TI Designs

Low-Power Micro Stepper Motor Driver Using FRAM MCU Design Guide

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

TIDM-LPSM	Tool Folder Containing Design Files
MSP-EXP430FR5969	Product Folder
DRV8836	Product Folder
INA199A1	Product Folder

TI E2E™ Community

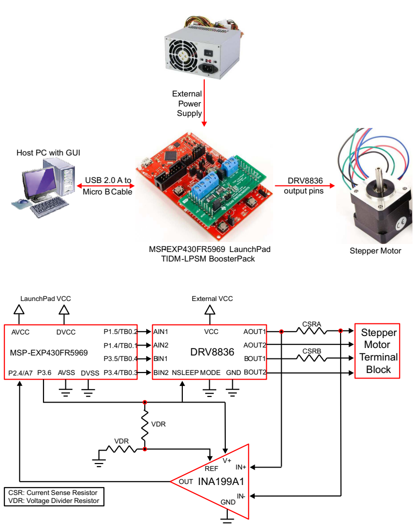
ASK Our E2E Experts
WEBENCH® Calculator Tools

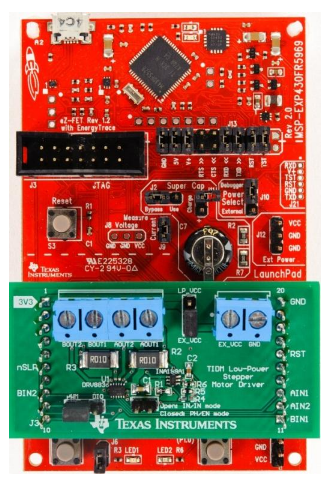
Design Features

- DRV8836 Sleep State And The MSP430FR5969 MCU's LPM3 And LPM3.5 Modes Provide Longer Battery Discharge Times With Average Current Consumption of 20 µA and 10 µA
- Beneficial Motor Feedback Produced by the Bi-Directional INA199A1 Current Shunt Monitor Featuring Low Offset and Minimal Error
- Motor States Uninterrupted by Power Cycles due to MSP FRAM Technology
- EnergyTrace™++ Technology Measures and Optimizes Low Power Consumption
- Java-based GUI Allow the User to Design a Repetitive 24-Hour Motor Operation Schedule by Employing The MSP RTC

Featured Applications

- Timing-Based Automated Operation
- Cameras And Consumer Products
- Stepper Or Single / Dual Brushed DC Motor Control





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 Description

The TIDM-LPSM (Low-Power Stepper Motor) is a 20-pin BoosterPack™ that, when used with the MSP-EXP430FR5969 LaunchPad™, offers exceptional micro-stepping control of up to one sixteenth of a step, translating to 1 / 3200th of a revolution for 200 steps per revolution low-voltage stepper motors. Through the utilization of MSP low-power modes and the DRV8836 stepper-motor-driver sleep state, power consumption has been minimized to an average of 20 µA and 10 µA for LPM3 and LPM3.5, respectively, resulting in optimal battery life. An INA1991A1 current shunt monitor is also included to provide a feedback loop to the MSP430FR5969 regarding the motor's current draw, allowing for close monitoring of motor behavior. With MSP430™ FRAM technology, motor states and schedules can be saved and restored in the event of a power cycle or supply disconnection. This design is intended for any application where low-voltage stepper motors can be applied, including automated operation (such as valve control), small robotics, and handheld devices like powered tools. The DRV8836 also allows for substitution of single- or dual-brushed DC motors in place of a stepper motor if desired. Although anticipated for the MSP-EXP430FR5969 hardware, the TIDM-LPSM adheres to the BoosterPack pin standard and therefore can be operated with any TI LaunchPad available.

This TI design delivers all resources necessary to begin a stepper motor project with the TIDM-LPSM, including a Code Composer Studio™ C-code application demo and java-based GUI. When employed congruently, these two projects can either command motor movement immediately or program and upload scheduled moves based on a repeating 24-hour clock. Figure 1 shows a typical stepper-motor system diagram using the TIDM-LPSM with provided application GUI.

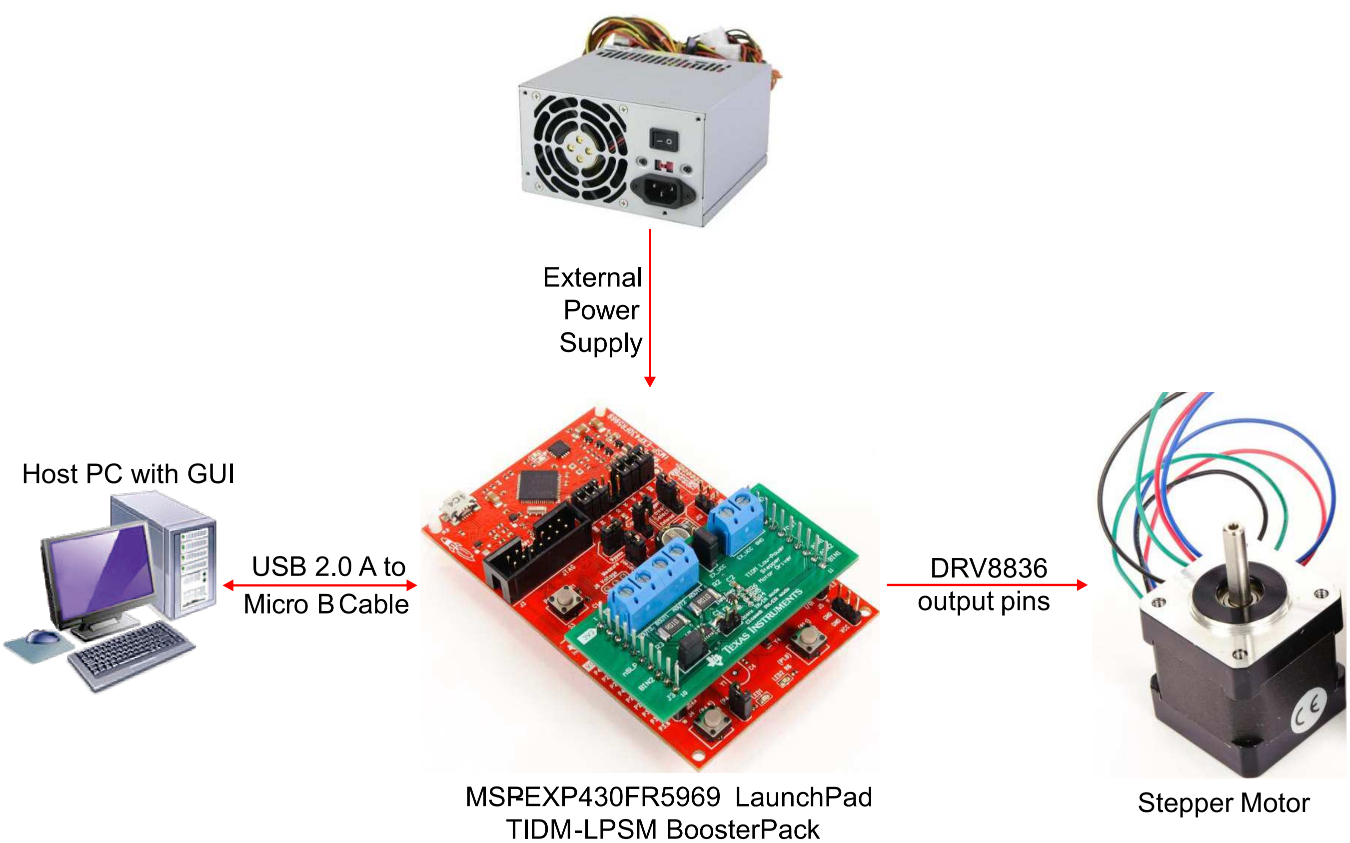


Figure 1. TIDM-LPSM System Diagram

Several current waveforms measured from an oscilloscope have been inserted to exhibit the expected operation using the contributed firmware. Power, energy, and current readings were also recorded using EnergyTrace++™ during active and low-power modes to prove the solution's benefit towards low-power battery-operated motor applications.

1.1 MSP-EXP430FR5969

The MSP-EXP430FR5969 (also known as the FR5969 LaunchPad) is an inexpensive evaluation module for evaluating and applying MSP430™ FRAM technology. Beginning development is simple with an on-board eZ-FET emulator used for programming and debugging the MSP430, providing back-channel UART via USB to PC, and communicating with EnergyTrace++. Two user buttons and LEDs are incorporated for user interaction without the need for additional hardware. This LaunchPad showcases the MSP430FR5969 16-bit MCU with 64 KB FRAM, 2 KB SRAM, up to 16-MHz CPU speed, 5 timer blocks, a 16-channel 12-bit analog-to-digital converter (ADC), and additional digital peripherals including AES256, CRC, MA, and HW MPY32. A 0.1-F SuperCap grants the option for standalone power, and rapid prototyping is accomplished with the use of 20-pin BoosterPack expansion headers along with many available plug-in modules encompassed in the BoosterPack ecosystem.

Table 1 shows the necessary pin connections required for the MSP-EXP430FR5969 to control the TIDM-LPSM BoosterPack.

Table 1. MSP-EXP430FR5969 Pin Connections to the TIDM-LPSM BoosterPack

DEVICE	DEVICE FUNCTION	FR5969 PIN USED	BOOSTERPACK PIN NUMBER	PIN FUNCTION
DRV8836	AIN1 / APHASE	1.5	13	TB0.2
	AIN2 / AENBL	1.4	12	TB0.1
	BIN1 / BPHASE	3.5	9	TB0.4
	BIN2 / BENBL	3.4	8	TB0.3
	NSLEEP	3.6	10	GPIO
INA199A1	V+	3.6	10	GPIO
	REF	3.6	10	GPIO
	OUT	2.4	6	A7

1.2 DRV8836

The DRV8836 provides an integrated motor-driver solution with two H-bridge drivers capable of driving two DC motors or one stepper motor, as well as other devices like solenoids. The output driver block for each H-bridge driver consists of N-channel power MOSFETs configured as an H-bridge to drive the motor winding. The DRV8836 can supply up to 1.5-A peak output current per H-bridge and operates on a power-supply voltage from 2 to 7 volts. Standard PHASE / ENABLE and IN / IN interfaces can be selected, and activation of a low-power sleep mode turns off all unnecessary logic to provide a very-low-current state. Internal shutdown functions are provided for over-current protection, short-circuit protection, under-voltage lockout, and overtemperature.

The TIDM-LPSM application demo firmware only uses the DRV8836's IN / IN mode for device operation; Table 2 shows the logic for this mode. Although not used here, the option for PHASE / ENABLE mode is left available through a jumper on the BoosterPack.

Table 2. DRV8836 IN / IN Mode Logic

MODE	xIN1	xIN2	xOUT1	xOUT2	FUNCTION (DC MOTOR)
0	0	0	Z	Z	Coast
0	0	1	L	H	Forward
0	1	0	H	L	Reverse
0	1	1	L	L	Brake

1.3 INA199A1

The INA199 series of voltage-output current-shunt monitors can sense drops across shunts at common-mode voltages from -0.3 to 26 volts, independent of the supply voltage. The low offset of the zero-drift architecture enables current sensing with maximum drops across the shunt as low as 10 mV full-scale. These devices operate from a single +2.7- to 26-volt power supply and draw a maximum of 100 μA of supply current.

The A1 subset of INA199 current-shunt monitors was chosen for this TI design for a gain of 50 V / V, though the A2 or A3 devices can easily be substituted for 100 V / V or 150 V / V gains, respectively.

NOTE: Modifying the gain value might require changes to the ADC segment of the MCU firmware.

2 Block Diagrams

2.1 TIDM-LPSM System Block Diagram

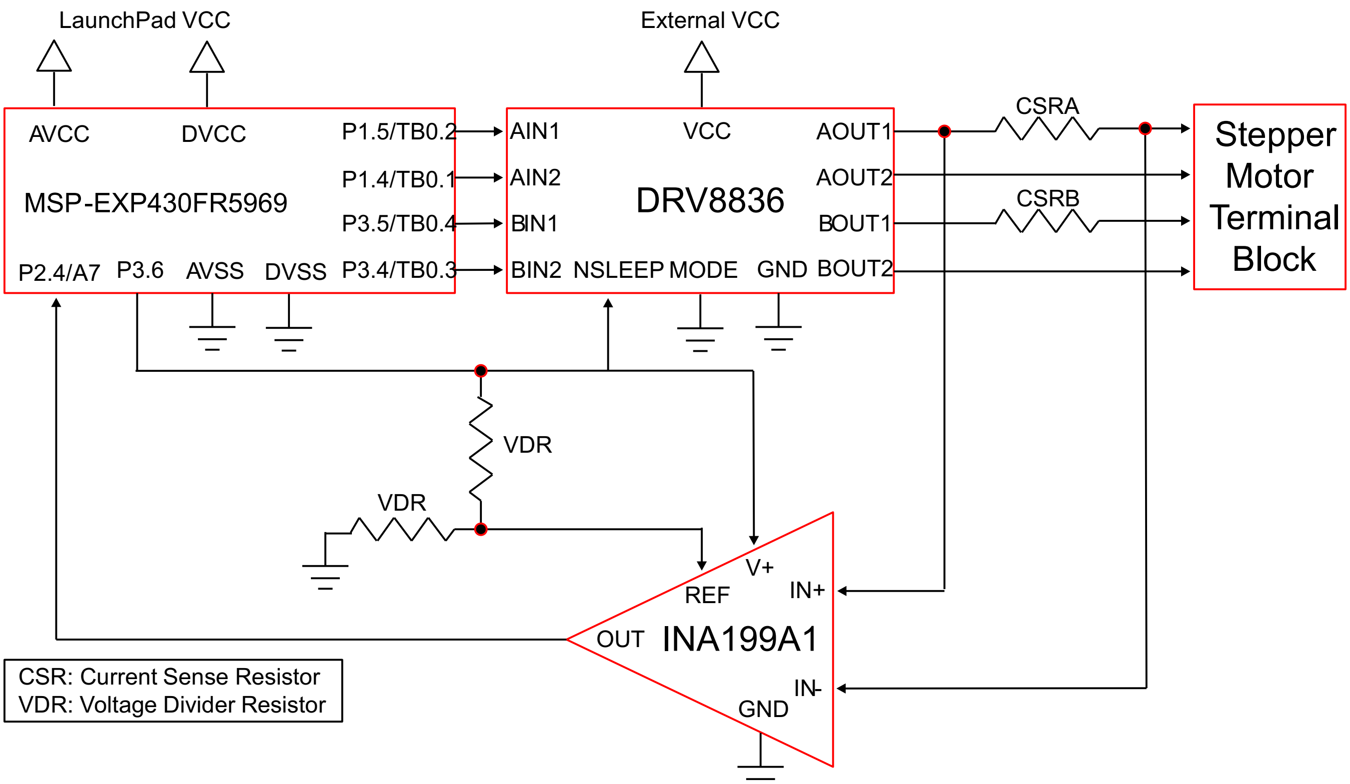


Figure 2. TIDM-LPSM Block Diagram

2.1.1 MSP-EXP430FR5969 Functional Diagrams

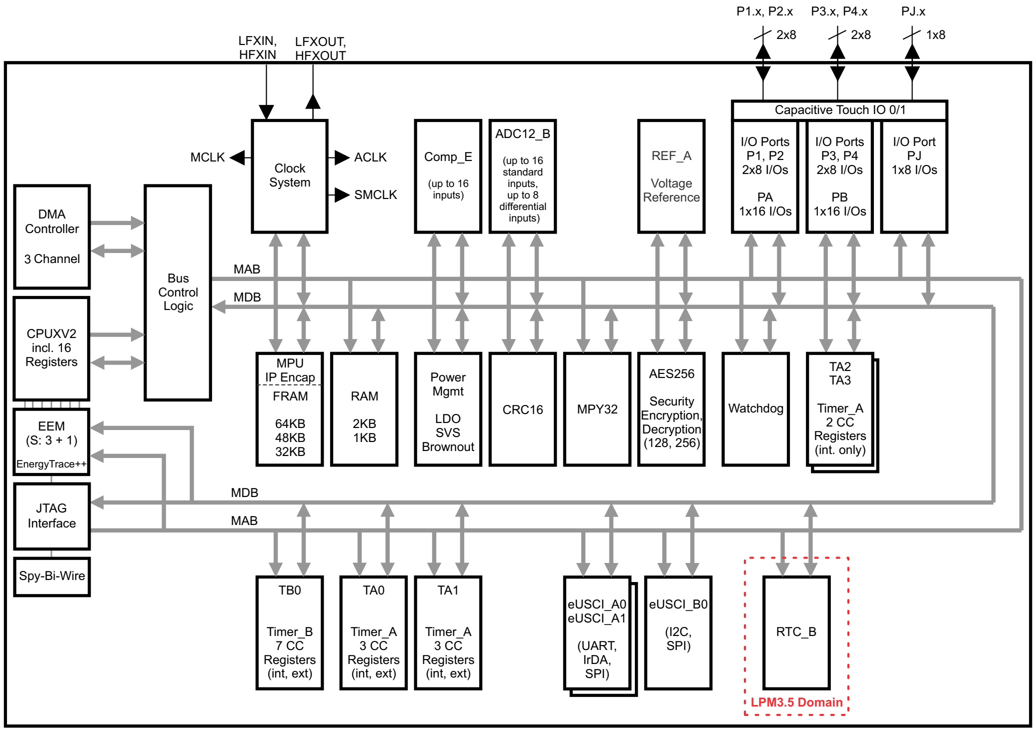


Figure 3. MSP430FR5969 Functional Diagram

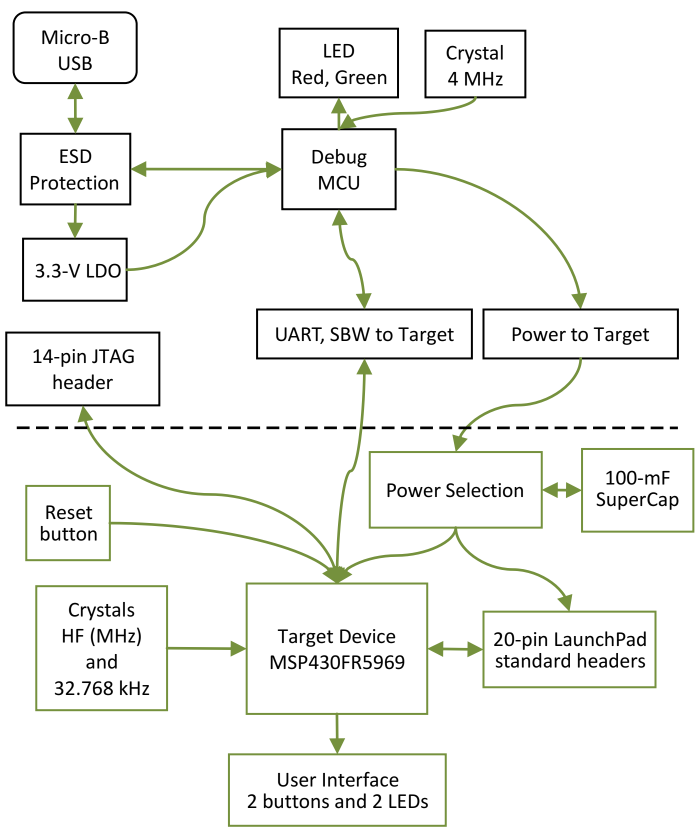


Figure 4. MSP-EXP430FR5969 Functional Diagram

2.1.2 DRV8836 Functional Diagram

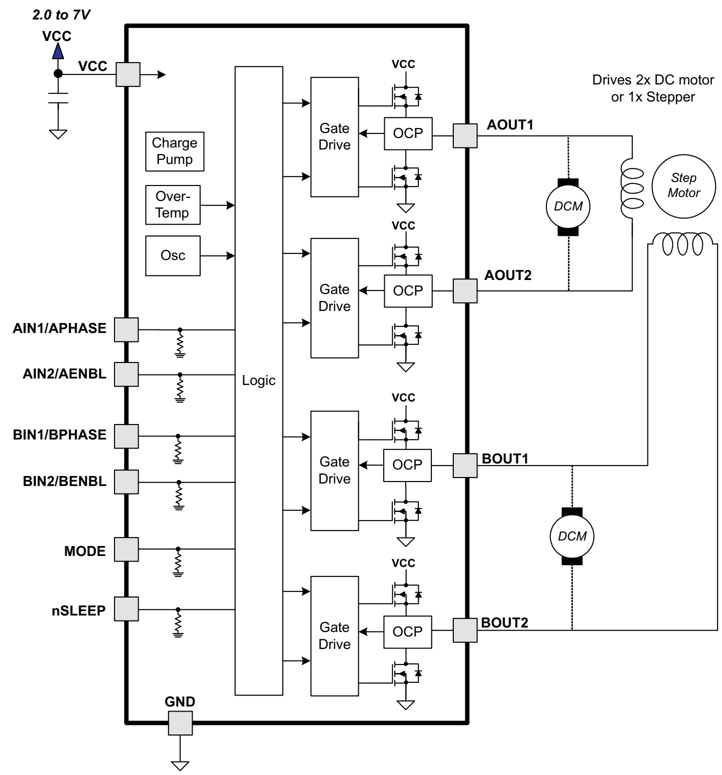


Figure 5. DRV8836 Functional Diagram

2.1.3 INA199A1 Functional Diagram

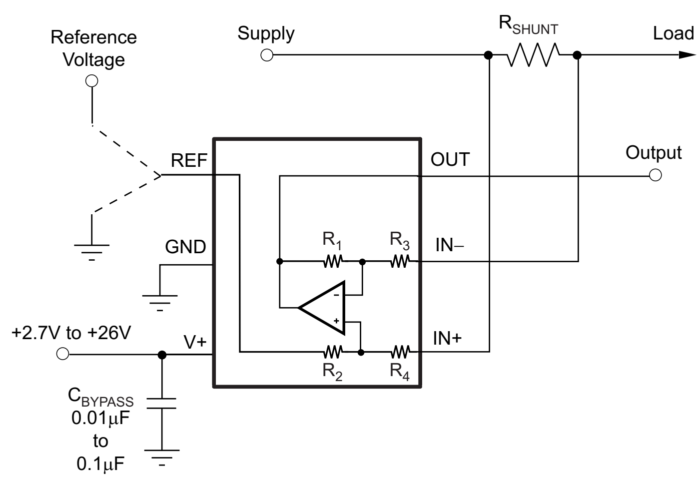


Figure 6. INA199A1 Functional Diagram

3 Hardware Considerations

3.1 BoosterPack Jumpers

A bird's-eye view of the TIDM-LPSM BoosterPack is provided in Figure 7.

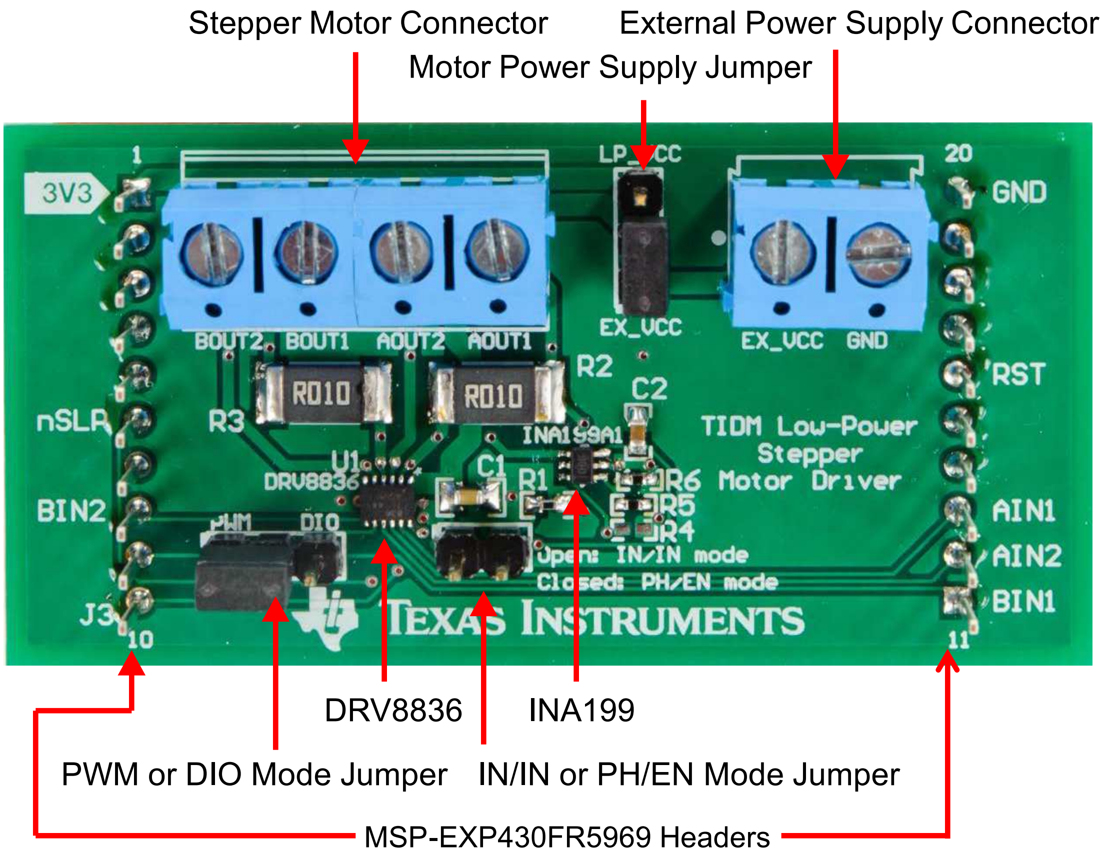


Figure 7. TIDM-LPSM BoosterPack Map

Included on this board are three jumper locations whose placement determines certain aspects of the system's operation. Their functionality is defined as such:

Motor power supply—This jumper determines if the power to the stepper motor is sourced from the LaunchPad Vcc or an external supply. In most cases the jumper should be placed in the EX_VCC position as the motor will often be operated using an external supply with a high current draw allowance. This is especially true if the LaunchPad is being powered by a small battery or through a USB connection, in which case only 500 mA of current is obtainable.

PWM or DIO mode— Using PWM (pulse width modulation) outputs through timer CCRx pins, as is done with the TIDM-LPSM application demo firmware for the MSP-EXP430FR5969, is the easiest and most efficient way to drive a stepper motor as the timer peripheral is capable of quickly transitioning pin states to create smooth waveforms. Therefore it is recommended to always have PWM functionality selected on this jumper while using the FR5969 LaunchPad. Unfortunately other 20-pin LaunchPads do not contain access to enough CCRx pins and therefore must operate this BoosterPack from DIO (digital input / output) mode. The logic to drive the pins will need to be changed accordingly and high-resolution micro stepping should not be expected if using the TIDM-LPSM application demo firmware from DIO mode.

IN / IN or PH / EN mode— Both modes are perfectly acceptable for driving stepper motors with the DRV8836 and more information regarding operation of either mode can be found in the DRV8836 device datasheet. IN / IN mode was selected for the TIDM-LPSM demo application firmware, therefore this jumper is left unpopulated to drive the MODE pin of the DRV8836 low. If using PH / EN mode (by populating the jumper headers) with the demo firmware, then the device logic will need to be adjusted appropriately.

3.2 INA199 Variance

Although originally designed with an INA199A1 current-shunt monitor, this IC can be replaced with any device in the INA199 family if a different gain is desired. V_{OUT} (voltage output from the INA199 to be read as analog input by the LaunchPad MCU) is calculated using the following equation:

$$V_{OUT} = V_{REF} + I_S \times R_{SHUNT} \times GAIN \tag{1}$$

where

- V_{REF} is the reference voltage input to the REF pin of the INA199 device
- I_S is the supply current across R_{SHUNT} towards the load
- R_{SHUNT} is the shunt resistor (represented by R2 on the TIDM-LPSM BoosterPack)
- GAIN represents the device gain dependent upon the internal resistors shown in Figure 6 and discussed in the INA199 device datasheet

Gain and internal resistor values for variances of the INA199 device are given in Table 3.

Table 3. INA199 Product Family Table

PRODUCT	GAIN	R_3 AND R_4	R_1 AND R_2
INA199A1 / B1	50	20 kΩ	1 MΩ
INA199A2 / B2	100	10 kΩ	1 MΩ
INA199A3 / B3	200	5 kΩ	1 MΩ

The value of the REF pin is determined by the output state of a LaunchPad MCU pin as well as a voltage divider created by resistors R5 and R6 on the TIDM-LPSM BoosterPack. By default, these resistor values are equivalent for a reference voltage equal to half that of the supply pin. The INA199 itself is powered by the same LaunchPad pin for energy conservation when the MCU enters a low-power state.

The R_{SHUNT} measured by the INA199 (R_2) is placed in a series with the DRV8836's AOUT1 output trace. Another shunt resistor (R_3) is populated in a series with BOUT1, but this is done solely for motor-output symmetry and in reality is not employed. If the user is changing the desired value of R_{SHUNT}, replace both R_2 and R_3 with the new resistance.

3.3 Stepper Motor Connection

All appropriate stepper-motor-driver outputs to the motor terminal block are clearly labeled on the TIDM-LPSM BoosterPack, and Figure 5 provides a visual demonstration of how these outputs should connect to the windings of a stepper motor. Cautiously inspect the winding orientation of the stepper motor chosen and ensure that the stepper motor is properly connected before trying to operate. If the motor is found driving in the direction opposite than intended, users can fix this either through firmware altercation or by switching the output connections respectively (AOUT1 ← → BOUT1 and AOUT2 ← → BOUT2).

4 Firmware Considerations

4.1 Code Description

Included with this TI Design is a software package that contains a Code Composer Studio (CCS) project designed for the MSP430FR5969 and an accompanying GUI. For proper evaluation of the TIDM-LPSM, the CCS project should be imported using CCS V6.0.1 or later and uploaded onto a MSP-EXP430FR5969 device. Although these code examples reflect operation for a specific application, the TIDM-LPSM is not bound by any range of operation and can be utilized wherever stepper or single- or dual-brushed DC motors are desired. The block diagram of this firmware is provided in Figure 8.

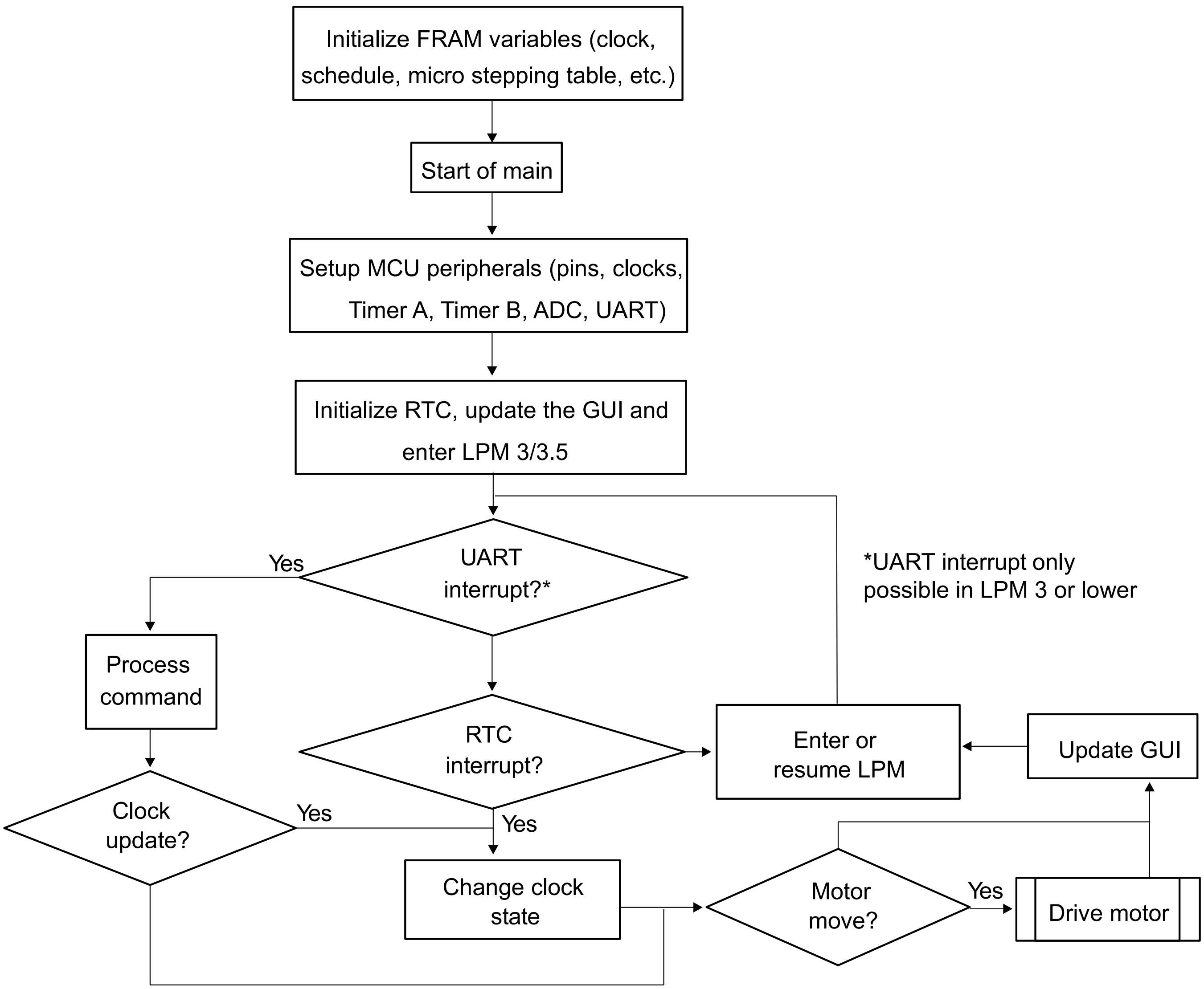


Figure 8. Firmware Block Diagram

During launch, all variables including the clock, clock schedule, necessary micro-stepping table, and position flag are initialized inside of the FRAM. At the start of main, the MCU peripherals (specifically pin states, clock assignments, and speeds), timers, analog-to-digital converter (ADC), and universal asynchronous receiver / transmitter (UART) are then established followed by real-time clock (RTC) setup dependent upon the reset state of the MSP430FR5969. If this is the first time the program has been run since the code was uploaded, an introductory move can be performed. A while loop is then entered permanently in which a low-power mode (LPM) is entered, then a series of actions is performed upon wakeup, which is designed to occur every minute based off of the RTC or from a UART interrupt if the device is in LPM3 or lower. Only LPM3 and 3.5 will be discussed because they offer maximum energy savings for this specific design.

4.2 Altering the Initial Clock Schedule

Included amongst FRAM variables is a 1440-length array of values (type unsigned long) that represents the 1440 minutes in a 24-hour period starting at midnight. Each time a minute is counted by the RTC, it forces a wakeup from LPM, and the code will refer to the value in this array that corresponds to the updated clock to perform a motor movement command if necessary. This array must be initialized in accordance with the end application desired, especially if operating in LPM3.5 because the GUI will be unable to use UART interrupts for programming new commands.

The values themselves represent the number of steps a motor is to take in a given direction. The actual number of degrees translated depends on the exact motor's steps or revolution specification. Direction of movement is reliant on orientation of connection to the DRV8836 outputs. Movement in one direction is simply stated by setting the variable to a two-byte value, giving a range of 216 steps for a single move. To move in the opposite direction, the two least significant bits are filled with 0xFFFF, whereas the two most significant bits represent the amount of steps to move, once again producing a maximum of 65,535 steps possible. If no movement is desired, use a value of zero. Examples of each case are given in the provided CCS project. By utilizing this nomenclature, users can create a customized schedule of motor movements that will repeat itself every day.

4.3 Operating from LPM3 vs LPM3.5

Selecting the correct low-power mode is a matter of choosing optimal power savings versus expanded GUI control. LPM3.5 operates at less than half the current consumption when compared to LPM3, as is evident from Section 6, but the GUI is limited to only providing updates of motor status upon RTC wakeup every minute. Conversely, LPM3 grants UART interrupt capabilities, hence allowing for immediate interaction with the stepper motor, whether that interaction is performing a command immediately or programming new commands into the clock schedule. Deciding between the two is fully dependent on the desired goals of the end application.

Switching between either mode inside of the CCS project is straightforward. LPM3.5 is achieved through the compute through power loss (CTPL, included in the MSP-FRAM-UTILITIES software package) library for ease of use. Located inside of the TIDM-LPSM-CTPL.c file at the beginning of the forever while loop are some lines of code that determine the operating LPM. By commenting in or out specific lines of this section of code, users can shift from one mode to the other. Figure 9 presents further explanation of this process.

```
    while(1)
    {
        // Go into sleep and wait for Clock interrupt
        // RTC_B makes an interrupt every 60s to wake up MCU from sleep
        __bis_SR_register(LPM3_bits + GIE);
//      ctpl_enterLpm35(CTPL_DISABLE_RESTORE_ON_RESET);
//      //Re-initialize system after LPM3.5
//        SystemInit();
```

Used for LPM3 operation

Used for LPM3.5 operation

Figure 9. Code for LPM Operation

4.4 Defining Motor Constants and Variables

Contained in the Motor Driver.c file are several constants that alter the way in which the stepper motor operates. Their names and purposes are as follows:

microstep— Determines the resolution of each step, a higher micro stepping value produces more smooth waveforms and better motor operation. Section 6 of this design guide gives waveform examples detailing the different operation each value provides. Every value corresponds to a required micro stepping table and at the time of this document's creation tables have been developed for microstep values of 1 (full step), 2 (half step), 4 (quarter step), 8, and 16.

uStepDelay— Represents the approximate amount of time required for a full step of the motor, used to adjust the motor's speed.

voltage— Corresponds to the voltage level being supplied to the MSP430FR5969 I/O pins, used in ADC calculations based off of the INA199 results to monitor motor behavior.

offset— If the motor is supposed to be moving but the ADC result from the INA199 is still within the allowable offset then something is amiss (motor is disconnected, power to the motor is not connected, etc.). A lower value produces a more loose tolerance level and vice versa.

4.5 Optional Settings

Two additional constants that are established in Motor Driver.c exploit the advantages of the TIDM-LPSM's current sensing capabilities. The useMonitor constant allows for the INA199 result to be read and interpreted by the MSP430FR5969 and can be used to cancel motor operation if the current readings are abnormal. The next constant, displaycurrent, is an extension of useMonitor and displays the maximum current reading, having been recorded from the last motor move, inside of the GUI text log. Either constant is deactivated with a value of zero but are otherwise active. Application examples of both settings are given in Section 5.

5 GUI Operation

5.1 Interface Overview

Before using the GUI, users must first download and install the RXTX java library necessary to perform serial communication through use of the COM port (copies of the 32-bit version of RXTXcomm.jar and rxtxSerial.dll have also been included with the project files). Download instructions can be found on this webpage and should be followed carefully. Once finished, the GUI is started by opening the TIDM-LPSM.jar file located in the dist folder of the GUI project folder.

As explained previously, the TIDM-LPSM GUI is packaged to operate alongside the example application firmware provided for the MSP430FR5969. By using the eZ-FET module included on all LaunchPads, users can utilize backchannel UART through a micro-USB connection for two-way communication between the MSP-EXP430FR5969 and a PC. The basic structure and design of the GUI is detailed in Figure 10.

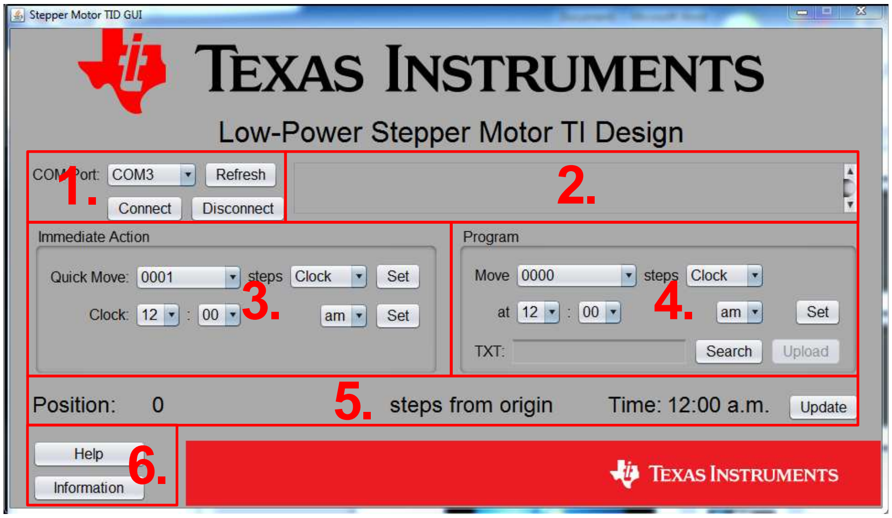


Figure 10. GUI Design Overview

An explanation of each numbered section of the GUI is provided as follows:

1. COM Port control, used to select the MSP Application UART COM port for which the MSP-EXP430FR5969 is connected.
2. Text log, provides the user with feedback such as COM port status, motor behavior, and schedule upload information.
3. Immediate Action box, for instantaneous resetting of the clock value or motor position.
4. Program box, for re-configuring the clock schedule.
5. Status line, shows the current motor position and clock time.
6. Help and Information, displays further GUI tips and guidelines.

5.2 *Connecting to a COM port*

For the GUI to properly communicate with the MSP430FR5969 device, the GUI must first know on which COM port the backchannel UART is enumerating. To find the correct COM port, first ensure that the MSP-EXP430FR5969 is the only FET-capable tool connected to the computer, and then view the available ports using the computer’s Device Manager. The desired COM port is labeled MSP Application UART1 as shown in Figure 11.

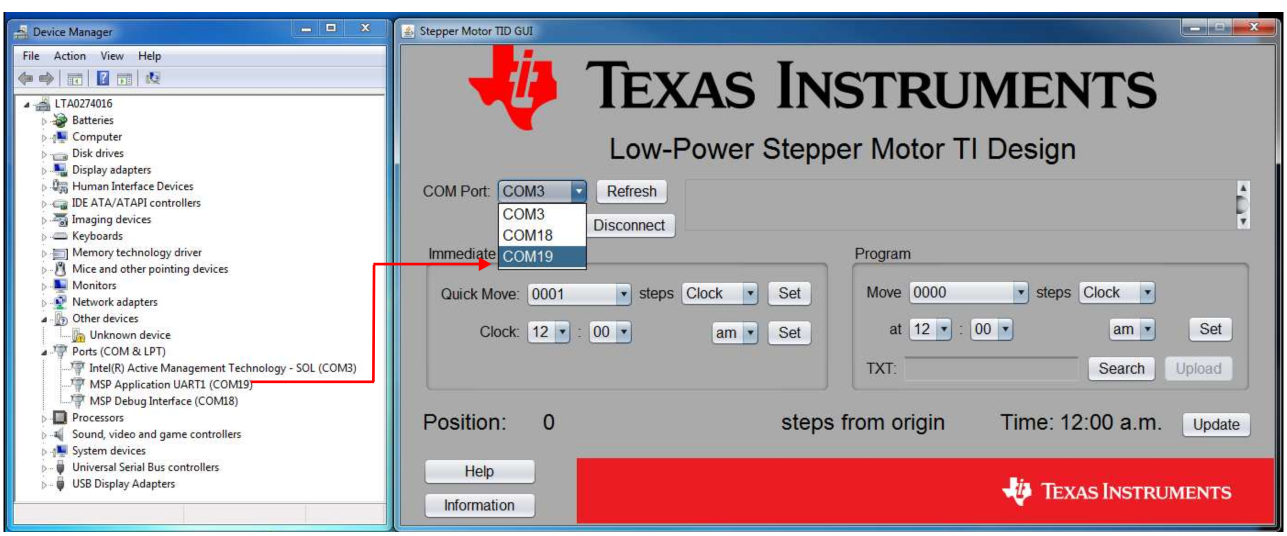


Figure 11. GUI COM Port Settings

After selecting the proper COM port, press the Connect button. At this point, the text log should update to say COM[x] opened successfully. If the text log says otherwise, make sure that the correct port has been selected and that the LaunchPad is not in use by any other computer program. Disconnect and refresh buttons are also provided for full control and interaction with the COM ports available, which is useful for locating the correct COM port if the LaunchPad was connected after opening the GUI.

5.3 Commands and Feedback

5.3.1 Immediate Action

The Immediate Action box is beneficial for instances where it is desirable to perform a move that exists outside of the movement schedule or if the clock needs to be set to a different value. Simply use the drop-down boxes to select the values needed and click the Set button when finished to have the commands processed and executed by the MSP430FR5969. Afterwards, the status line will be automatically updated.

5.3.2 Program

The layout of the Program box is very similar to that of the Immediate Action box, except that it adds or alters entries to the motor movement schedule. An extra feature that the Program box offers is the ability to upload a .txt file capable of updating the schedule with multiple commands at once. This action amends the designated motor movement of clock values defined in the .txt file but will not erase other pre-existing settings.

The .txt file is generated by following a strict set of rules. Instructions in the .txt file must be 10 characters long per line, and each character must be within its specific parameters or the line will be ignored. The first character is ‘f’ or ‘b’ to depict the direction of motor movement. The next four characters represent the number of motor steps in decimal form to be taken and are followed by the character ‘@’. Afterwards come four characters to define the clock time in minutes from midnight. Up to 1440 unique commands can be uploaded in this manner, one per minute in a 24-hour period. An example file TIDM-LPSMUploadExample.txt, in which four commands are written using the above instructions, has been provided with the GUI source files.

Once a .txt file has been created and saved on the computer, the file must then be selected by using the Search button. The file’s location will be displayed in the GUI’s TXT dialog box, in which case the file is now ready for sending to the MSP430FR5969 by clicking on the Upload button. The GUI’s text log will show whether the file was successfully uploaded. Figure 12 shows the aforementioned TIDM-LPSMUploadExample.txt program file being sent to the LaunchPad. Text file TIDM-LPSMClearMemory.txt has also been provided for completely resetting the schedule to all zero values.

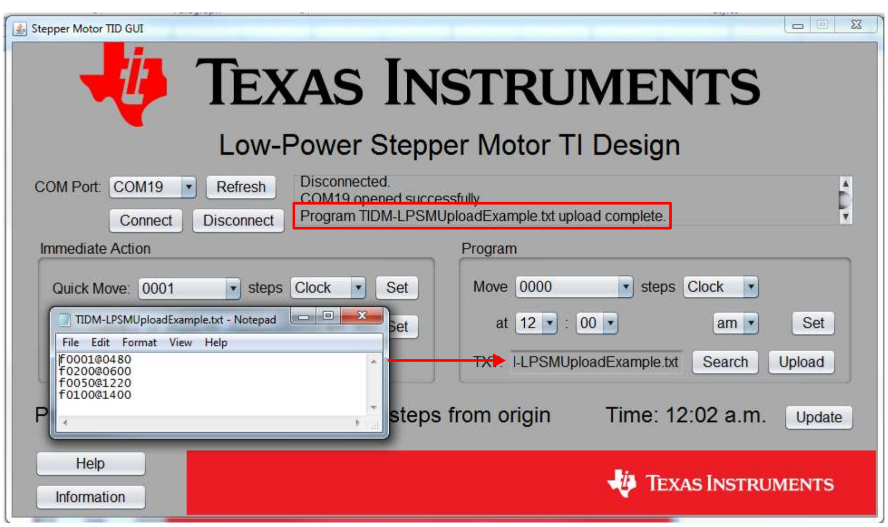


Figure 12. Uploading a Program File to the MSP-EXP430FR5969

5.4 Text Log

As shown in previous sections of this design guide, the text log is used to show helpful feedback to the user with information regarding COM port and program file upload status. If the useMonitor constant is defined inside of the CCS project, then the text log will alert the user if a move was not performed due to a system fault, most likely due to motor- or power-supply disconnection or if the offset variable was not set correctly. Likewise, defining the displaycurrent constant will show the maximum current drawn during the last move. Because the current of the motor is measured after each micro step has been completed, small movements such as those that produce less than a full wavelength will result in a smaller current draw reading than expected. The best outcomes are created from longer active-motor operation where several waveform peaks can be sampled. Figure 13 shows these optional comments displayed in the GUI's text log.

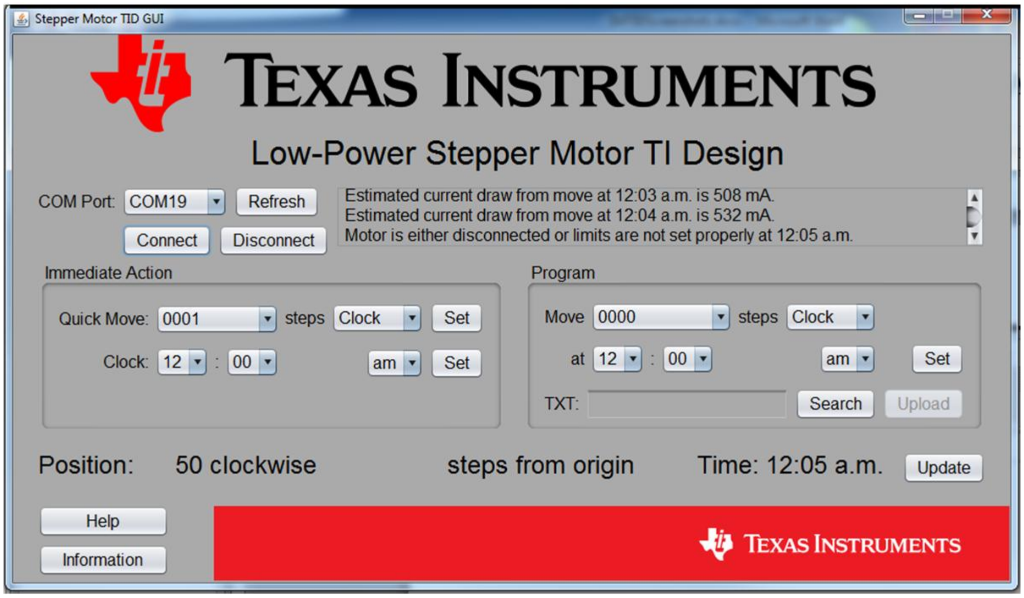


Figure 13. Additional Text Log Feedback

6 Test Data

6.1 Current Waveforms

Current measurement screenshots were taken using current probes connected to an oscilloscope. The probes themselves measured the current traveling through the phase wires of a NEMA 14-size hybrid-bipolar-stepper motor sold by Pololu Robotics & Electronics. This step motor has a 1.8° step angle (200 steps / revolution), and each phase draws 1 A at 2.7 V for an allowable holding torque of 1.4 kg-cm (20 oz-in).

Figure 14 illustrates a series of current waveform screenshots, one for each micro step value currently supported by the TIDM-LPSM firmware. This value portrays the number of steps taken to complete one fourth of a waveform period. As can be expected, the resolution of the waveform increases as the number of micro steps taken increases, therefore producing more precise and smooth steps by the motor. Peak-to-peak current measurements are left unaffected for similar frequencies.

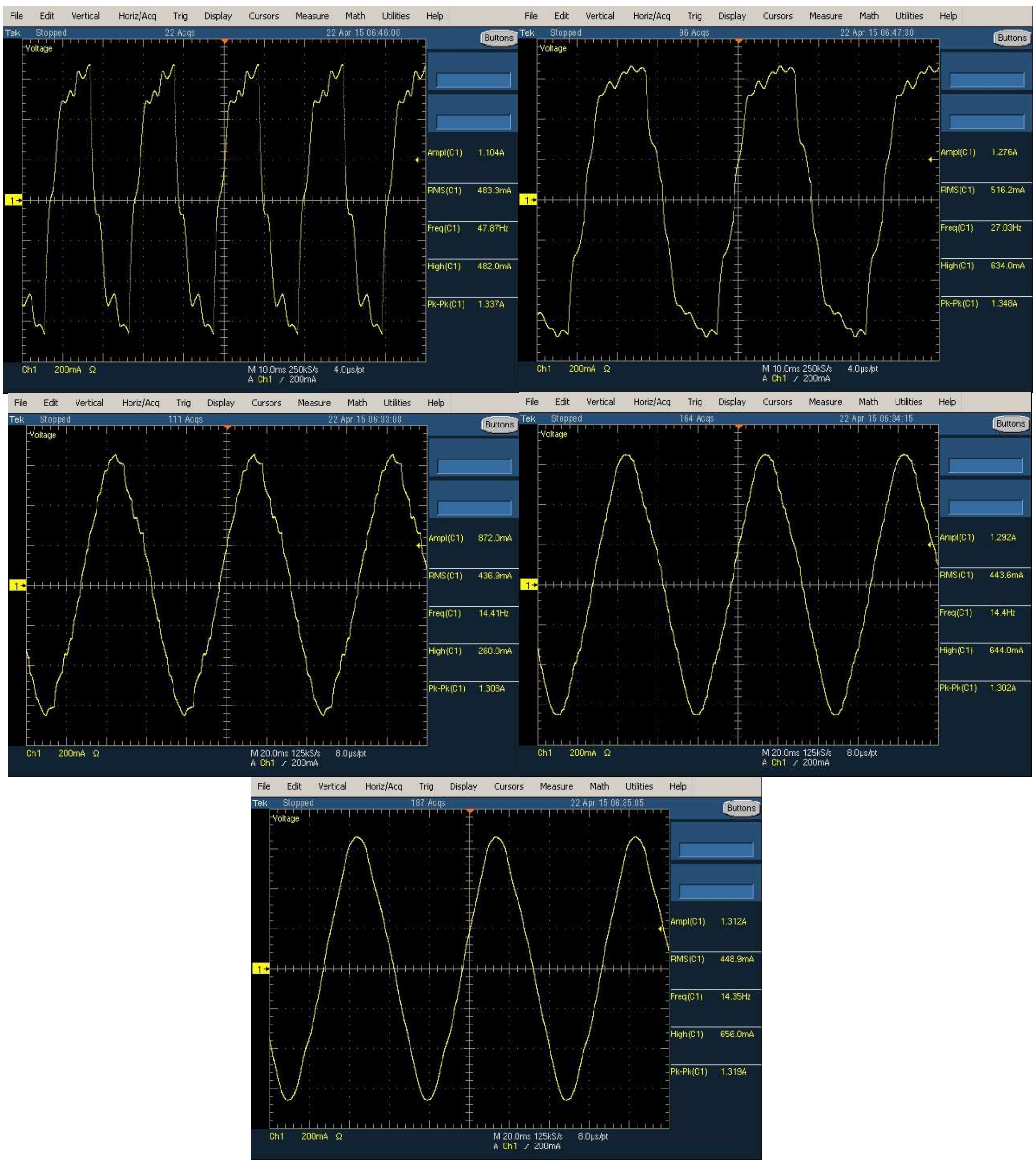


Figure 14. Current Measurement Micro Step Values (From Top Left to Bottom): 1, 2, 4, 8, 16

The delay between micro steps also factors greatly towards the stepper motor's performance. Decreasing this step delay will produce faster motor revolutions but can affect both the resolution and amplitude of the current waveforms. As seen in Figure 15, in which a micro-step value of 16 is used, too short of a delay results in decreased current draw and in extreme cases causes sporadic waveforms barely capable, if at all, of driving the stepper motor. The last screenshot in Figure 14 can be referenced for comparing a delay value of 16.

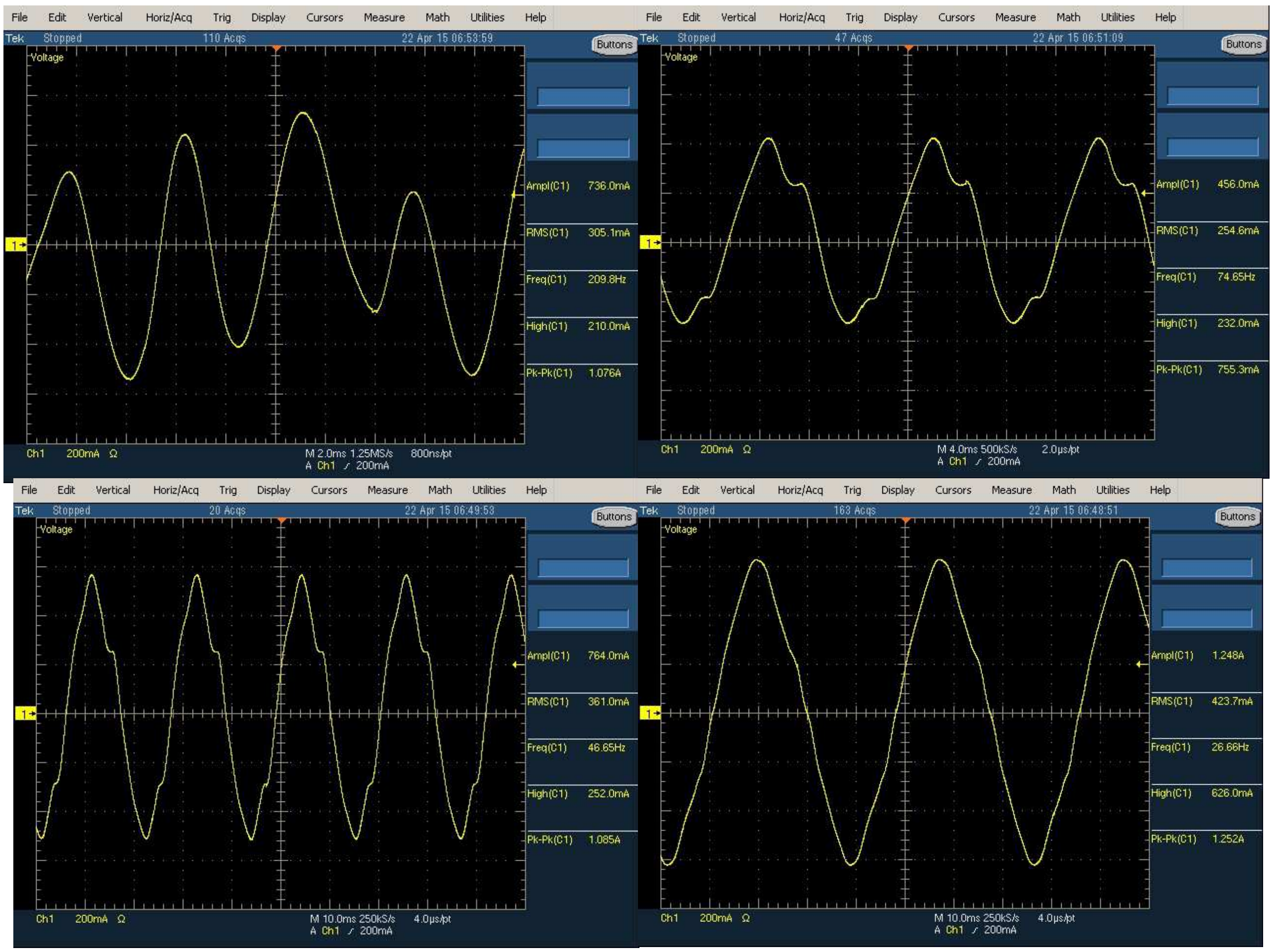


Figure 15. Current Measurement Delay Values (in Milliseconds, From Top Left to Bottom): 1, 2, 4, 8

Figure 16 analyzes a case where two current probes, one connected to AOUT1 and the other to BOUT1, were used at the same time to demonstrate the 90° phase shift expected for proper stepper motor operation. If the motor were to be run in the direction opposite shown, the following waveform would then be in the lead by 90°.

Figure 16. 90-Degree Phase Shift in Current Waveforms between Bridges A and B

6.2 EnergyTrace++ Recordings

Power consumption statistics for device startup, active mode, LPM3, and LPM3.5 were documented using EnergyTrace++ technology packaged with the CCS IDE. Figure 17 and Figure 18 show the power (in mW) and energy (in µJ) of the startup condition for a time span of ten seconds. Figure 19 gives EnergyTrace++ profiles for each device-mode condition and provides clear proof as to the current consumption advantages of LPM3.5 over LPM3. Active mode will operate for less than 5% of the total time (3 seconds per minute), which is when the motors are being driven.

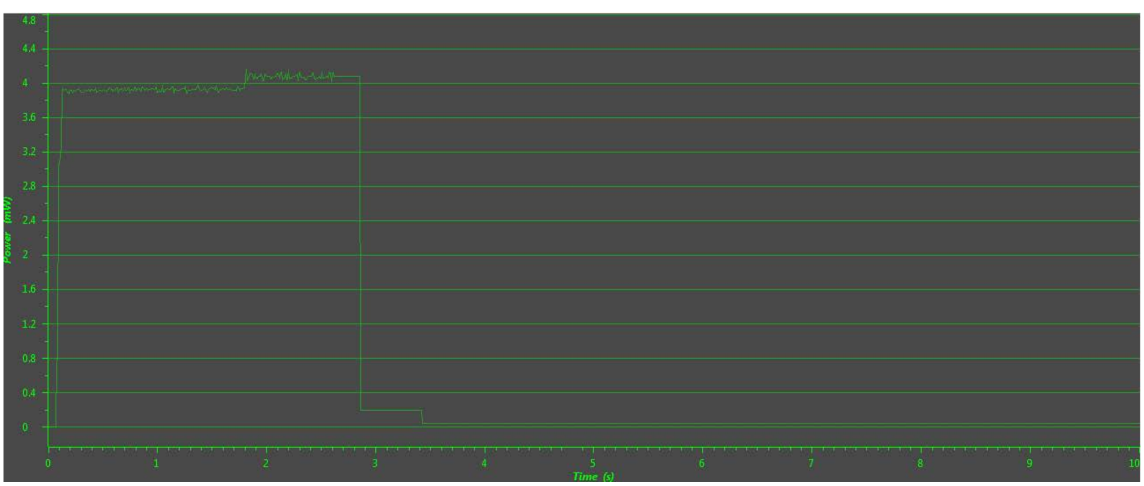


Figure 17. Power (in Milliwatts) of Device Startup

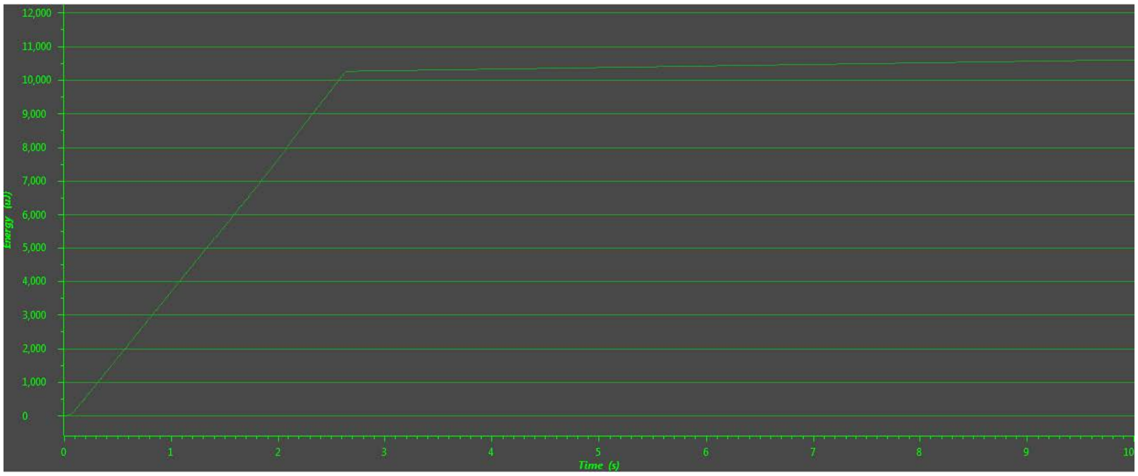


Figure 18. Energy (in Microjoules) of Device Startup

EnergyTrace™ Profile

Name	Live
System	
Time	10 sec
Energy	0.770 mJ
Power	
Mean	0.0735 mW
Min	0.0000 mW
Max	0.0804 mW
Voltage	
Mean	3.5870 V
Current	
Mean	0.0205 mA
Min	0.0000 mA
Max	0.0224 mA
Battery Life	CR2032: 357.3 day (est.)

EnergyTrace™ Profile

Name	Live
System	
Time	10 sec
Energy	0.371 mJ
Power	
Mean	0.0350 mW
Min	0.0000 mW
Max	0.0413 mW
Voltage	
Mean	3.5870 V
Current	
Mean	0.0097 mA
Min	0.0000 mA
Max	0.0115 mA
Battery Life	CR2032: 741.8 day (est.)

EnergyTrace™ Profile

Name	Live
System	
Time	3 sec
Energy	14.452 mJ
Power	
Mean	4.4705 mW
Min	4.4079 mW
Max	4.7086 mW
Voltage	
Mean	3.5890 V
Current	
Mean	1.2456 mA
Min	1.2284 mA
Max	1.2925 mA
Battery Life	CR2032: 5.7 day (est.)

Figure 19. EnergyTrace++ Profile of Active Mode (Left), LPM3 (Middle), and LPM3.5 (Right)

7 Design Files

7.1 Schematics

To download the schematics for each board, see the design files at TIDM-LPSM.

7.2 Bill of Materials

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

7.3 PCB Layout Recommendations

Due to high current consumption sometimes required from stepper motors, the respective trace widths from the output of the DRV8836 need to be adequately sized. PCB trace width calculators can be used to estimate the required trace width given current consumption and copper thickness. Utilizing traces on both the top and bottom of the board with several vias in between to patch the two sides together can effectively aid in increasing current allowance. If using a multilayer board, keep in mind that internal layers require wider traces to achieve the same effect as external layers. It is also recommended that the motor terminal blocks or connectors be placed relatively close to the DRV8836 device so that the trace distances are minimized.

The DRV8836 includes a PowerPAD™ that acts as a thermal heat sink for ground. It is highly recommended that this pad be connected to a ground plane through the use of multiple vias to correctly dissipate heat from the stepper-motor driver. Application brief SLMA004B provides more information on how to suitably connect the PowerPAD to a ground plane.

7.4 Layer Plots

To download the layer plots, see the design files at TIDM-LPSM.

7.5 Eagle Project

To download the Eagle Project files for each board, see the design files at TIDM-LPSM.

7.6 Gerber Files

To download the Gerber files, see the design files at TIDM-LPSM.

7.7 Software Files

To download the software files, see the design files at TIDM-LPSM.

8 References

1. RXTX website (http://rxtx.qbang.org)
2. Pololu Robotics & Electronics website (http://www.pololu.com)
3. Circuit Calculator website (http://circuitcalculator.com)
4. TI Application Brief, *PowerPAD Made Easy* (SLMA004B)

9 About the Author

RYAN M. BROWN is an MSP430 Customer Applications Engineer at Texas Instruments, where he is responsible for supporting development of customer designs in various applications. Ryan brings to this role his experience in PCB design software, MSP430 architecture, and Launchpad and BoosterPack protocol. Ryan earned his Master of Science in Electrical Engineering (MSEE) from Texas Tech University in Lubbock, TX.