Application Note

Brushed DC Motor Control Using the SimpleLink CC2340R5 MCU With Zigbee®

Ryan Brown

ABSTRACT

This application note explores the capability of adding the Zigbee® radio protocol to brushed DC (BDC) motor designs in a one-MCU design. BDC motors are commonly found in applications which can benefit from having embedded radios, including window shades, electric toothbrushes, and door locks. The material covered in this document shows how the CC2340R5 is capable of fulfilling this task with assistance of the DRV8251A motor driver and the TMAG5213 Hall-effect latch.

The design provided by this document uses hardware EVMs which are available for purchase on TI.com and firmware freely provided on the SimpleLink Low Power F3 Demos GitHub. A description of both required hardware connections and the operation of the firmware are given in detail so that, after obtaining a BDC motor, developers can be fully enabled to operate a demonstration and further modify the project to meet requirements. Additional test data is also provided so that readers can fully understand the operating conditions alongside options for application expansion.

Table of Contents

Trademarks

SimpleLink™, Code Composer Studio™, and EnergyTrace™ are trademarks of Texas Instruments.
Zigbee® is a registered trademark of Connectivity Standards Alliance.
Wi-Fi® is a registered trademark of Wi-Fi Alliance.

All trademarks are the property of their respective owners.

1 Introduction

The SimpleLink™ CC2340R5 is a powerful and inexpensive MCU with 512kB of flash and 36 or 64kB of SRAM, featuring an Arm® Cortex®-M0+ and 2.4GHz radio. This feature set is capable of achieving a multitude of end applications for a variety of radio protocols in a single-chip design. This application note highlights a single instance to prove the wider possibilities capable with this device. Furthermore, the CC2340R5 has been demonstrated to also support brushless DC (BLDC) motor and stepper motor applications.

Controlling a BDC motor with the CC2340R5 through Zigbee radio communication is possible when coupled with a DRV8251A and TMAG5213. This document details the hardware and software implementations necessary to realize this application, and optional features which have been enabled. Through reading this document, users can learn more about both BDC motor control and CC2340R5 development, and gain confidence to utilize similar concepts for designs.

1.1 CC2340R5

The CC2340R family is part of the SimpleLink™ MCU platform. This consists of Wi-Fi®, Bluetooth LE, Thread, Zigbee, Sub-1GHz MCUs, and host MCUs that all share a common, easy-to-use software development kit (SDK) and rich tool set. These devices are optimized for low-power wireless communication in building automation (wireless sensors, lighting control, beacons), asset tracking, medical, retail EPOS (electronic point of sale), ESL (electronic shelf), and personal electronics (toys, HID, stylus pens) markets.

The LP-EM-CC2340R5 development kit is used to speed up development with the CC2340R5 MCU with support for Bluetooth 5 Low Energy (LE), Zigbee, and 2.4GHz proprietary protocols. Software support is provided by the SimpleLink™ Low Power F3 software development kit (SDK) which can be built by using the Code Composer Studio™ integrated development environment (IDE). Features include access to all I/O signals with the BoosterPack™ plug-in module connectors and connecting the LaunchPad™ development kit to a smartphone using TI SimpleLink Connect. The LP-XDS110ET or LP-XDS110 debugger (sold separately) is required for software development and RF evaluation

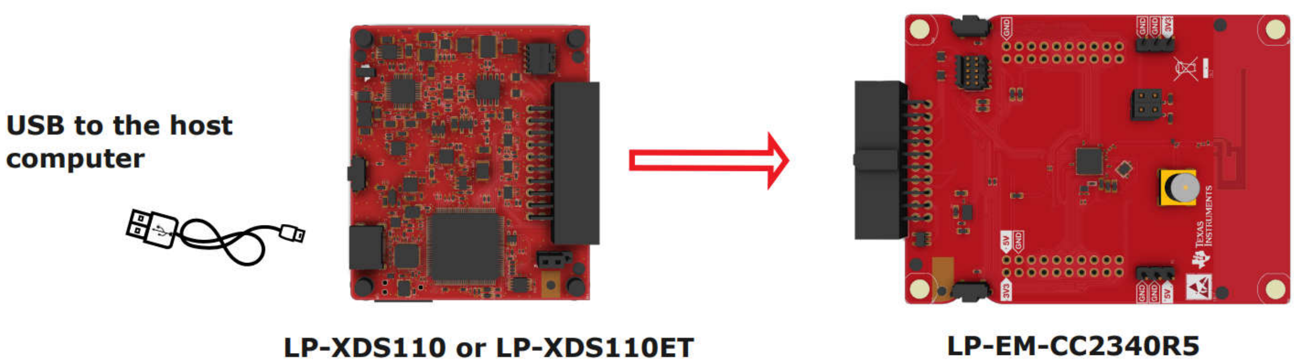

Figure 1-1. LP-XDS110ET and LP-EM-CC2340R5 Connection

1.2 DRV8251A

The DRV8251 family of devices are integrated motor drivers with N-channel H-bridge, charge pump, current regulation, and protection circuitry. The DRV8251A additionally has current-sense feedback. The charge pump improves efficiency by supporting N-channel MOSFET half bridges and 100% duty cycle driving. The external voltage reference pin, VREF, determines the threshold of current regulation during start-up and stall events without interaction from a microcontroller. A low-power sleep mode achieves ultra-low quiescent current draw by shutting down most of the internal circuitry. Internal protection features include supply undervoltage lockout, output overcurrent, and device overtemperature. The H-bridge driver architecture supports peak currents up to 4.1A. The DRV8251A can operate from a single power supply and supports a wide input supply range of 4.5V to 48V.

The DRV8251AEVM is a 3.7A, brushed DC drive stage based on the DRV8251A H-bridge driver for BDC motors. The device includes a current shunt amplifier for low-side current measurement, 150mA LDO, dead time control pin, VDS overcurrent level pin, and gate driver shutoff pin. The EVM comes equipped with two potentiometers which allow for adjustment of the control signal duty cycle. If desired, the on-board MCU can be disconnected so the DRV8251A can be controlled by an externally. Up to 48V can be supplied to the EVM, and the DRV8251AEVM's integrated LDO generates a 3.3V reference voltage. A status LED for the power supply is included for user feedback.

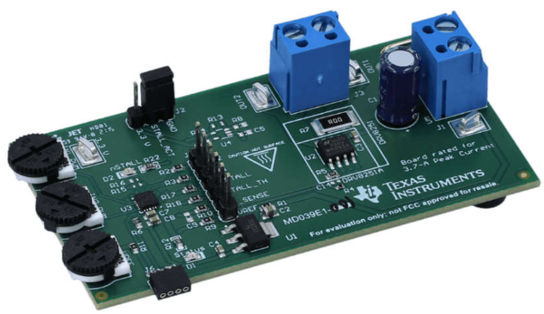

Figure 1-2. DRV8251AEVM

1.3 BDC Motor

A brushed DC (BDC) electric motor two-pole motor using a direct current (DC) power supply represents the general motor design enabled in this Application Note. In this report the interface circuit drives 2 PWMs to the high and low side to control both motor speed and direction. While some BDC motors contain an integrated Hall-effect sensor, the motor discussed in this report does not contain one. The red wire of the BDC motor is indicated in this report as M+ while the black wire is indicated as M-.

Figure 1-3. BDC Motor

1.4 TMAG5213

The TMAG5213 is a low-cost, chopper-stabilized Hall-effect sensor with excellent sensitivity stability over temperature and integrated protection features. The magnetic field is indicated through a digital bipolar latch output. This bipolar latch magnetic response allows the device output to be sensitive to positive and negative magnetic flux through the Z-axis of the package. The TMAG5213 has a wide operating voltage range from 2.5V to 28V and operating temperature range from –40°C to +125°C designed for a variety of industrial applications. Internal protection functions are provided for output short circuit or overcurrent, and the device has an open-drain output stage with 30mA current sink capability. The inclusion of the TMAG5213 is optional, but can be omitted in the case of using a BDC motor with an integrated Hall-effect latch or for a project using an alternative motor position tracking method.

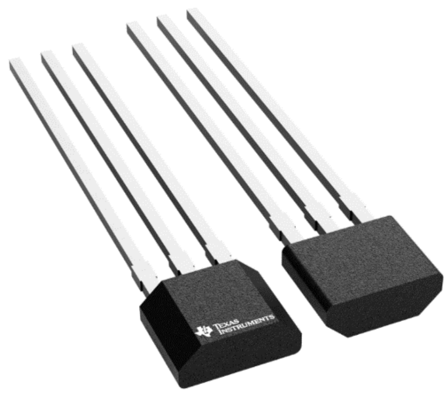

Figure 1-4. TMAG5213 Hall-Effect Latch

2 BDC Application

2.1 Hardware Setup

The following sections describe the hardware which must be procured, setup changes implemented on the EVMs, and then the necessary connections required to run the example without any modifications to the default firmware design.

2.1.1 DRV8251AEVM Settings

Below is a depiction of the DRV8251AEVM board with jumpers populated in the correct places. Also provided are some comments about how the board switches and sliders are positioned to avoid issues when trying to operate the BDC motor. Please refer to the DRV8251AEVM User's Guide for more instructions on how to interface with this hardware.

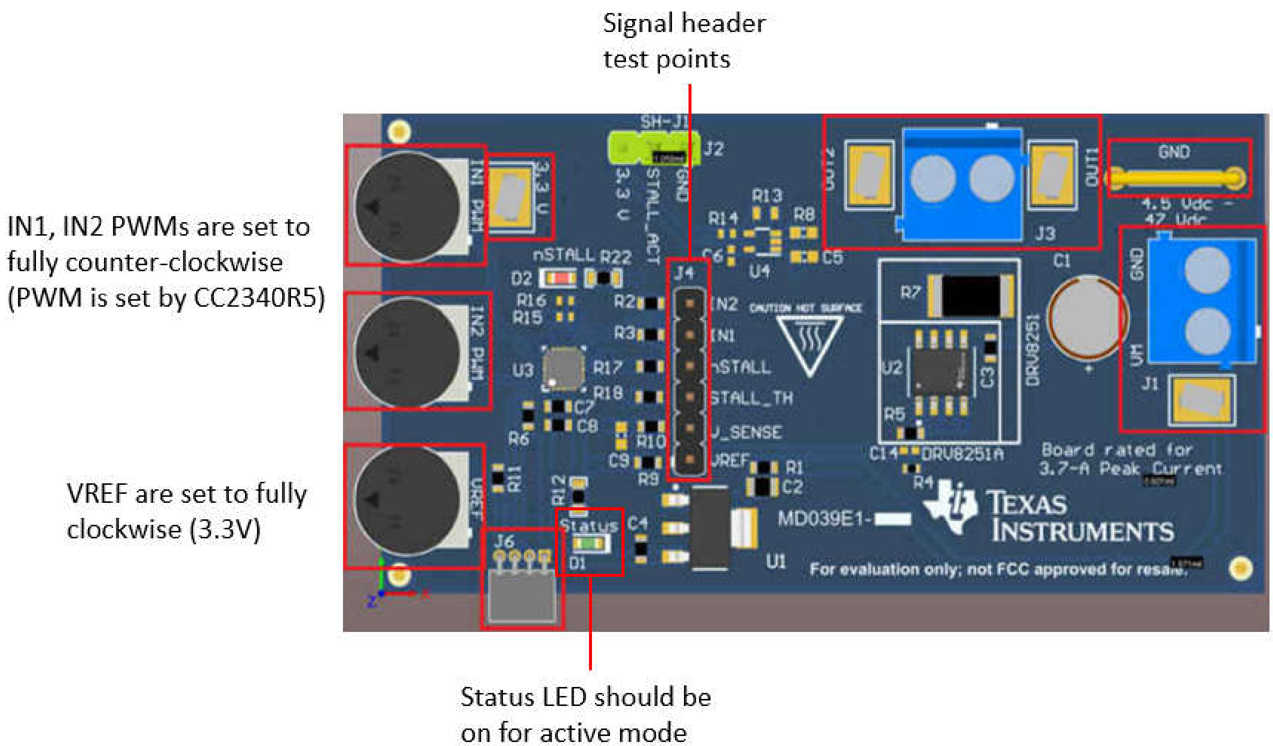

Figure 2-1. DRV8251AEVM Hardware Settings

2.2 Connection Diagram

This table contains the necessary connections between the LP-EM-CC2340R5 and the DRV8251AEVM and to realize the BDC motor demonstration.

Table 2-1. Connections Between the LP-EM-CC2340R5 and DRV8251AEVM

Connection	CC2340R5 Function	LP-EM-CC2340R5 Pin	DRV8251AEVM Pin
Forward motor control	PWM output	DIO1	IN1
Reverse motor control	PWM output	DIO5	IN2
Current sense	ADC input	DIO24	VSENSE
Common ground	Common ground	GND	GND

The BDC motor wires must be connected to the specified DRV8251AEVM pins. The necessary motor wires and the corresponding pin connections are listed below. For a specific motor, verify which wires implement these functions and connect them accordingly.

Table 2-2. DRV8251AEVM BDC Motor Connections

Connection	BDC Motor Wire	DRV8251AEVM Pin
High side	M+ (red)	OUT1
Low side	M- (black)	OUT2

The TMAG5213 wires must be connected to the specified CC2340R5 pins. The necessary Hall-effect latch wires and the corresponding pin connections are listed as shown. Powering the TMAG5213 using a CC2340R5 digital output enables the MCU to disable power to the TMAG5213 when the BDC motor is not being driven, reducing the power consumption during sleep mode. The TMAG5213ADQLPG package was used for this project, with the pinout and location highlighted in the following resources

Table 2-3. Connections Between the LP-EM-CC2340R5 and TMAG5213ADQLPG

Connection	CC2340R5 Function	LP-EM-CC2340R5 Pin	TMAG5213ADQLPG Pin	TMAG5213ADQLPG Pin Number
Hall-effect latch power	Digital output	DIO23	VCC	1
Hall-effect latch ground	Common ground	GND	GND	2
Hall-effect latch output	Digital interrupt input	DIO0	OUT	3

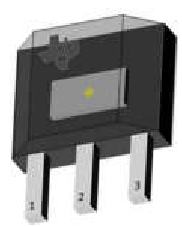

Figure 2-2. TMAG5213ADQLPG Pinout with Hall-Effect Element Highlighted

The following image highlights the hardware settings of the entire system, combining the LP-EM-CC2340R5, DRV8251AEVM, and the TMAG5213 which is positioned directly above the BDC motor. While this depicts a general BDC motor setup with an external Hall-effect latch, the system discussed in later sections was developed for the specific case of a window covering and controller application. The provided code design is also tailored to a window covering and controller application, however, this can be easily changed to support any specific BDC motor application.

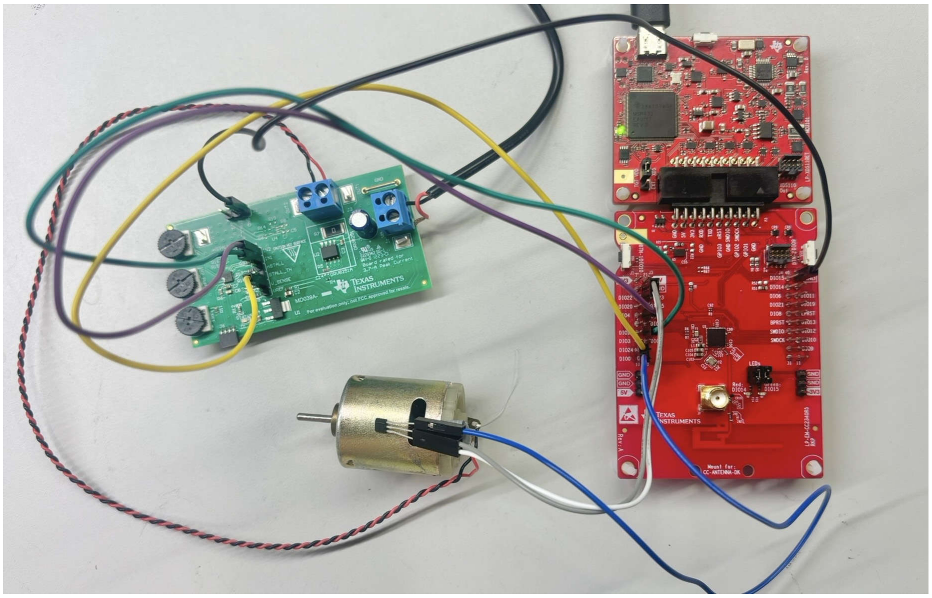

Figure 2-3. Physical Hardware Setup

3 Running the Example

The next sections discuss the firmware details and how each component works towards driving the BDC motor and collecting data. For the window covering and controller use case that is supported in this demonstration, the code is divided up into two separate CCS projects, with window_covering_LP_EM_CC2340R5_freertos_ticlang functioning as the sleepy Zigbee End Device (ZED) and window_controller_LP_EM_CC2340R5_freertos_ticlang functioning as the Zigbee Coordinator (ZC). The window covering project must be flashed onto the CC2340R5 LaunchPad that is connected to the DRV8251A, TMAG5213, and BDC motor. The window controller project must be flashed onto a separate, standalone CC2340R5 LaunchPad. More information on these CCS projects and the corresponding operating modes are provided in *Zigbee Network Formation*.

3.1 Dependencies

The code project supplied on the SimpleLink Low Power F3 Demos GitHub has the following dependencies:

- SimpleLink F3 SDK v9.10.0.83
- SysConfig v1.23.1
- TI CLANG v4.0.2 compiler

Make sure that all of these dependencies are installed on the machine before attempting to import the project into Code Composer Studio™ (CCS) v20 or later. For more examples for setting up your environment, refer to SimpleLink Academy for CC23xx. Note that users are responsible for migrating and supporting any dependency versions not listed above. See the Project Properties of window_covering_LP_EM_CC2340R5_freertos_tclang, which must be identical to the dependencies of window_controller_LP_EM_CC2340R5_freertos_ticlang.

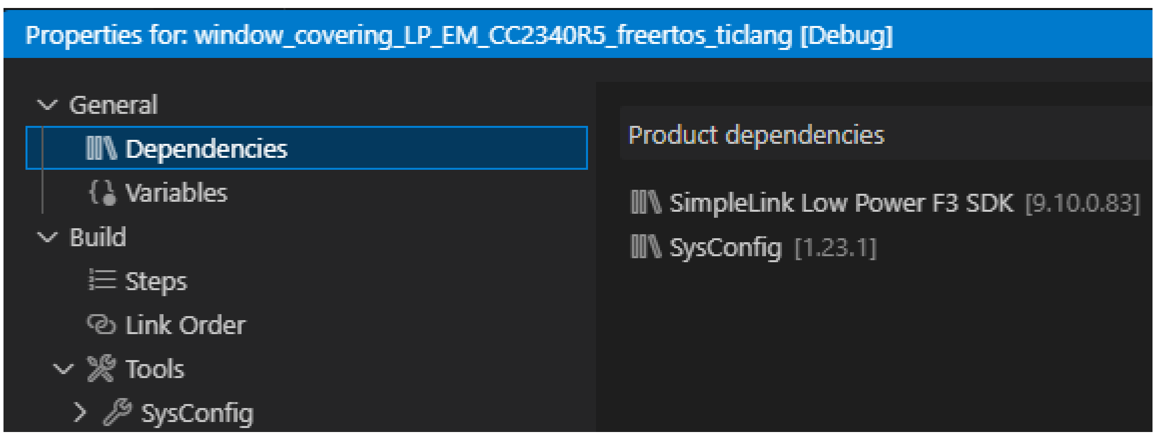

Figure 3-1. CCS Properties

3.2 Loading Firmware

Projects built inside of CCS can be loaded either directly in this IDE by selecting Run -> Flash Project (Ctrl + F5) or Debug Project (F5). TI recommends to exit Debug Mode to allow free-running if the project is not actively being debugged. Consider using the UNIFLASH software tool to load binary images.

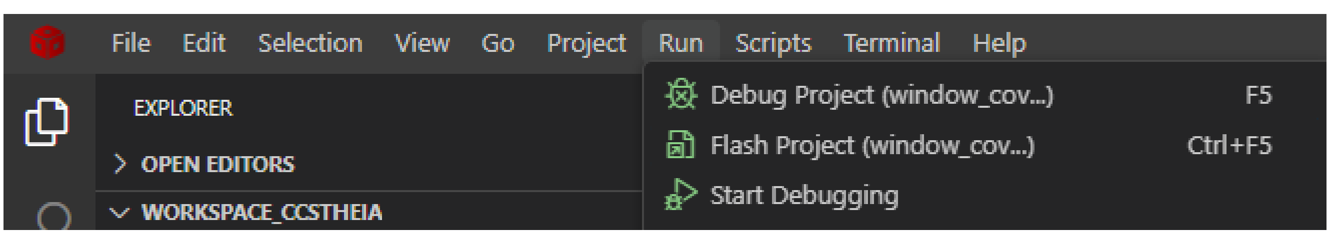

Figure 3-2. CCS Load Options

3.3 Zigbee Network Formation

The CC2340R platform supports the 2.4GHz IEEE 802.15.4-2011 physical layer (PHY) used in the Zigbee protocol. See the Zigbee User's Guide regarding protocol specific questions. Relevant Zigbee network parameters are configured in the SysConfig tool, including device type and channel selection. These settings must be consistent across devices to verify proper network operation. Network establishment follows the Zigbee Base Device Behavior (BDB) specification. To initiate the network, first, press the reset button on the window controller (ZC) device, which creates the network with configured parameters and opens the network for joining. Then press the reset button on the window covering (ZED) device, which enables the ZED to scan for and join the open network, establishing a communication path with the ZC. This simple reset button sequence automates the entire commissioning process. Once joined, devices can communicate using the supported Zigbee clusters, enabling immediate functionality without additional configuration steps. Using BTN-1 on the window controller LaunchPad toggles between window covering up and down commands sent to the window covering device. Developers are cautioned to fully test and understand local window operations, covered in Section 4.2 below, before evaluating Zigbee radio window commands.

This project uses the window covering/controller clusters within the Zigbee Cluster Library (ZCL) for Home Automation (HA). For better results, reset the ZC first and allow one to two seconds before resetting the ZED. For additional documentation on Zigbee network formation, see the Zigbee Light and Switch Project Zero SimpleLink Academy Lab and ZBOSS User's Guide.

4 Firmware Design

The firmware of the window_covering projects which operates the BDC motor project is further analyzed in the following sections.

4.1 Code Flow Description

Figure 4-1 shows a code block diagram of the processes used inside the CC2340R5 code. The Zigbee functionality is available from *window_covering.c* while the motor operation is realized in the *bdc_motor.c* file.

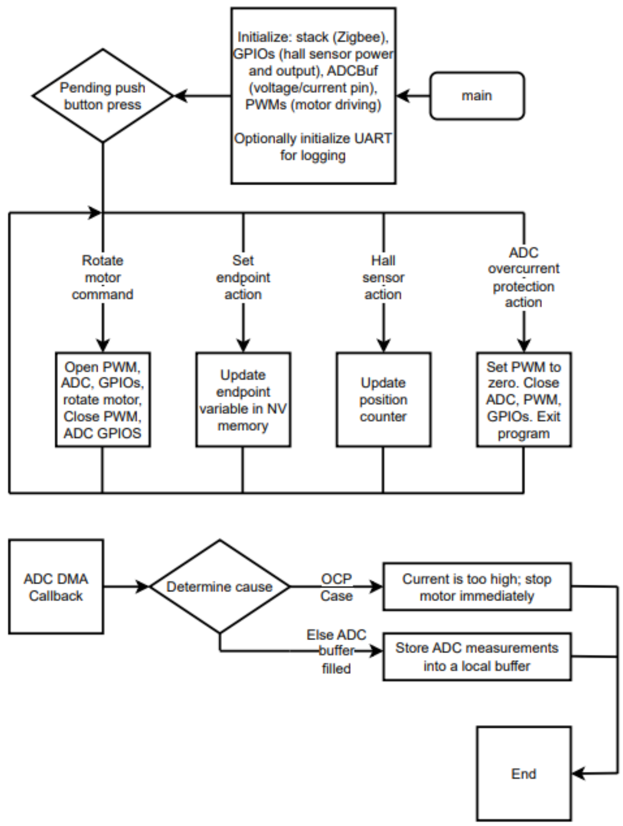

Figure 4-1. BDC Motor Code Diagram

The main function initializes all TI drivers and timers necessary for the BDC motor example to operate. Once entering the main while loop, this pends further action on an event being set by hardware callbacks. For the provided application, these hardware callbacks are triggered by pressing the push buttons on the LaunchPads. After servicing the corresponding action through a subroutine, the events are reset and the process repeats.

All hardware callbacks, with a few exceptions, simply post an event for the main application to process. An exception is the ADCBuf callback which processes the status immediately and does not call for any further action from the main application loop.

There are several definitions implemented which define the motor behavior and are referenced in the following sections. These configurable definitions are located in *bdc_motor.c*.

Table 4-1. BDC Motor Application Defined

Definition	Default	Units	Function
PWM_PERIOD	100	µs	Duration of one PWM cycle
PWM_DUTY_INC	100	µs	Duration to increment PWM duty cycle during acceleration or decrement during deceleration
PWM_START_POINT	0	µs	PWM duty value to accelerate from/to decelerate to. Generally, is set to zero
PWM_END_POINT	100	µs	PWM duty value to accelerate to/ to decelerate from. In this case, this is set as the same value as the PWM_PERIOD for 100% duty cycle
ADCBUF_SAMPLE_SIZE	100	Integer	Size of ADC buffer transferred by DMA
ADCBUF_SAMPLING_FREQ	1000	Hz	Sampling frequency of the ADC. Note that the frequency of ADC buffers completed is equal to ADCBUF_SAMPLING_FREQ / ADCBUF_SAMPLE_SIZE
USE_HALL	Defined	N/A	Determines whether the hall effect sensor is used during functionality or not
STALL_TIMEOUT	1000000	µs	Timeout to stop motor if the motor has not been moving for this amount of time as detected by the hall effect sensor when USE_HALL is defined
MOVEMENT_TIME	3000000	µs	Timeout period to stop motor if the motor has been spinning longer than this duration, if USE_HALL is not defined
OCP_THRESHOLD	800000	µV	Minimum value of averaged ADC buffer to trigger an overcurrent condition

4.2 LaunchPad Button Functionality

To interact with the BDC motor of the device window covering locally, pushbutton operations are implemented for several events which allow the user to configure the motor endpoints and rotate the motor clockwise or counter-clockwise. This table summarizes those events.

Table 4-2. LaunchPad Button Operation for BDC Motor

Button Operation	BTN-1 Function	BTN-2 Function
Double Click	Set high endpoint for current position	Set low endpoint for current position
Single Click	Move until reached high endpoint location (USE_HALL defined) or movement timeout (USE_HALL not defined)	Move until reached low endpoint location (USE_HALL defined) or movement timeout (USE_HALL not defined)
Pressed Down	Move up continuously until released	Move down continuously until released

When USE_HALL is defined, TI recommends that users rotate the motor manually (that is, button press) and set endpoints (that is, button double clicks) before using Zigbee or single button click commands to control the motor position. When USE_HALL is not defined, a single click only moves in the desired direction for MOVEMENT_TIME.

4.3 Non-Volatile Memory

Non-volatile (NV) Flash memory is implemented to save both the high and low endpoints as well as the current motor rotation position whenever a new value is recorded. Therefore, when the device is power cycled or reset these variables are restored to the former values. High and low endpoints must be re-established through pushbutton operations each time the device erases flash memory, such as during a programming operation. NV memory is only used when USE_HALL is defined, and cannot account for any physical motor rotation position changes that occur while the CC2340R5 device is not running.

4.4 Bidirectional PWM Motor Control

This BDC motor application uses the PWM driver to implement variable speed motor control in both the forward and reverse direction. This requires the use of two separate PWM timers to enable bidirectional control. Motor acceleration and deceleration features are implemented to smoothly ramp up and ramp down the duty cycle to reduce sudden from zero to PWM_DUTY_VALUE.

4.5 ADC Overcurrent Protection Feature

In this BDC motor application, overcurrent protection (OCP) works by continuously monitoring current through ADC sampling of the VSENSE pin of the DRV8251A. The VSENSE pin outputs a voltage signal directly proportional to the current draw of the motor, providing real-time feedback on the electrical load of the motor. The CC2340R5's ADC periodically samples this voltage at one-millisecond intervals during motor operation, converting the analog VSENSE reading to a digital value that represents actual motor current. When this measured current exceeds the predefined ocpThreshold value, the protection mechanism immediately disables both PWM channels, stopping the motor before damage occurs. Possible causes of an overcurrent condition include a stalled motor, mechanical obstruction, or electrical fault. This implementation leverages the integrated current sensing capability of the DRV8251A, eliminating the need for external current sense resistors while providing accurate, responsive protection that safeguards both the motor and the mechanical system during operation in either direction. The implementation features a dedicated callback function that processes ADC samples and triggers the OCP functionality when necessary, which additionally logs the fault condition and increments a fault counter for diagnostic purposes.

4.6 Hall-Effect Based Motor Position Tracking Feature with Configurable Endpoints

This BDC motor design implements an external Hall-effect latch to enable precise tracking of the rotational position of the BDC motor. Correct positioning of the TMAG5213 relative to the BDC motor is essential to accurately monitor the motor rotation. As highlighted in the following image, the magnet within the BDC motor must spin such that the poles flip around a center or rotation above the defined *Z-axis*. Within the CCS project, a GPIO input interrupt is set to increment a counter every time the Hall-effect latch is set to high. Additionally, the Hall-effect latch is powered by a GPIO output, allowing one to disable it when the motor is not actively spinning to minimize power consumption. This Hall-effect based motor position tracking enables more precise motor position tracking than a time-based design. Use of configurable endpoints can be enabled to start and stop motor operation by either setting variables in the code at compile time or though button presses in real time. The window covering project can be configured to operate with other motor position tracking methods by removing the USE_HALL definition.

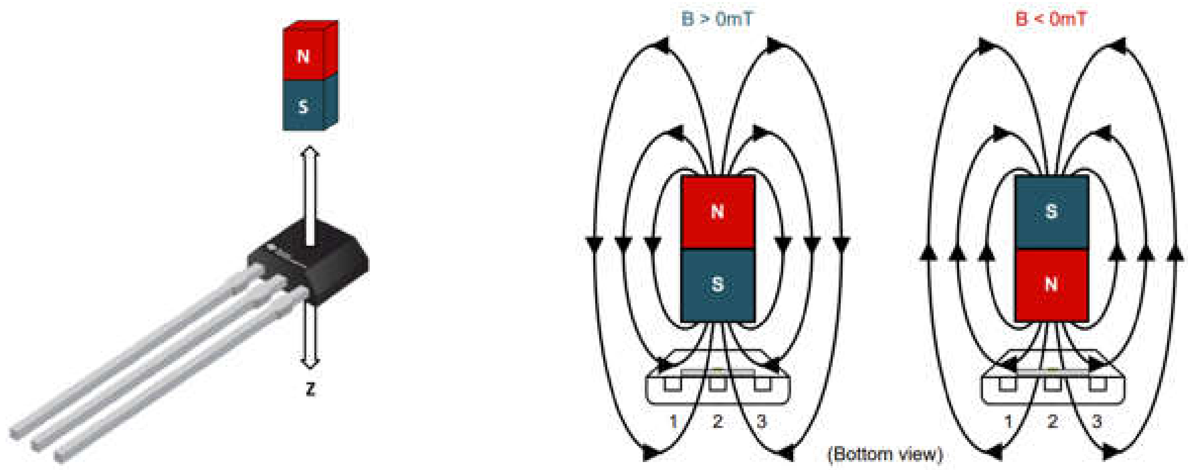

Figure 4-2. Hall-Effect Latch Magnetic Sensing Orientation

4.7 Logging Functionality

The optional logging feature serves as a useful debugging tool in this BDC motor application, providing real-time visibility into system operation through the UART log sink. The implementation uses a structured logging system with different severity levels (DEBUG, VERBOSE, INFO, WARNING, ERROR) to categorize messages based on importance. During motor operations, key events are logged with timestamps, such as motor start/stop actions, current measurements, position updates, and fault conditions such as overcurrent detection. This logging capability proves invaluable during development and troubleshooting, as this provides insights into the timing of operations, the sequence of events during motor control, and the specific conditions that can trigger protection mechanisms. The log sink can be enabled or disabled at compile time through configuration settings, allowing developers to include comprehensive debugging information during development while removing logging overhead for production releases to optimize performance and reduce code size. See Logging with the SimpleLink F3 SDK for more information.

4.8 Interoperability with Third-Party Smart Hub Devices

The CC2340R5-based Zigbee implementation offers seamless interoperability with popular third-party smart home ecosystems such as Amazon Echo, Apple Homepod, and Google Nest, through standardized Zigbee 3.0 communication protocols. This BDC motor application has been tested with Amazon Echo. These smart hub devices act as a ZC. During the commissioning process, the CC2340R5 device (ZED) joins the Zigbee network of the Amazon Echo hub through standard network steering procedures, after which Alexa automatically recognizes the device type based on the simple descriptor and supported clusters. This enables natural voice commands such as *Alexa, turn on the light* or *Alexa, set the fan to 50%*. Setting up a BDC motor application with a third-party smart hub device requires use of that app for a third party for ZC-side configuration.

5 Tests and Results

This section presents comprehensive performance analysis of the CC2340R5-based BDC motor control design, focusing on PWM characteristics, motor position tracking, power consumption metrics, and system reliability.

5.1 PWM and Hall-effect Signal Analysis

Using a logic analyzer, the two PWM channels and the Hall-effect latch OUT pin can be monitored during motor operation. As captured, PWM undergoes a linear acceleration from 0% duty cycle to 100% duty cycle. As a result of driving the motor at an accelerating duty cycle, the Hall-effect latch monitors the motor starting to turn. The Hall-effect latch frequency increases during this time until the latch frequency reaches a steady state value. A similar behavior is observable during motor deceleration after the motor control is complete.

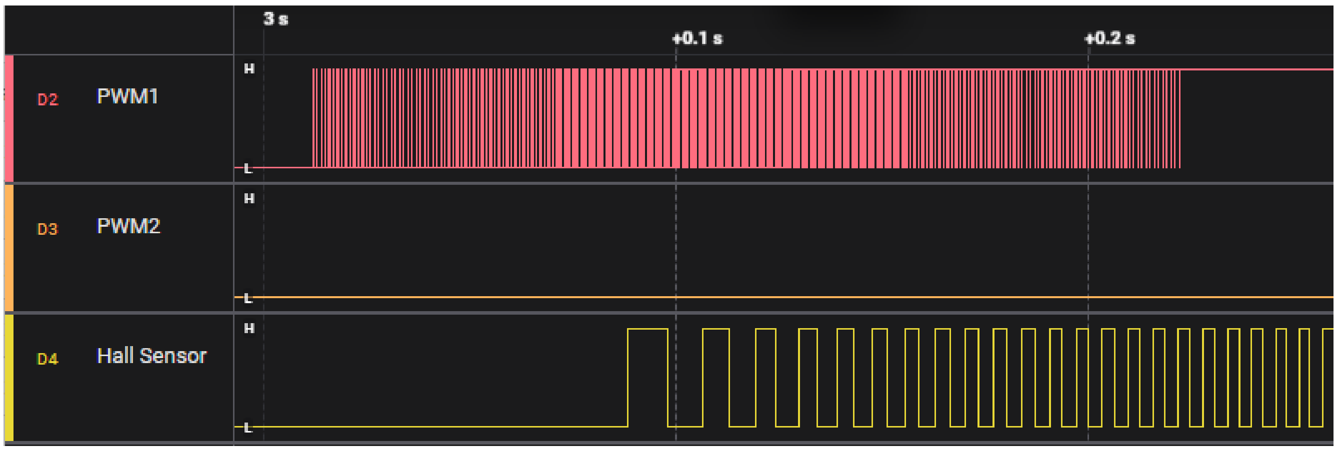

Figure 5-1. PWM and Hall-Effect Latch Logic Analyzer Capture

5.2 Power Consumption Analysis with EnergyTrace™

EnergyTrace™ technology provides invaluable insights into the power characteristics of a CC2340R5-based BDC motor control design. The EnergyTrace technology is provided through the LP-XDS110ET. By capturing real-time power consumption across different operational states, users can evaluate the energy efficiency of the system and identify opportunities for optimization. In the EnergyTrace capture, the window covering current is monitored during one motor operation event with a duty cycle of 100%. Since the window covering application is functioning as a sleepy ZED with a poll period of 3 seconds, a small spike occurs as the radio looks to receive any potential transmission from the ZC. With these spikes, the average current draw is < 10μA, which is an expected current draw value. This value can be lowered even further with the use of a longer poll period. Note that the standby current of the CC2340R5 with the radio disabled is < 1μA.

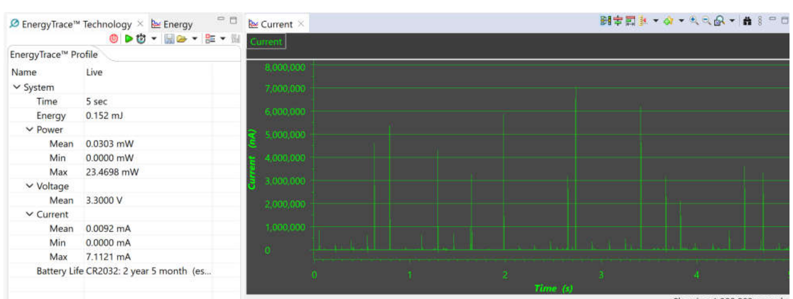

Figure 5-2. EnergyTrace Capture of Window Covering During Radio Standby

During motor operation, the device enters active mode, resulting in a current draw of several milliamps. This enables use of timers such as LGPT for PWM. In this case, with the use of a 12V BDC motor, the draw is 27during the period of motor operation. This value depends on the specific motor being used and the resistance characteristics. Current draw during motor operation also depends on the duty cycle as the CC2340R5 must drive the IN1 and IN2 inputs on the DRV8251EVM. After motor operation concludes, the device drivers are closed to allow the CC2340R5 to reenter a low current standby mode, waiting for further motor control instructions.

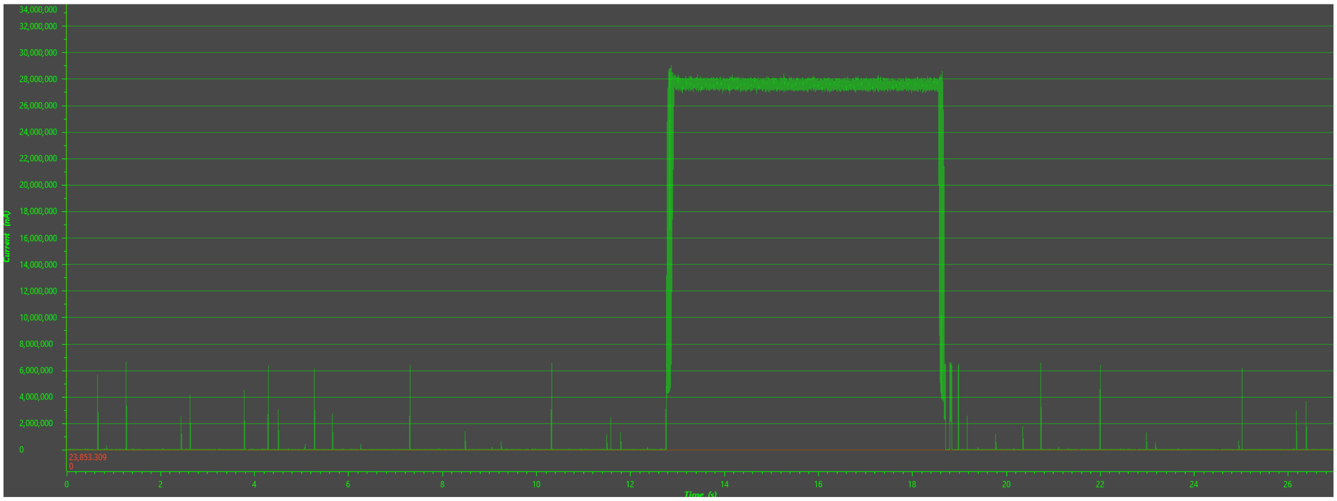

Figure 5-3. EnergyTrace Capture of Window Covering Motor Operation

6 Summary

This application note has fully defined a BDC motor design using the CC2340R5 and DRV8251AEVM and TMAG5213. A description of the required hardware connections and MCU programming instructions have been provided so that users are empowered to operate the out-of-box demonstration. Test results have been provided to confirm the stability and robustness of the design. The source code is freely accessible and the code flow has been detailed so that developers are familiar with how the project works and are enabled to further modify the project to fit unique application requirements. Readers are encouraged to post to the E2E forum concerning any additional questions or support needs that pertain to these resources provided.

7 References

1. Texas Instruments, CC2340R SimpleLink™ Family of 2.4GHz Wireless MCUs, data sheet.
2. Texas Instruments, CC2340R5 LaunchPad Development Kit, quick start guide.
3. Texas Instruments, DRV8251A 4.1-A Brushed DC Motor Driver with Integrated Current Sense and Regulation, data sheet.
4. Texas Instruments, DRV8251/AEVM User's Guide, user's guide
5. Texas Instruments, TMAG5213 Hall-Effect Latch for Cost-Optimized Designs, data sheet.
6. Texas Instruments, Zigbee Quick Start Guide, quick start guide.
7. Texas Instruments, SimpleLink Academy Labs, training.
8. GitHub, SimpleLink Low Power F3 Demos, example code.