than that specifically described in the published documentation for a particular reference design.** TI may make corrections, enhancements, improvements and other changes to its reference designs.

Buyers are authorized to use TI reference designs with the TI component(s) identified in each particular reference design and to modify the reference design in the development of their end products. HOWEVER, NO OTHER LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE TO ANY OTHER TI INTELLECTUAL PROPERTY RIGHT, AND NO LICENSE TO ANY THIRD PARTY TECHNOLOGY OR INTELLECTUAL PROPERTY RIGHT, 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 components or services are used. Information published by TI regarding third-party products or services does not constitute a license to use such products or services, or a warranty or endorsement thereof. Use of such information 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 REFERENCE DESIGNS ARE PROVIDED "AS IS". TI MAKES NO WARRANTIES OR REPRESENTATIONS WITH REGARD TO THE REFERENCE DESIGNS OR USE OF THE REFERENCE DESIGNS, EXPRESS, IMPLIED OR STATUTORY, INCLUDING ACCURACY OR COMPLETENESS. TI DISCLAIMS ANY WARRANTY OF TITLE AND ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, QUIET ENJOYMENT, QUIET POSSESSION, AND NON-INFRINGEMENT OF ANY THIRD PARTY INTELLECTUAL PROPERTY RIGHTS WITH REGARD TO TI REFERENCE DESIGNS OR USE THEREOF. TI SHALL NOT BE LIABLE FOR AND SHALL NOT DEFEND OR INDEMNIFY BUYERS AGAINST ANY THIRD PARTY INFRINGEMENT CLAIM THAT RELATES TO OR IS BASED ON A COMBINATION OF COMPONENTS PROVIDED IN A TI REFERENCE DESIGN. IN NO EVENT SHALL TI BE LIABLE FOR ANY ACTUAL, SPECIAL, INCIDENTAL, CONSEQUENTIAL OR INDIRECT DAMAGES, HOWEVER CAUSED, ON ANY THEORY OF LIABILITY AND WHETHER OR NOT TI HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES, ARISING IN ANY WAY OUT OF TI REFERENCE DESIGNS OR BUYER'S USE OF TI REFERENCE DESIGNS.

TI reserves the right to make corrections, enhancements, improvements and other changes to its semiconductor products and services per JESD46, latest issue, and to discontinue any product or service per JESD48, latest issue. Buyers should obtain the latest relevant information before placing orders and should verify that such information is current and complete. All semiconductor products are sold subject to TI's terms and conditions of sale supplied at the time of order acknowledgment.

TI warrants performance of its components to the specifications applicable at the time of sale, in accordance with the warranty in TI's terms and conditions of sale of semiconductor products. Testing and other quality control techniques for TI components are used to the extent TI deems necessary to support this warranty. Except where mandated by applicable law, testing of all parameters of each component is not necessarily performed.

TI assumes no liability for applications assistance or the design of Buyers' products. Buyers are responsible for their products and applications using TI components. To minimize the risks associated with Buyers' products and applications, Buyers should provide adequate design and operating safeguards.

Reproduction of significant portions of TI information in TI data books, data sheets or reference designs is permissible only if reproduction is without alteration and is accompanied by all associated warranties, conditions, limitations, and notices. TI is not responsible or liable for such altered documentation. Information of third parties may be subject to additional restrictions.

Buyer acknowledges and agrees that it is solely responsible for compliance with all legal, regulatory and safety-related requirements concerning its products, and any use of TI components in its applications, notwithstanding any applications-related information or support that may be provided by TI. Buyer represents and agrees that it has all the necessary expertise to create and implement safeguards that anticipate dangerous failures, monitor failures and their consequences, lessen the likelihood of dangerous failures and take appropriate remedial actions. Buyer will fully indemnify TI and its representatives against any damages arising out of the use of any TI components in Buyer's safety-critical applications.

In some cases, TI components may be promoted specifically to facilitate safety-related applications. With such components, TI's goal is to help enable customers to design and create their own end-product solutions that meet applicable functional safety standards and requirements. Nonetheless, such components are subject to these terms.

No TI components are authorized for use in FDA Class III (or similar life-critical medical equipment) unless authorized officers of the parties have executed an agreement specifically governing such use.

Only those TI components that TI has specifically designated as military grade or "enhanced plastic" are designed and intended for use in military/aerospace applications or environments. Buyer acknowledges and agrees that any military or aerospace use of TI components that have **not** been so designated is solely at Buyer's risk, and Buyer is solely responsible for compliance with all legal and regulatory requirements in connection with such use.

TI has specifically designated certain components as meeting ISO/TS16949 requirements, mainly for automotive use. In any case of use of non-designated products, TI will not be responsible for any failure to meet ISO/TS16949.