Document Number: KTFRDM34931UG
Rev. 2.0, 9/2015

FRDM-34931S-EVB / FRDM-34931-EVB Evaluation Board

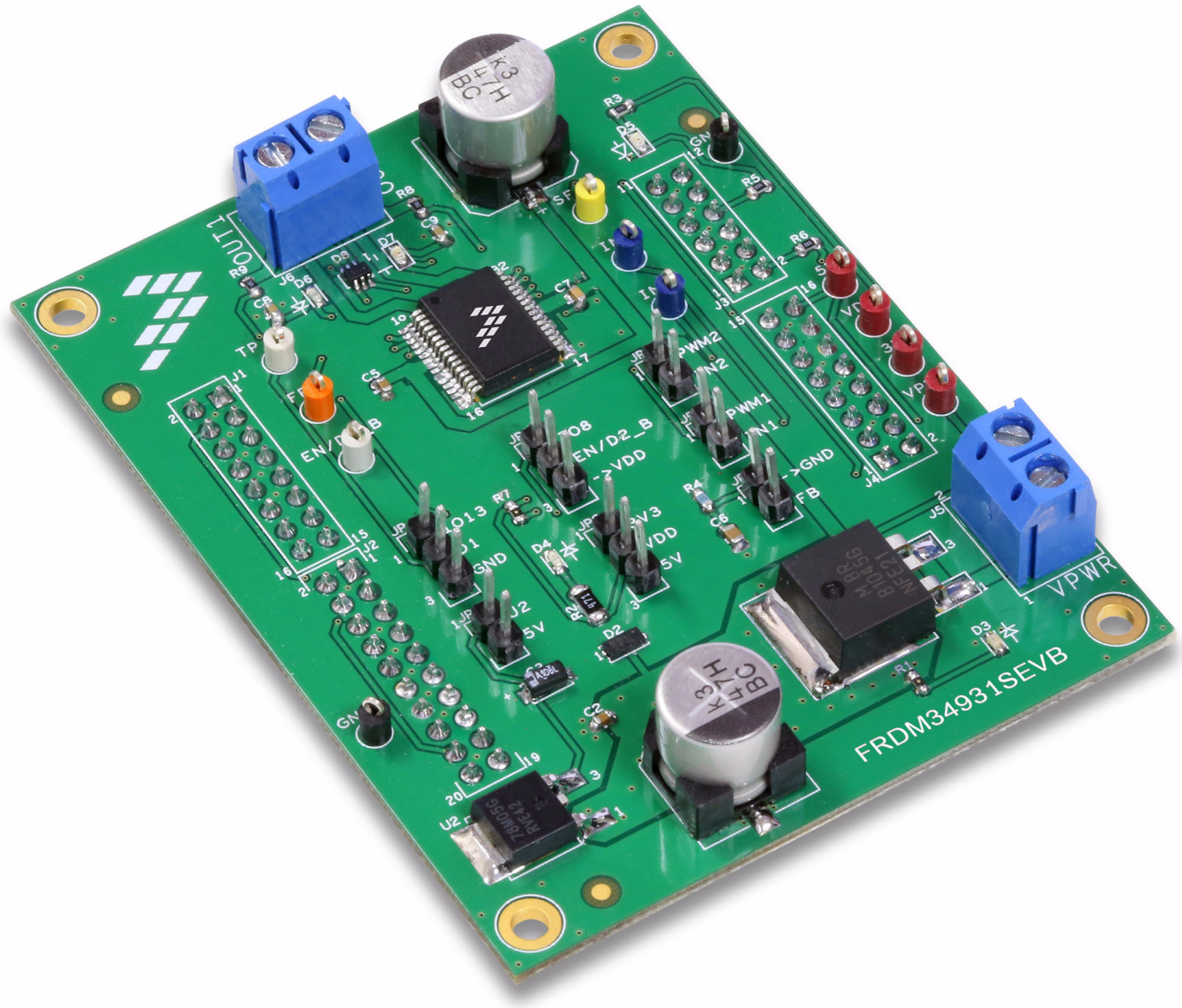

Figure 1. FRDM-34931S-EVB / FRDM-34931-EVB

© Freescale Semiconductor, Inc., 2015. All rights reserved.

Table of Contents

1 Important Notice . . . 3
2 Getting Started . . . 4
3 Getting to Know the Hardware . . . 5
4 FRDM-KL25Z Freedom Development Platform . . . 14
5 Setting up the Hardware and the Graphical User Interface (GUI) . . . 18
6 Installing Processor Expert Software . . . 30
7 Schematic . . . 49
8 Board Layout . . . 50
9 Board Bill of Materials . . . 51
10 Accessory Item Bill of Materials . . . 53
11 References . . . 54
12 Revision History . . . 55

1 Important Notice

Freescale provides the enclosed product(s) under the following conditions:

This evaluation kit is intended for use of ENGINEERING DEVELOPMENT OR EVALUATION PURPOSES ONLY. It is provided as a sample IC pre-soldered to a printed circuit board to make it easier to access inputs, outputs, and supply terminals. This evaluation board may be used with any development system or other source of I/O signals by simply connecting it to the host MCU or computer board via off-the-shelf cables. This evaluation board is not a Reference Design and is not intended to represent a final design recommendation for any particular application. Final device in an application will be heavily dependent on proper printed circuit board layout and heat sinking design as well as attention to supply filtering, transient suppression, and I/O signal quality.

The goods provided may not be complete in terms of required design, marketing, and or manufacturing related protective considerations, including product safety measures typically found in the end product incorporating the goods. Due to the open construction of the product, it is the user's responsibility to take any and all appropriate precautions with regard to electrostatic discharge. In order to minimize risks associated with the customers applications, adequate design and operating safeguards must be provided by the customer to minimize inherent or procedural hazards. For any safety concerns, contact Freescale sales and technical support services.

Should this evaluation kit not meet the specifications indicated in the kit, it may be returned within 30 days from the date of delivery and will be replaced by a new kit.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typical”, must be validated for each customer application by customer’s technical experts.

Freescale does not convey any license under its patent rights nor the rights of others. Freescale products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale product could create a situation where personal injury or death may occur.

Should the Buyer purchase or use Freescale products for any such unintended or unauthorized application, the Buyer shall indemnify and hold Freescale and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges Freescale was negligent regarding the design or manufacture of the part.Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © Freescale Semiconductor, Inc. 2015

2 Getting Started

2.1 Kit Contents/Packing List

The **FRDM-34931S-EVB / FRDM-34931-EVB** contents include:

- Assembled and tested evaluation board/module in anti-static bag
- Warranty card

2.2 Jump Start

Freescale's analog product development boards help to easily evaluate Freescale products. These tools support analog mixed signal and power solutions including monolithic ICs using proven high-volume SMARTMOS mixed signal technology, and system-in-package devices utilizing power, SMARTMOS and MCU dies. Freescale products enable longer battery life, smaller form factor, component count reduction, ease of design, lower system cost and improved performance in powering state of the art systems.

- For FRDM-34931-EVB, go to www.freescale.com/FRDM-34931-EVB
- For FRDM-34931S-EVB, go to www.freescale.com/FRDM-34931S-EVB
- Review your Tool Summary Page
- Look for



- Download documents, software and other information

Once the files are downloaded, review the user guide in the bundle. The user guide includes setup instructions, BOM and schematics. Jump start bundles are available on each tool summary page with the most relevant and current information. The information includes everything needed for design.

2.3 Required Equipment and Software

To use this kit, you need:

- DC Power supply: 5.0 V to 40 V with up to 10 A current handling capability, depending on motor requirements.
- USB Standard A (male) to mini-B (male) cable
- Typical loads (brushed DC motor, power resistors or inductive load with up to 5.0 A and 36 V operation)
- Function generator (optional)
- FRDM-KL25Z Freedom Development Platform (optional)
- ARM®mbed™ firmware loaded on FRDM-KL25Z board (To compile the code, you need to have an account in www.mbed.org.)
- MC34931 microcode loaded on FRDM-KL25Z
- Graphical User Interface required for use with FRDM-KL25Z

2.4 System Requirements

The kit requires the following to function properly with the software:

- A USB enabled computer with Windows® XP or later (required only if FRDM-KL25Z is used)

3 Getting to Know the Hardware

3.1 Board Overview

The FRDM-34931S-EVB / FRDM-34931-EVB Evaluation Board (EVB) provides a development platform that exercises all the functions of the MC34931 H-Bridge IC. The EVB is designed for use in conjunction with the FRDM-KL25Z board (not included with the evaluation board.) In this configuration, the FRDM-KL25Z must be prepped and the hardware configured as described in Section 5. To control the MCU outputs, use the graphical user interface available on Freescale website "GUI Brushed DC FRDM-34931S-EVB" for FRDM-34931S-EVB and "GUI Brushed DC FRDM-34931-EVB" for FRDM-34931-EVB. Alternatively, the EVB can be used without the FRDM-KL25Z, in which case the parallel inputs in the device must be controlled through 3.3/5 V compatible GPIO of the MCU or by connecting the board to a function generator.

3.2 Board Features

The board allows evaluation of Freescale part MC34931 and all its functions. The board features the following.

- Compatibility with Freescale's all Freedom Development Platform
- Built in reverse battery protection
- Test points to allow signal probing
- Built in voltage regulator to supply logic level circuitry
- LEDs to indicate the supply status and direction of motor
- Transient voltage suppressor to handle system level transients

3.3 Block Diagram

The hardware block diagram is shown in Figure 2

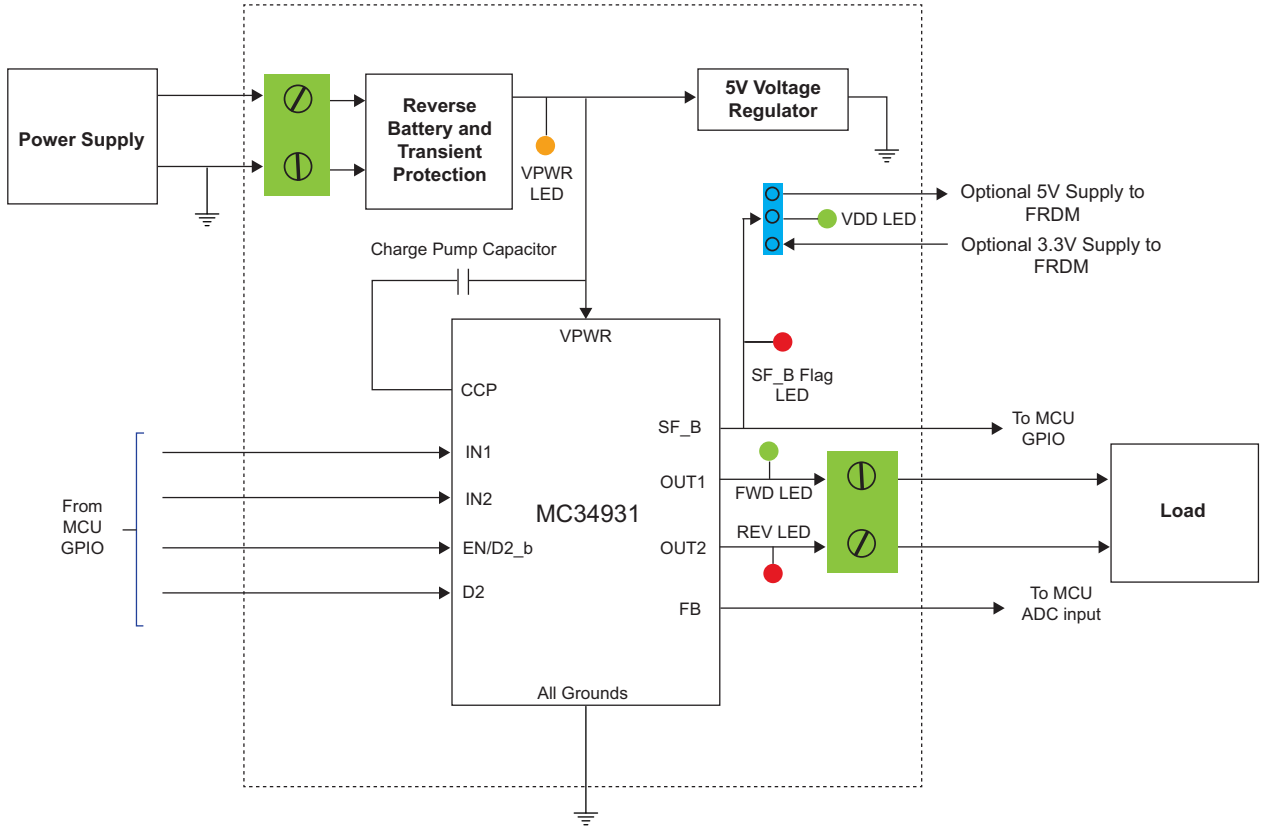

Figure 2. Block Diagram

3.4 Device Features

This evaluation board features the following Freescale product:

Table 1. MC3491 Device Features

Device	Description	Features
MC34931	The 34931 is a monolithic H-Bridge Power IC in a robust thermally enhanced 32 pin SOIC-EP package.	• 5.0 V to 36 V continuous operation (transient operation from 5.0 V to 40 V) • 3.0 V and 5.0 V TTL / CMOS logic compatible inputs • 235 mΩ maximum RDS(on) at T_J = 150 °C (Each H-Bridge MOSFET) • Overcurrent limiting (regulation) via internal constant-off-time PWM • Output short-circuit protection (short to VPWR or GND) • Temperature-dependent current-limit threshold reduction • Sleep mode with current draw < 20 µA

3.5 Operation Modes

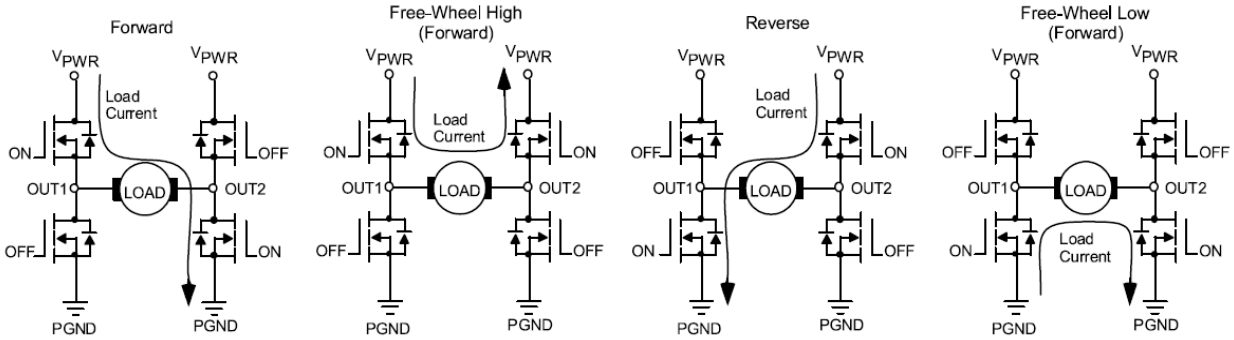

Figure 3. Operation Modes

3.6 Architecture

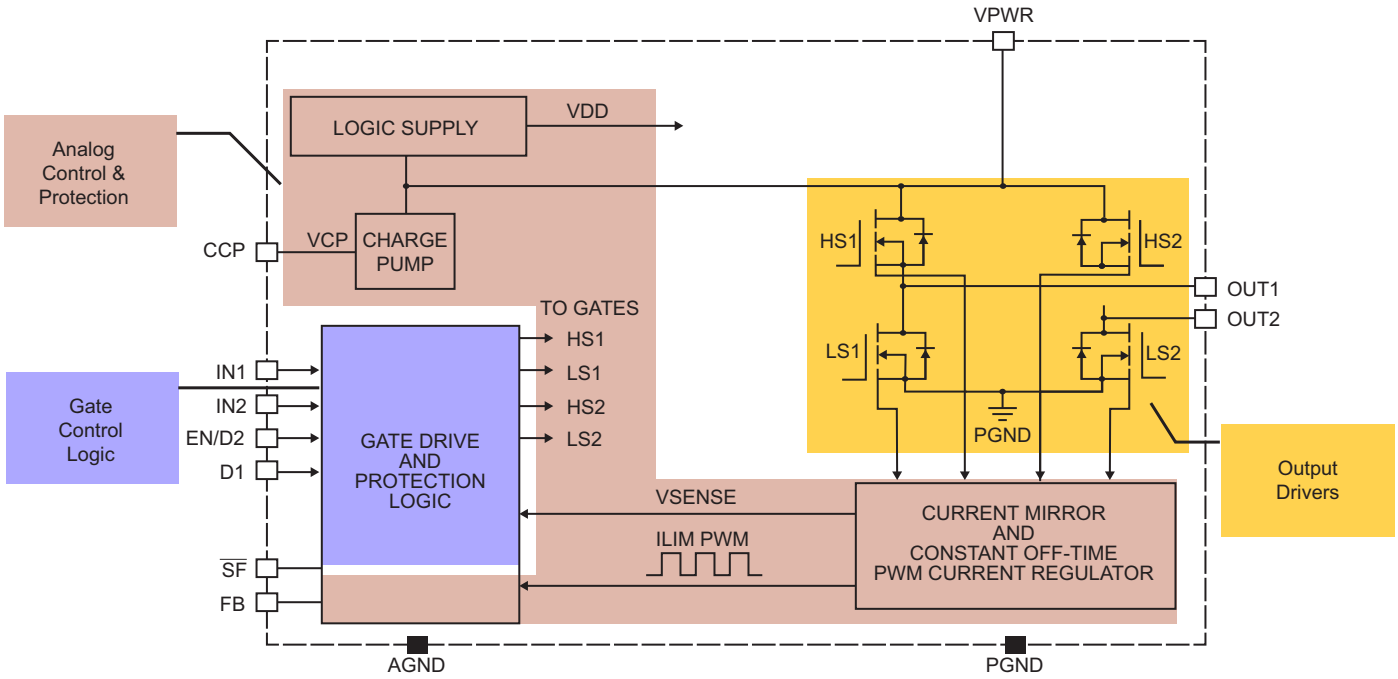

Figure 4. General Architecture Diagram

3.7 Thermal Management

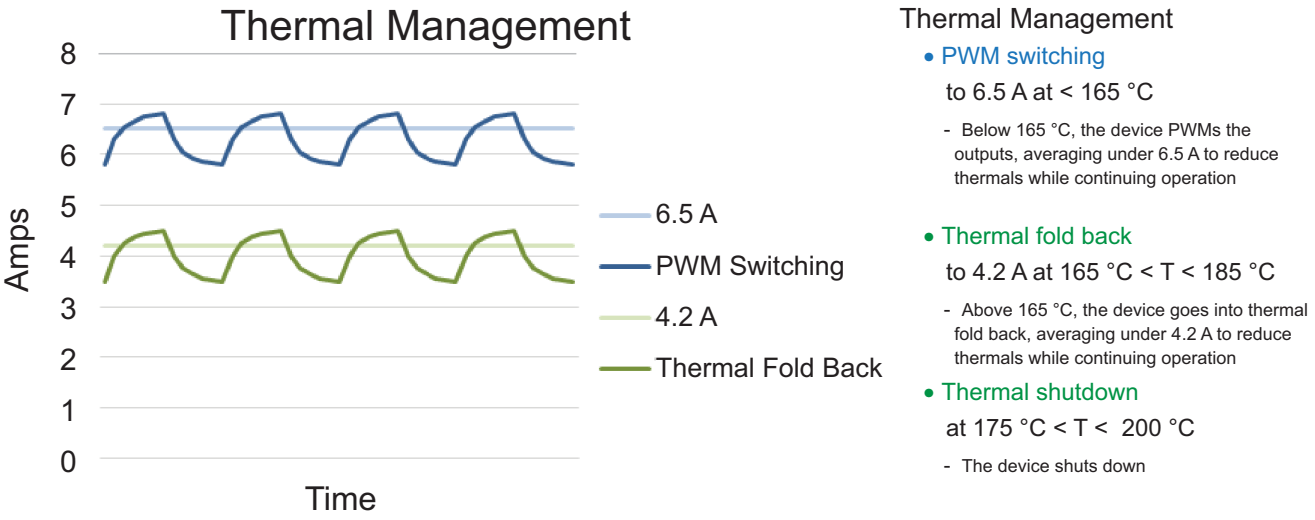

Figure 5. Thermal Management (Thermal Fold-back)

3.8 Board Description

Figure 6 describes the main blocks of the evaluation board.

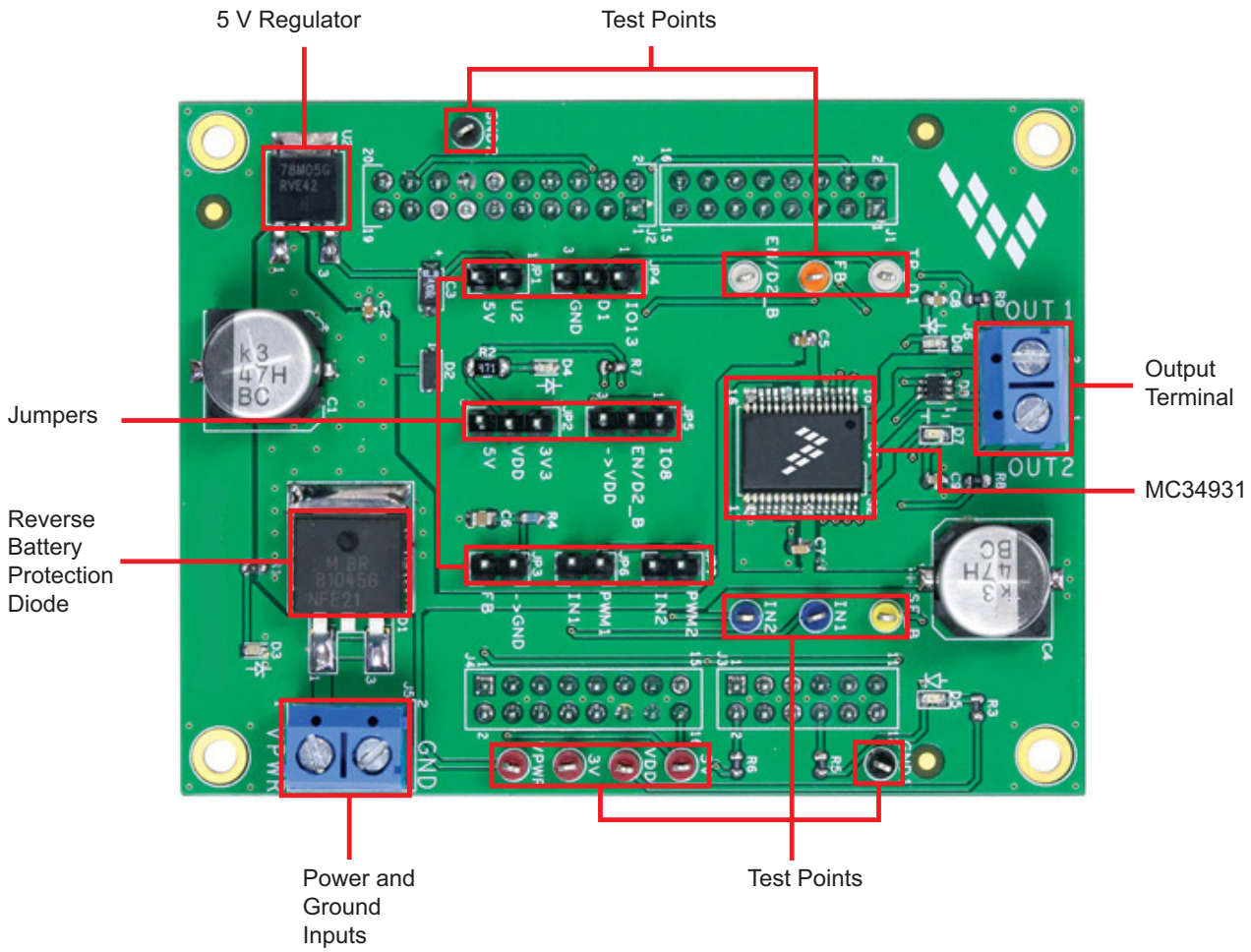

Figure 6. Board Description

Table 2. Board Description

Name	Description
MC34931	Monolithic H-Bridge Power IC in a robust thermally enhanced 32 pin SOIC-EP package
5.0 V Regulator	5.0 V regulator for VDD and supply.
Jumpers	Jumpers for configuring the board for different modes of operation
Reverse Battery Protection Diode	Diode for protecting MC34931 in reverse battery condition
Power and Ground inputs	Power supply terminal to connect the battery/power supply with the board
Test Points	Test points to probe different signals
Output terminal	Output connector to connect a load to the MC34931 output

3.9 LED Display

The following LEDs are provided as visual output devices for the evaluation board:

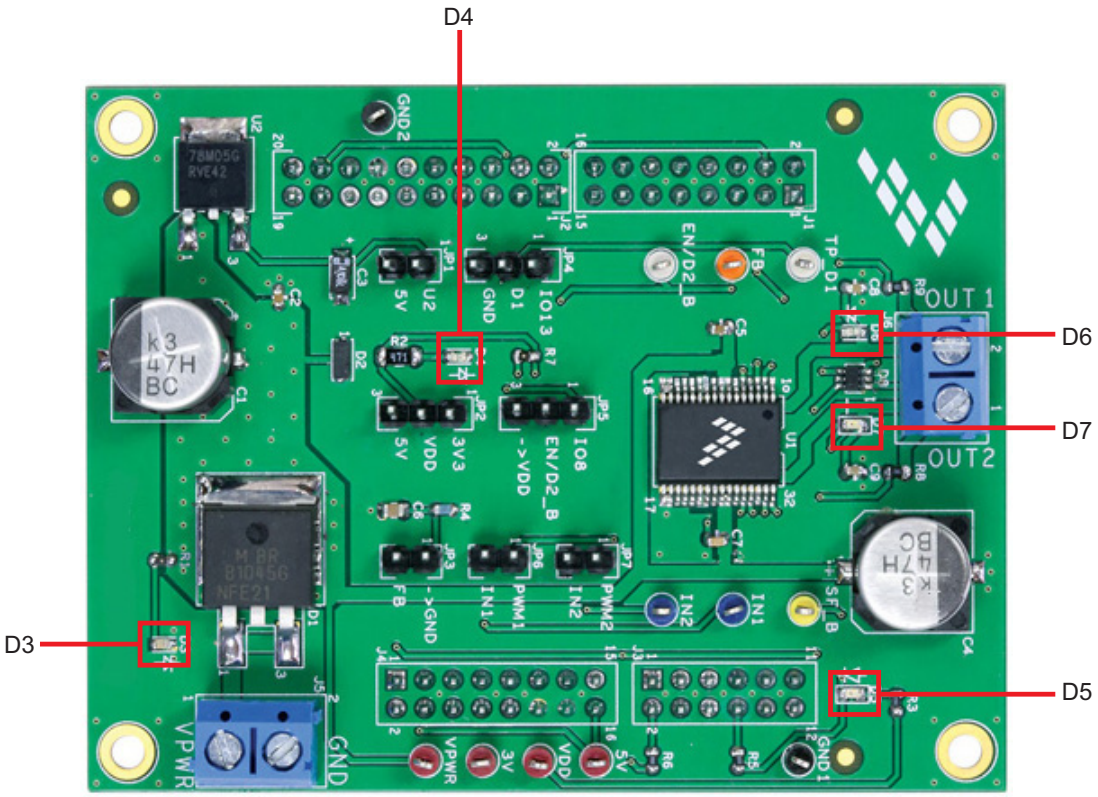

Figure 7. LED Display

Table 3. LED Display

LED ID	Description
D3	YELLOW LED, indicates when main/battery supply is connected
D4	GREEN LED, indicates when +5.0 V supply is connected
D5	RED LED, illuminates when the H-Bridge detects a fault
D6	Green LED, indicates current flowing in forward direction
D7	RED LED, indicates current flowing in reverse direction

3.10 Jumper Definitions

Figure 8 shows the jumper locations on the board.

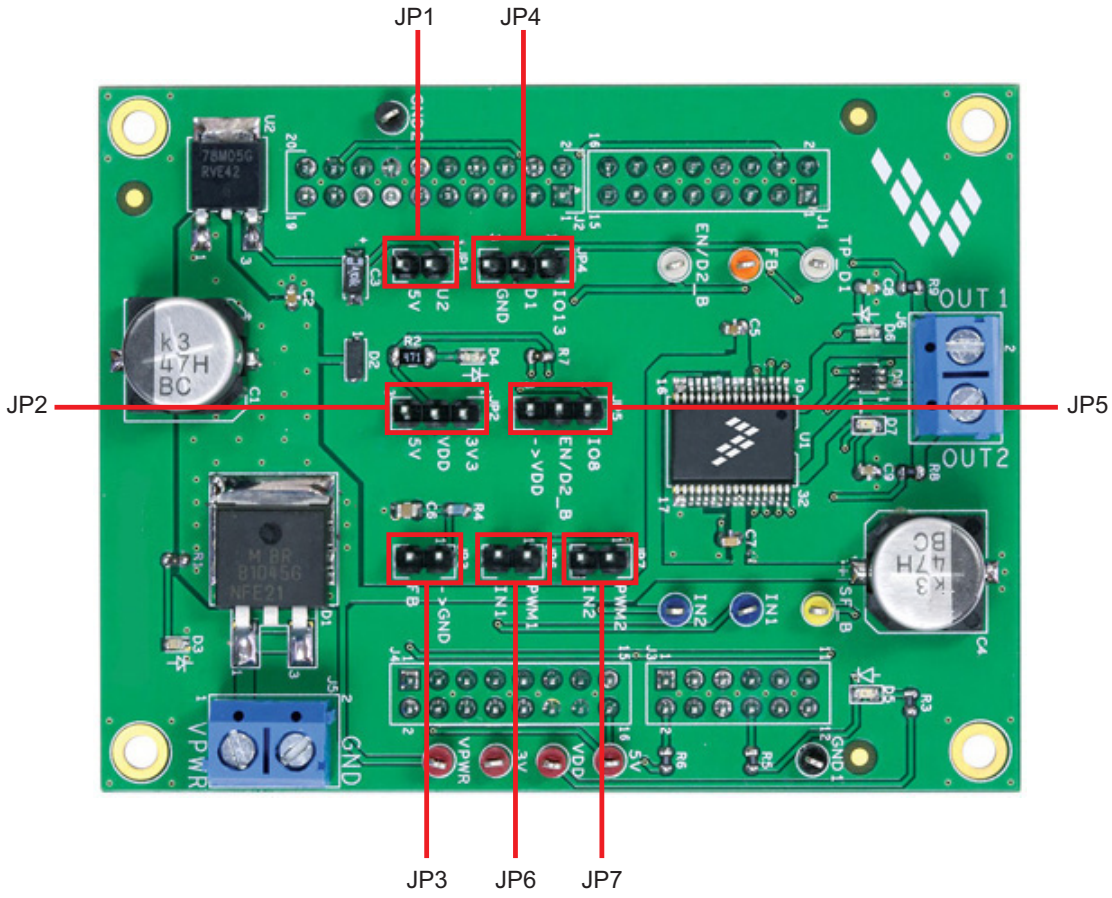

Figure 8. Board Jumpers

The following table defines the evaluation board jumper positions and explains their functions. (The default settings are shown in blue.)

.

Table 4. Jumper Definitions

Jumper	Description	Setting	Connection
JP1	5.0 V Regulator output	**1-2**	5.0 V regulator connected / External or USB 5.0 V
JP2	VDD Select	1-2	3.3 V as VDD
		2-3	5.0 V as VDD
JP3	FB	**1-2**	Feedback to MCU ADC / NC
JP4	D1	**1-2**	MCU GPIO
		2-3	GND
JP5	EN/D2_B	**1-2**	MCU GPIO
		2-3	VDD
JP6	IN1	**1-2**	MCU GPIO / EXT Signal to IN1
JP7	IN2	**1-2**	MCU GPIO / EXT Signal to IN2

3.11 Input Signal Definition

The board has the following input signals which are used to control the outputs or functions inside the circuit.

Table 5. Input Signals

Input Name	Description
D1	Disable signal to tri-state the outputs (Active High)
EN/D2_b	Disable signal to tri-state the output and put the part in sleep mode (Active Low)
IN1	Logic input to control OUT1
IN2	Logic input to control OUT2

3.12 Output Signal Definition

The board has the following output signals which are used to drive a load such as a brushed DC motor. It provides an analog output for real time load current monitoring. This signal allows closed loop control of the load.

Table 6. Output Signals

Output Name	Description
OUT1	Output 1 of H-Bridge controlled by the logic input IN1
OUT2	Output 2 of H-Bridge controlled by the logic input IN2
SF_B	Open drain Active Low status flag output to indicate fault
FB	Current mirror output for real time load current monitoring

3.13 Test Point Definition

Figure 9 shows the location of the test points on the board.

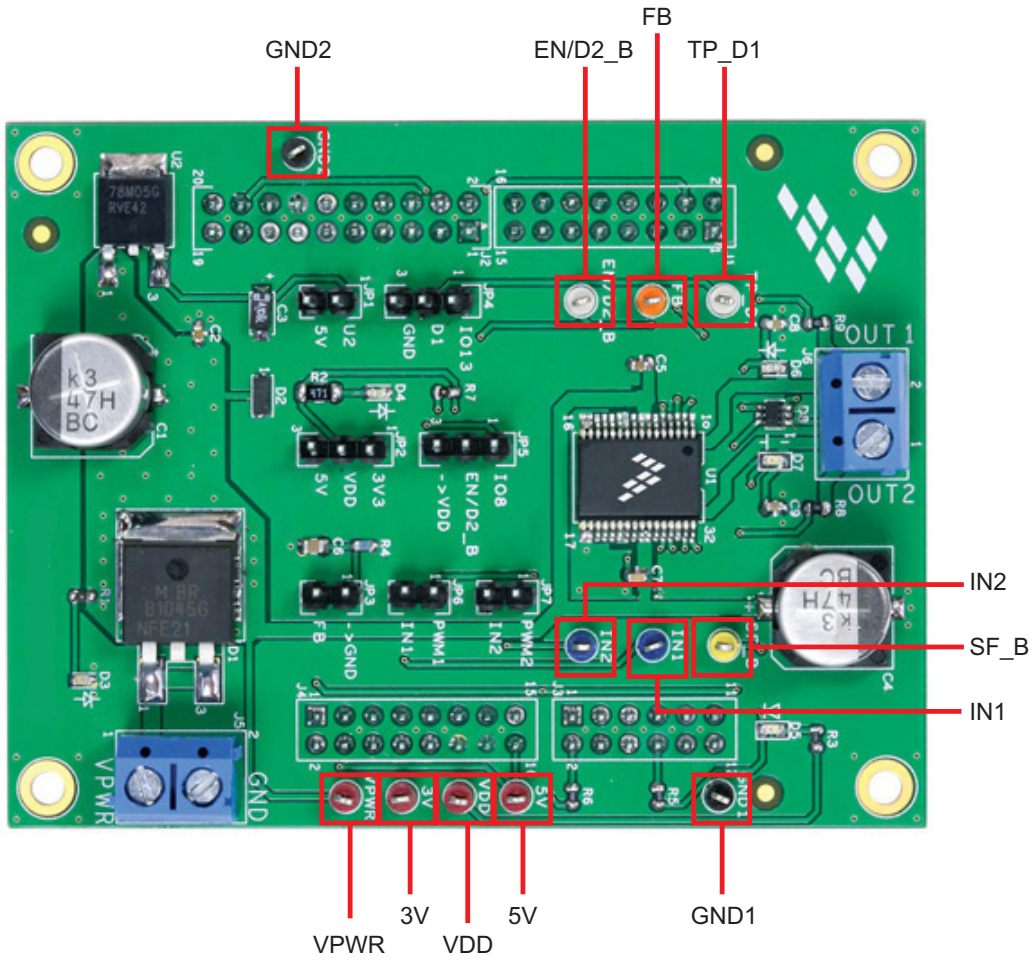

Figure 9. Test Points

The following test points provide access to various signals to and from the board.

Table 7. Test Points

Test Point Name	Signal Name	Description
TP_D1	D1	Disable signal to tri-state the outputs (Active High)
EN/D2_B	EN/D2_b	Disable signal to tri-state the output and put the part in sleep mode (Active Low)
FB	FB	Current mirror output for real time load current monitoring
IN1	IN1	Logic input to control OUT1
IN2	IN2	Logic input to control OUT2
SF_B	SF_b	Open drain Active Low status flag output to indicate fault
GND1	GND	Common Ground
GND2	GND	Common Ground
VPWR	VBAT	Battery or power supply input voltage
5V	5V	5.0 V signal from regulator

Table 7. Test Points (continued)

Test Point Name	Signal Name	Description
3V	3V3	3.3 V supply from the FRDM board
VDD	VDD	VDD supply for the FS_B pull-up resistor

3.14 Screw Terminal Connections

The board has following screw terminal connections to connect the power supply and the load. Figure 10 shows the location of the screw terminal connectors.

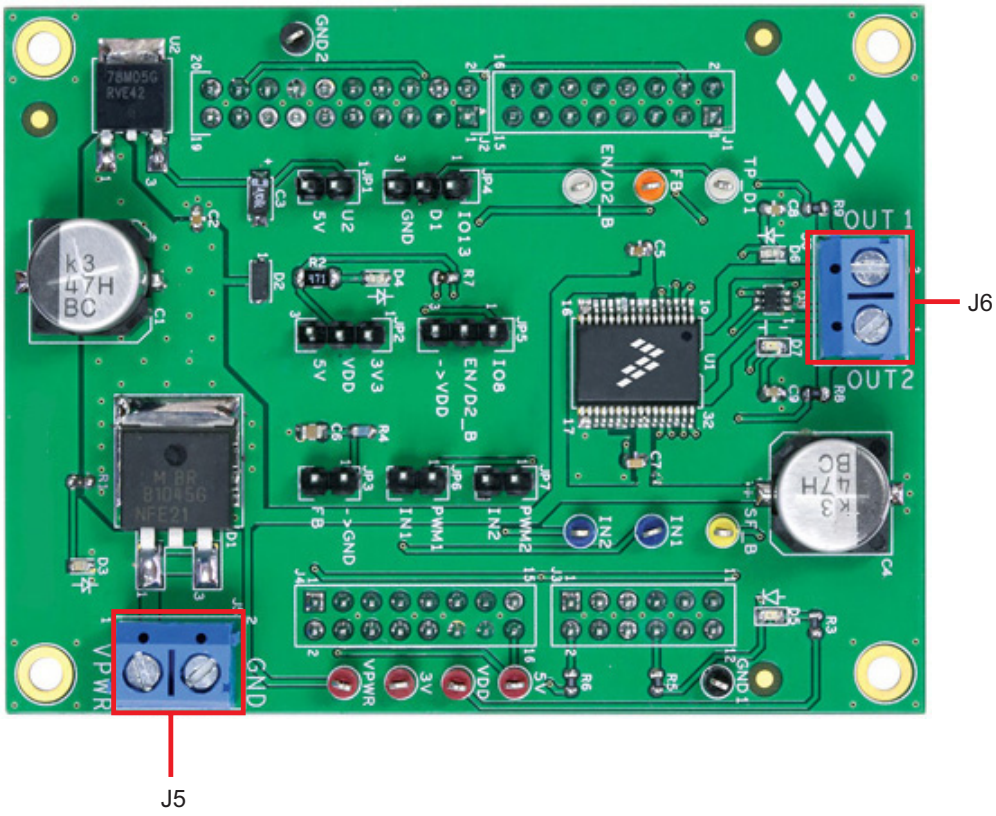

Figure 10. Screw Terminal Connectors

Table 8. Screw Terminal Connections

Screw Terminal Name	Description
J5	Power supply connector for MC34931
J6	Output connector for connecting to a load

4 FRDM-KL25Z Freedom Development Platform

The Freescale Freedom development platform is a set of software and hardware tools facilitating rapid prototyping of designs based on the Kinetis family of microcontrollers. The Freescale Freedom KL25Z board serves as the basic hardware component of the development platform. The FRDM-KL25Z implements a Kinetis L Series microcontroller and makes use of the device's built-in USB, LED, and I/O port features. The board can be loaded with application specific firmware and can be configured with Graphical User Interface software that supports development and testing.

The Freescale FRDM-34931S-EVB / FRDM-34931-EVB may be mounted to the FRDM-KL25Z as a shield board. When used in conjunction with the FRDM-34931S-EVB /FRDM-34931-EVB, the FRDM-KL25Z provides basic functions, such as PC communication, that support the application-specific features of the evaluation board.

For use with the FRDM-34931S-EVB / FRDM34931-EVB, the FRDM-KL25Z must have ARM®mbed™ firmware installed (see Section 5.2.2), MC34931 microcode installed (see Section 5.2.3), and must use the Freescale "GUI Brushed DC FRDM-34931S-EVB" or "GUI Brushed DC FRDM-34931-EVB" as the software interface (see Section 5.2.4).

For complete information on the FRDM-KL25Z, access the documentation available on the FRDM-KL25Z Tool Summary page.

Figure 11 illustrates the primary components of the FRDM-KL25Z which apply when the board is used in conjunction with the evaluation board.

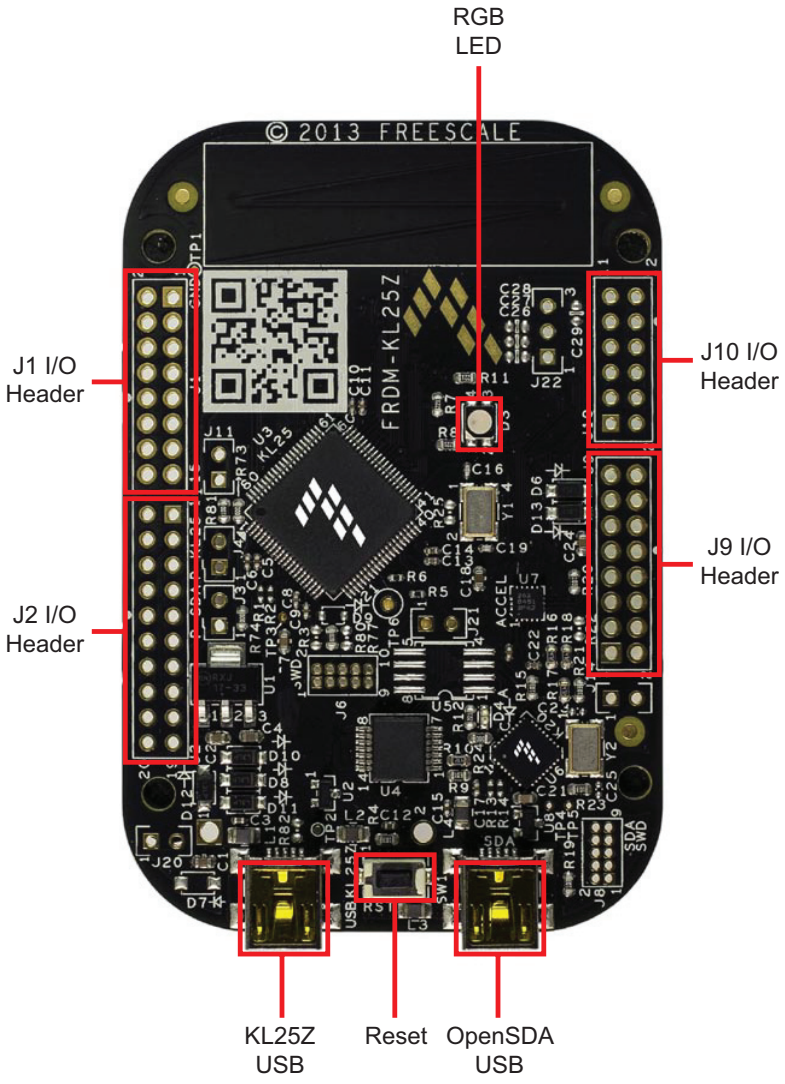

Figure 11. FRDM-KL25Z Board

4.1 Connecting the FRDM-KL25Z to the Evaluation Board

The FRDM-KL25Z provides an ideal support platform for the FRDM-34931S-EVB/FRDM-34931-EVB kit. In this configuration, the FRDM-KL25Z connects to a PC and allows the user—via the GUI—to set parameters that control the operation of the motor. The FRDM-34931S-EVB connects to the FRDM-KL25Z using the four dual row Arduino™ R3 connectors on the bottom of the board. The connections are as follows:

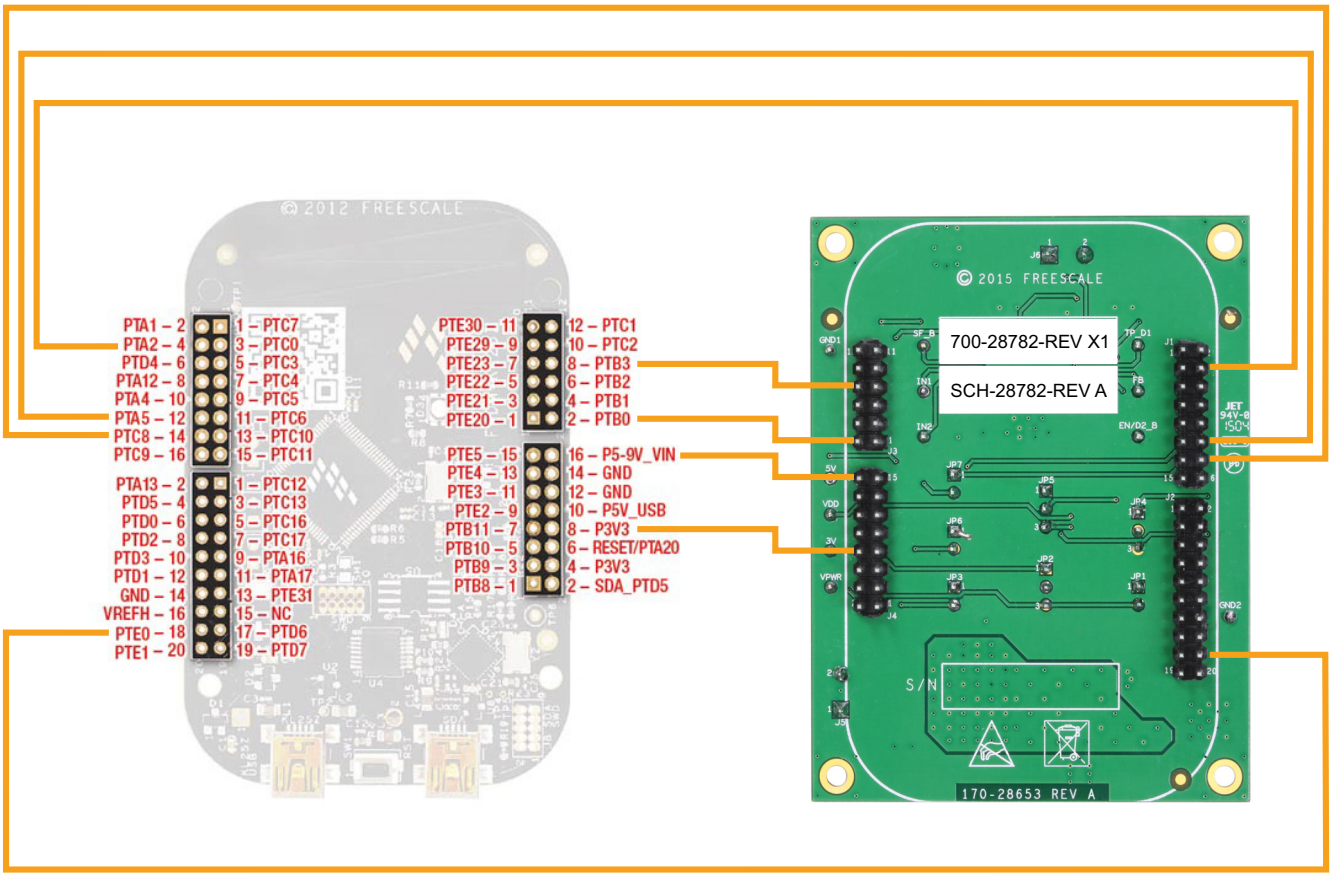

Figure 12. FRDM-KL25Z to FRDM-34931S-EVB / FRDM-34931-EVB Connections

Table 9. FRDM-34932S-EVB / FRDM-34931-EVB to FRDM-KL25Z Connections

FRDM-34931S-EVB/ FRDM-34931-EVB		FRDM-KL25Z		Pin Hardware Name		Description
Header	Pin	Header	Pin	FRDM-34931S-EVB FRDM-34931-EVB	FRDM-KL25Z	
J1	1	J1	1	N/C	PTC7	Not Connected
J1	2	J1	2	N/C	PTA1	Not Connected
J1	3	J1	3	N/C	PTC0	Not Connected
J1	4	J1	4	IO13 (D1)	PTA2	Disable signal to tri-state the outputs (Active High)
J1	5	J1	5	N/C	PTC3	Not Connected
J1	6	J1	6	N/C	PTD4	Not Connected
J1	7	J1	7	N/C	PTC4	Not Connected

Table 9. FRDM-34932S-EVB / FRDM-34931-EVB to FRDM-KL25Z Connections (continued)

FRDM-34931S-EVB/ FRDM-34931-EVB		FRDM-KL25Z		Pin Hardware Name		Description
Header	Pin	Header	Pin	FRDM-34931S-EVB FRDM-34931-EVB	FRDM-KL25Z	
J1	8	J1	8	N/C	PTA12	Not Connected
J1	9	J1	9	N/C	PTC5	Not Connected
J1	10	J1	10	N/C	PTA4	Not Connected
J1	11	J1	11	N/C	PTC6	Not Connected
J1	12	J1	12	PWM1 (IN1)	PTA5	Logic input to control OUT1 using PWM signal
J1	13	J1	13	N/C	PTC10	Not Connected
J1	14	J1	14	PWM1 (IN2)	PTC8	Logic input to control OUT2 using PWM signal
J1	15	J1	15	N/C	PTC11	Not Connected
J1	16	J1	16	N/C	PTC9	Not Connected
J2	1	J2	1	N/C	PTC12	Not Connected
J2	2	J2	2	N/C	PTA13	Not Connected
J2	3	J2	3	N/C	PTC13	Not Connected
J2	4	J2	4	N/C	PTD5	Not Connected
J2	5	J2	5	N/C	PTC16	Not Connected
J2	6	J2	6	N/C	PTD0	Not Connected
J2	7	J2	7	N/C	PTC17	Not Connected
J2	8	J2	8	N/C	PTD2	Not Connected
J2	9	J2	9	N/C	PTA16	Not Connected
J2	10	J2	10	N/C	PTD3	Not Connected
J2	11	J2	11	N/C	PTA17	Not Connected
J2	12	J2	12	N/C	PTD1	Not Connected
J2	13	J2	13	N/C	PTE31	Not Connected
J2	14	J2	14	N/C	GND	Not Connected
J2	15	J2	15	N/C	N/C	Not Connected
J2	16	J2	16	N/C	VREFH	Not Connected
J2	17	J2	17	N/C	PTD6	Not Connected
J2	18	J2	18	IO8 (EN/D2_b)	PTE0	Disable signal to tri-state the output and put the part in Sleep mode (Active Low)
J2	19	J2	19	N/C	PTD7	Not Connected
J2	20	J2	20	N/C	PTE1	Not Connected
J3	1	J10	1	N/C	PTE20	Not Connected
J3	2	J10	2	FB	PTB0	Current mirror output for real time load current monitoring
J3	3	J10	3	N/C	PTE21	Not Connected
J3	4	J10	4	N/C	PTB1	Not Connected

Table 9. FRDM-34932S-EVB / FRDM-34931-EVB to FRDM-KL25Z Connections (continued)

FRDM-34931S-EVB/ FRDM-34931-EVB		FRDM-KL25Z		Pin Hardware Name		Description
Header	Pin	Header	Pin	FRDM-34931S-EVB FRDM-34931-EVB	FRDM-KL25Z	
J3	5	J10	5	N/C	PTE22	Not Connected
J3	6	J10	6	N/C	PTB2	Not Connected
J3	7	J10	7	N/C	PTE23	Not Connected
J3	8	J10	8	SF_B	PTB3	Open drain Active Low status flag output to indicate fault
J3	9	J10	9	N/C	PTE29	Not Connected
J3	10	J10	10	N/C	PTC2	Not Connected
J3	11	J10	11	N/C	PTE30	Not Connected
J3	12	J10	12	N/C	PTC1	Not Connected
J4	1	J9	1	N/C	PTB8	Not Connected
J4	2	J9	2	N/C	SDA_PTD5	Not Connected
J4	3	J9	3	N/C	PTB9	Not Connected
J4	4	J9	4	N/C	P3V3	Not Connected
J4	5	J9	5	N/C	PTB10	Not Connected
J4	6	J9	6	N/C	RESET/PTA20	Not Connected
J4	7	J9	7	N/C	PTB11	Not Connected
J4	8	J9	8	FSD 3V3 OUT	P3V3	3.3 V logic output from FRDM-KL25Z board to FRDM34931S-EVB
J4	9	J9	9	N/C	PTE2	Not Connected
J4	10	J9	10	N/C	P5V_USB	Not Connected
J4	11	J9	11	N/C	PTE3	Not Connected
J4	12	J9	12	GND	GND	Not Connected
J4	13	J9	13	N/C	PTE4	Not Connected
J4	14	J9	14	N/C	GND	Not Connected
J4	15	J9	15	N/C	PTE5	Not Connected
J4	16	J9	16	FSD 5V IN	P5-9V_VIN	5.0 V logic input to FRDM-KL25Z board from FRDM-34931S-EVB

5 Setting up the Hardware and the Graphical User Interface (GUI)

The evaluation board is designed to work in conjunction with Freescale's FRDM-KL25Z board with the PC-based GUI providing direct access to the MC34931S/MC34931 MCU for testing and analysis. Alternatively, the board may be used as a stand-alone component, in which case lab hardware, such as a function generator, must be used to support testing and analysis.

The evaluation board consists of an H-Bridge, a parallel interface, power conditioning circuitry, and a set of two Input Select jumpers. All +5.0 V V_{DD} power required by the board is obtained via the parallel interface.

WARNING

To avoid damaging the board, the following restrictions must be observed:

- The motor supply voltage (V_{PWR}) must be at least 5.0 V, but must not exceed 40 V.
- The peak operating current of the load must not exceed 5.0 A.

5.1 Setting up the FRDM-34931S-EVB/FRDM-34931-EVB as a Stand-alone Component

This section describes how to configure the FRDM-34931S-EVB / FRDM-34931-EVB for use as a stand-alone component. The procedure assumes that you are using a four-channel function generator to do testing and analysis. The same connections apply if the board is connected to a microcontroller instead of a function generator. Consult the board description (Section 3), the schematic (Section 7), and the MC34931S/MC34931 datasheet to determine how best to configure the board for use in your environment.

1. Connect the function generator to the board. There are two options, depending on whether you want to control the enabling and disabling of the MC34931S/MC34931 H-bridge outputs (Option 1), or whether you want the H-bridge outputs continuously enabled (Option 2) while the board is connected to the function generator. Figure 14 illustrates how to set the jumpers and connect to a function generator (or an MCU) for each of these options.
2. With the power switched off, attach the DC power supply to the VPWR and GND screw connector terminals on the evaluation board (J5 in Figure 10).
3. Attach one set of coils of the brushed motor to the OUT 1 and OUT 2 screw connector terminals on the evaluation board (J6 in Figure 10).

Figure 14 illustrates the hardware configuration.

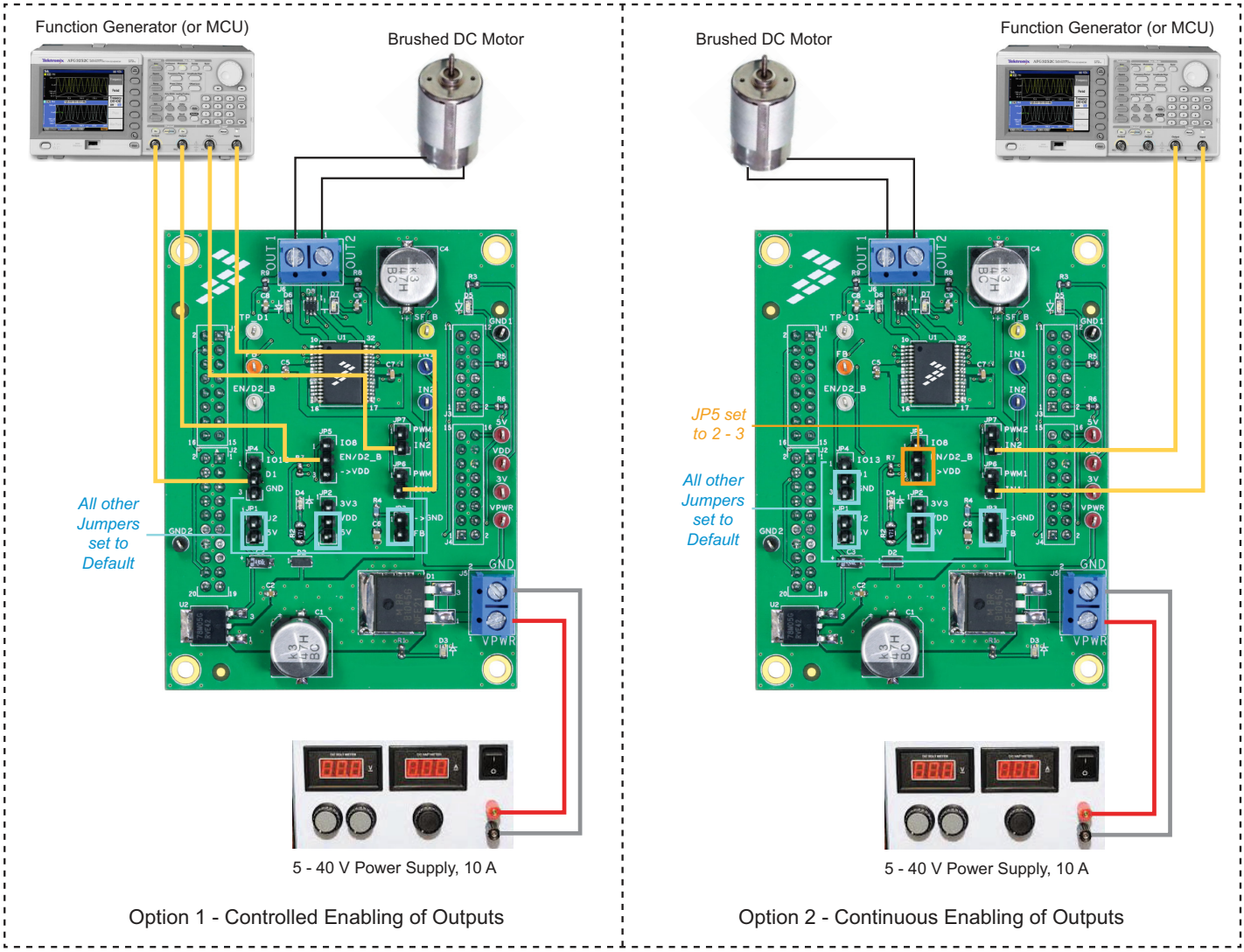

Figure 13. Hardware Configuration - Stand-alone

5.2 Setting up the FRDM-34931S-EVB/FRDM-34931-EVB for Use with the FRDM-KL25Z

To configure the evaluation board for use with the FRDM KL25Z and the Graphical User Interface (GUI) you must:

- Connect the hardware
- Download the mbed firmware to the FRDM-KL25Z board
- Download the MC34931 microcode to the FRDM-KL25Z board
- Install the Graphical User Interface **GUI Brushed DC FRDM-34931S-EVB** for FRDM-34931S-EVB or **GUI Brushed DC FRDM-34931-EVB** for FRDM-34931-EVB

5.2.1 Connecting the Hardware

The FRDM-34931S-EVB / FRDM-34931-EVB consists of an H-Bridge, a parallel interface, power conditioning circuitry, and a set of two Input Select jumpers. All +5.0 V V_{DD} power required by the board is obtained via the parallel interface.

WARNING

To avoid damaging the board, the following restrictions must be observed:

- The motor supply voltage (V_{PWR}) must be at least 5.0 V, but must not exceed 40 V.
- The peak operating current of the load must not exceed 5.0 A.

1. Connect the FRDM-34931S-EVB or FRDM-34931-EVB to the FRDM-KL25Z.
2. With the power switched off, attach the DC power supply to the VPWR and GND screw connector terminals on the evaluation board (J5 in Figure 10).
3. Attach one set of coils of the brushed motor to the OUT 1 and OUT 2 screw connector terminals on the evaluation board (J6 in Figure 10).

Figure 14 illustrates the hardware configuration.

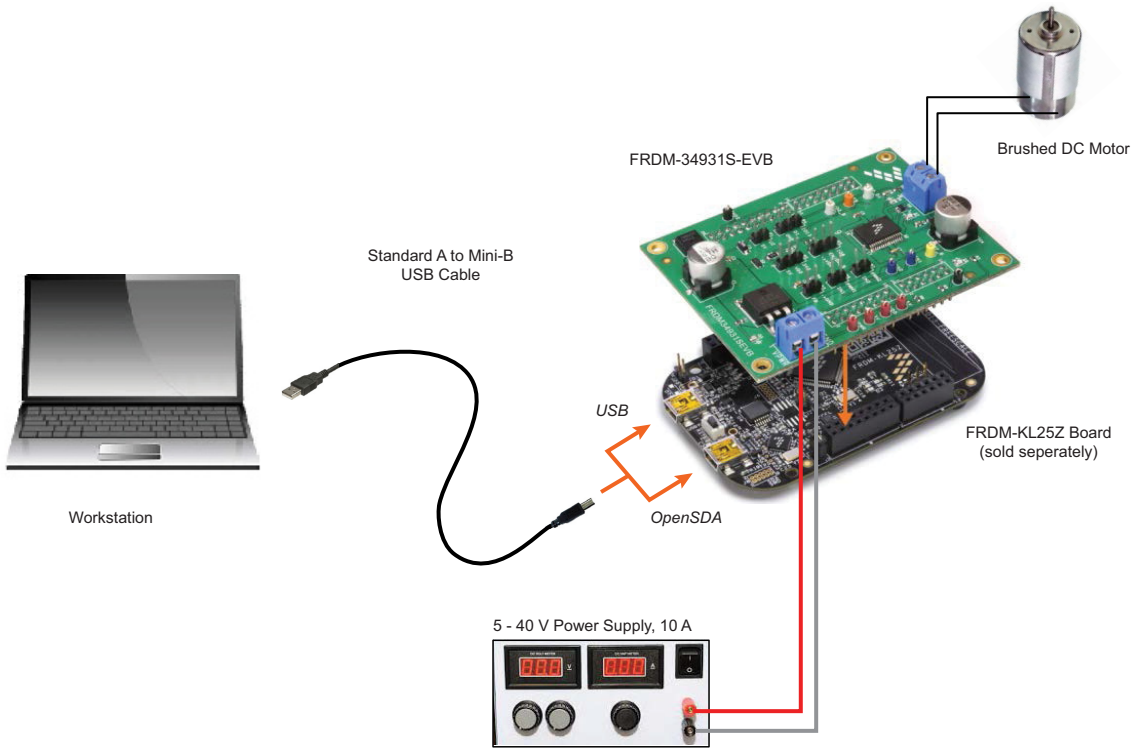

Figure 14. FRDM-34931S-EVB / FRDM-34931-EVB with FRDM-KL25Z Hardware Configuration

5.2.2 Downloading mbed® Firmware to the FRDM-KL25Z Board

You must install mbed® firmware on the FRDM-KL25Z board to enable downloading of the MC34931 microcode. The procedure is as follows:

1. Connect the USB cable between your PC and the OpenSDA USB port on the FRDM-KL25Z board.
2. Download the mbed firmware onto the FRDM-KL25Z board. The instructions are on the ARM®mbed™ website at the following url: https://developer.mbed.org/handbook/Firmware-FRDM-KL25Z
3. After downloading the mbed firmware, power cycle the board (by disconnecting then reconnecting the USB cable to the OpenSDA port) to initiate the firmware update. When this process completes, a USB drive named “mbed” should appear on your PC.

5.2.3 Downloading the MC34931 Microcode to the FRDM-KL25Z Board

The MC34931 microcode provides the firmware interface between the MC34931 device, the Freedom platform and the GUI. The procedure is as follows:

1. Connect the USB cable between your PC and the OpenSDA USB port on the FRDM-KL25Z board.
2. Go to https://developer.mbed.org/teams/Freescale/code/Brushed_DC_Motor_Control_MC34931_MC33931/ and click on the **Import this Program** tab.

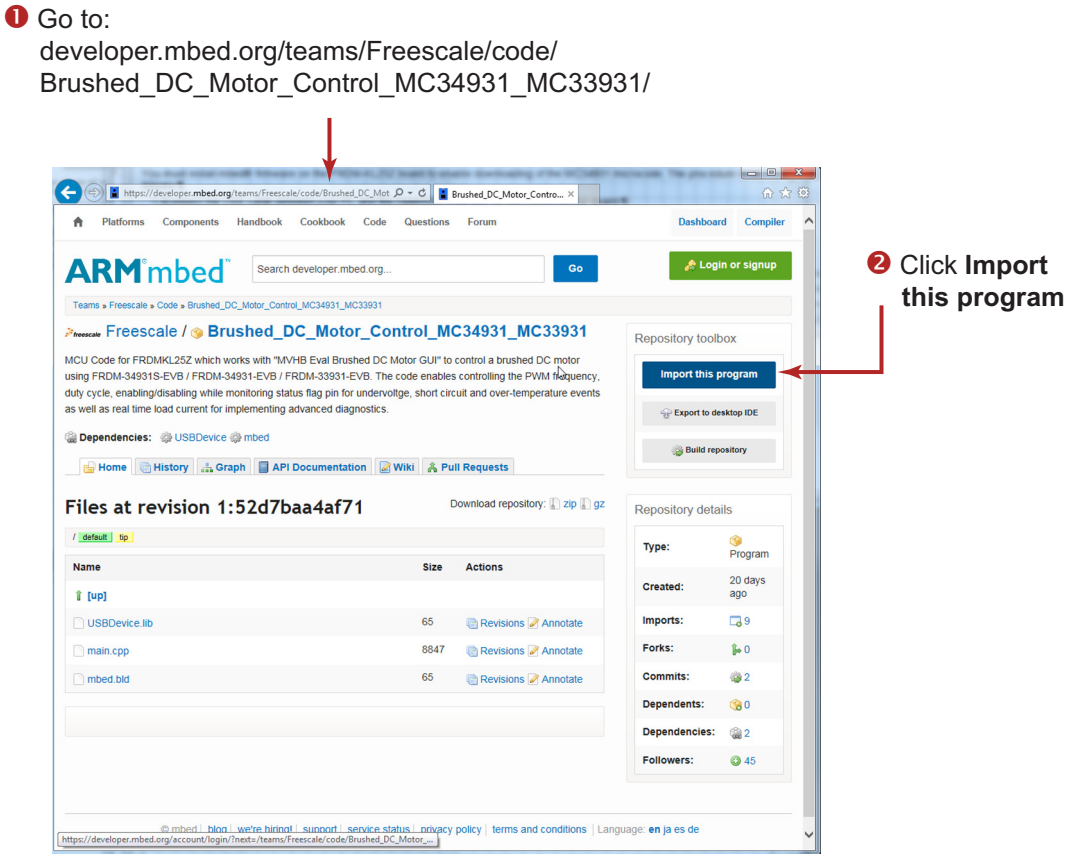

Figure 15. MC34931/MC33931 mbed Import Screen

3. Log into your mbed account. (If you do not have an mbed account, you must create one.) After logging in, you will be returned to the screen in Figure 15. Click on **Import this program** again.

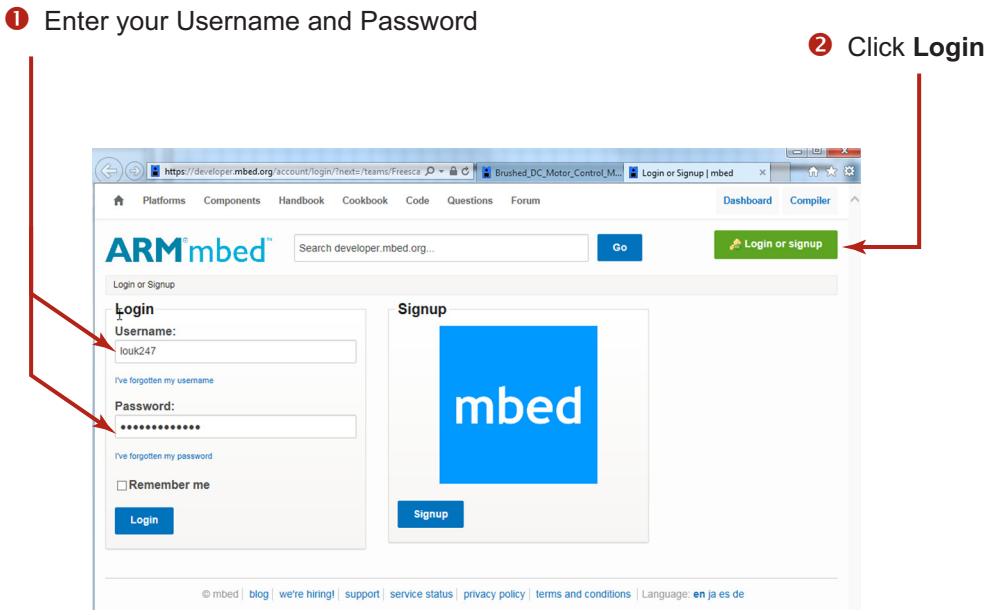

Figure 16. mbed Login Screen

4. The mbed compiler opens with the Import Program window displayed. Click on the **Import** button.

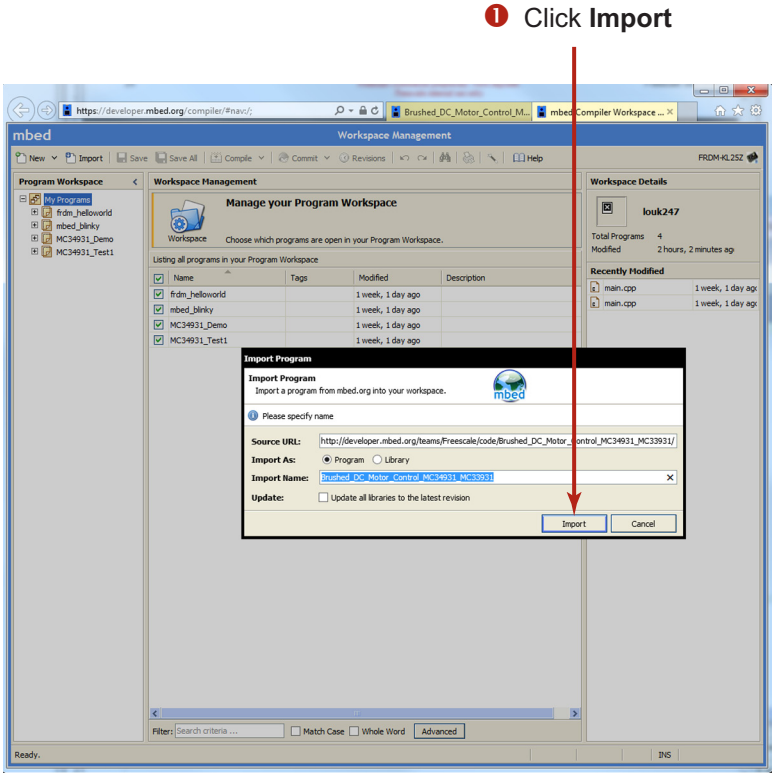

Figure 17. mbed Compiler Import Program Screen

5. When the import completes, the mbed compiler screen should look like Figure 18. Click on the **main.cpp** item.

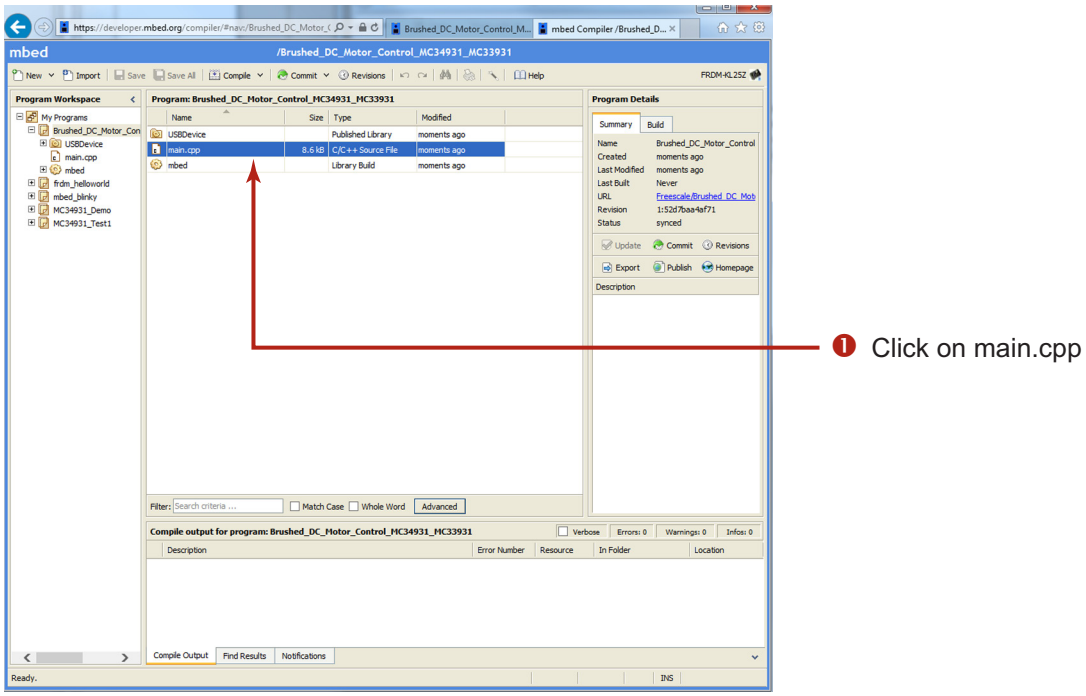

Figure 18. mbed Compiler Select Screen

6. The source code for **main.cpp** appears in the code editor. Click on the **Compile** button to compile the **main.cpp** source code.

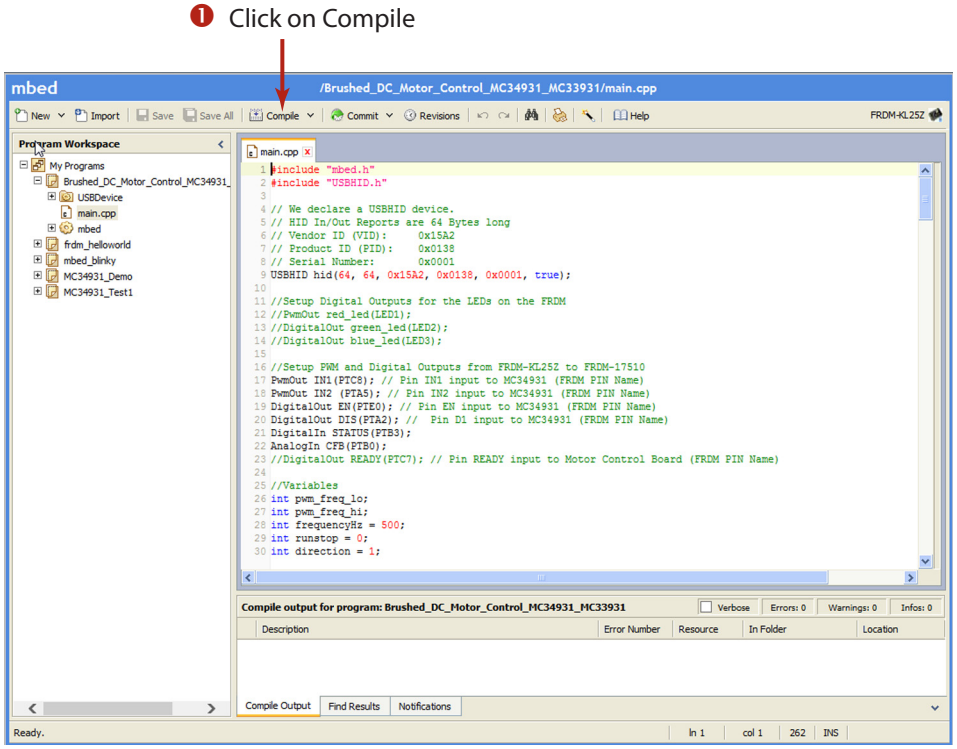

Figure 19. mbed Compiler New Program Screen

7. When the compiler completes, an executable file named **Brushed_DC_Motor_Control_MC34931_MC33931_KL25Z.bin** downloads to your system download folder. Drag and drop this file to the mbed device which appears as a USB drive on your system.

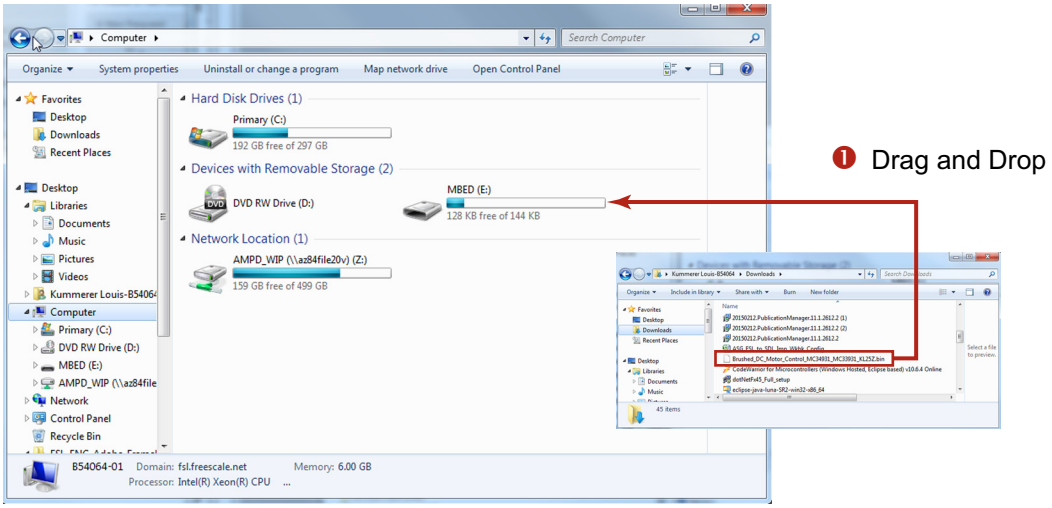

Figure 20. Downloading Brushed_DC_Motor_Control_MC34931_MC33931_KL25Z.bin to FRDM-KL25Z

8. Remove the USB connector from the FRDM-KL25Z OpenSDA USB port and insert it in the KL25Z USB port.

The KL25Z board is now ready for use with the FRDM-34931-EVB/FRDM-34931S-EVB and the Motor Control GUI.

5.2.4 Installing the Graphical User Interface

The Graphical User Interface provides a PC-based interface allowing you to easily exercise FRDM-34931S-EVB/ FRDM-34931-EVB functions to control a DC Brushed Motor. The GUI runs on any Windows 8, Windows 7, Vista, or XP-based operating system.

Two variations of the GUI are available. The GUI Brushed DC FRDM-34931S-EVB GUI (with a maximum PWM frequency of 20 kHz) supports users with the FRDM-34931S-EVB. The GUI Brushed DC FRDM-34931-EVB (with a maximum PWM frequency of 10 kHz) is intended for use with the FRDM-34931-EVB.

To install the software:

1. Go to the evaluation board Tool Summary Page
   - for FRDM-34931-EVB, go to www.freescale.com/FRDM-34931-EVB
   - for FRDM-34931S-EVB, go to www.freescale.com/FRDM-34931S-EVB
2. Under Jump Start Your Design, click on the **Get Started with the FRDM-34931*x*-EVB** link.
3. From the list of files that appear, click on the link for the **GUI Brushed DC FRDM-34931S-EVB** or **GUI Brushed DC FRDM-34931-EVB** software.

The software automatically downloads to your PC and initiates the installation process. An Installation Wizard guides you through the rest of the process.

5.2.5 Using the Graphical User Interface

To start the GUI, do the following:

1. Configure the hardware as described in Section 5.2.1.
2. Click on the Freescale **GUI Brushed DC FRDM-34931S-EVB** icon to launch the GUI.

3. Make sure the GUI recognizes the FRDM-KL25Z. Check the USB connection in the upper left corner of the GUI. The hex Vendor ID value should display as 0x15A2 and the Part ID value should display as 0x138. If these value do not appear, the GUI has failed to establish a connection with the FRDM-KL25Z. You may need to disconnect and reconnect the USB cable to the board's KL25Z USB port. If the connection still fails, press the reset button on the FRDM-KL25Z board.
4. Click the **Enable Target** checkbox on the GUI screen. The Target parameter on the GUI screen should change from DISABLED to ENABLED.
5. Set the **DI**, **EN/D2_B**, **Direction** and **Braking** as desired (See Section 5.2.6 - Section 5.2.9.) Adjust the **PWM Frequency** and **Duty Cycle** to meet your requirements.
6. Click **Run** to run the motor. Notice that some options of the GUI are disabled while the motor is running. To make changes, click **Stop**, make the desired changes, and then click **Run** to continue.
7. When finished, de-select the **Enable Target** button on the GUI, and click **Quit**. Turn off DC power supply and remove the USB cable.

The GUI is shown in Figure 21. The hex address numbers at the top are loaded with the vendor ID for Freescale (0x15A2), and the part ID (0x138). The left side panel displays these numbers only if the PC is communicating with the FRDM-KL25Z via the USB interface.

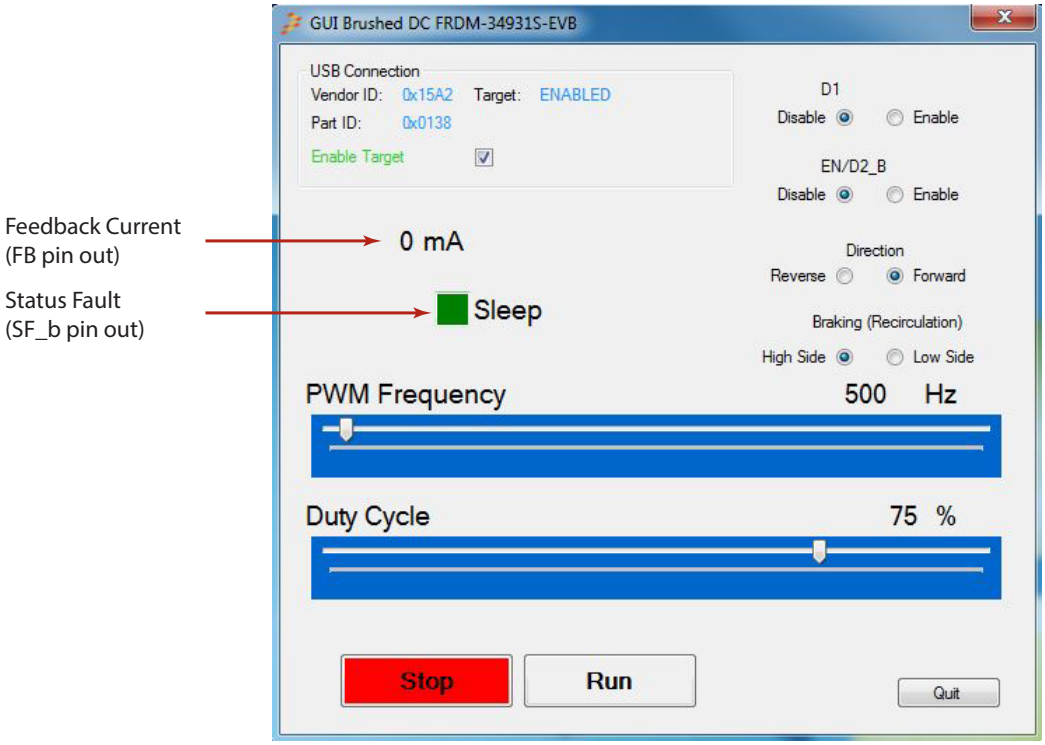

Figure 21. GUI Screen

5.2.6 Forward with High-side Re-circulation

To test the FRDM-34931S-EVB/FRDM-34931-EVB in the forward with high-side re-circulation mode, configure the GUI as follows:

D1: Enable

EN/D2_B: Enable

Direction: Forward

Braking: High-side

Figure 22 shows this configuration with the motor running.

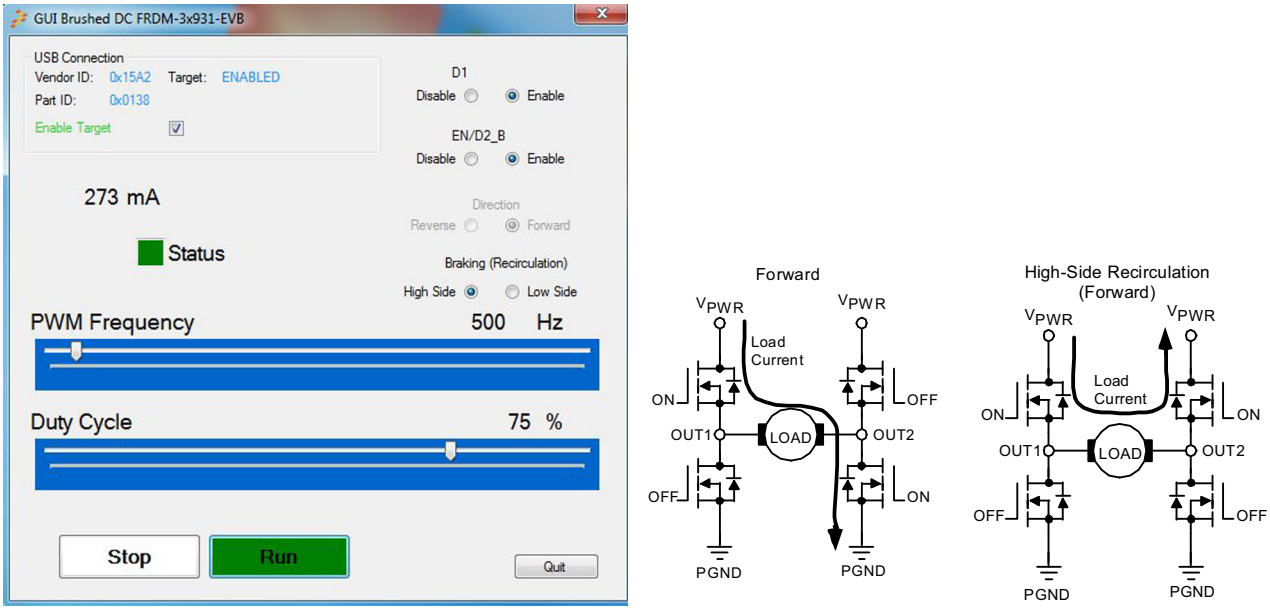

Figure 22. Forward with High-side Recirculation

5.2.7 Forward with Low-side Recirculation

To test the FRDM-34931S-EVB/FRDM-34931-EVB in the forward with low-side re-circulation mode, configure the GUI as follows:

D1: Enable

EN/D2_B: Enable

Direction: Forward

Braking: Low-side

Figure 23 shows this configuration with the motor running.

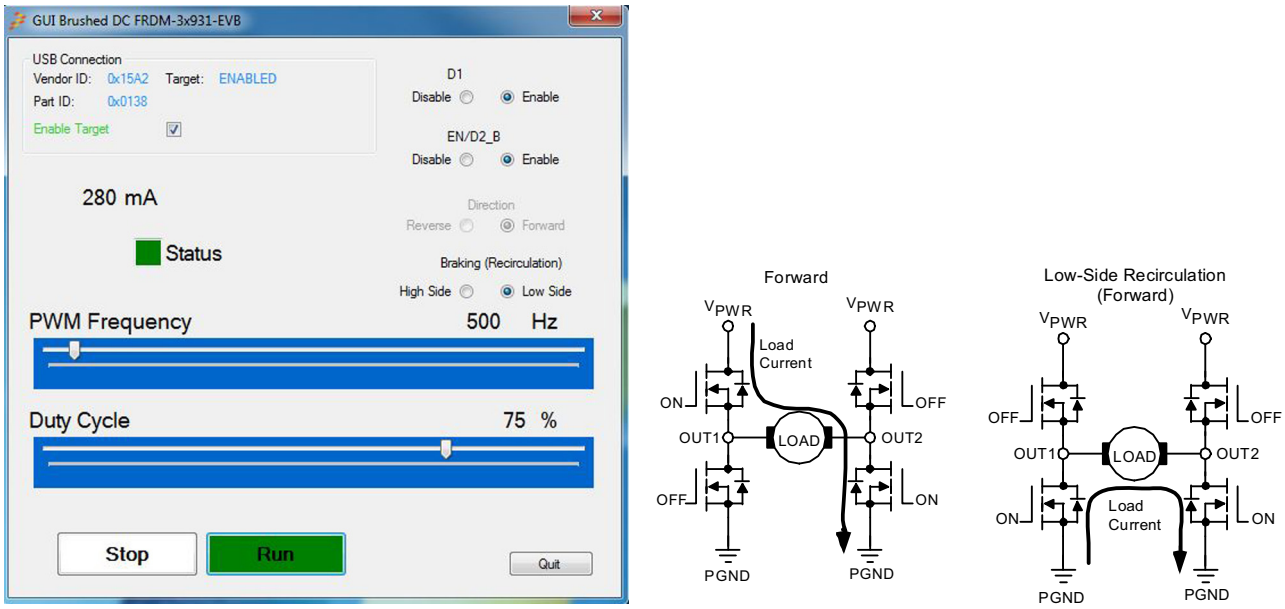

Figure 23. Forward with Low-side Re-circulation

5.2.8 Reverse with High-side Recirculation

To test the FRDM-34931S-EVB/FRDM-34931-EVB in the reverse with high-side re-circulation mode, configure the GUI as follows:

D1: Enable

EN/D2_B: Enable

Direction: Reverse

Braking: High-side

Figure 24 shows this configuration with the motor running.

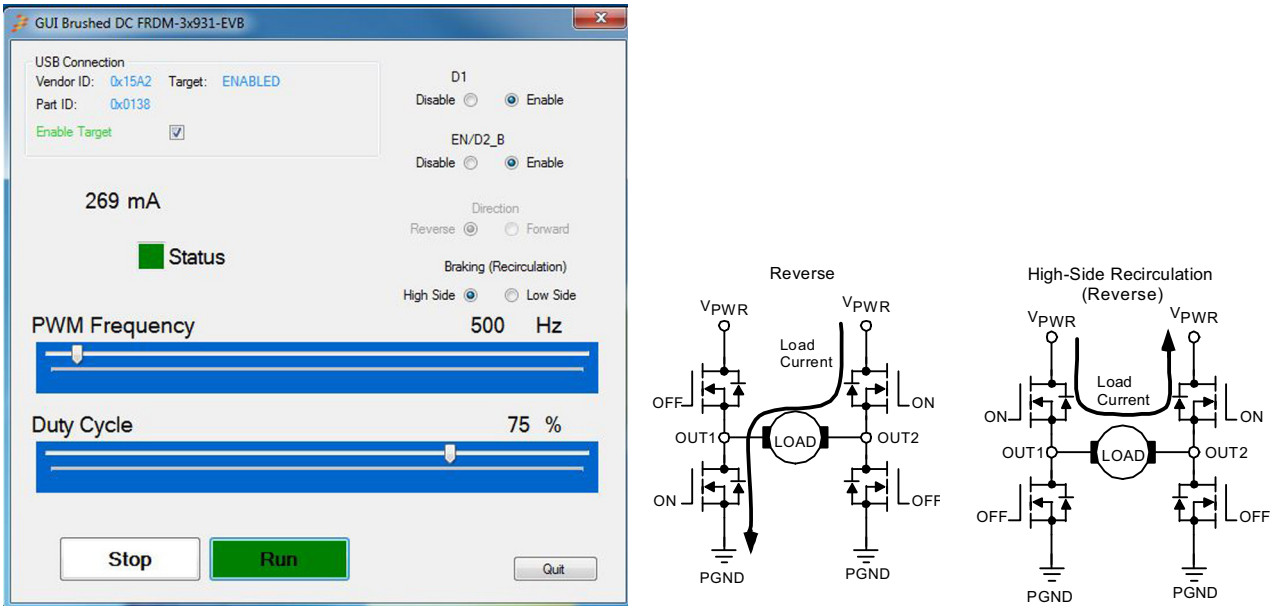

Figure 24. Reverse with High-side Re-circulation

5.2.9 Reverse with Low-side Recirculation

To test the FRDM-34931S-EVB/FRDM-34931-EVB in the reverse with low-side re-circulation mode, configure the GUI as follows:

D1: Enable

EN/D2_B: Enable

Direction: Reverse

Braking: Low-side

Figure 25 shows this configuration with the motor running.

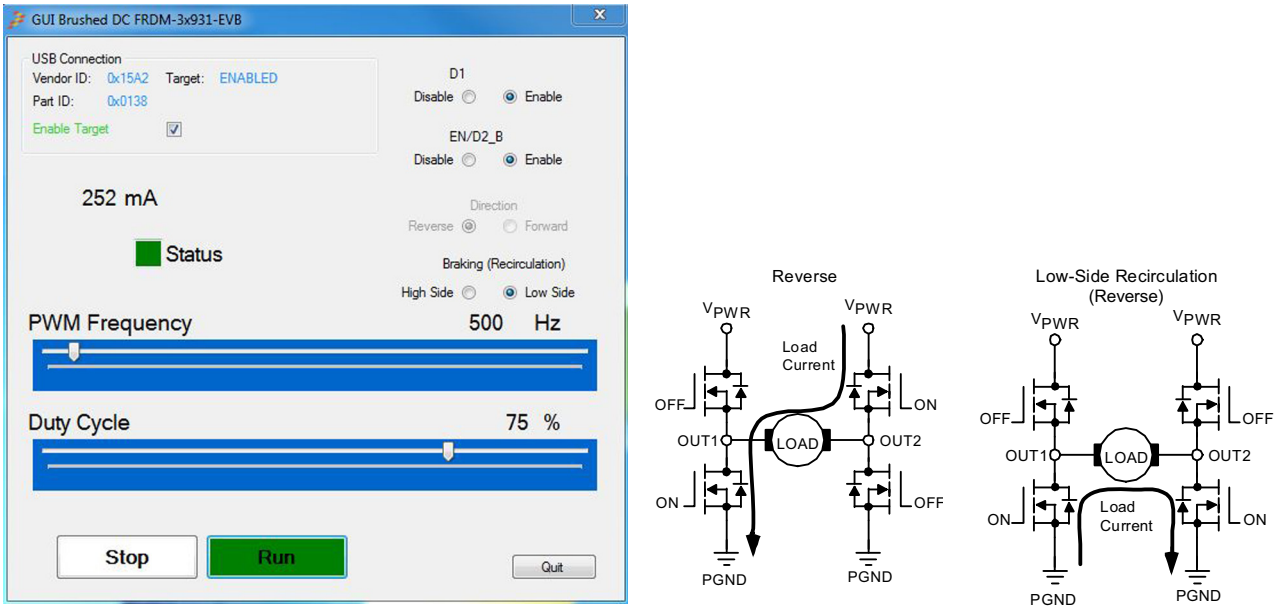

Figure 25. Reverse with Low-side Re-circulation

5.2.10 Direction Control with High-side vs. Low-side Recirculation

Table 10 illustrates the login behind direction control with high-side versus low-side recirculation.

Table 10. H-Bridge Operation Logic

1	Forward - High-side re-circulation
	IN1 = 1 IN2 = PWM signal with selected duty cycle and frequency
2	Reverse - High-side re-circulation
	IN1 = 0 IN2 = PWM signal with selected duty cycle and frequency
3	Forward - Low-side re-circulation
	IN1 = PWM signal with selected duty cycle frequency IN2 = 0
4	Reverse - Low-side re-circulation
	IN1 = 0 IN2 = PWM signal with selected duty cycle

6 Installing Processor Expert Software

6.1 Installing CodeWarrior on your Computer

This procedure explains how to obtain and install the latest version of CodeWarrior (version 10.6 in this guide).

NOTE

The sample software in this kit requires CodeWarrior 10.6 or newer. The component and some examples in the component package are intended for Kinetis Design Studio 3.0.0. If you have CodeWarrior 10.6 and Kinetis Design Studio 3.0.0 already installed on your system, skip this section.

1. Obtain the latest CodeWarrior installer file from the Freescale CodeWarrior website here: www.freescale.com/webapp/sps/site/homepage.jsp?code=CW_HOME&tid=vanCODEWARRIOR.
2. Run the executable file and follow the instructions.
3. In the **Choose Components** window, select the Kinetis component and click on **Next** to complete the installation.

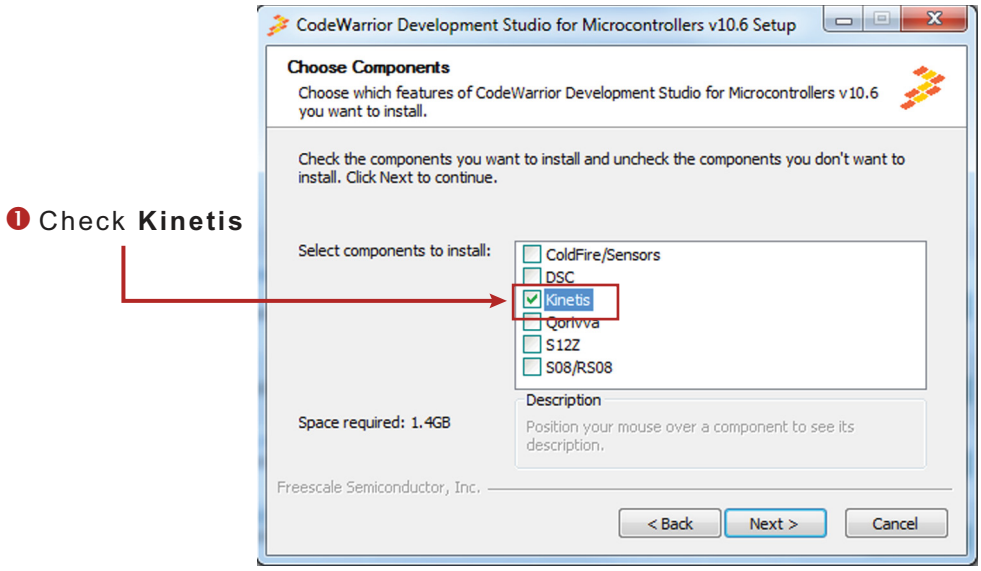

Figure 26. Select Components GUI

6.2 Downloading the MVHBridge Component and Example Projects

The examples used in this section are based on a pre-configured CodeWarrior project. You must first download the project and its associated components:

1. Go to the Freescale website www.freescale.com/MVHBRIDGE-PEXPERT
2. Download example projects and H-Bridge component zip file.
3. Unzip the downloaded file and check that the folder contains the files listed in Table 11.

Table 11: MVHBridge Example Project and Components

Folder Name	Folder Contents
CodeWarrior_Examples	Example project folder for CodeWarrior.
MVH_K20D72M_brushed	Example project for DC brush motor control.
MVH_K20D72M_brushed_FreeMaster	Example project intended for control of brushed motor using FreeMaster tool. Latest Freemaster installation package: www.freescale.com/freemaster
MVH_K20D72M_step_FreeMaster	Example project intended for control of stepper motor using FreeMaster tool.
MVH_K20D72M_stepper	Example project for stepper motor control using full-stepping and micro-stepping mode.
MVH_K20D72M_stepper_fullstep	Example project for stepper motor control demonstrating full-step mode.
MVH_K20D72M_stepper_ramp	Example project for stepper motor control demonstrating acceleration and deceleration ramp.
MVH_K64F120M_brushed_2component	Example project for DC brush motor control using two H-Bridges (i.e. MC33932 and MC33926).
MVH_K70F120M_brushed	Example project for TWR-K70F120M with DC brushed motor control.
MVH_K70F120M_stepper	Example project for TWR-K70F120M with stepper motor control using full-stepping and micro-stepping mode.
MVH_KL25Z48M_brushed_2component	Example project for DC brushed motor control using a dual H-Bridge devce (e.g. MC33932 and 33926).
MVH_KL25Z48M_fullstep_ramp	Example project for stepper motor control demonstrates acceleration and deceleration ramp in full-step mode.
Component	Processor Expert component folder.
DriverSuite_Examples	Example project folder for Driver Suite.
MVH_K20D72M_stepper	Example project for stepper motor control uses full-stepping and micro-stepping mode.
KDS_Examples	Example project folder for Kinetis Design Studio.
MVH_K20D72M_stepper	Example project for stepper motor control, which uses full-stepping and micro-stepping mode.
MVH_K20D72M_stepper_ramp	Example project for stepper motor control demonstrating usage of acceleration and deceleration ramp.
FRDM34931SEVB_Examples	Example project folder for CodeWarrior and H-Bridge board FRDM-34931SEVB
MVH_KL25Z_brushed	Example project for DC brush motor control.

6.2.1 Importing the MVHBridge Component into the Processor Expert Library

1. Launch CodeWarrior by clicking on the CodeWarrior icon (located on your desktop or in **Program Files** -> **Freescale Codewarrior** folder.)
2. When the CodeWarrior IDE opens, go to the menu bar and click **Processor Expert** -> **Import Component(s)**.
3. In the pop-up window, locate the component file (.PEupd) in the example project folder MVHBridge_PEx_SW\Component. Select **MVHBridge_b**xxx**.PEupd** and **ChannelAllocator_b**xxx**.PEupd** files then click **Open** (see Figure 27).

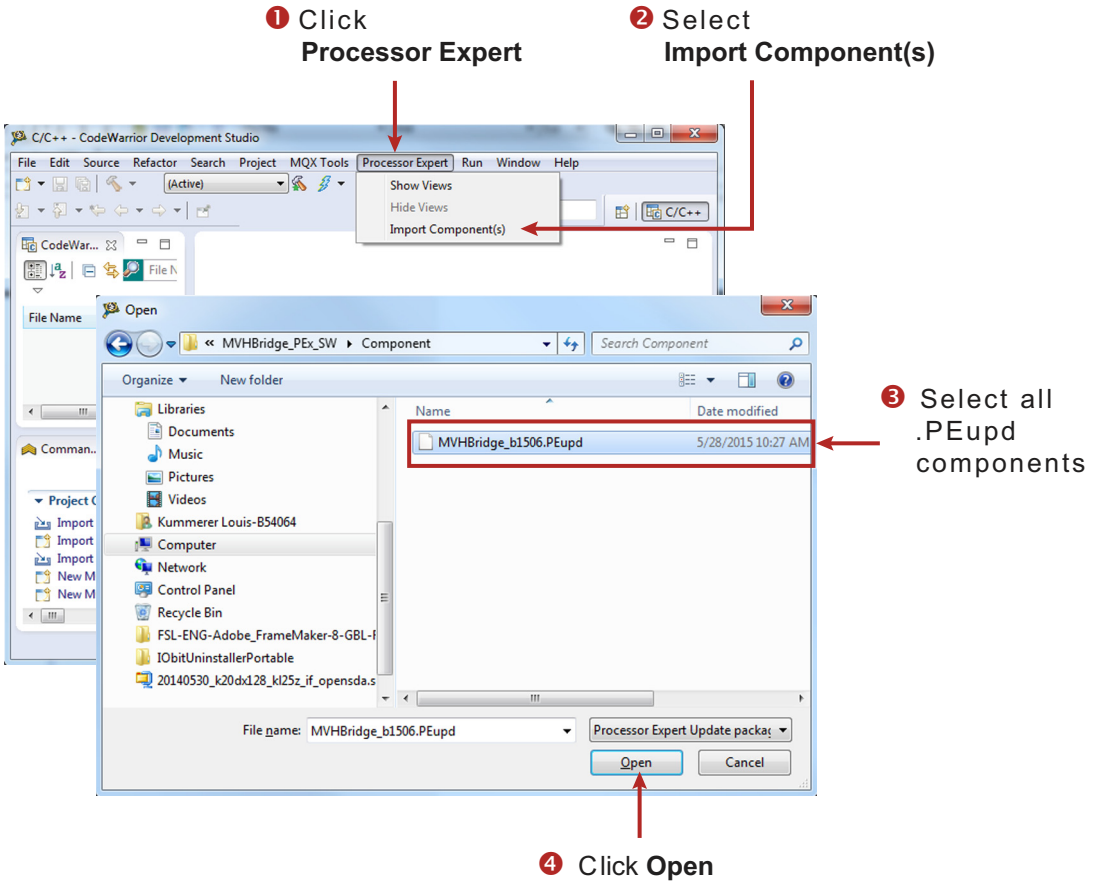

Figure 27. Import MVHBridge Component

4. If the import is successful, the MVHBridge component appears in **Components Library** -> **SW** -> **User Component** (see Figure 28). The MVHBridge component is ready to use.

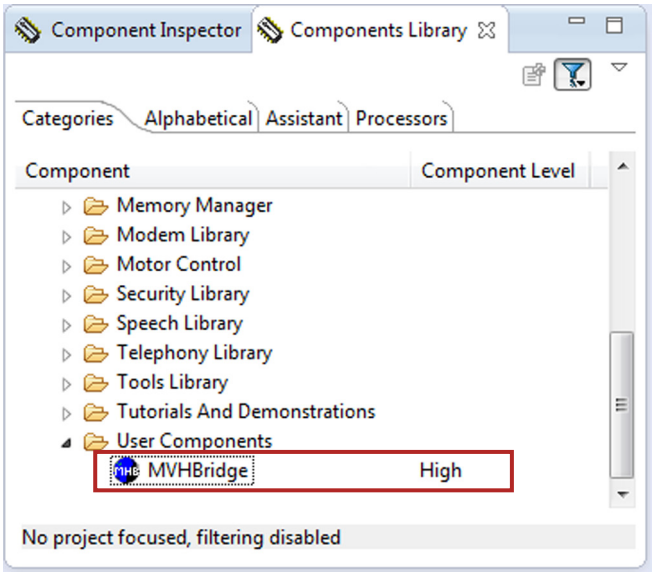

Figure 28. MVHBridge Component Location after CodeWarrior Import

6.2.2 Importing an Example Project into the Processor Expert Library

The following steps show how to import an example from the downloaded zip file into CodeWarrior.

1. In the CodeWarrior menu bar, click **File** -> **Import**… In the pop-up window, select **General** -> **Existing Projects into Workspace** and click **Next**.

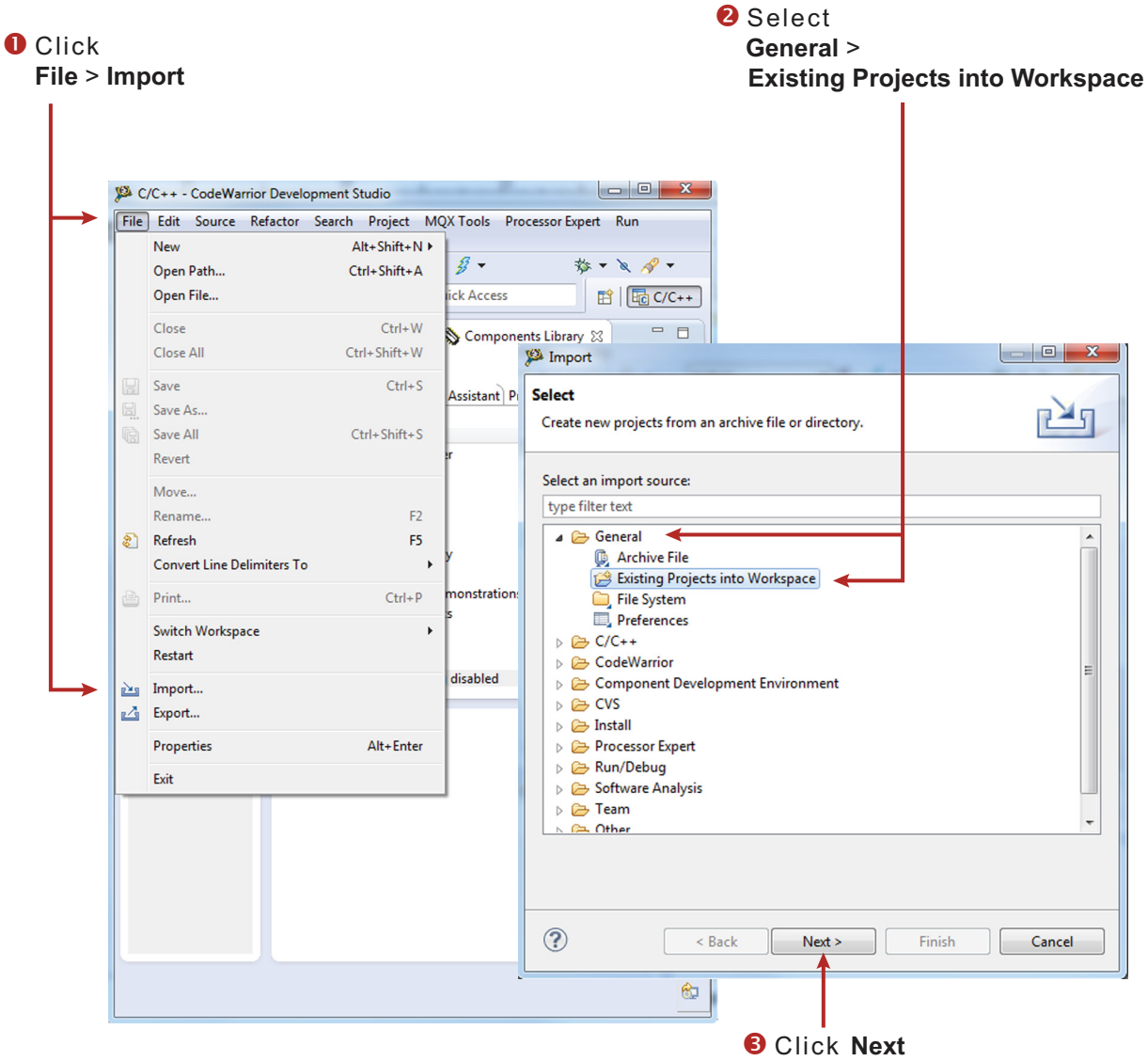

Figure 29. Importing an example file (a)

2. Click **Browse** and locate the folder where you unzipped the downloaded example files. Find the folder MVHBridge_PEx_SW\CodeWarrior_Examples and select a project to import. (see Figure 30, which shows MVH_K20D72M_step_FreeMaster as the imported project). Then click **OK**.

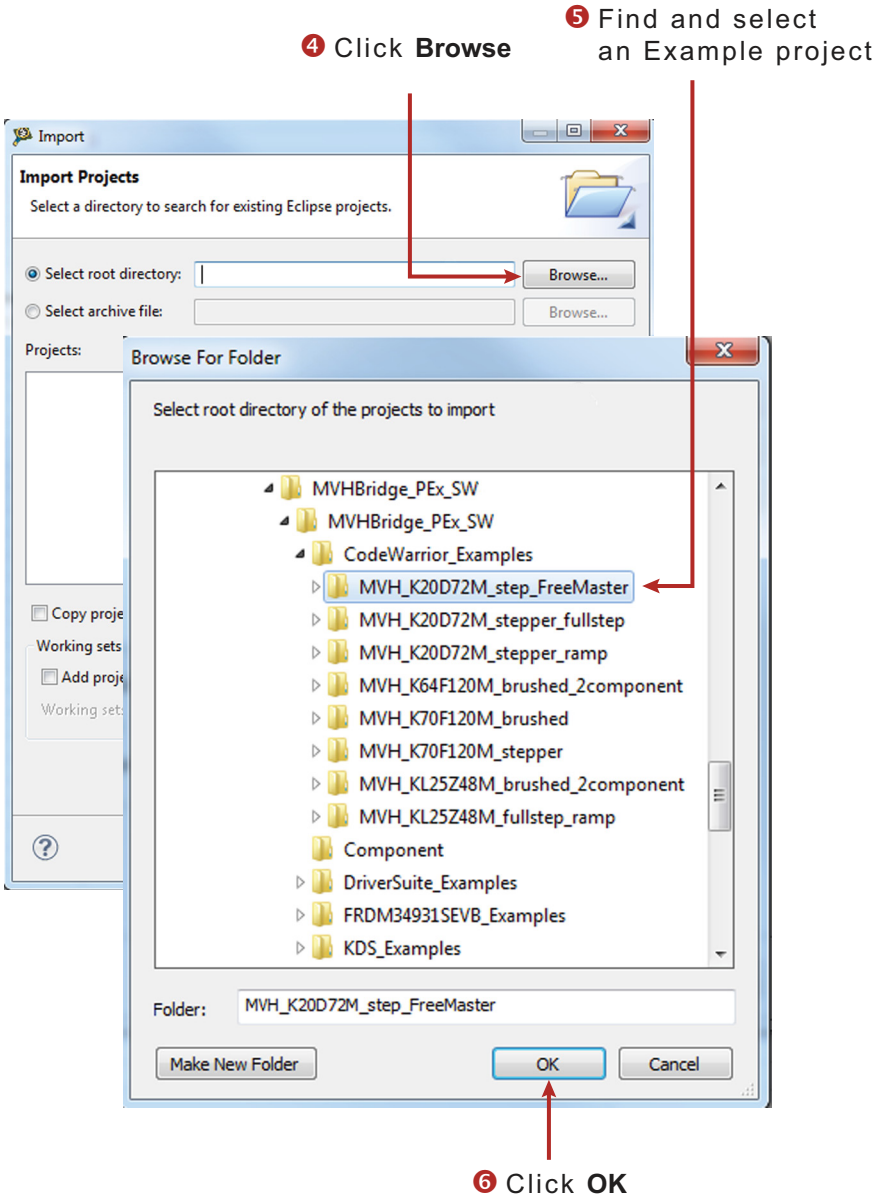

Figure 30. Importing an example file (b)

3. With your project now loaded in the **Select root directory** box, click on the **Copy projects into workspace** checkbox. Then click **Finish**. Figure 31 shows the CodeWarrior **Projects** panel and the **Components** panel after the project has been successfully imported.

The project is now in the CodeWarrior workspace where you can build and run it.

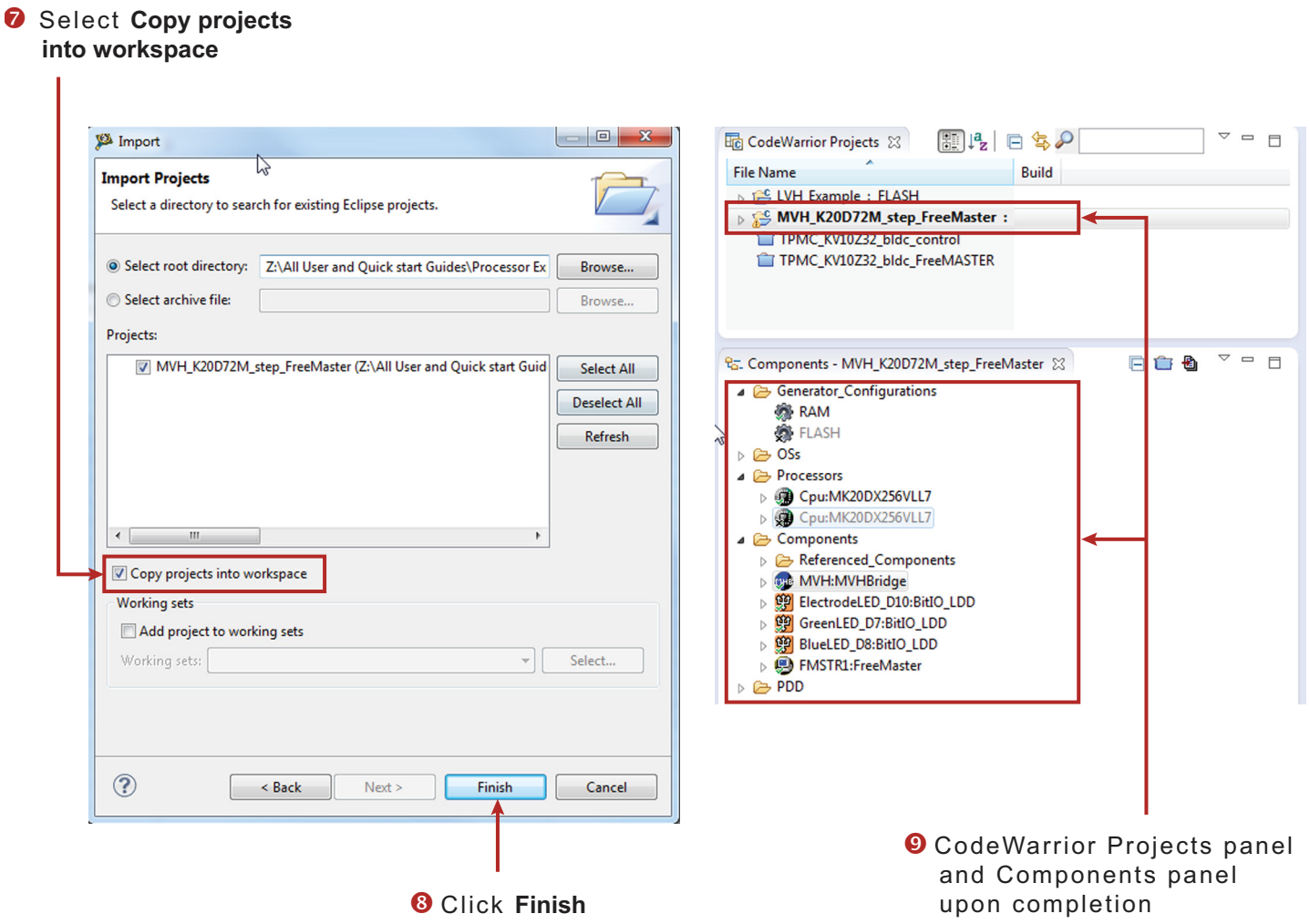

Figure 31. Importing an example file (c)

6.3 Creating a New Project with Processor Expert and the MVHBridge Component

If you choose not to use the example project, the following instructions describe how to create and setup a new project that uses the MVHBridge component. If you do not have the MVHBridge component in the Processor Expert Library, please follow steps in Section 6.2.1.

To creat a new project do the following:

1. In the CodeWarrior menu bar, select **File** -> **New** -> **Bareboard Project**. When the **New Bareboard Project** dialog box opens, enter a project name into the text box and then click **Next**. (see Figure 32).

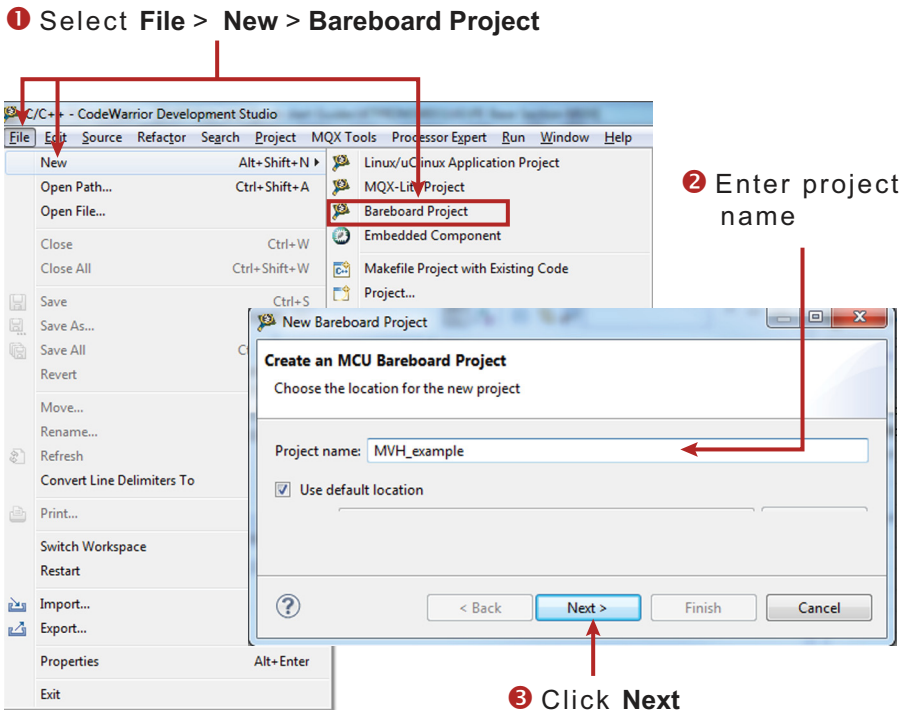

Figure 32. Creating an MCU Bare-board Project

2. In the **Devices** dialog box, select the MCU class your project is using in the MCU board (In Figure 33, MK20DX256 has been selected). Then click **Next**.
3. In the **Connections** dialog box, select the type of connection your project uses. (In Figure 33 **P&E USB Multilink Universal [FX]/USB MultiLink** has been selected). Then click **Next**.

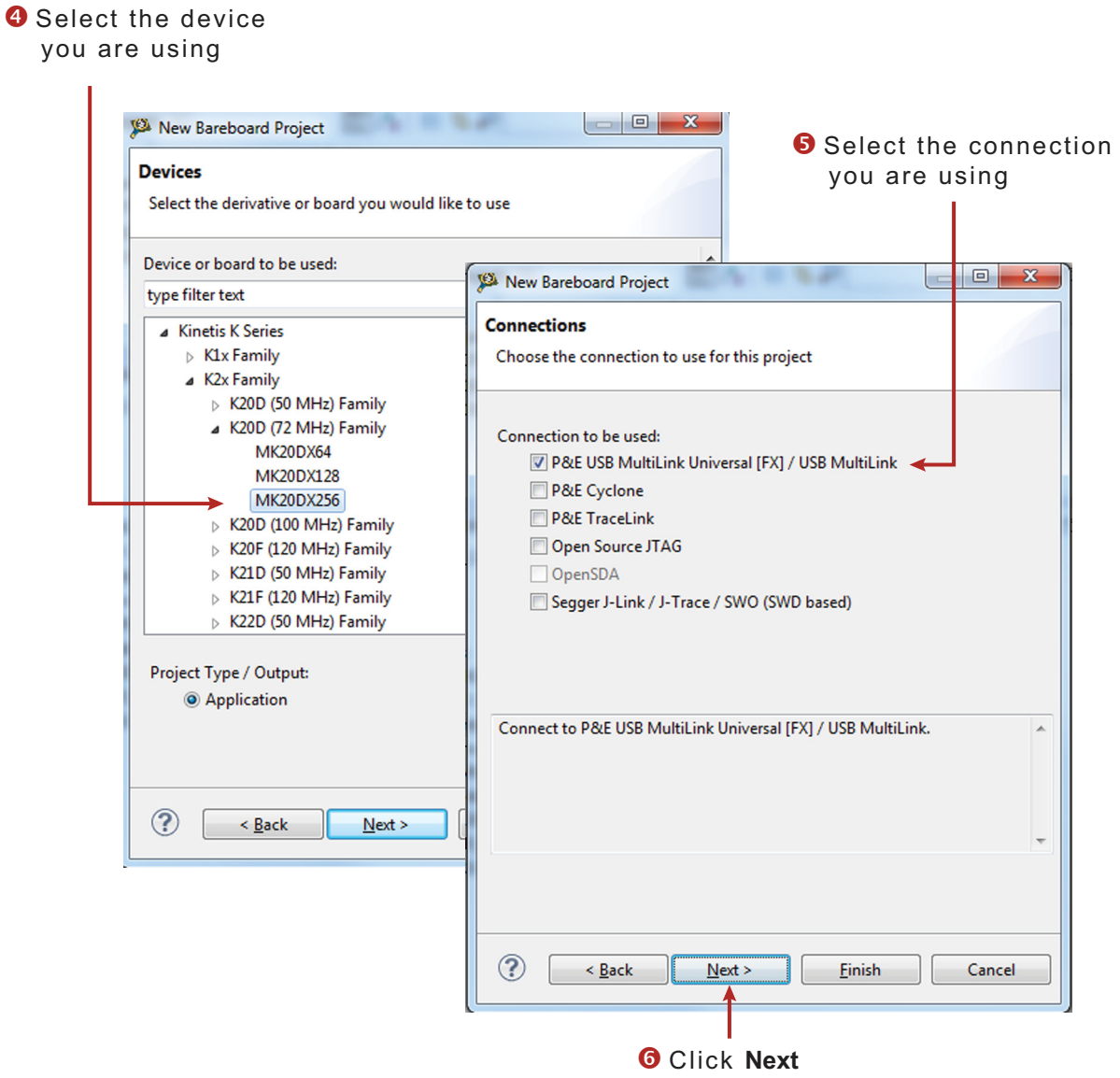

Figure 33. Selecting a device and a connection

4. In the **Language and Build Tools Options** dialog box, select the options that apply to your project. (In Figure 34, the default options are selected.) Then click **Next**.
5. In the **Rapid Application Development** dialog box, make sure that the **Processor Expert** button is selected. Then click **Finish**

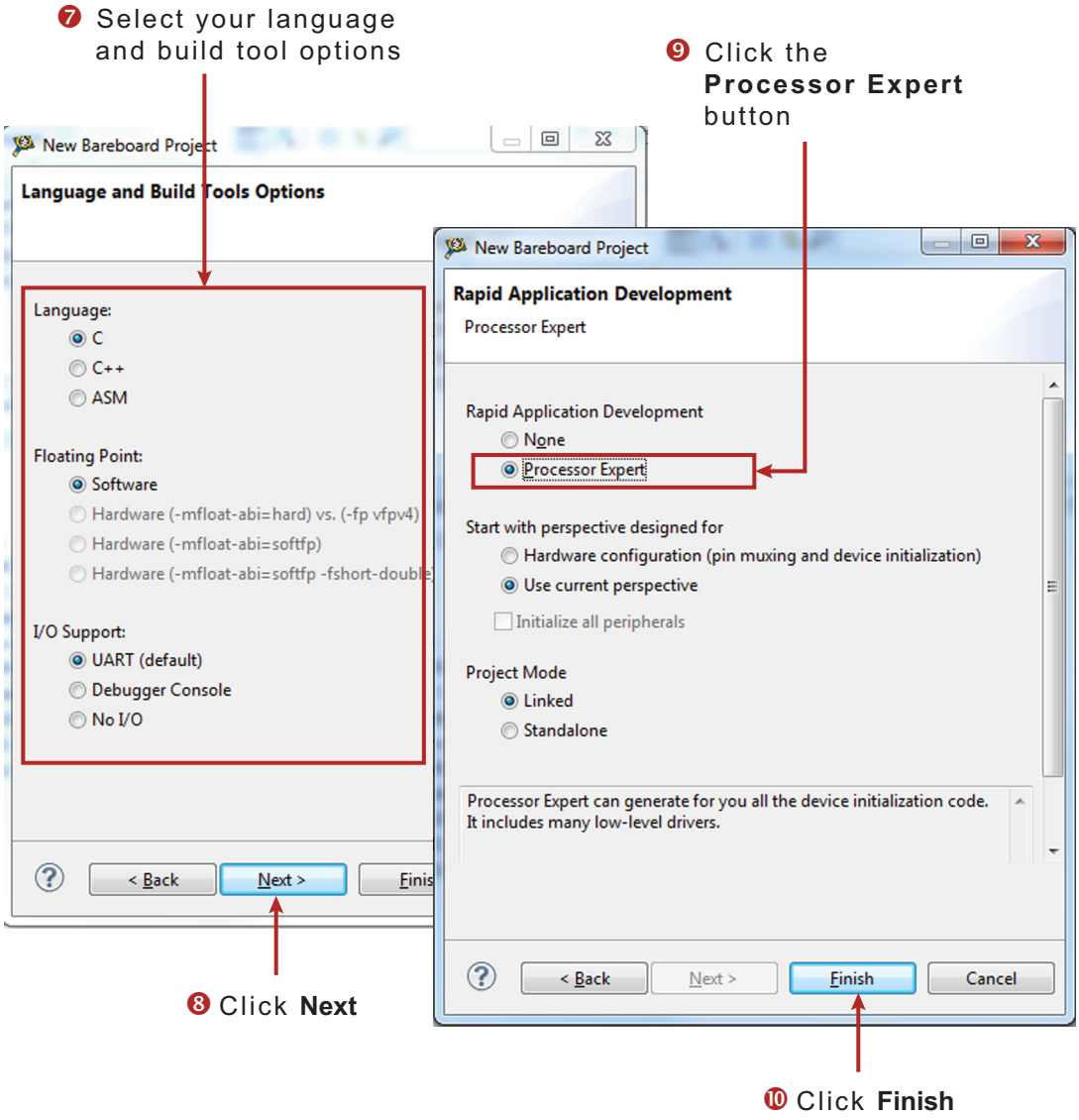

Figure 34. Selecting the language, build tools, and the rapid application development options

6. Figure 35 shows the CodeWarrior Projects panel and the Components panel after the project has been successfully created.

Before you can build and run your project, you must add the MVHBridge component (imported in Section 6.2.1) into your project. Section 6.3.1 outlines this procedure.

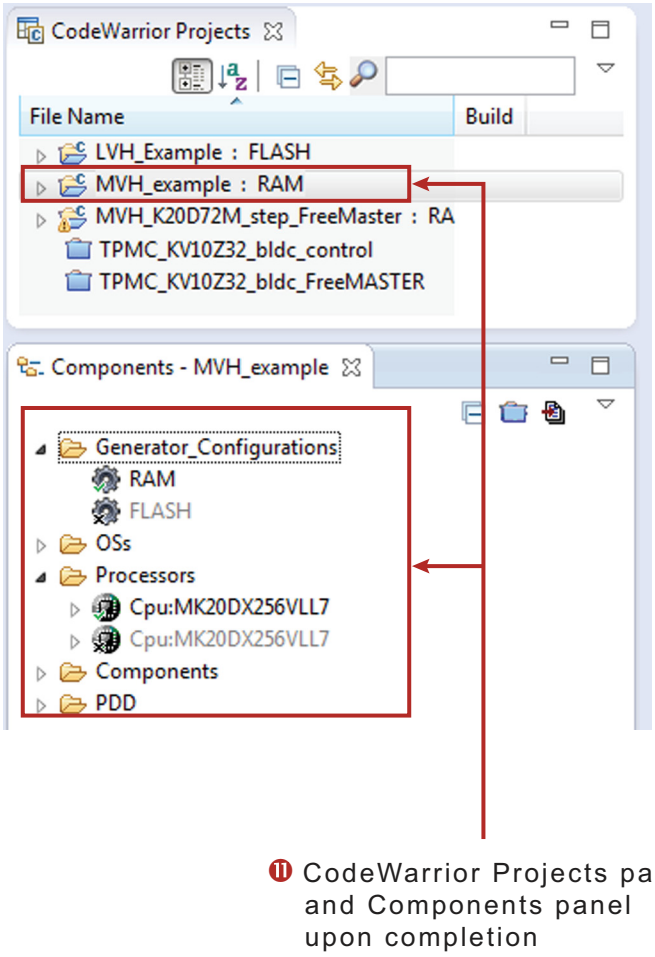

Figure 35. CodeWarrior Projects and Components panels with project created

6.3.1 Adding the MVHBridge Component into the Project

1. Find MVHBridge in the Components Library and add it into your project (see Figure 36).

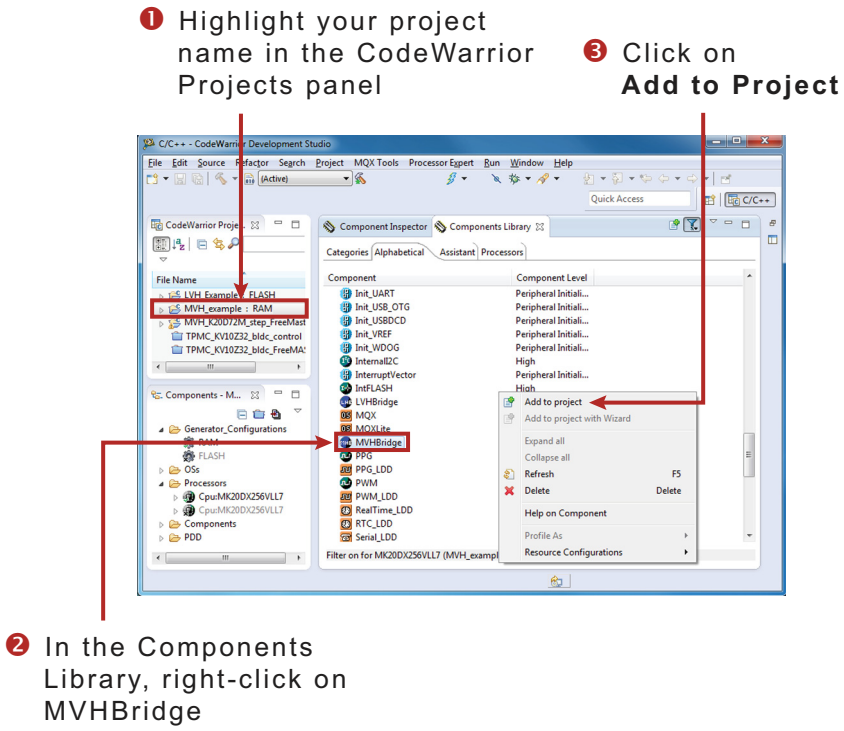

Figure 36. Add the MVHBridge Component to the Project

2. Figure 37 shows the Components panel after the component has been added. To view the Component Inspector options, double click on the MVHBridge component in the Components panel.

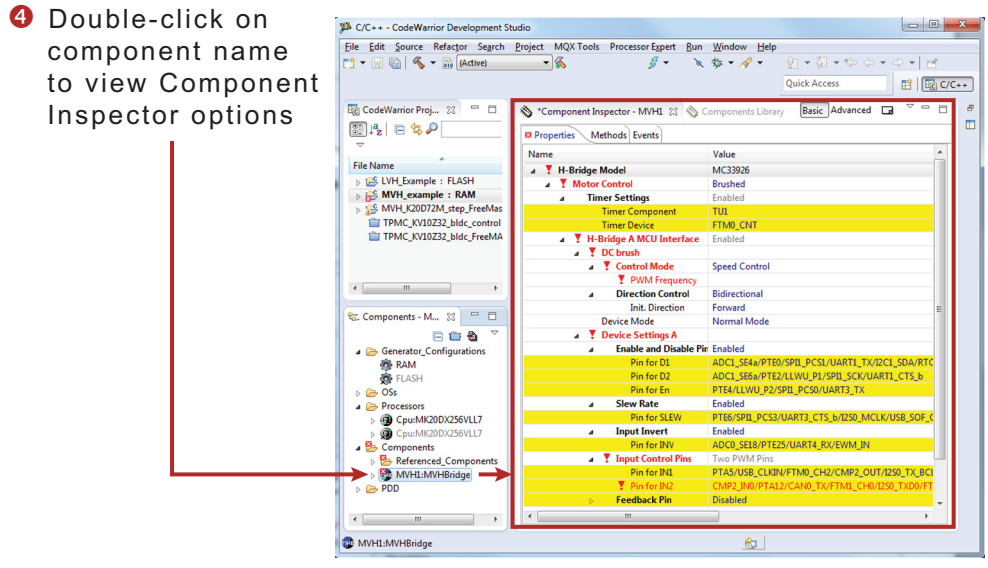

Figure 37. Select the Component

6.3.2 General Settings of MVHBridge Component

The Component Inspector view provides a means of accessing and modifying component properties. When CodeWarrior is set to the **Classic** view, properties in the Component Inspector are arranged in a collapsible tree-structure. Property names appear in the **Name** column. The **Values** column lists the current value assigned to the property. Values that are not greyed-out in this column may be modified. The **Details** column contains additional information (including error conditions) about the selected property. (If you have CodeWarrior preferences set to the **Tab** view, properties will be arranged differently in the Component Inspector; However, the same definitions apply.)

Figure 38 shows typical Component Inspector properties for a project using a DC brushed motor and an MC34931S MCU with a single H-Bridge. Different components and settings may apply when other types of motors and MCU's are used.

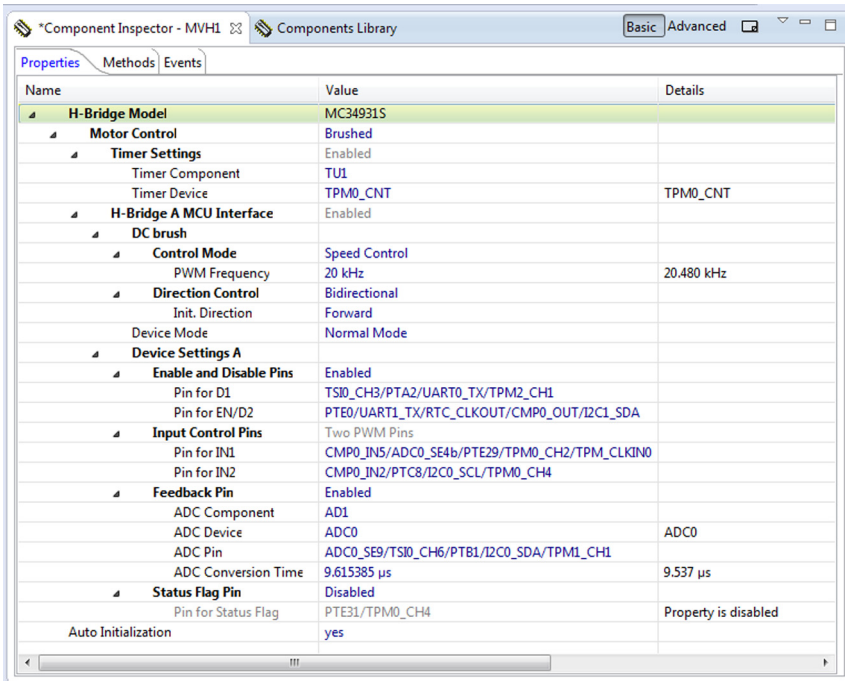

Figure 38. Component Inspector - brushed DC motor project

For the project in Figure 38 the **H-Bridge Model** is the top node in the tree structure. A drop-down menu in the **Value** column allows you to select the H-Bridge model your project uses.

The **Motor Control** group is directly below the **H-Bridge Model** node. The group contains two child nodes: **Timer Setting** and **H-Bridge A MCU Interface**. An MCU with dual H-Bridges would have an **H-Bridge B MCU Interface** group with settings similar to H-Bridge A. The settings in each of these groups are detailed below:

Timer Setting when enabled, defines timer settings for the project. (For the MC34931S used in this example, the timer is enabled by default.) The group contains the following settings:

Timer Component defines the name of the linked **TimerUnit_LDD Component**.

Timer Device defines the name of the hardware timer being used.

H-Bridge A MCU Interface defines H-Bridge interface setting. The group contains three child nodes:

DC Brush allows you to select the motor control mode and the motor direction:

Control Mode allows you to select whether your settings control the motor speed (**Speed Control**) or whether the motor is controlled by GPIO pin signals (**State Control**).

PWM Frequency sets the Pulse Width Modulation frequency.

Direction Control determines in which direction the motor is allowed to rotate. **Forward** means the motor can rotate only in the clockwise direction. **Reverse** allows movement in the counterclockwise direction only. **Bidirectional** allows the motor to rotate in either direction.

Init Direction determines which direction (forward or reverse) the motor moves at startup.

Device Mode defines the H-Bridge operational mode for the selected device. The mode specifics depend on the device, but Normal, Sleep, and Stand-by are typical. For more information, see the data sheet for your device. **Device Mode** is controlled by enabling and disabling pins. The mode can be changed in your C code using the **SetMode** method.

Device Settings A associates each of the output pins with a corresponding input pin name.

Enable and Disable Pins settings control the Device Mode. The number and the names of pins in this group depends on the H-Bridge model you have selected. In all cases, you must assign the appropriate value to each pin name in the group.

Input Control Pins settings define H-Bridge outputs. These pins are controlled by timer channels or by GPIO pins according to other settings in the component.

Feedback Pin settings define current measurements on the feedback pin. H-Bridge feedback provides ground-referenced 0.24% of the high side output current.

ADC Component sets the name of the linked ADC_LDD component.

ADC Device defines the device used for current measurement.

ADC Pin defines the pin used for ADC current sensing.

ADC Conversion Time specifies the time interval in micro-seconds allowed for a single analog to digital conversion.

Status Flag Pin allows tracking of the H-Bridge status flag. Method **GetStatusFlag** provides current device status. Method **ClearStatusFlag** clears the status flag. Use Event **OnStatusFlagA** or **OnStatusFlagB** (depending on the H-Bridge interface) to handle errors indicated by the status flag.

Auto Initialization when set, causes Processor Expert to automatically make an initialization call. If this option is not set, your code must make the Init call.

6.3.3 Setting up a Project to Control a DC Brushed Motor

1. Select the H-Bridge model you want to configure and set the **Motor Control** property to **Brushed**.

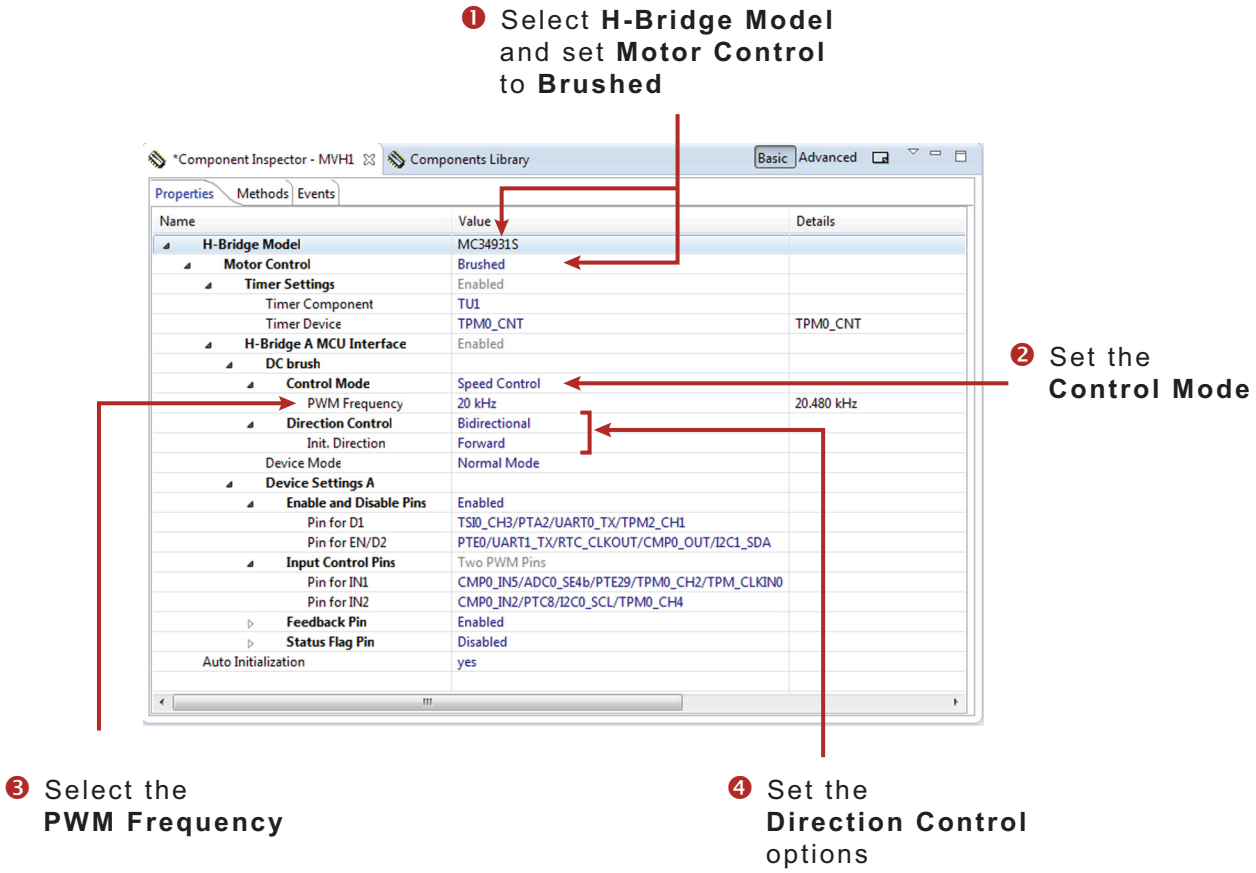

Figure 39. Brushed motor control setup

2. Set the **Control Mode** property. There are two ways to control the DC brushed motor:

   Speed Control - motor speed is controlled by your settings. The **TimerUnit_LDD** component is used to generate the PWM signal. The **PWM Frequency** property is visible in this mode only. If you set the **Speed Control** mode on both interfaces (i.e. Interface A and Interface B), the **PWM Frequency** property on Interface B will be set automatically to the same value as Interface A (because Interface B uses the same timer.)

   State Control - motor is controlled by GPIO pins (**BitIO_LDD** components). This means you can switch the motor on or off without speed adjustments. The advantage of this mode is that you do not need timer channels. If you set **State Control** on both interfaces or you have only a single H-Bridge model (one interface) with **State Control**, the T**imerUnit_LDD** component is not required by the MVHBridge component and you can remove it from the project.

3. Set the **PWM Frequency**.
4. Set the **Direction Control** property. The **Direction Control** property determines what direction the motor is allowed to move in. Setting the property to **Forward** restricts the motor's movement to the forward direction only. Setting the property to **Reverse** restricts movement to the reverse direction only. A **Bidirectional** setting allows the motor to move in either direction. The Bidirectional mode requires two timer channels. **Forward** or **Reverse** requires only one timer channel and one GPIO port. This setting is available only when **Speed Control** mode is set in the **Control Mode** property.

6.3.4 Generating Driver Source Code

After you have completed configuring the components, you are ready to generate the driver code that will be incorporated into your application. The process is as follows

1. Click on the **Generate Processor Expert Code** icon in the upper right corner of the **Components** panel.

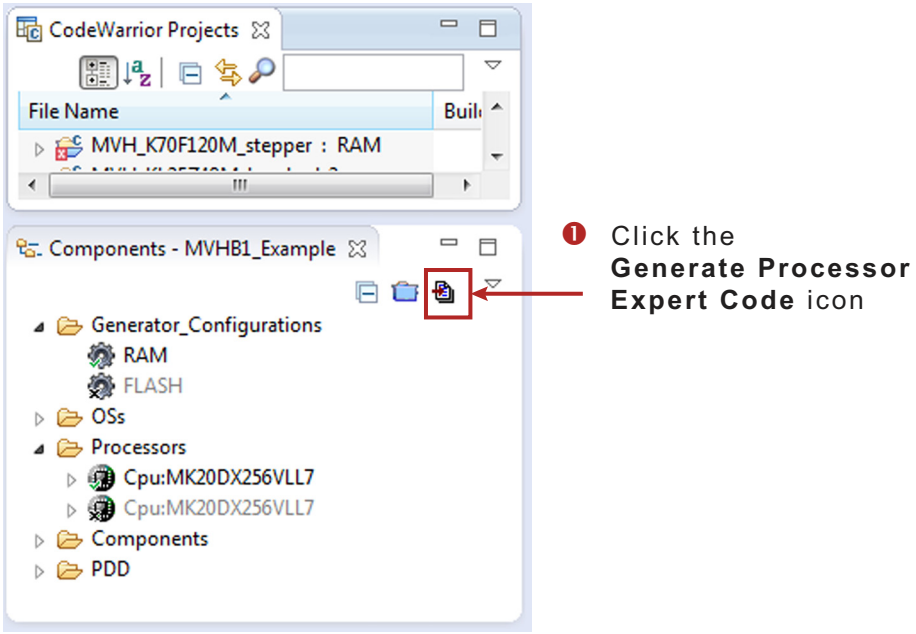

Figure 40. Generating the Source Code

2. The driver code for the H-Bridge device is generated into the **Generated_Code** folder in the **Project** panel. The component only generates the driver code. It does not generate application code. Figure 41 shows the locations of the generated driver source and the application code.

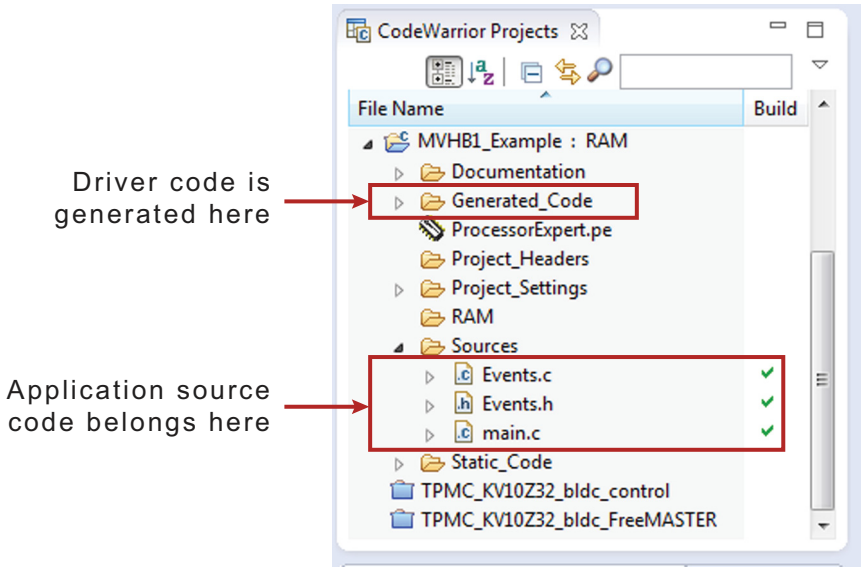

Figure 41. Source code locations

6.3.5 Developing Application Code in Processor Expert

Processor Expert allows you to write application code, add component methods, and build your application without leaving the CodeWarrior environment.

6.3.5.1 Writing your Application Code

All of your application code must reside in the **Sources** folder in your project directory. You may modify the code in **main.c** and **Events.c**, but retain the original comments related to usage directions.

6.3.5.2 Adding Component Methods

To add a component method into your application source code:

1. In the **Components** panel for your project, click on **Components**. Find the method you wish to add to your code.
2. Drag and drop the Method directly into the source code panel
3. Add the appropriate parameters to the method. (Hovering your mouse over the ethod displays a a list of the required parameters.)

For example, you can open the MVHBridge component method list, drag and drop **RotateProportional** to **main.c** and add the necessary parameters (see Figure 42).

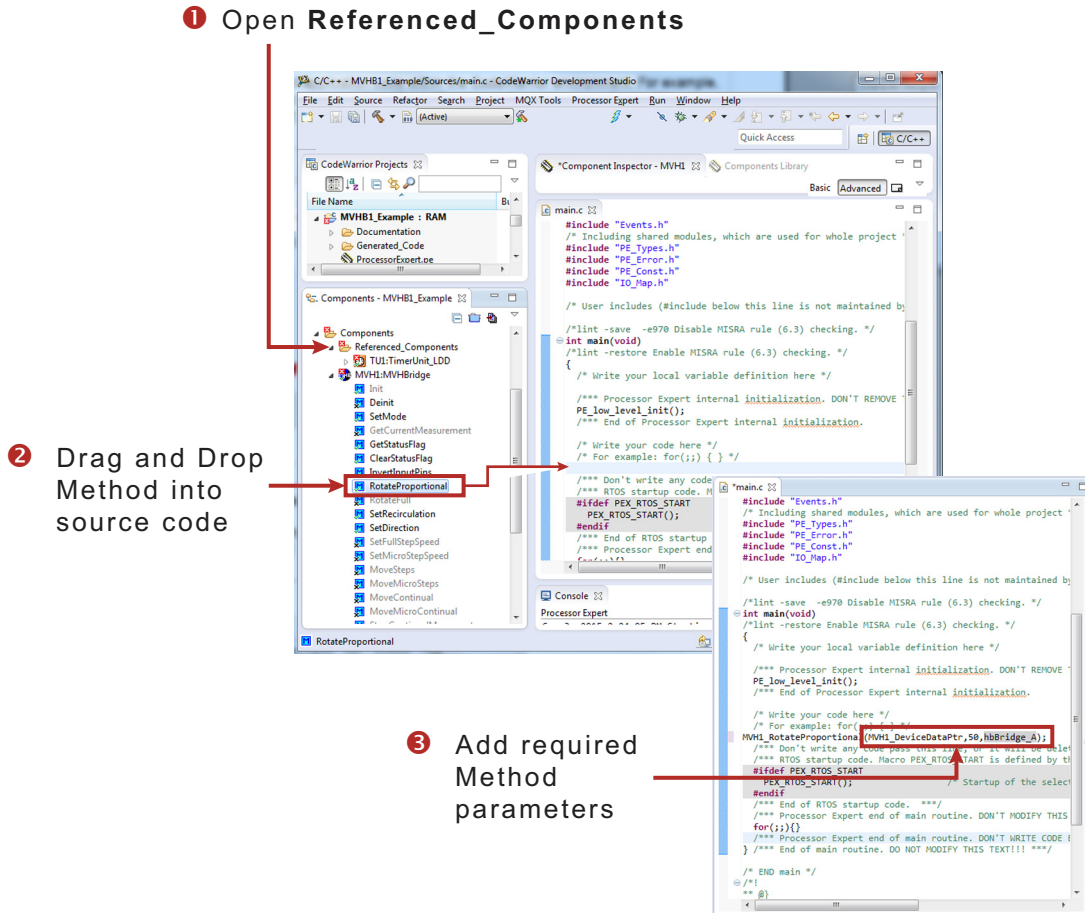

Figure 42. Adding Component Methods

6.3.5.3 Finding Descriptions of the MVHBridge Methods

Hovering your mouse over any of the Methods displays a description of the Method, including a list of required parameter. See Figure 43.

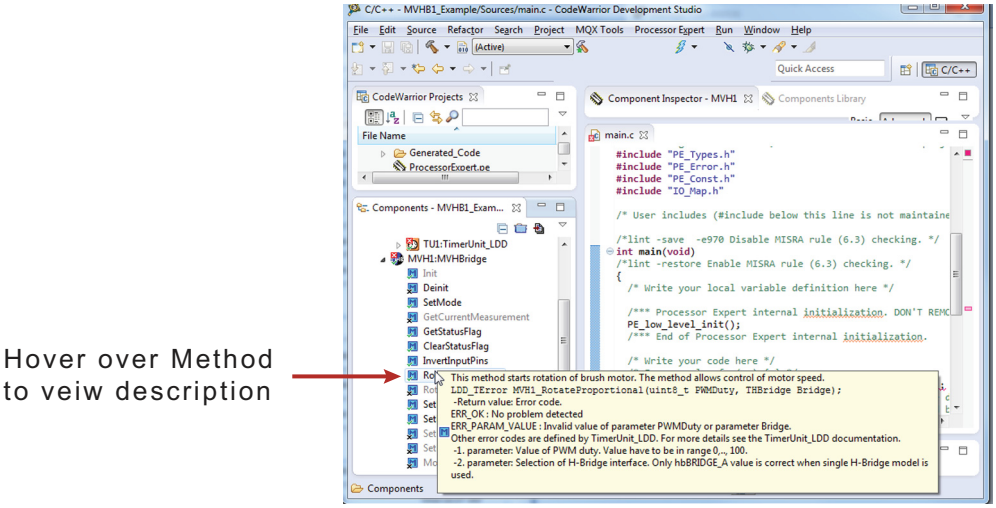

Figure 43. MVHBridge Method Descriptions

6.3.5.4 Jumping into Function Source Code

CodeWarrior is based on the Eclipse IDE which allows you to jump directly into the source code of a function from within the main routine while you are editing. To do so, move your mouse cursor over the function name and click. The source code appears in the edit window.

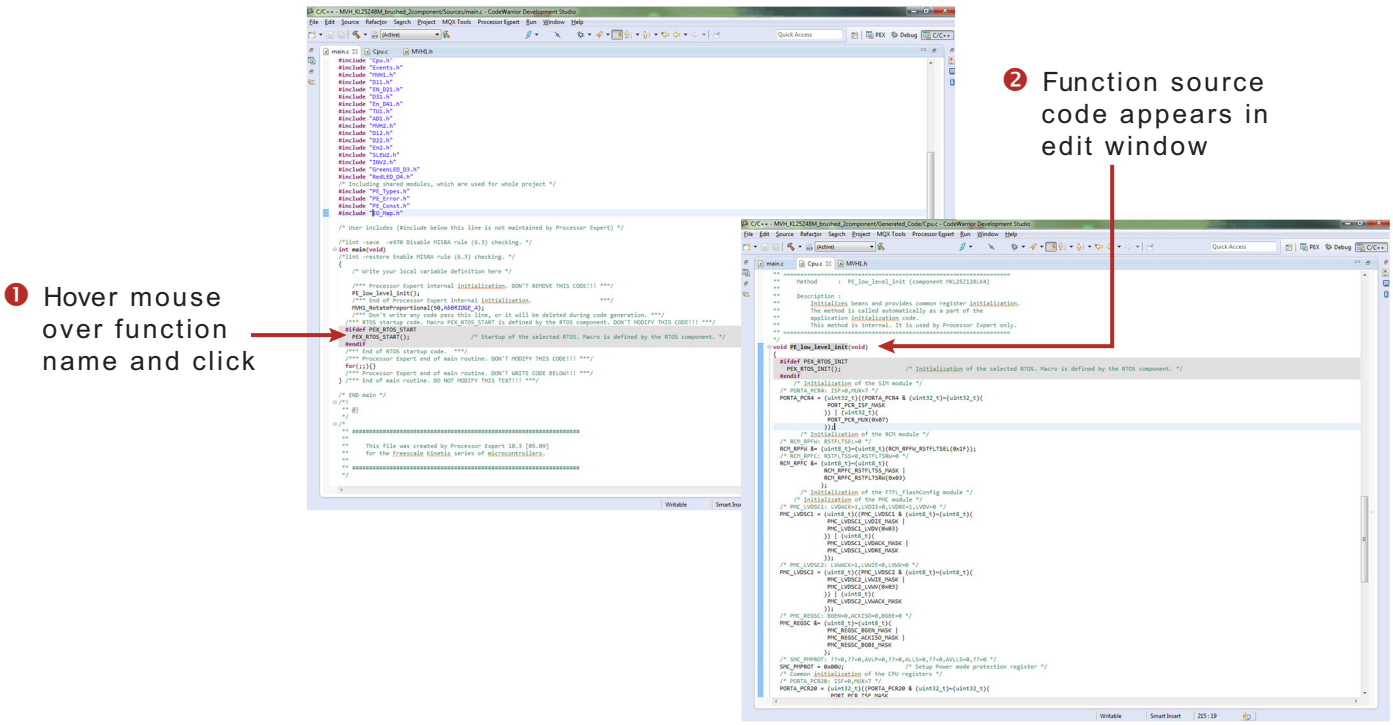

Figure 44. Jumping into a function's source code

6.3.5.5 Compiling, Downloading and Debugging

To compile, download and debug on board, click compile, then click the debug icon in the toolbar. CodeWarrior will download and launch the program on board (see Figure 45).

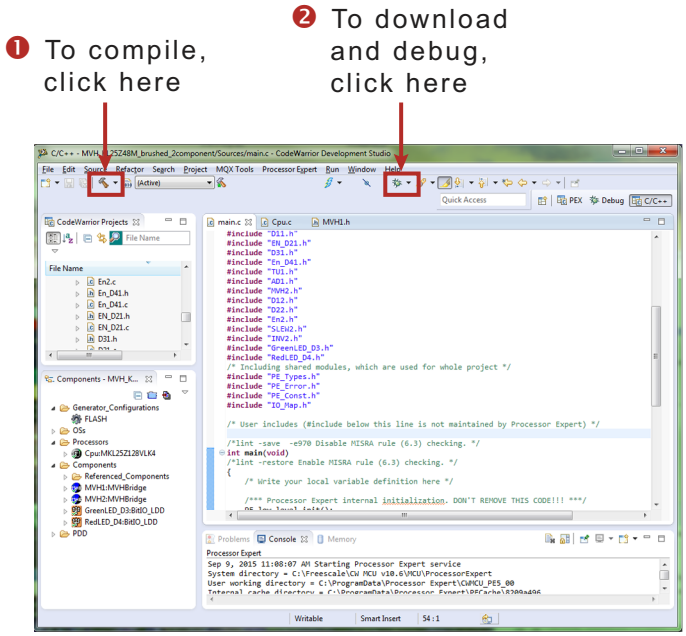

Figure 45. Compiling and Downloading the Application

6.4 Frequently Asked Questions

Q: Why do I occasionally unexpected behavior in my DC brushed motor?

A: Check the value of the signals on the enable and disable pins (D1, EN/D2, D3, EN/D4). These signals affect the H-Bridge device mode. To provide a wider range of MCU compatibility, some pins are wired to more than one MCU board pin using 0 Ω resistors. Check your schematic and remove resistors as needed to disconnect unused pins.

Q: How do I set up the **MVHBridge** component when two or more components with conflicting values are configured to control brushed motors? (See Figure 46)

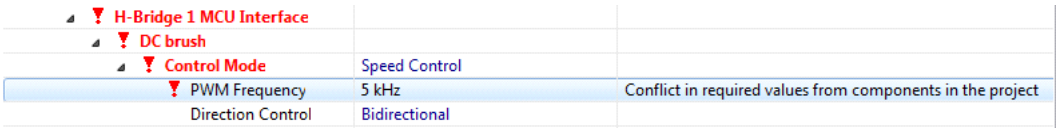

Figure 46. Conflict in the Required Values for Components in the Project

A: You can use more than one **MVHBridge** component in same project. These components can share the same timer device in brushed motor control mode, but the **PWM Frequency** and **Timer Device** properties must conform in all of the components.

Q: Can I use both a stepper motor and a brushed DC motor on a single timer?

A: The stepper motor control needs a dedicated timer because the timer period can be dynamically changed. Using a stepper motor and a brushed DC motor on the same timer pins is possible only when the **Control Mode** property of the brushed DC motor is set to **State Control**.

Q: The **TimerUnit_LDD** component used by MVHBridge is not set properly and shows some errors.

A: The reason could be that the **TimerUnit_LDD** component channels are not allocated correctly. You must change some property of the **MVHBridge** component to force allocation of the channels. Change the **Output Control** property to **GPIO** and back to **PWM** if you are configuring a stepper motor (**Motor Control** property set to **Stepper**). For a brushed motor (**Motor Control** property set to **Brushed**) change the **Control Mode** property to **State Control** and back to **Speed Control** on interface A or interface B.

Q: I sometimes get the following unexpected error while generating Processor Expert code: "Generator: FAILURE: Unexpected status of script: Drivers\Kinetis\TimerUnit_LDD.drv, please contact Freescale support". What causes this?

A: Occasionally, when you enable the **MVHBridge** component in your project, the **TimerUnit_LDD** component channels have not been allocated. If this occurs, changing certain MVHBridge properties will force allocation of the channels. When you are configuring a brushed motor (**Motor Control** property set to **Brushed**), change the **Control Mode** property to **State Control** and then back to **Speed Control** on interface A or interface B.

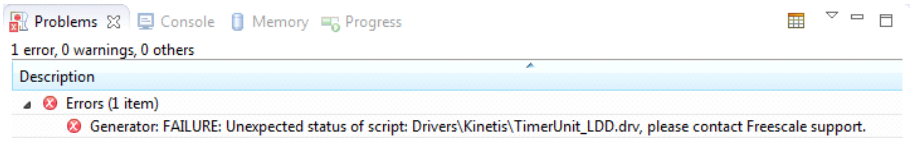

Figure 47. Unexpected Error Related to the MVHBridge TimerUnit_LDD Component

Q: I have set up several CPU clock configurations (via the Clock configurations property of the CPU component.) Sometimes during runtime, when I switch between these configuration (using the CPU **SetClockConfiguration** method), the speed of the motor appears to be inaccurate. Why does this occur?

A: Switching to a different configuration results in the use of a different input frequency by a timer device. The MVHBridge component may not pick up the new value and continues to use the previous value in its calculations.

7 Schematic

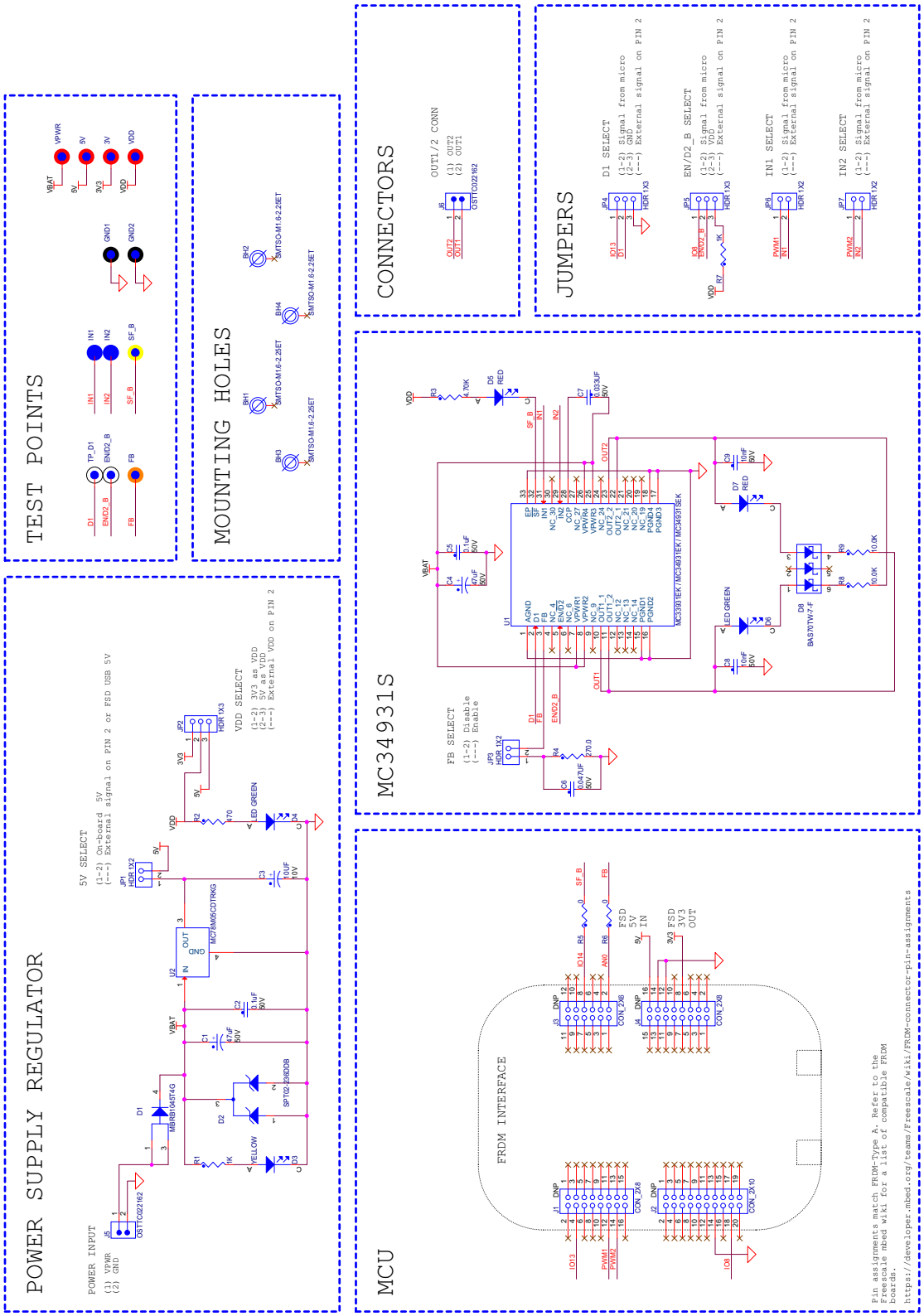

Figure 48. Schematic

8 Board Layout

8.1 Silkscreen

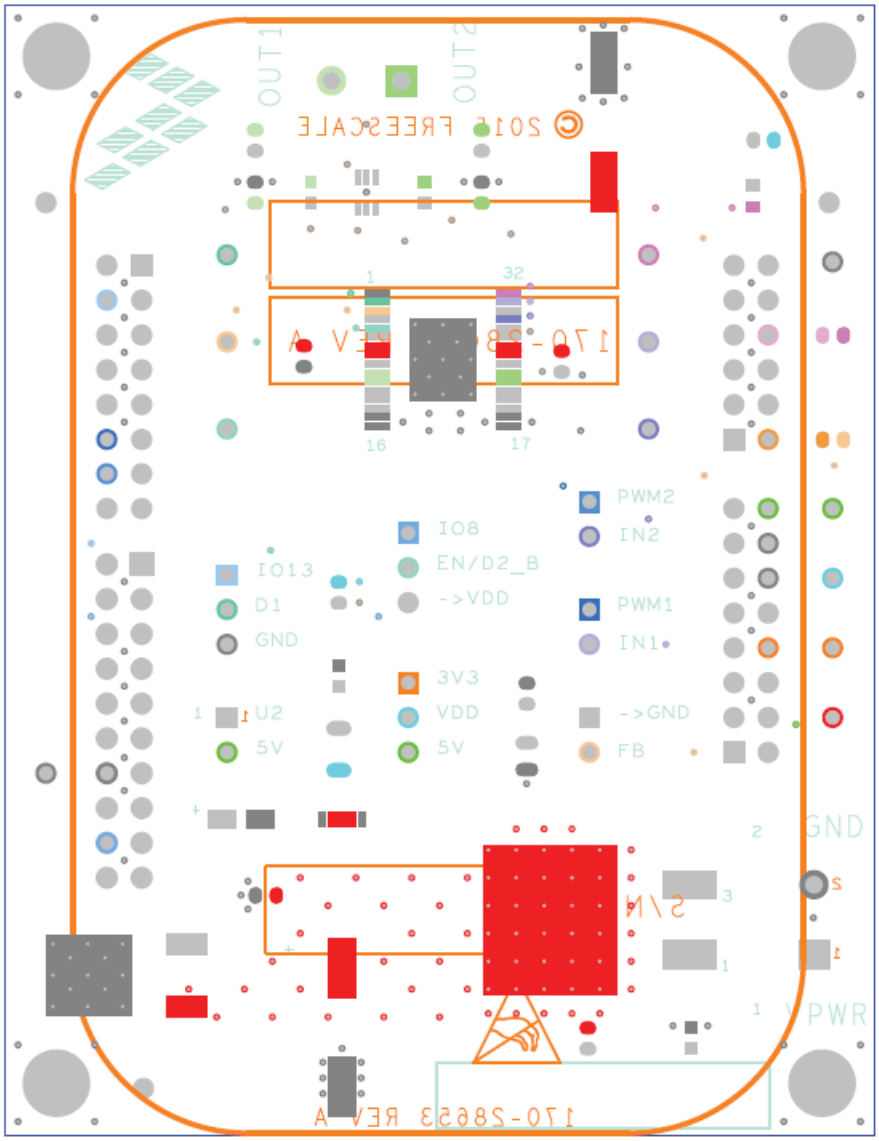

Figure 49. Silkscreen

9 Board Bill of Materials

Table 12. Bill of Materials [1]

Item	Qty	Schematic Label	Value	Description	Part Number	Assy Opt
Freescale Components						
1	1	U1		Freescale device	MC34931EK/MC34931SEK	[2]
Voltage Regulator						
2	1	U2		IC LIN VREG LDO 5 V 0.5 A 35 V DPAK	MC78M05CDTRKG	
Diodes						
3	1	D1		DIODE SCH PWR RECT 10 A 45 V D2PAK	MBRB1045T4G	
4	1	D2		DIODE DUAL ARRAY 2 A 6-36 V uQFN-2L	SPT02-236DDB	
5	1	D3		LED YEL SGL 25 MA SMT 0603	LY Q976-P1S2-36-0-20-R18	
6	2	D4, D6		LED GRN SGL 20 MA 0603	LG L29K-G2J1-24-Z	
7	2	D5, D7		LED SM RED 0603 ROHS COMPLIANT	QTLP600CRTR	
8	1	D8		DIODE SCH TRIPLE 70 MA 70 V / 200 MW SOT363	BAS70TW-7	
Capacitors						
9	2	C1, C4	47 µF	CAP ALEL 47 µF 50 V 20% AUTO SMD	UBC1H470MNS1GS	
10	2	C2, C5	0.1 µF	CAP CER 0.1 uF 50 V 5% X7R AEC-Q200 0603	C0603C104J5RACAUTO	
11	1	C3	10 µF	CAP TANT 10 µF 10 V 10% — 3216-18	293D106X9010A2TE31	
12	1	C6	0.047 µF	CAP CER 0.047 µF 50 V 5% X7R 0805	C0805C473J5RAC	
13	1	C7	0.033 µF	CAP CER 0.033 µF 50 V 5% X7R 0603	06035C333JAT2A	
14	2	C8, C9	0.1 µF	CAP CER 0.01 µF 50 V 5% X7R 0603	06035C103JAT2A	
Resistors						
15	2	R1, R7	1.0 KΩ	RES -- 1 KΩ 1/4 W 1% AEC-Q200 0603 ANTISURGE	ESR03EZPF1001	
16	1	R2	470 Ω	RES MF 470 Ω 1/4 W 5% AEC-Q200 1206	CRCW1206470RJNEA	
17	1	R3	4.7 KΩ	RES MF 4.7 KΩ 1/4 W 1% AEC-Q200 0603	CRCW06034K70FKEA	
18	1	R4	270 Ω	RES MF 270.0 Ω 1/10 W 1% 0603	RK73H1JTTD2700F	
19	2	R5, R6	0 Ω	RES MF ZERO Ω 1/10 W — AEC-Q200 0603	RK73Z1JTTD	
20	1	R8, R9	10 KΩ	RES MF 10.0 KΩ 1/10 W 1% 0603	RK73H1JTTD1002F	

Table 12. Bill of Materials [(1)] (continued)

Item	Qty	Schematic Label	Value	Description	Part Number	Assy Opt
Switches, Connectors, Jumpers and Test Points						
21	12	TP_D1,EN/D2_B, FB, GND1,GND2, IN1,IN2, SF_B, 3V,5V,VPWR,VDD		TEST POINT 40 MIL DRILL 180 MIL		
22	4	JP1,JP3,JP6,JP7		HDR 1X2 TH 100 MIL SP 338H SN 100L	TSW-102-07-T-S	
23	3	JP2,JP4,JP5		HDR 1x3 TH 100 MIL SP 343H SN 100L	TSW-103-07-T-S	
24	2	J1, J4		HDR 2X8 TH 100MIL CTR 338H SN 100L	TSW-108-07-T-D	
25	1	J2		HDR 2X10 TH 100MIL CTR 343H SN 100L	TSW-110-07-T-D	
26	1	J3		HDR 2X6 TH 100MIL CTR 338H SN 100L	TSW-106-07-T-D	
27	2	J5, J6		CON 1X2 TB 5.08 MM SP 406H SN 138L	OSTTC022162	

Notes

1. Freescale does not assume liability, endorse, or warrant components from external manufacturers are referenced in circuit drawings or tables. While Freescale offers component recommendations in this configuration, it is the customer's responsibility to validate their application.
2. **Critical components.** For critical components, it is vital to use the manufacturer listed.

10 Accessory Item Bill of Materials

Table 13. Bill of Materials [3]

Item	Qty	Part Number	Description
1	1	FRDM-KL25Z	Freescale Freedom Development Platform for Kinetis KL14/15/24/25 MCUs

Notes

3. Freescale does not assume liability, endorse, or warrant components from external manufacturers are referenced in circuit drawings or tables. While Freescale offers component recommendations in this configuration, it is the customer's responsibility to validate their application.

11 References

Following are URLs where you can obtain information on related Freescale products and application solutions:

Freescale.com Support Pages	Description	URL
FRDM-34931S-EVB	Tool Summary Page	www.freescale.com/FRDM-34931S-EVB
FRDM-34931-EVB	Tool Summary Page	www.freescale.com/FRDM-34931-EVB
MC34931	Product Summary Page	www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MC34931
FRDM-KL25Z	Tool Summary Page	www.freescale.com/FRDM-KL25Z
ARM®mbed™	mbed FRDM-KL25Z Upgrade Page	mbed.org/handbook/mbed-FRDM-KL25z-Upgrade
CodeWarrior	Tool Summary Page	www.freescale.com/webapp/sps/site/homepage.jsp?code=CW_HOME&tid=vanCODEWARRIOR
Processor Expert Code Model	Code Walkthrough Video	www.freescale.com/video/processor-expert-code-model-codewarrior-code-walkthrough:PROEXPCODMODCW_VID

11.1 Support

Visit www.freescale.com/support for a list of phone numbers within your region.

11.2 Warranty

Visit www.freescale.com/warranty to submit a request for tool warranty.

12 Revision History

Revision	Date	Description of Changes
1.0	7/2015	• Initial Release
2.0	9/2015	• Added processor expert section
	9/2015	• Fixed invalid Section reference • Fixed duplicate section title • Added Processor Expert, CodeWarrior, Kinetis to tradmark citations in last page

How to Reach Us:

Home Page:
freescale.com

Web Support:
freescale.com/support

Information in this document is provided solely to enable system and software implementers to use Freescale products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits based on the information in this document.

Freescale reserves the right to make changes without further notice to any products herein. Freescale makes no warranty, representation, or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale data sheets and/or specifications can and do vary in different applications, and actual performance may vary over time. All operating parameters, including “typicals,” must be validated for each customer application by customer’s technical experts. Freescale does not convey any license under its patent rights nor the rights of others. Freescale sells products pursuant to standard terms and conditions of sale, which can be found at the following address: freescale.com/SalesTermsandConditions.

Freescale, the Freescale logo, Processor Expert, CodeWarrior, and Kinetis are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. SMARTMOS is a trademark of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners.

© 2015 Freescale Semiconductor, Inc

Document Number: KTFRDM34931UG
Rev. 2.0
9/2015