

# **BOOST-DRV8711**

This document is provided with the BOOST-DRV8711 Stepper Motor BoosterPack as a supplement to the [DRV8711 datasheet](#) and [CSD88537ND datasheet](#) to detail the hardware setup and operation of the BoosterPack.

### Contents

1	BOOST-DRV8711 Views .....	2
2	Introduction to the BOOST-DRV8711 .....	3
	2.1 Features.....	3
	2.2 Pinout.....	3
3	Getting Started .....	4
	3.1 Requirements .....	4
	3.2 Configuring the LaunchPad .....	4
	3.3 Connecting the Hardware.....	7
4	Stepper Motor Demo Application .....	8
	4.1 Introduction .....	8
	4.2 Setting up the BOOST-DRV8711 Firmware.....	8
	4.3 Setting up the BOOST-DRV8711 GUI.....	11
	4.4 Spinning Your Stepper Motor.....	12
5	Hardware Files (Schematic/Gerber) .....	14

### List of Figures

1	BOOST-DRV8711 Stand-Alone.....	2
2	BOOST-DRV8711 on MSP-EXP430G2 .....	2
3	BOOST-DRV8711 Pinout.....	3
4	MSP-EXP430G2 Jumper Location .....	5
5	Target Configuration .....	8
6	Target Configuration Setup.....	9
7	Launch Selected Configuration .....	9
8	Connect Target Selection .....	10
9	Load Program Selection.....	10
10	Existing CCS Eclipse Projects.....	11
11	BOOST-DRV8711 GUI Screen.....	12
12	BOOST-DRV8711 GUI CONTROL Tab.....	13
13	BOOST-DRV8711 GUI REGISTERS Tab .....	14

# 1 BOOST-DRV8711 Views

The views in Figure 1 and Figure 2 are of the BOOST-DRV8711 Stepper Motor BoosterPack.

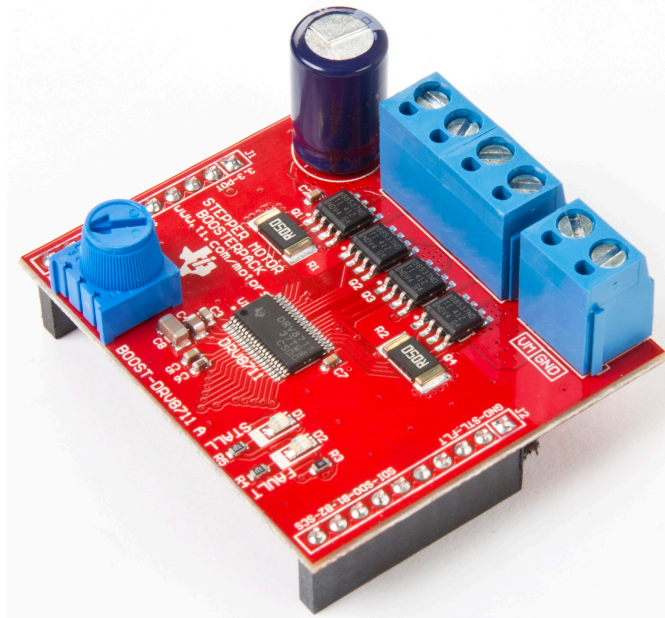


Figure 1. BOOST-DRV8711 Stand-Alone

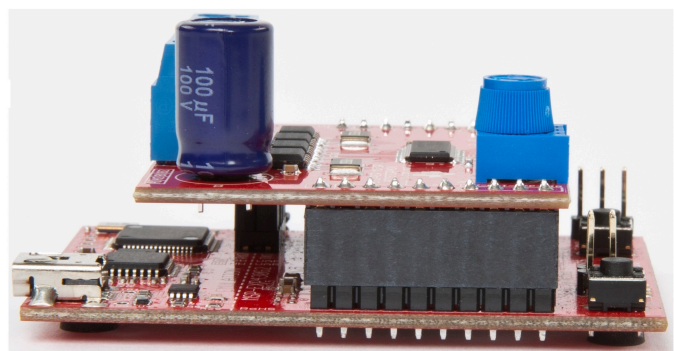
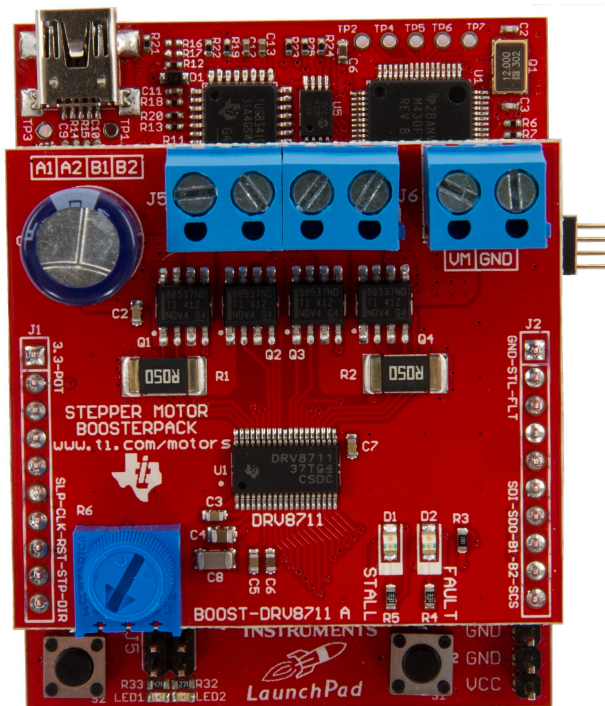


Figure 2. BOOST-DRV8711 on MSP-EXP430G2

## 2 Introduction to the BOOST-DRV8711

The 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 in order to evaluate your motor applications.

### 2.1 Features

- Complete stepper motor drive stage in a small form factor (1.75 in × 2.00 in)
- Supports 8.2–52 V and up to 4.5 A continuous for each H-Bridge
- 4x CSD88537ND Dual 60-V N-Channel NexFET Power MOSFETs (12.5 mΩ)
- Motor stall and device fault LED indicators
- Fully protected drive stage including overcurrent, overtemperature, undervoltage, and motor stall detect
- Combine with TI LaunchPad kits to create a complete stepper motor drive and control platform
- Optimized for the MSP-EXP430G2 LaunchPad with a user-friendly application to get your motor spinning in minutes

### 2.2 Pinout

The BOOST-DRV8711 brings out a mixture of power, control, and feedback signals to the LaunchPad headers.

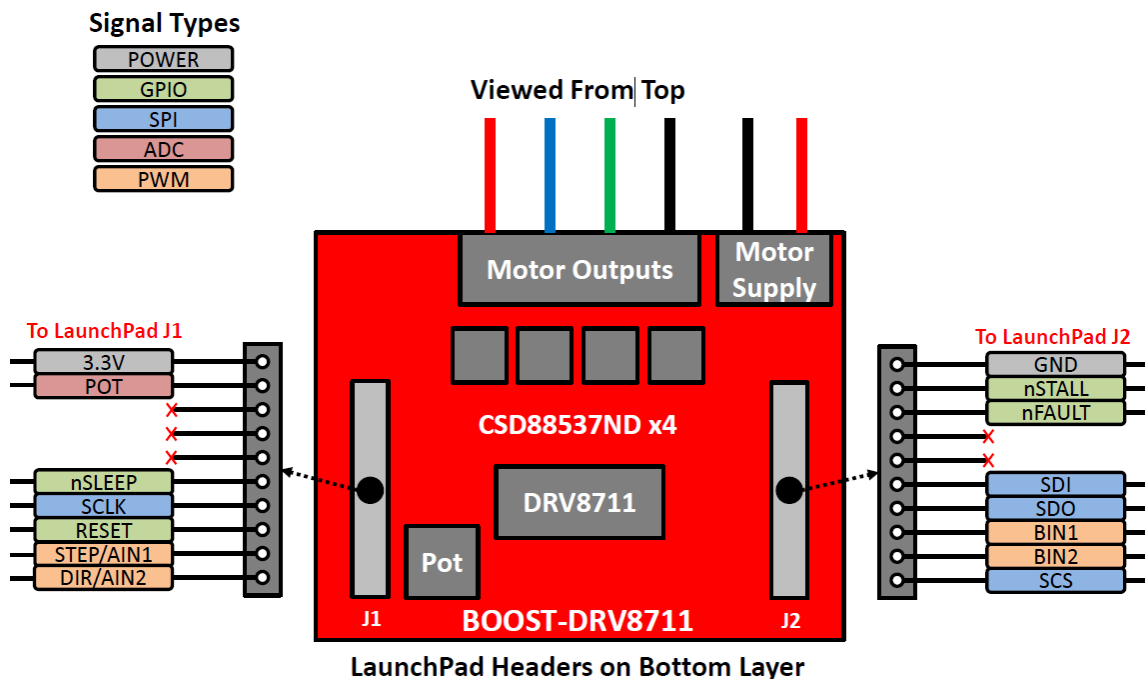


Figure 3. BOOST-DRV8711 Pinout

- Powered by an external power supply (8.2–52 V) that can be connected to the terminal block header (J6)
- 4-pin terminal block header (J5) for connecting a bipolar stepper motor
- Fault and motor status reporting through the nFAULT and nSTALL signals
- SPI interface to set device configuration, operating parameters, and read out diagnostic information
- Built-in microstepping indexer through a STEP/DIR interface or H-Bridge control through an IN/IN interface
- Onboard potentiometer for creating your own, easy-to-use, demo application

## 3 Getting Started

### 3.1 Requirements

The Stepper Motor BoosterPack is not a standalone evaluation board and requires a compatible LaunchPad kit to provide the appropriate control signals. In addition to the Stepper Motor BoosterPack and a compatible LaunchPad, a stepper motor and sufficient power supply are required.

### 3.2 Configuring the LaunchPad

#### 3.2.1 Example Using MSP-EXP430G2

For the MSP-EXP430G2 LaunchPad, remove the P1.0 and P1.6 (LED1 and LED2) jumpers for correct operation of the POT and SDO signals of the BoosterPack. The TXD and RXD jumpers should be aligned horizontally (yellow arrows in [Figure 4](#)) for operation of the hardware UART with a software application.



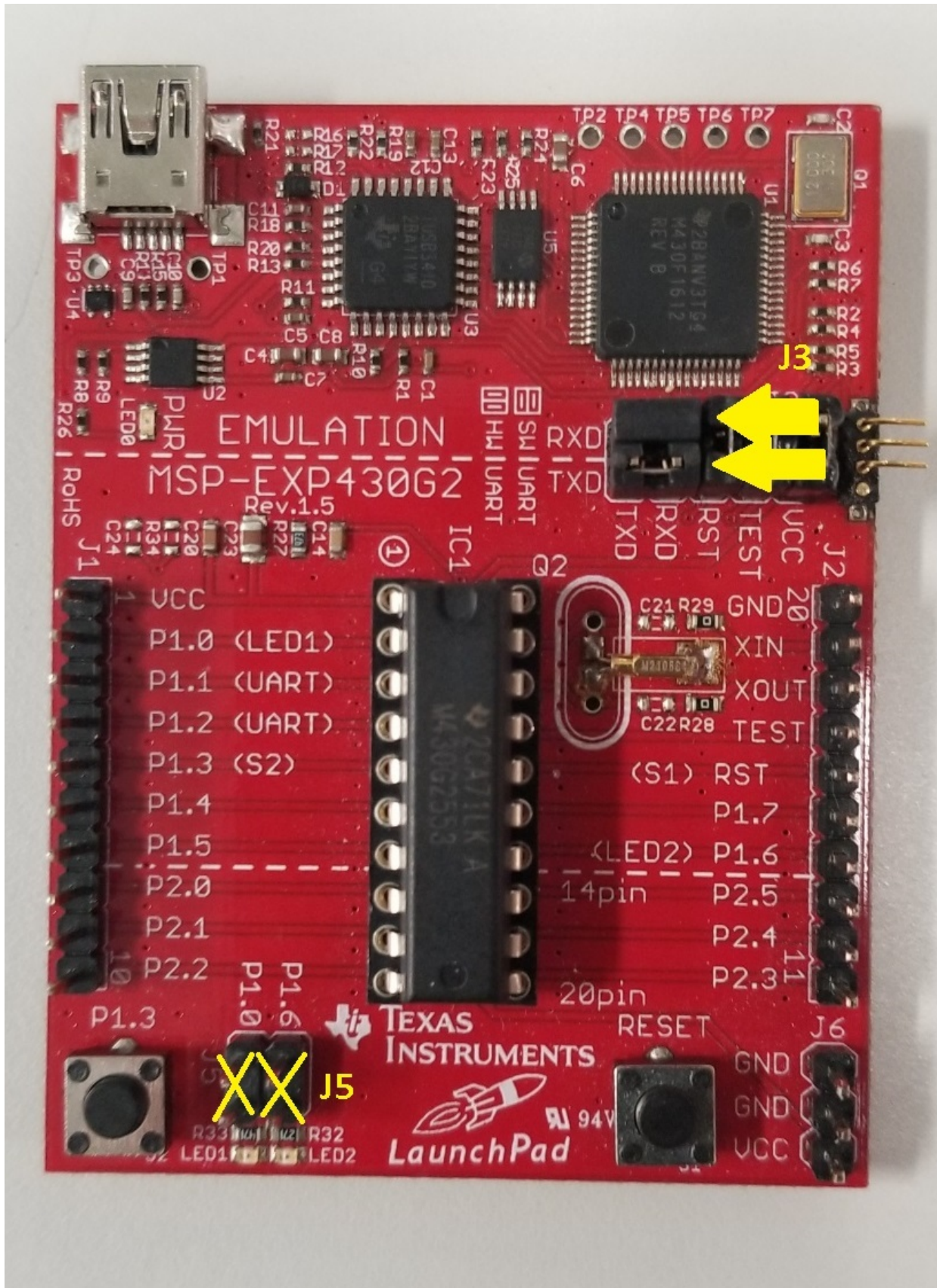


Figure 4. MSP-EXP430G2 Jumper Location

---

For other LaunchPads, ensure that the appropriate jumpers are configured on pins that the BOOST-DRV8711 BoosterPack utilizes.

### 3.3 Connecting the Hardware

1. Plug the Stepper Motor BoosterPack onto the LaunchPad as shown in [Figure 2](#). **The terminal block headers should be oriented towards the USB connector.**
2. Connect your stepper motor to the **terminal block header J5**. The motor should have two windings, each with a + and a – termination. Connect one winding to A1/A2 and the other to B1/B2 (polarity does not matter). For questions on the motor wire coloring coding, please see the motor's datasheet.
3. Connect the power supply that will power the Stepper Motor BoosterPack's DRV8711 and Drive Stage to the terminal block header J6. The connections have been labeled VM and GND. For full performance, ensure you can supply as much current as your motor may demand. The Stepper Motor BoosterPack has a designed operating range from 8.2–52 V up to 4.5 A continuous for each H-Bridge.

#### **WARNING**

**At high currents the drive stage can increase to high temperatures**

4. Enable the power supply
5. Enable your controller and spin the motor. The BOOST-DRV8711 Stepper Motor BoosterPack combined with a TI LaunchPad provides a complete stepper motor evaluation platform. With the MSP-EXP430G2 LaunchPad and a MSP430G2553 you can take full advantage of TI's **pre-written stepper motor control application** (see [Section 4](#) for additional details).

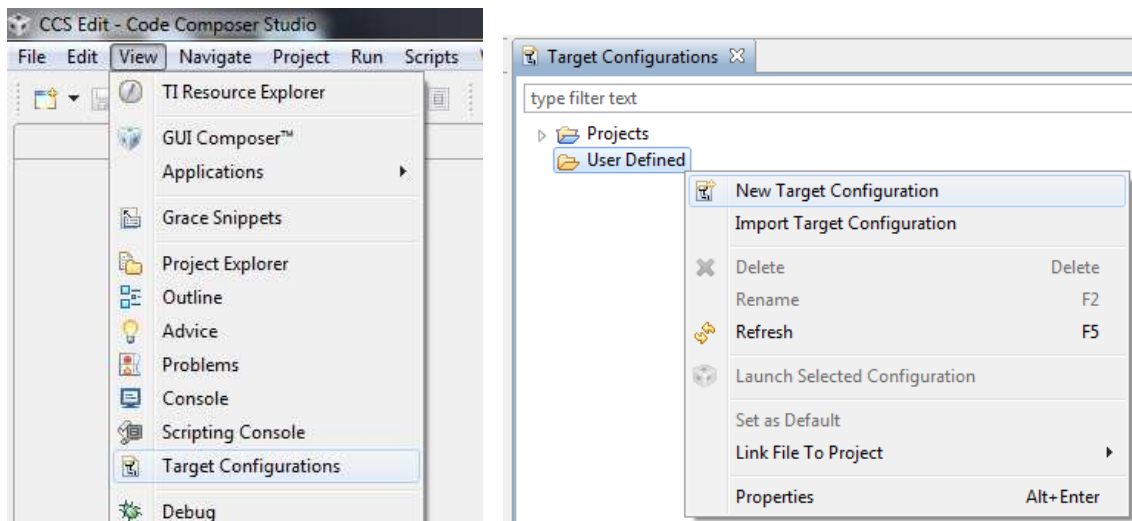
## 4 Stepper Motor Demo Application

### 4.1 Introduction

As mentioned earlier, the BOOSTXL-DRV8301 Motor Drive BoosterPack has been optimized to work together with the MSP430G2 Launchpad, MSP-EXP430G2, and MSP430G2553 to provide a complete stepper motor evaluation platform. With the demo application provided, you can have your stepper motor up and spinning in minutes. **Get started with TI's pre-written stepper control application by following the steps outlined in Section 4.2 through Section 4.4.**

### 4.2 Setting up the BOOST-DRV8711 Firmware

1. Download the latest version of Code Composer Studio to load the BOOST-DRV8711 stepper motor control application onto the MSP430G2553. The application was developed in CCS v5.5.0.  
[http://processors.wiki.ti.com/index.php/Download\\_CCS](http://processors.wiki.ti.com/index.php/Download_CCS)
2. To obtain the BOOST-DRV8711 firmware and GUI, download the BOOST-DRV8711 Hardware and Software Files from the [tool folder](#). This zip folder contains the complete hardware design files, including the Altium source files, Gerbers, BOM, schematic, as well as the Stepper Motor Demo firmware and GUI.
3. Flash the MSP430G2553 on the MSP430G2 LaunchPad with the firmware provided. This will require the MSP430G2 LaunchPad, a MSP430G2553, Mini USB cable, and can be done in one of two ways which are outlined in [Step 4](#) and [Step 5](#).
4. Method 1: Loading the binary .out file
  - Open Code Composer Studio.
  - Select **View** → **Target Configuration** ([Figure 5](#), left side of image)
  - Right click on the **User Defined** folder and select **New Target Configuration** ([Figure 5](#), right side of image)



**Figure 5. Target Configuration**

- Give the Target Configuration a name and select **Finish**
- [Figure 6](#) illustrates how the Target Configuration should be set up. Save the Target Configuration file.



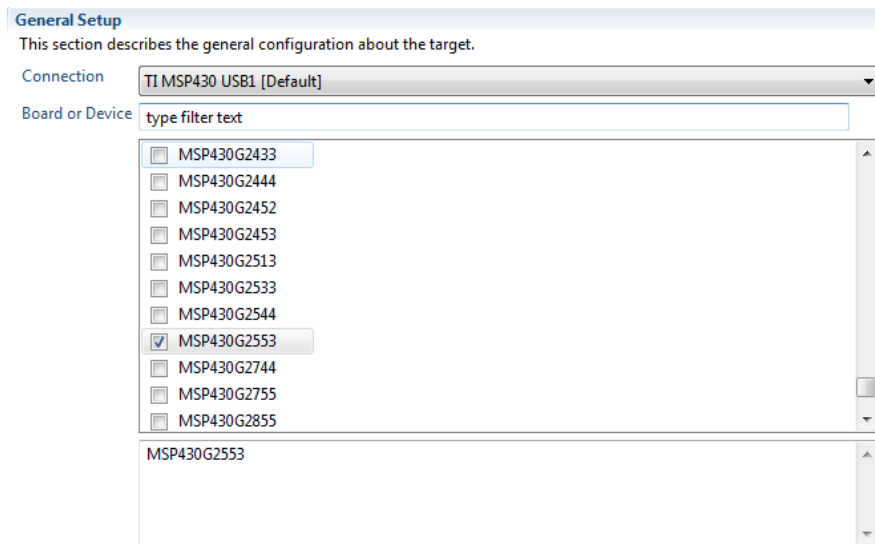


Figure 6. Target Configuration Setup

- Go back to **View** → **Target Configurations**
- Right click your newly created Target Configuration file and select **Launch Selected Configuration** (Figure 7)

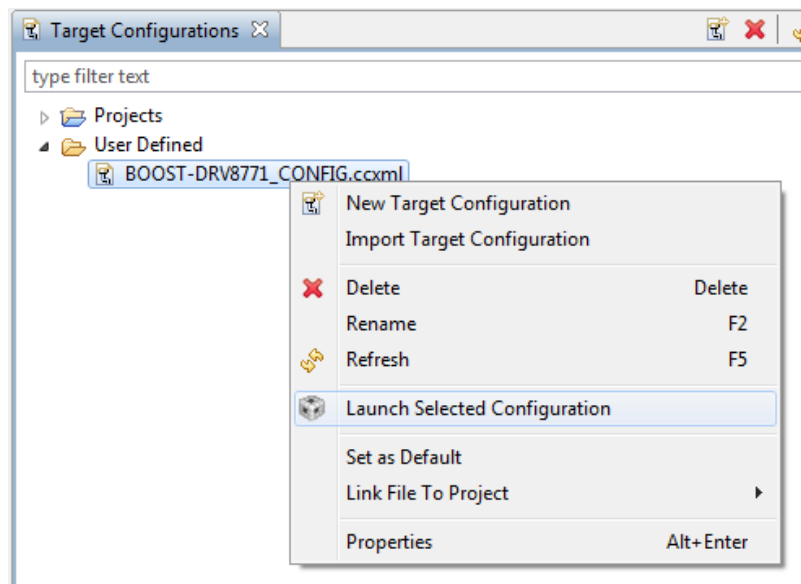
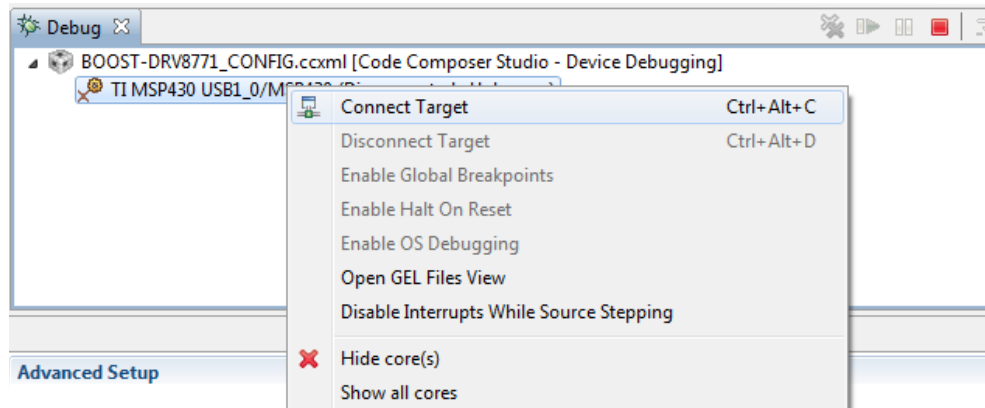


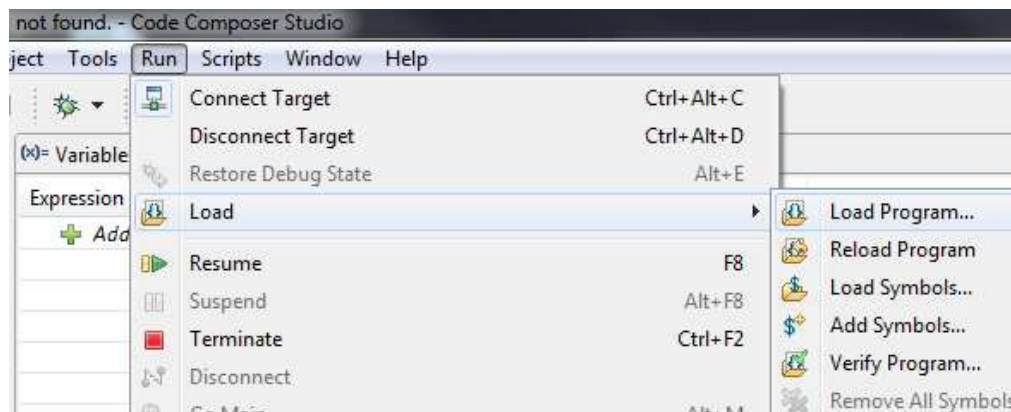
Figure 7. Launch Selected Configuration

- In the Debug Menu, right click on the **TI MSP430 USB1\_0** connection and select **Connect Target** (Figure 8)



**Figure 8. Connect Target Selection**

- After the device connects, go to **Run** → **Load** → **Load Program** (Figure 9). Browse to the **BOOST-DRV8711 Hardware and Software Files** folder that was downloaded from the tool folder. Go into the Application subdirectory and then go into the GUI subdirectory. Select the **appProgram.out** file (Application\BOOST-DRV8711\_GUIvX.X\appProgram.out).



**Figure 9. Load Program Selection**

- After a short delay, the appProgram.out file is loaded onto the MSP430G2553
  - Close CCS
5. Method 2: Flashing the project through CCS debugger
    - Open Code Composer Studio
    - Go to **File** → **Import**

- Select **Existing CCS Eclipse Projects** under the **Code Composer Studio** tab (Figure 10)

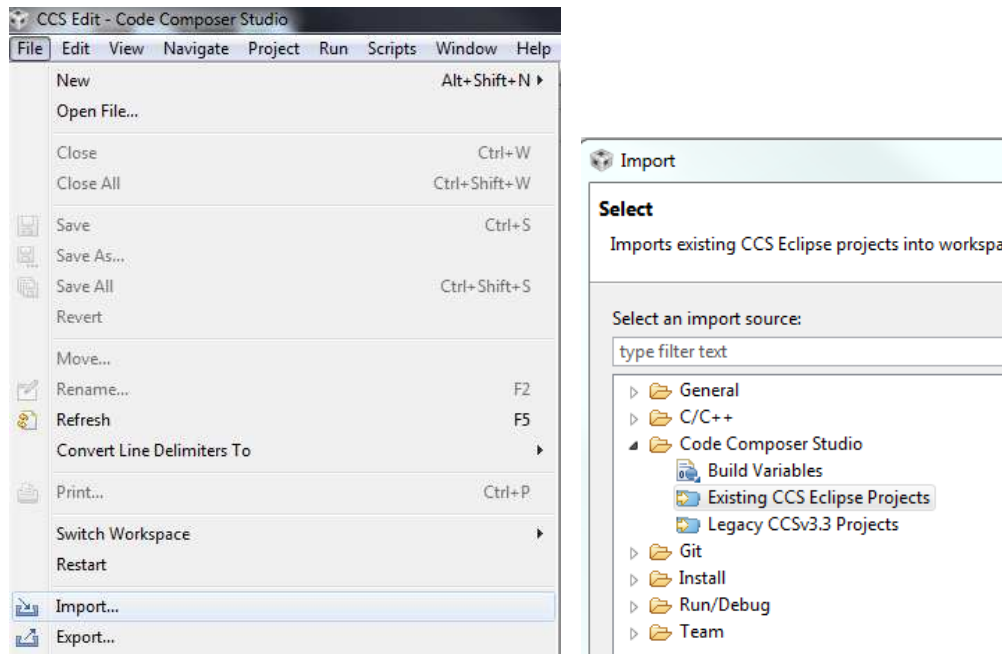


Figure 10. Existing CCS Eclipse Projects

- Check the **Copy projects into workspace** option and then browse to the **BOOST-DRV8711\_FIRMWAREvX.X** directory located with the Application folder within the **BOOST-DRV8711 Hardware and Software Files** folder. The project should now show up in the **Discovered Projects** section. Ensure that it is checked and select **Finish**.
- Select the **BOOST-DRV8711\_FIRMWAREv1.0** project in the **Project Explorer** and click the **Debug** icon
- CCS will now build the project and load it onto the **MSP430G2533**
- Close CCS

### 4.3 Setting up the BOOST-DRV8711 GUI

1. Download the latest version of the GUI Composer Runtime to initially run the BOOST-DRV8711 GUI. You must register for a TI account if you don't already have one. Select the appropriate version for your operating system and follow the install instructions.  
[http://processors.wiki.ti.com/index.php/Category:GUI\\_Composer#GUI\\_Composer\\_Downloads](http://processors.wiki.ti.com/index.php/Category:GUI_Composer#GUI_Composer_Downloads)
2. After installing the GUI Composer Runtime, copy the **BOOST-DRV8711\_GUIvX.X** folder, located in the Application directory of the BOOST-DRV8711 Hardware and Software Files folder and paste this folder into the GUI Composer webapps folder located in the **C:\ti\guicomposer\webapps\** directory.  
**(Note: if you chose a non-default installation directory in Step 1, the top-level directory may differ)**
3. To run the GUI, double click the **BOOST-DRV8711\_GUIvX.X.exe** file within the **BOOST-DRV8711\_GUIvX.X** folder of the webapps directory. You can make a shortcut to this .exe in order to start it from other file locations.

**\*\*Ensure that the GUI "exe" is exactly two levels below the GUI Composer "webapps" folder. The GUI will not start if this is incorrect. The path should look similar to this**

**C:\ti\guicomposer\webapps\BOOST-DRV8711\_GUIvX.X\BOOST-DRV8711\_GUIvX.X.exe.**

**(Note: if you chose a non-default installation directory in Step 1, the top-level directory may differ)**

### 4.4 Spinning Your Stepper Motor

After a successful launch of the *BOOST-DRV8711\_GUIvX.X.exe*, [Figure 11](#) pops up. It may take a small period of time before the GUI connects and the GUI Widgets populate (red X's appear on the widgets while the GUI is connecting). If the GUI does not load after a few minutes (the X's disappear), a connection issue may have occurred and TI recommends restarting the application.

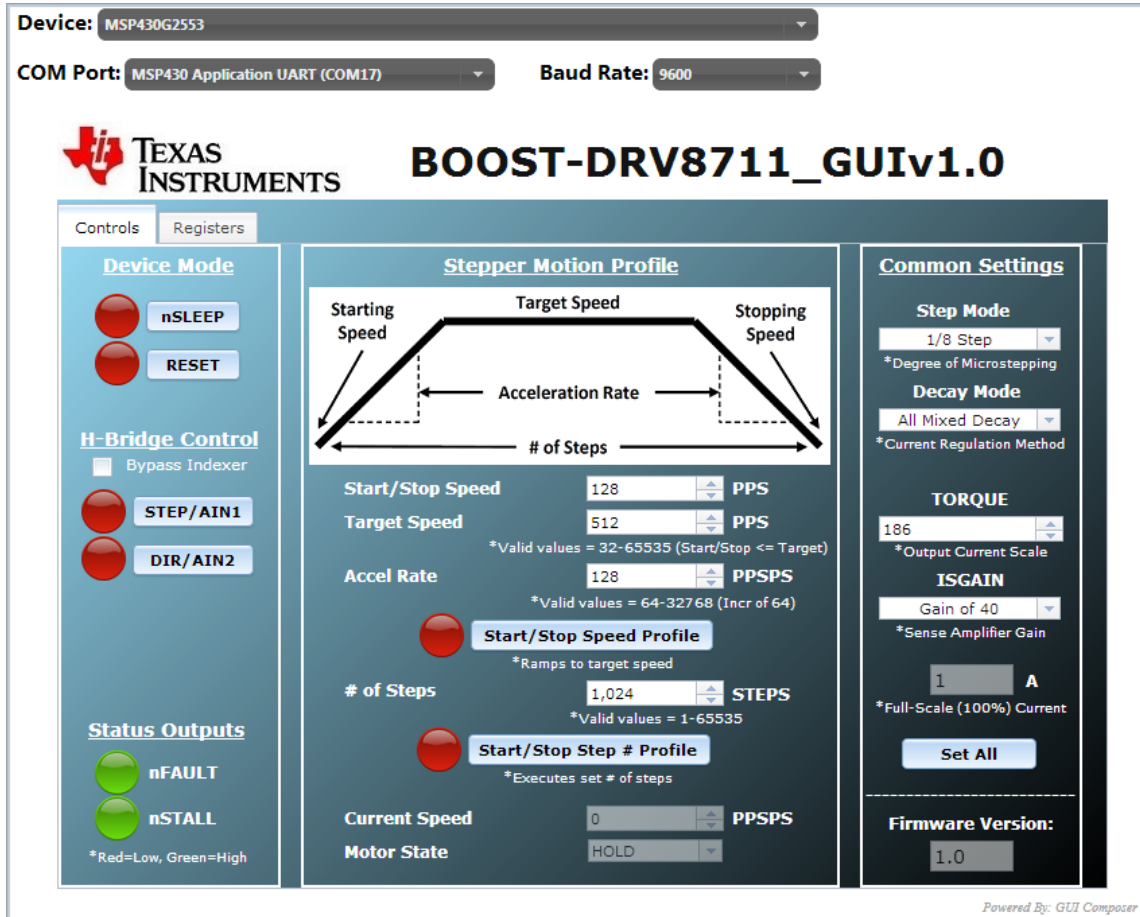


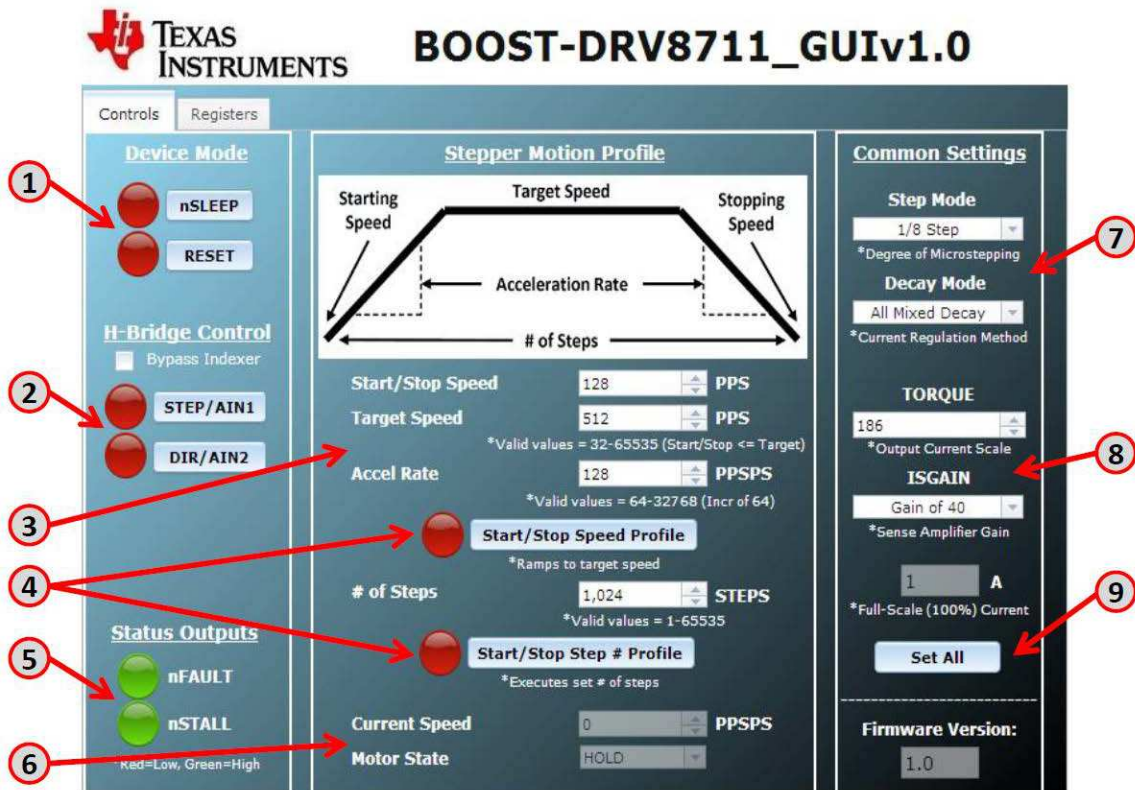
Figure 11. BOOST-DRV8711 GUI Screen

#### 4.4.1 Quick Start

1. Set Your **Full Scale Current** level appropriately by adjusting the **TORQUE** and **ISGAIN** settings. Your **Full Scale Current** level is determined by your stepper motor's current rating and power supply capability. Click the **Set All** button after choosing the appropriate settings.
2. Select your **Step Mode**. This determines the level of microstepping applied to the motor.
3. Set the **nSLEEP** pin high to bring the DRV8711 out of sleep mode. The DRV8711 now begins regulating current.
4. Adjust the **Stepper Motion Profile** parameters to the desired values. The units are **Pulses Per Second**, otherwise known as **Steps Per Second**.
5. Enable the **Speed Profile** or **Step # Profile** button depending on the desired mode.

The BOOST-DRV8711 GUI provides two tabs. The first tab controls the Stepper Motor and the second tab sets the registers of the DRV8711.

#### 4.4.2 CONTROL Tab Walkthrough



**Figure 12. BOOST-DRV8711 GUI CONTROL Tab**

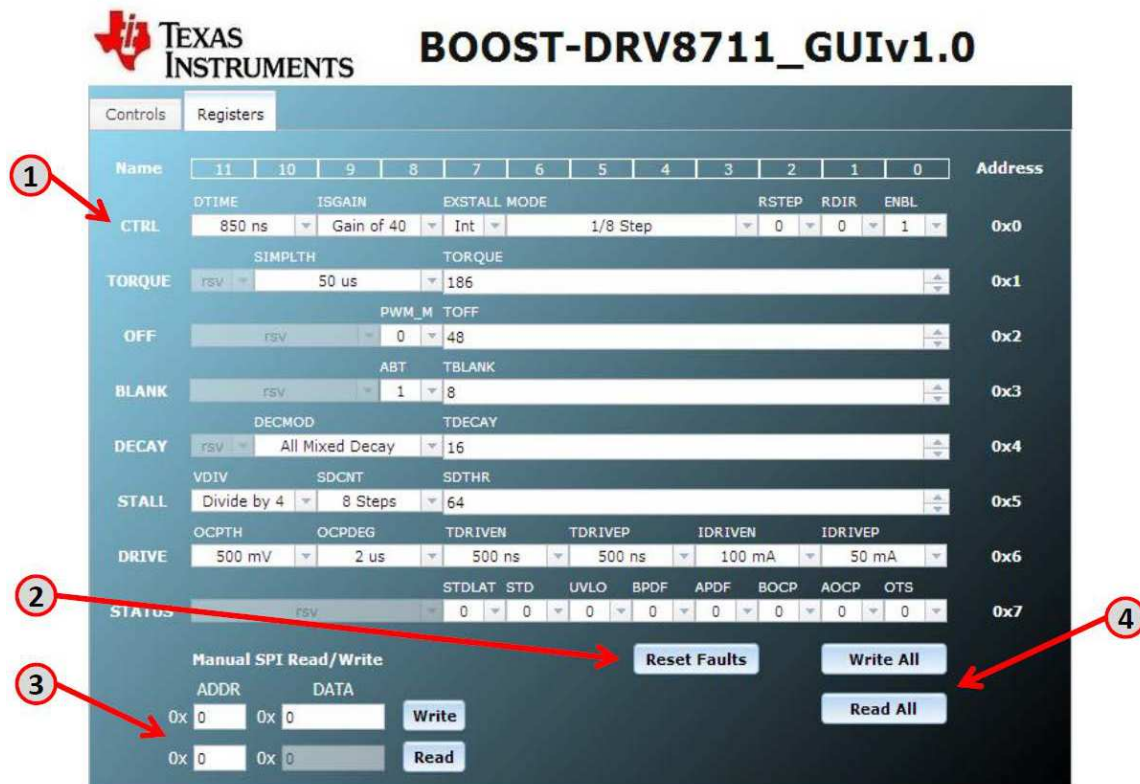
1. The nSLEEP and RESET buttons directly control the nSLEEP and RESET pins of the DRV8711. Red indicates LOW (0 V) and green indicates HIGH (3.3 V). nSLEEP = LOW puts the DRV8711 in a low power sleep mode. RESET = HIGH resets the internal logic and disable the H-bridge outputs.
2. The STEP and DIR buttons give you command of the stepper motor. STEP moves the motor one step, independent of the Stepper Motion Profile. DIR selects which direction the motor is spinning.
3. The Stepper Motion Profile provides a method to spin the stepper motor in a variety of ways. It gives you command of the starting/stopping speed, acceleration rate, target speed, and number of steps (if using the step # profile). The units are pulses per second, or steps per second, as the DRV8711 will move a step with every rising edge it sees.
4. The Speed Profile button (once selected) accelerates the stepper motor from the starting speed to the target speed. The motor remains at this speed until the Speed Profile button is selected again. The Step # Profile (once selected) moves the stepper motor the specified number of steps while attempting to maintain the speed profile. Due to approximation errors on the MCU, the target speed may not exactly match the target speed.
5. The nFAULT and nSTALL provide status about the motor and motor driver. The nFAULT reports on a variety of faults for the DRV8711. A more detailed description of the various faults can be found in the datasheet. nSTALL is a feature of the DRV8711 to detect a motor stall. This feature must be calibrated to function properly. Please see the datasheet for additional information.
6. The Current Speed box provides information about the current speed of the speed motor. The Motor State box indicate the status of the motor, whether it is stopped, accelerating, decelerating, and so forth. Note that there is a slight time delay in the GUI (~ 1 second).
7. The Step Mode setting determines the microstepping level of the DRV8711. The Decay Mode determines the decay method of the current regulation scheme. The recommended modes for stepper motors are All Mixed and Auto Mixed Decay. The decay method is fine tuned in the Registers tab.
8. The TORQUE and ISGAIN settings, in combination with the hardware SENSE resistor, determine the full scale current of the current regulation scheme. By adjusting these settings you can adjust the full



scale current level appropriately for your specific motor. Refer to the datasheet for the exact formula for full scale current.

- The Set All button takes the settings in the GUI and writes them to the DRV8711. Click this button to update the DRV8711 once you have made the desired settings.

#### 4.4.3 REGISTERS Tab Walkthrough



**Figure 13. BOOST-DRV8711 GUI REGISTERS Tab**

- The BOOST-DRV8711 GUI provides access to all of the register settings of the DRV8711. Use this page to fine tune the motor driver settings. The register name as well as its hexadecimal address is shown. Refer to the [DRV8711 datasheet](#) for a more detailed description of each setting.
- The Reset Faults button resets any faults that have occurred while driving your stepper motor. If a fault occurs while spinning your motor (FAULT LED lights ups), select this button to clear the fault. Selecting the button only clears the fault if the fault condition has been removed. Please refer to the DRV8711 datasheet for a detailed description of possible fault conditions.
- The Manual SPI Read/Write section allows you to manually read or write hexadecimal values to the DRV8711.
- The Write All button updates the DRV8711 with the values set in the GUI. The Read All button updates the GUI with the values from the DRV8711.

## 5 Hardware Files (Schematic/Gerber)

The complete design files are found in the tool folder (<http://www.ti.com/tool/boost-drv8711>) including the schematic, Gerbers, layout files, PCB views, and bill of materials.

## IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES "AS IS" AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI's products are provided subject to TI's Terms of Sale ([www.ti.com/legal/termsofsale.html](http://www.ti.com/legal/termsofsale.html)) or other applicable terms available either on [ti.com](http://ti.com) or provided in conjunction with such TI products. TI's provision of these resources does not expand or otherwise alter TI's applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265  
Copyright © 2019, Texas Instruments Incorporated