

Optimized H-Bridge Driver Control for Stepper and Brushed-DC (BDC) Motors Using MSPM0 MCUs

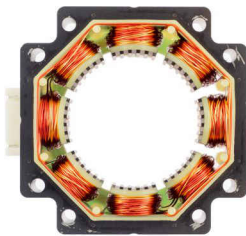


In the modern world, motors are used in many products. Stepper motors are widely used in slide rails, workbenches, medical machines, and many other applications because of small size, low cost, position control, ease of use, and no cumulative error. Brushed DC (BDC) motors are also widely used in lathes, starters, and electric locomotives because of good starting and speed performance. For these two kinds of motors, the H-bridge structure is used in hardware to control the speed and direction of the motor. MSPM0 MCUs can fill this role with its low price, broad portfolio, abundant analog features, and software resources optimized for brushed-DC and stepper control.

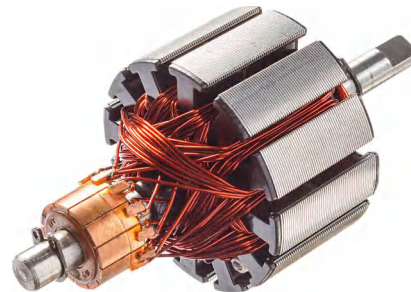
What is an H-bridge driver for stepper and brushed-DC motors?

A brushed-DC motor (BDC) is the simplest kind of motor to actuate. To operate a BDC motor, a voltage is applied across the motor terminals to change the magnetic field on the rotor and create a continuous spinning motion. Despite having drawbacks such as heat dissipation, high rotor inertia, and electromagnetic interference, Brushed DC motors do not require current feedback and are easy to control, making them a simple low-cost solution for many applications.

Stepper motors function using a series of electromagnets, alternating which are turned on/off to cause rotation of a central rotor and turn the motor shaft. The stepper motor can provide continuous motion, or maintain a fixed rotor position depending on the system requirements. To control the torque or audible noise of the stepper motor, "stepping" patterns are implemented based on the complexity of the stepping algorithm.



Stepper Motor



BDC Motor

H-bridge drive circuits are widely used to control of stepper and BDC motors. An H-bridge is built with four switches (solid-state or mechanical). The rotation, stop and reverse of the motor can be realized by controlling the on or off of those tubes. H-bridges are available as [integrated circuits](#), or can be built from [discrete components](#).

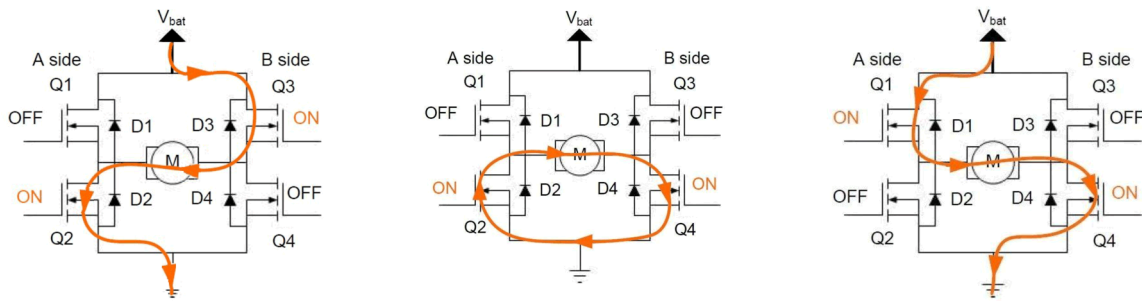


Figure 1. H-Bridge Driver (Left: Clockwise, Center: Stop, Right: Counter-Clockwise)

Across those stepper and BDC related applications, users need to accurately control the motor speed, torque or other variables to meet the requirements of actual applications. Therefore, the main resources used are as follows:

Hardware

- **Microcontroller (MCU):** As the motor driver controller, it controls analog data acquisition, runs control algorithms, monitors the motor status, and communicates with other products.
- **Pre-driver:** Pre-driver for power stage
- **Power stage:** Up to 8 MOSFETs (or IGBTs) in dual H-bridge configuration
- **Analog front end:** Collect motor voltage, current, speed, and more

Software

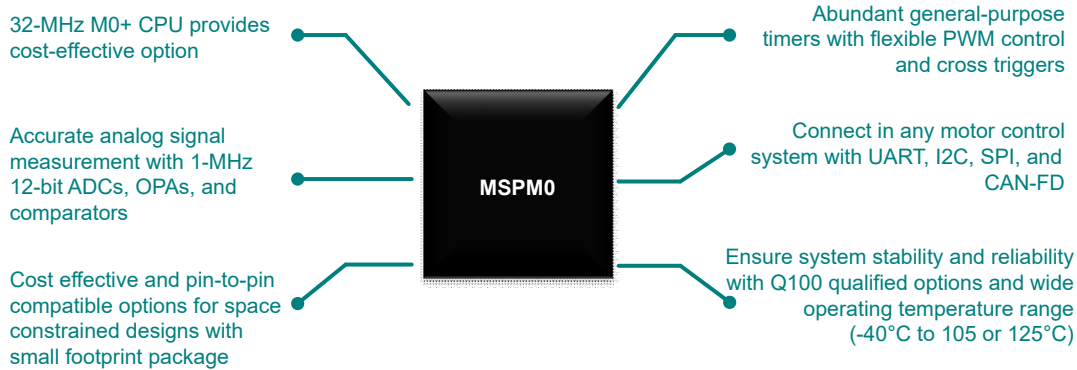
- **Application:** Task management and scheduling
- **Algorithm:** Calculate the drive signal output of the motor in real time according to the feedback of the motor and the user input signal

You can realize all these functions using TI devices, pre-drivers, and MOSFETs.

Motor	MCU	Motor Voltage	Pre-driver (Device)	Power Stage (Switch Type)
Brushed-DC	MSPM0Lxx ARM Cortex M0+ 32 MHz MCUs	1.65 V to 50 V	H-bridge Motor Driver (DRV82xx series)	MOSFET (CSD series)
		5.5 V to 60 V	H-bridge Gate Driver (DRV870x series)	
		5 V to 115 V	H-bridge Gate Driver (DRV877x series)	
		4.5 V to 70 V	Quad Half-bridge driver (DRV89xx series)	MOSFET, IGBT
		5 V to 115 V	Half bridge drivers (UCC2xxx series, LMxxx series)	
Stepper	MSPM0Lxx ARM Cortex M0+ 32 MHz MCUs	1.65 V to 70 V	Dual H-Bridge Stepper Driver (DRV84xx series)	MOSFET, IGBT
		2 V to 65 V	Dual H-Bridge Intelligent Stepper Driver (DRV88xx series)	

Why is MSPM0 suitable for stepper and brushed-DC control?

TI's scalable M0+ MSPM0Lx mainstream MCUs with on-chip motor control peripherals provides a great low-cost solution for a variety of motor control applications. With up to 32MHz CPU speed and a portfolio from 8KB to 64KB of flash memory with scalable analog integration and motor control peripherals, MSPM0Lx is an option for stepper and brushed DC motor solutions.



- **32-MHz M0+ CPU** – Reduce process time for control and sensing signal
- **1-Msps 12-bit ADC module (up to 10 channels)** – Sensing H-Bridge current
- **Two zero-drift chopper op-amps** – Accurately amplify the 2 H-Bridge current
- **High-speed comparators** – Enable fast current protection for motor
- **Four general-purpose timers** – Flexible PWM control and cross triggers
 - Synchronization and cross trigger to generate multiphase motor control PWMs
 - Up to 8 PWM (can drive 4 BDC or 2 stepper motors)
- **Robust IO design with glitch filter** – Provide a reliable system with motor noise
- **Comprehensive communication interfaces** – Including UART, I2C, SMBus, SPI, meet all the communication requirements with the motor control system.
- Scalable MCU portfolio with pin-to-pin compatible devices cross wide range of flash memory options.
- Low cost and small footprint packages provide cost effective options for space constrained designs.

What can MSPM0 do in brushed-DC motor control?

In brushed-DC applications, MSPM0 can monitor the motor status, run algorithms, and generate PWM to drive the motor (through a pre-driver IC). With the help of scalable analog Integration, the MCU can calculate accurate values of bus voltage, motor current, and speed quickly, then provide some input to control algorithm. The MSPM0L13xx can generate 8 PWMs and can therefore drive 4 BDC motors at the same time.

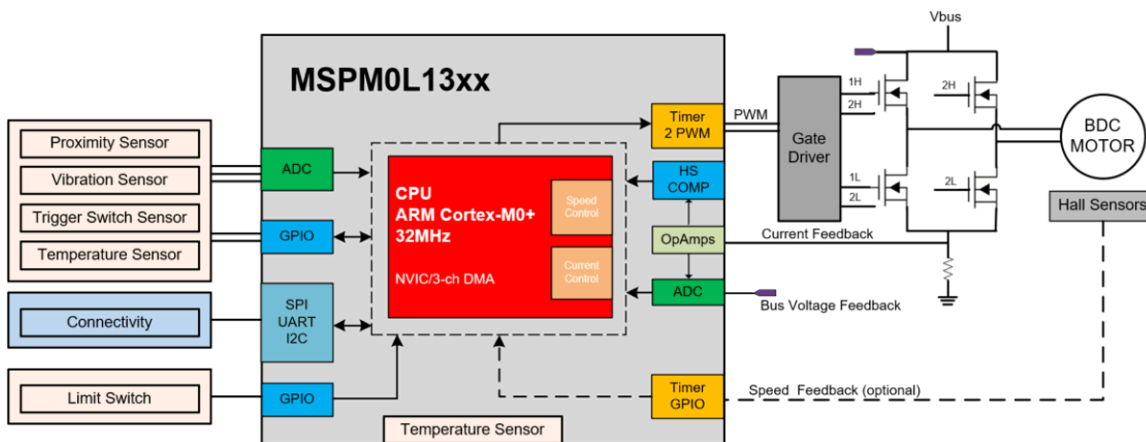


Figure 2. MSPM0L13xx Motor Control Block Diagram for Brushed-DC Control

What can MSPM0 do in stepper motor control?

In stepper applications, MSPM0 can monitor the motor status (optional), run algorithms, communication with a gate driver (set subdivision parameters) (optional), and generate PWM to drive the motor (through a pre-driver IC). With the help of scalable analog integration, the MCU can calculate accurate values for the bus voltage, motor current, and speed quickly (optional), then provide inputs to the control algorithm. The MSPM0L13xx can generate up to 8 PWM signals, and can therefore drive two stepper motors at the same time.

MSPM0L1xxx Stepper Control Using Stepper Driver With PWM Interface

Basic stepper drivers often use a PWM interface for stepper control, where specific PWM patterns can provide torque control while controlling the stepper motor's position. To accomplish this, the MSPM0 provides 4 PWM input signals using a full-step or half-step commutation pattern to control the respective currents through the stepper motor's phases. Additionally, many stepper drivers include current regulation from an analog input signal, which can be provided using the MSPM0's 8-bit DAC output from an integrated comparator to smoothen the current profile. This topology is suitable high torque or low-precision stepper applications such as toys, smart locks, robotics, and security cameras.

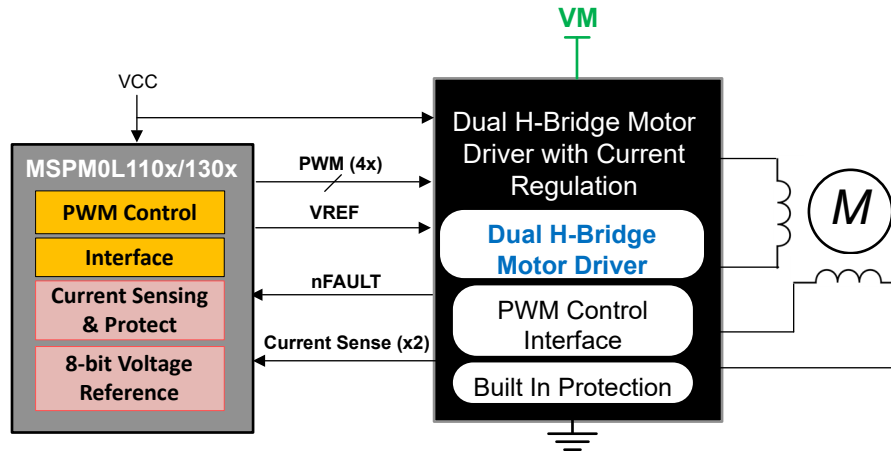


Figure 3. MSPM0L1xxx Stepper Control Using Stepper Driver With PWM Interface

MSPM0L1xxx Stepper Control Using Intelligent Stepper Driver With STEP Interface

In higher performance stepper applications that require low noise, precise control, or stall detection, it is better to use an intelligent stepper driver capable of up to 1/256 microstepping. These drivers include indexers and advanced stepper control and protection algorithms and typically only require one PWM signal advance the stepper motor. Additionally, these drivers can accept an analog input for current regulation and often use a SPI interface to configure the control algorithm, device settings, and diagnose system-level faults. The MSPM0 can interface with intelligent stepper drivers by providing the PWM signals, an 8-bit DAC output voltage for the current regulation, and communicate to the driver using SPI interface. This topology is suitable in low-torque and low-noise stepper applications such as printers, ATM machines, stage lighting, office and home automation, medical applications, and 3D printers.

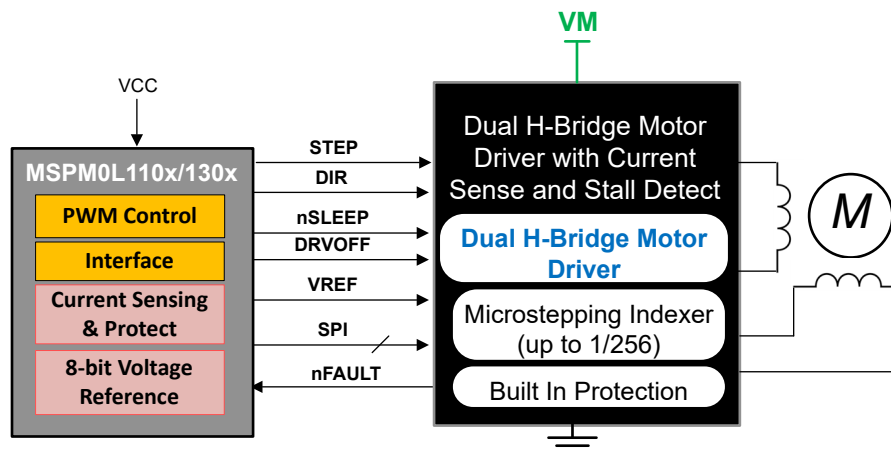


Figure 4. MSPM0L1xxx Stepper Control Using Stepper Driver With PWM Interface

Solution Details

- Code examples
 - [MSPM0 Brushed-DC Library](#)
 - Supports DRV82xx (motor driver) and DRV87xx (gate driver) devices
 - Scalable to discrete driver topologies for high-voltage Brushed DC applications
- Stepper algorithm libraries
 - [MSPM0 Stepper Library](#)
 - Supports DRV84xx integrated drivers for basic full-/half-stepping control
 - Supports DRV88xx intelligent drivers for up to 1/256 microstepping
- Peripheral features
 - Timer – Generate PWM and detect Hall sensor feedback
 - Generate PWM (TIMG)
 - Configure the cross trigger between TIMG instances to generate up to 8 PWMs
 - Detect Hall sensor feedback (optional)
 - Use capture/compare (CC) inputs to XOR digital hall signals and create frequency generator (FG) pulses for speed computation
 - ADC – Sensing current and voltage
 - Enable the 1-Msps ADC with multiple channels to sense motor current and bus voltage
 - SPI – Communicate with the gate driver for the stepper motor (optional)
 - Communicate with the gate driver to set subdivision parameters
 - OPA – Standard (STD) mode

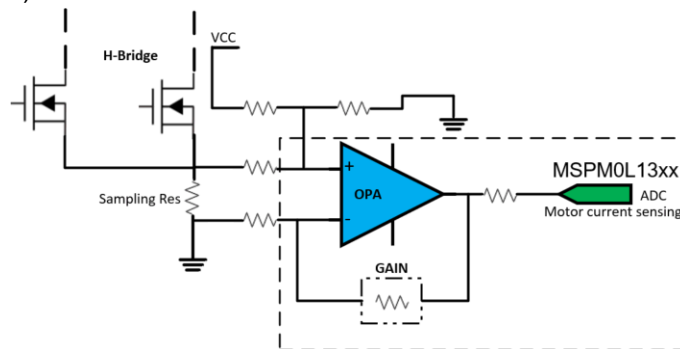


Figure 5. MSPM0L13xx OPA Block Diagram for Current Sensing

- 6-MHz GBW in standard (STD) mode
- 4-V/ μ s slew rate in STD mode
- 6- μ V/ $^{\circ}$ C input offset voltage temperature drift in STD mode
- COMP – High-speed mode

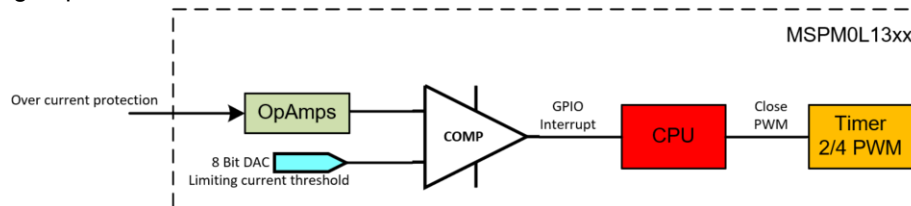


Figure 6. MSPM0L13xx COMP Block Diagram for Current Protection

- The 8-bit DAC in COMP sets the current limit threshold, and COMP generates an interrupt to the CPU if the current is over the threshold. The CPU stops the PWM at this interrupt.
- DAC settling time: 1 μ s in static mode
- COMP response time: 40 ns in high-speed mode

Resources

Order an [MSPM0 LaunchPad development kit](#) and DRV8xxx EVM today to start evaluating MSPM0 for your motor control system. Jump start your motor control design with MSPM0 code examples and interactive online trainings.

- [MSPM0 overview page](#)
- [LP-MSPM0L1306](#) LaunchPad development kit
- [MSPM0 Academy](#)
- DRV8xxx EVM
 - [DRV8706S-Q1EVM](#) (Brushed-DC Control)
 - [DRV8411AEVM](#) (Basic Stepper Control)
 - [DRV8889-Q1EVM](#) (Intelligent Stepper Control)
- [TI Precision Labs - Motor Drivers: Stepper Motor Driver Basics](#)
- [TI Precision Labs - Motor Drivers: Brushed-DC Basics](#)
- [TI Precision Labs - Motor Drivers: The H-Bridge](#)

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATA SHEETS), 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, regulatory 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](#) or other applicable terms available either on [ti.com](https://www.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.

TI objects to and rejects any additional or different terms you may have proposed.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265
Copyright © 2023, Texas Instruments Incorporated