FRDM-17529EV-EVB Evaluation Board
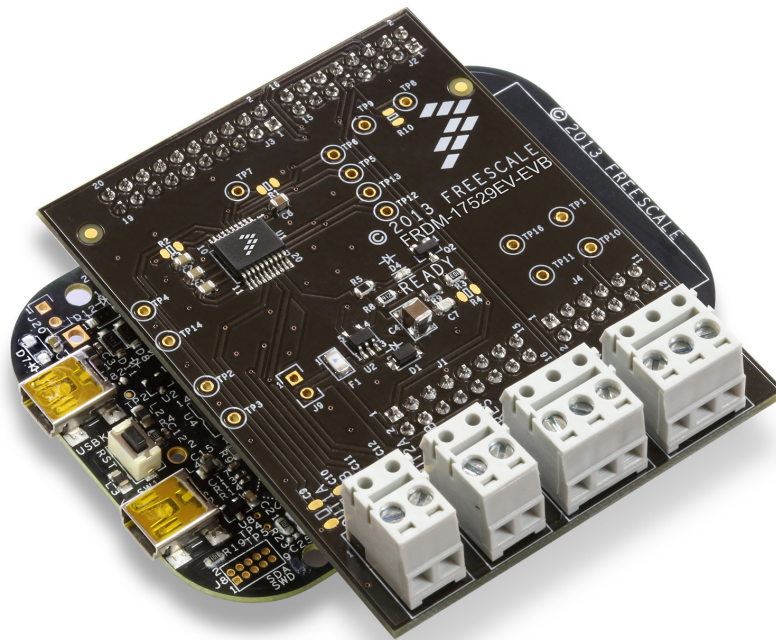

Figure 1. FRDM-17529EV-EVB

Table of Contents

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 kit 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. 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 that 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-17529EV-EVB contents include:

- Assembled and tested evaluation board/module in an anti-static bag
- Quick Start Guide, Analog Tools
- 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.

- Go to http://www.freescale.com/FRDM-17529EV-EVB
- Review your Tool Summary Page
- Look for

Jump Start Your Design

- 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 (2.0 V to 6.8 V, 0.1 A to 0.7 A, depending on stepper motor requirements)
- USB A to mini-B cable
- Oscilloscope (preferably 4-channel) with current probe(s)
- Digital multimeter
- FRDM-KL25Z Freedom Development Platform
- Typical loads (stepper motor, brushed DC motors, or power resistors)
- 3/16" blade screwdriver
- One 12-pin (PPTC062LFBN-RC), two 16-pin (PPTC082LFBN-RC), and one 20-pin (PPTC102LFBN-RC) female connector, by Sullins Connector Solutions, or equivalent soldered to FRDM-KL25Z

2.4 System Requirements

The kit requires the following:

- USB-enabled PC with Windows® XP or higher

3 Getting to Know the Hardware

3.1 Board Overview

The FRDM-17529EV-EVB evaluation board features the MPC17529EV dual H-Bridge IC, which features the ability to drive either a single two phase stepper motor or two brushed DC motors. The MPC17529EV incorporates internal control logic, a charge pump, gate drive, and high current, low $R_{DS(on)}$ MOSFET output circuitry.

3.2 Board Features

The FRDM-17529EV-EVB evaluation board is designed to easily evaluate and test the main component, the MPC17529EV. The board's main features are as follows:

- Compatible with Freedom series evaluation boards such as FRDM-KL25Z
- Built in fuse for both part and load protection
- Screw terminals to provide easy connection of power and loads
- Test points to allow probing of signals
- Built in voltage regulator to supply logic level circuitry
- LED to indicate status of Logic power supply of the evaluation board, as well as a general purpose indicator

3.3 FRDM-KL25Z Features

The FRDM-KL25Z board features are as follows:

- MKL25Z128VLK4 MCU - 48 MHz, 128 KB Flash, 16 KB SRAM, USB OTG (FS), 80LQFP
- Capacitive touch slider, MMA8451Q accelerometer, Tri-color LED
- Flexible power supply options - USB, coin cell battery, external source
- Easy access to MCU I/O
- Battery-ready, power-measurement access points
- Form factor compatible with Arduino™ R3 pin layout
- New, OpenSDA debug interface
- Mass storage device flash programming interface (default) - no tool installation required to evaluate demonstration applications
- P&E Debug interface provides run-control debugging and compatibility with IDE tools
- CMSIS-DAP interface: new ARM standard for embedded debug interface

Additional reference documents are available on freescale.com/FRDM-KL25Z.

3.4 Device Features

This evaluation board features the following Freescale product:

Table 1. Device Features

Device	Description	Features
MPC17529EV	The MPC17529EV is a dual H-Bridge motor driver IC intended for operating stepper motors	• Voltage range of operation from 2.0 V to 6.8 V • Output Current of 0.7 A (DC) continuous, 1.4 A peak • 700 mΩ $R_{DS(on)}$ H-Bridge MOSFET outputs • 3.3/5.0 V TTL/CMOS compatible inputs • PWM frequencies up to 200 kHz • Undervoltage shutdown • Cross conduction (shoot through) suppression

3.5   Board Description

This evaluation board consists mainly of an MPC17529EV. The following sections describe the additional hardware used to support the dual H-Bridge driver.
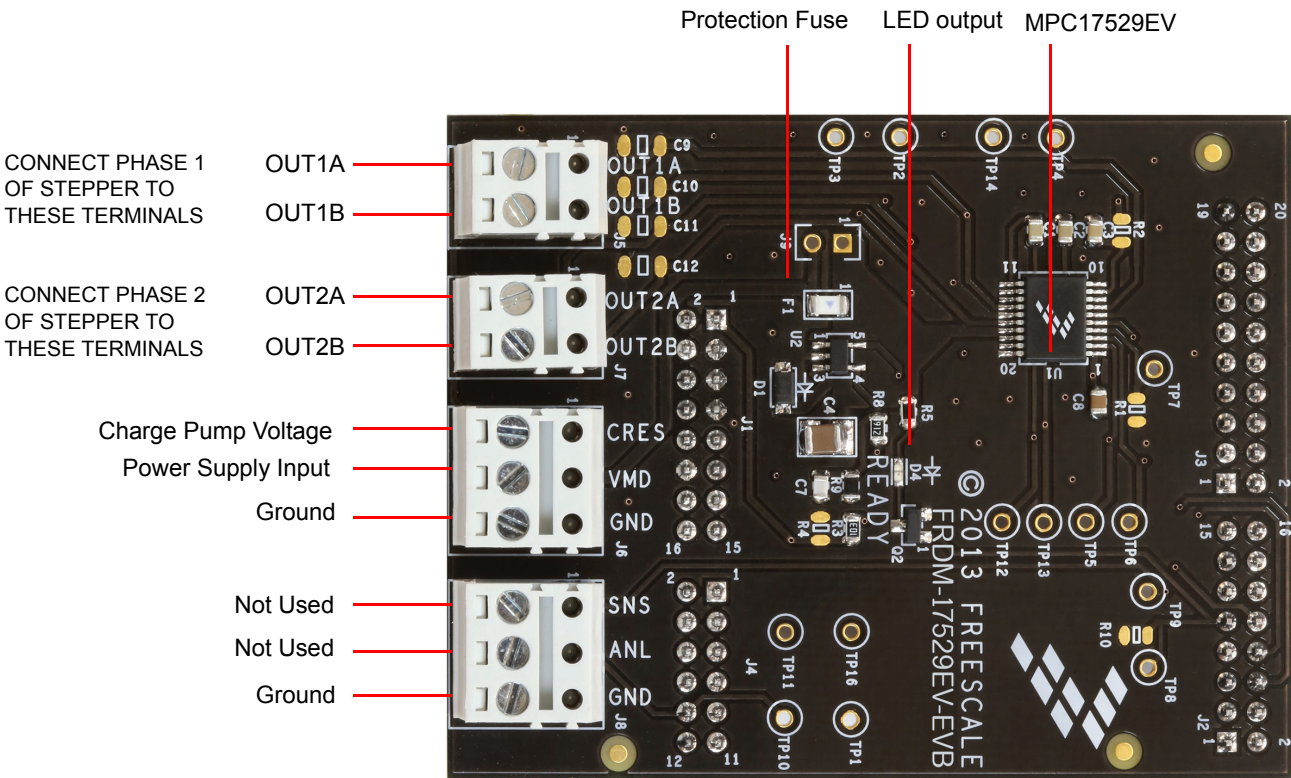

Figure 2. Board Description

Table 2. Board Description

Name	Description
U1	MPC17529EV H-Bridge motor drive IC
F1	Overcurrent protection fuse
D4	User defined LED output
OUT1A	Output 1A Connect motor phase 1 lead to this terminal
OUT1B	Output 1B Connect motor phase 1 lead to this terminal
OUT2A	Output 2A Connect motor phase 2 lead to this terminal
OUT2B	Output 2B Connect motor phase 2 lead to this terminal
CRES	Charge pump voltage
VM	Power supply input
GND	Ground terminal
SNS	Not used – connection to FRDM-KL25Z input
ANL	Not used – connection to FRDM-KL25Z input
GND	Ground terminal

3.6   LED Display

An LED is provided as a visual output device for the FRDM-17529EV-EVB evaluation board:

Table 3. Board Description

Name	Description
LED1 (D4 board designator)	Illuminated with an output from the FRDM-KL25Z. Note the on board voltage regulator must be operating for the LED to operate

3.7   Test Point Definitions

The following test-points provide access to signals on the FRDM-17529EV-EVB. These signals are:

Table 4. Test Point Definitions

TP#	Signal Name	Description
TP1	GND	Ground
TP2	OUT2A	H-Bridge 2 Output A
TP3	OUT2B	H-Bridge 2 Output B
TP4	OUT1A	H-Bridge 1 Output A
TP5	IN1A	H-Bridge 1 Input A
TP6	IN1B	H-Bridge 1 Input B
TP7	OE	Output Enable pin
TP8	READY	Logic signal from microcontroller. This signal causes the green LED to operate
TP9	SNSIN	Not Used
TP10	ANLIN	Not Used
TP11	VDDPWRGOOD	Signal to the Microcontroller indicating the voltage regulator is operating (3.3 V)
TP12	IN2A	H-Bridge 2 Input A
TP13	IN2B	H-Bridge 2 Input B
TP14	OUT1B	H-Bridge 1 Output B
TP16	VDD	Logic power supply from the voltage regulator on the evaluation board

3.8   Input Signal Definitions

The MPC17529EV IC has five input signals that are used to control certain outputs or functions inside the circuit. These signals are:

Table 5. Input Signal Definitions

Name	Description
IN1A	Controls OUT1A
IN1B	Controls OUT1B
IN2A	Controls OUT2A
IN2B	Controls OUT2B
OE	Enables Outputs 1B, 1B and Outputs 2A, 2B

3.9 Output Signal Definitions

The MPC17529EV IC has four output signals used to drive a two phase stepper motor. These signals are:

Table 6. Output Signal Definitions

Name	Description
OUT1A	Output A of H-Bridge 1
OUT1B	Output B of H-Bridge 1
OUT2A	Output A of H-Bridge 2
OUT2B	Output B of H-Bridge 2

3.10 Screw Terminal Connections

There are four connectors on the FRDM-17529EV-EVB which provide connections to the following signals:

Table 7: Screw Terminal Connections

Name	Signal	Signal Description
J5	OUT1A	H-Bridge 1 output A
	OUT1B	H-Bridge 1 output B
J6	CRES	Charge pump voltage for H-Bridge gate drive
	VM	Motor supply input (this is also the supply for the on board voltage regulator)
	GND	This is the primary ground connection for the motor power supply
J7	OUT2A	H-Bridge 2 output A
	OUT2B	H-Bridge 2 output B
J8	SNS	Not used
	ANL	Not used
	GND	Additional ground

3.11 Jumper J9

The FRDM-17529EV-EVB has provision (not populated) for a jumper to accommodate higher currents than the on board fuse is capable of handling (1.25 A). If the fuse is bypassed, use extreme care to make sure the maximum current for the MPC17529 is not exceeded (0.7 A continuous, 1.4 A peak/transients).

4   FRDM-KL25Z Freedom Development Platform

The Freescale Freedom development platform is a set of software and hardware tools for evaluation and development. It is ideal for rapid prototyping of microcontroller-based applications. The Freescale Freedom KL25Z hardware, FRDM-KL25Z, is a simple, yet sophisticated design featuring a Kinetis L Series microcontroller, the industry's first microcontroller built on the ARM® Cortex™-M0+ core.

4.1   Connecting FRDM-KL25Z to the Board

The FRDM-17529EV-EVB kit may be used with many of the Freedom platform evaluation boards featuring Kinetis processors. The FRDM-KL25Z evaluation board has been chosen specifically to work with the FRDM-17529EV-EVB kit because of its low cost and features. The FRDM-KL25Z board makes use of the USB, built in LEDs, and I/O ports available with Freescale's Kinetis KL2x family of microcontrollers. The main functions provided by the FRDM-KL25Z are to allow control of a stepper motor using a PC computer over USB, and to drive the necessary inputs on the FRDM-17529EV-EVB evaluation kit to operate the motor.

The FRDM-17529EV-EVB is connected to the FRDM-KL25Z using four dual row headers. The connections are as follows:

Table 8: FRDM-17529EV-EVB to FRDM-KL25Z Connections

FRDM-17529EV-EVB		FRDM-KL25Z		Pin Hardware Name		Description
Header	Pin	Header	Pin	FRDM-17529EV-EVB	FRDM-KL25Z	
J1	1	J9	1	RUNPWRGD	PTB8	Regulator voltage present
J1	2	J9	2	N/C	SDA_PTD5	No connection
J1	3	J9	3	GND	PTB9	System ground
J1	4	J9	4	N/C	P3V3	No connection
J1	5	J9	5	GND	PTB10	System ground
J1	6	J9	6	N/C	RESET/PTA20	No connection
J1	7	J9	7	GND	PTB11	System ground
J1	8	J9	8	N/C	P3V3	No connection
J1	9	J9	9	N/C	PTE2	No connection
J1	10	J9	10	N/C	P5V_USB	No connection
J1	11	J9	11	N/C	PTE3	No connection
J1	12	J9	12	GND	GND	System ground
J1	13	J9	13	N/C	PTE4	No connection
J1	14	J9	14	N/C	GND	No connection
J1	15	J9	15	N/C	PTE5	No connection
J1	16	J9	16	N/C	P5-9V_VIN	No connection
J2	1	J1	1	OE	PTC7	Enable
J2	2	J1	2	N/C	PTA1	No connection
J2	3	J1	3	N/C	PTC0	No connection
J2	4	J1	4	N/C	PTD4	No connection
J2	5	J1	5	N/C	PTC3	No connection
J2	6	J1	6	IN1A	PTD4	Input 1A
J2	7	J1	7	N/C	PTC4	No connection
J2	8	J1	8	IN1B	PTA12	Input 1B
J2	9	J1	9	READY	PTC5	No connection Green LED (from KL25Z)
J2	10	J1	10	IN2A	PTA4	Input 2A
J2	11	J1	11	SNSIN	PTC6	Not used

Table 8: FRDM-17529EV-EVB to FRDM-KL25Z Connections (continued)

FRDM-17529EV-EVB		FRDM-KL25Z		Pin Hardware Name		Description
Header	Pin	Header	Pin	FRDM-17529EV-EVB	FRDM-KL25Z	
J2	12	J1	12	IN2B	PTA5	Input IN2B
J2	13	J1	13	N/C	PTC10	No connection
J2	14	J1	14	N/C	PTC8	No connection
J2	15	J1	15	N/C	PTC11	No connection
J2	16	J1	16	N/C	PTC9	No connection
J3	1	J2	1	N/C	PTC12	No connection
J3	2	J2	2	N/C	PTA13	No connection
J3	3	J2	3	N/C	PTC13	No connection
J3	4	J2	4	N/C	PTD5	No connection
J3	5	J2	5	N/C	PTC16	No connection
J3	6	J2	6	N/C	PTD0	No connection
J3	7	J2	7	N/C	PTC17	No connection
J3	8	J2	8	N/C	PTD2	No connection
J3	9	J2	9	N/C	PTA16	No connection
J3	10	J2	10	N/C	PTD3	No connection
J3	11	J2	11	N/C	PTA17	No connection
J3	12	J2	12	N/C	PTD1	No connection
J3	13	J2	13	N/C	PTE31	No connection
J3	14	J2	14	N/C	GND	No connection
J3	15	J2	15	N/C	N/C	No connection
J3	16	J2	16	N/C	VREFH	No connection
J3	17	J2	17	N/C	PTD6	No connection
J3	18	J2	18	N/C	PTE0	No connection
J3	19	J2	19	N/C	PTD7	No connection
J3	20	J2	20	N/C	PTE1	No connection
J4	1	J10	1	N/C	PTE20	No connection
J4	2	J10	2	N/C	PTB0	No connection
J4	3	J10	3	N/C	PTE21	No connection
J4	4	J10	4	N/C	PTB1	No connection
J4	5	J10	5	N/C	PTE22	No connection
J4	6	J10	6	N/C	PTB2	No connection
J4	7	J10	7	N/C	PTE23	No connection
J4	8	J10	8	N/C	PTB3	No connection
J4	9	J10	9	N/C	PTE29	No connection
J4	10	J10	10	ANLIN	PTC2	Not used
J4	11	J10	11	N/C	PTE30	No connection
J4	12	J10	12	N/C	PTC1	No connection

KTFRDM17529EVUG, Rev. 2.0

5 Installing the Software and Setting up the Hardware

5.1 Installing the Motor Control Graphical User Interface (GUI) on your Computer

The latest version of the Motor Control GUI is designed to run on any Windows 8, Windows 7, Vista, or XP-based operating system. To install the software, go to www.freescale.com/analogtools and select your kit. Click on that link to open the corresponding Tool Summary Page. Look for "Jump Start Your Design". Download to your computer desktop the Motor Control GUI software.

Run the installed program from the desktop. The Installation Wizard will guide you through the rest of the process.

To use the Motor Control GUI, go to the Windows Start menu, then Programs, then Motor Control GUI, and click on the Freescale icon. The Motor Control Graphic User Interface (GUI) will appear. The GUI is shown in Figure 3. 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.
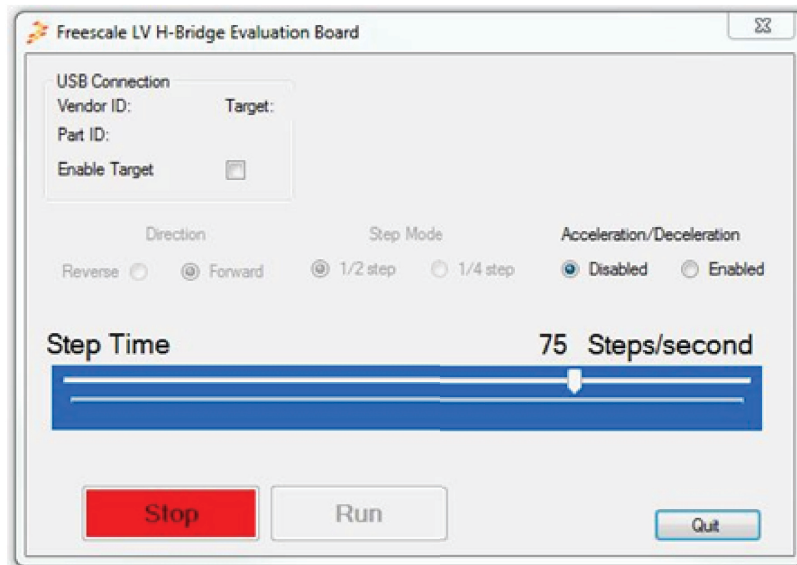
Figure 3. Motor Control GUI

5.2 Configuring the Hardware

Figure 4 shows the configuration diagram for FRDM-17529EV-EVB.
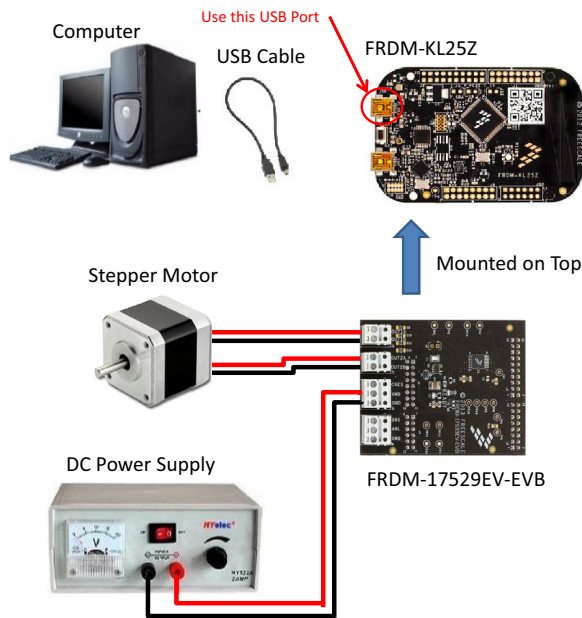
Figure 4. FRDM-17529EV-EVB plus FRDM-KL25Z Board Setup

5.2.1 Step-by-step Instructions for Setting Up the Hardware Using Motor Control GUI

When using the FRDM-17529EV-EVB make sure that the following operating parameters are followed or damage may occur.

- The maximum motor supply voltage (VM) cannot exceed 6.8 V, and must be at least 3.3 V
- The nominal operating current of the stepper motor cannot exceed 0.7 A (1.4 A peak)

In order to perform the demonstration example, first set up the evaluation board hardware and software as follows:

1. Setup the FRDM-KL25Z to accept code from the mbed online compiler. The instructions are at mbed.org (mbed.org/handbook/mbed-FRDM-KL25Z-Upgrade) (you will need to switch to the other USB port on the FRDM-KL25Z, and back after you load the project). mbed is a developer site for ARM based microcontrollers.

2. Go to the Freescale page on mbed.org and look for the repository named "LVHB DC Motor Drive". (developer.mbed.org/teams/Freescale/code/LVHB-Stepper-Motor-Drive/) Save the compiled code on your local drive, and then drag and drop it onto the mbed drive (which is the FRDM-KL25Z). Move the USB connector back to the other USB port on the FRDM-KL25Z.
Note: You may be asked to create a user before you can download the code.

3. Connect the FRDM-17529EV-EVB to the FRDM-KL25Z. This is best accomplished by soldering the female connectors to the FRDM-KL25Z, and then connecting to the male pins provided on the FRDM-17529EV-EVB.

4. Ready the computer, install the **Stepper Motor Driver GUI Software** (See Section 5.1).

5. Attach DC power supply (without turning on the power) to the VM and GND terminals.

6. Attach one set of coils of the stepper motor to the OUT 1A and OUT 1B output terminals. Attach the other phase coil of the stepper motor to terminals OUT2A and OUT2B. Launch the **Stepper Motor Driver GUI** Software.

7. Make sure the GUI recognizes the FRDM-KL25Z. This is determined by seeing the hex Vendor ID (0x15A2), and Part ID (0x138) under USB connection in the upper left hand corner of the GUI. If the GUI does not recognize the FRDM-KL25Z, you need to disconnect and reconnect the USB cable to the FRDM-KL25Z.

8. Turn on the DC power supply.

KTFRDM17529EVUG, Rev. 2.0

9. Click on the **Enable Target** checkbox on the GUI. The demo is now ready to run.

10. Click the **Run** button to run the motor. Notice that some options of the GUI are disabled while the motor is running. To make changes, click the **Stop** button on the GUI, make the desired changes, and then click **Run** on the GUI to continue.

11. When finished, click **Enable Target** button on the GUI, and then **Quit**. Turn off DC power supply. Remove USB cable.

6 Installing the 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.
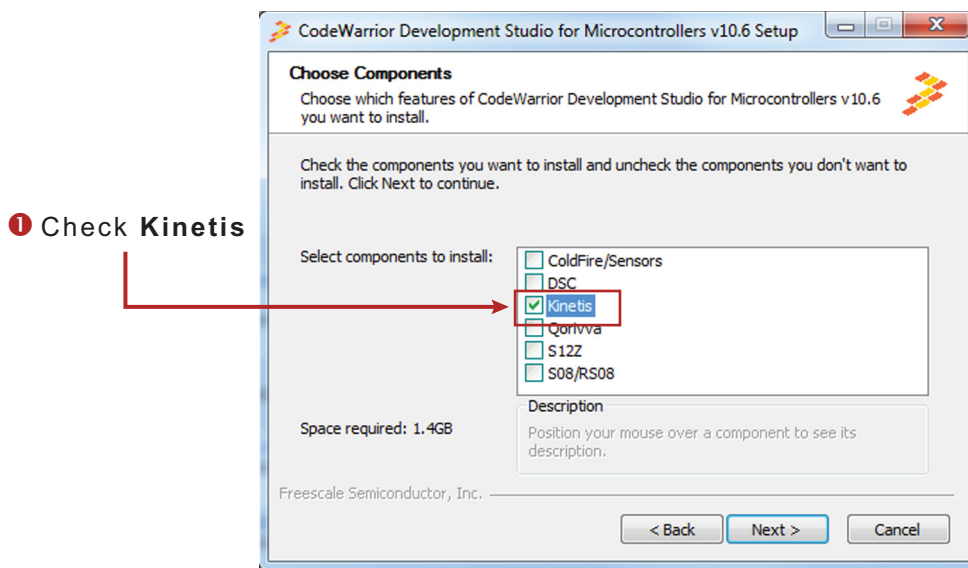
Figure 5. Select Components GUI

6.2 Downloading the LVHBridge 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/LVHBRIDGE-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 9.

Table 9. LVHBridge Example Project and Components

Folder Name	Folder Contents
CodeWarrior_Examples	Example project folder for CodeWarrior.
LVH_KL25Z_brush_MC34933	Example project for DC brush motor control using FRDM-34933EP-EVB H-Bridge board and FRDM-KL25Z MCU board
LVH_KL25Z_brush_MPC17510	Example project for DC brush motor control using FRDM-17510EJ-EVB H-Bridge board and FRDM-KL25Z MCU board
LVH_KL25Z_stepper	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM-KL25Z MCU board
LVH_KL25Z_stepper_ramp	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM-KL25Z MCU board. Acceleration ramp is enabled
Component	Processor Expert component folder
KDS_Examples	Example project folder for Kinetis Design Studio 3.0.0 or newer.
LVH_K20D50M_brush_MC34933	Example project for DC brush motor control using FRDM-34933EP-EVB H-Bridge board and FRDM-K20D50M MCU board
LVH_K20D50M_brush_MPC17510	Example project for DC brush motor control using FRDM-17510EJ-EVB H-Bridge board and FRDM-K20D50M MCU board
LVH_K20D50M_stepper_bitIO	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM- K20D50M MCU board
LVH_K20D50M_stepper_ramp_bitIO	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM- K20D50M MCU board. Acceleration ramp is enabled
LVH_KL25Z_brush_MC34933	Example project for DC brush motor control using FRDM-34933EP-EVB H-Bridge board and FRDM-KL25Z MCU board
LVH_KL25Z_brush_MPC17510	Example project for DC brush motor control using FRDM-17510EJ-EVB H-Bridge board and FRDM-KL25Z MCU board
LVH_KL25Z_brush_FreeMASTER	Example project intended to control DC brush motor using FreeMASTER tool. Latest Freemaster installation package: www.freescale.com/freemaster
LVH_KL25Z_step_FreeMASTER	Example project intended to control stepper motor using FreeMASTER tool
LVH_KL25Z_stepper	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM-KL25Z MCU board
LVH_KL25Z_stepper_ramp	Example project intended to control stepper motor using MC34933 H-Bridge freedom board and FRDM-KL25Z MCU board. Acceleration ramp is enabled
LVH_KL26Z_stepper	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM-KL26Z MCU board
LVH_KL26Z_stepper_iar	Example project intended to control stepper motor using FRDM-34933EP-EVB H-Bridge board and FRDM-KL26Z MCU board. IAR compiler is used instead of GNU C compiler

6.2.1 Import the LVHBridge Component into Processor Expert Library

1. Launch CodeWarrior by clicking on the CodeWarrior icon (located on your desktop or in Program Files -> Freescale Codewarrior folder). When the CodeWarrior IDE opens, go to the menu bar and click **Processor Expert** -> **Import Component(s)**.

2. In the pop-up window, locate the component file (.PEupd) in the example project folder LVHBridge_PEx_SW\Component. Select **LVHBridge_b1508.PEupd** and **ChannelAllocator_b1508.PEupd** files then click **Open** (see Figure 6).

❶ Click **Processor Expert** ❷ Select **Import Component(s)**

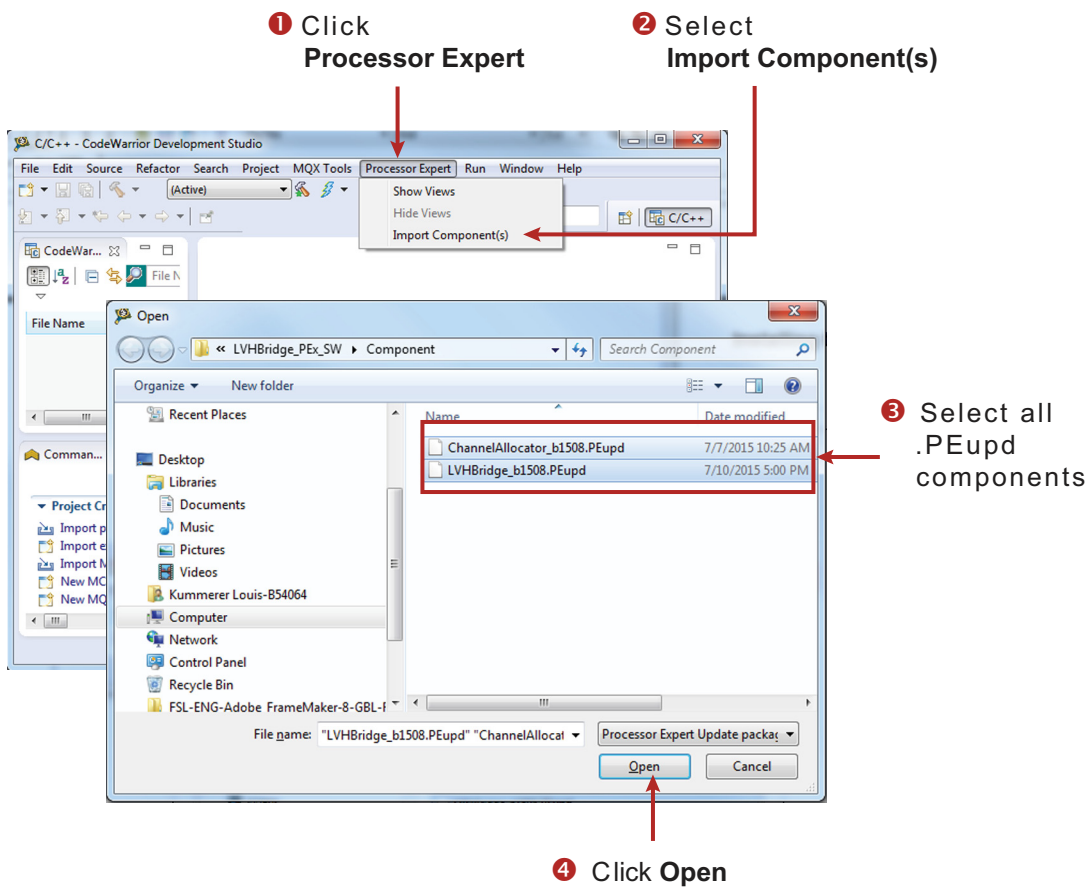❸ Select all .PEupd components

❹ Click **Open**

Figure 6. Import LVHBridge Component

3. If the import is successful, the LVHBridge component appears in Components Library -> SW -> User Component (see Figure 7). Note that the component **ChannelAllocator** is not visible, because it is not designed to be users accessible.
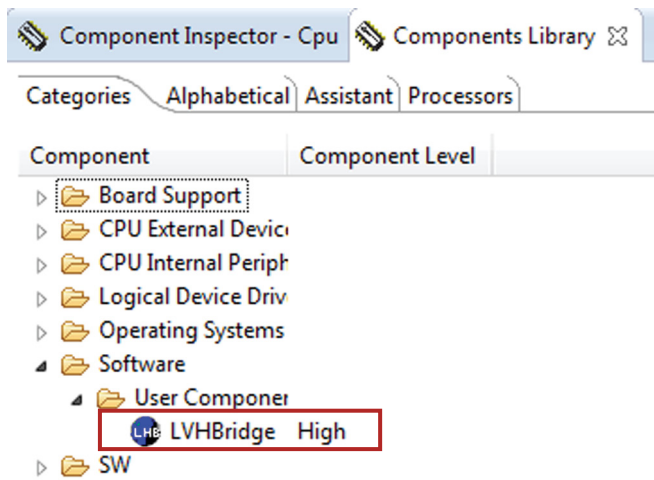

Figure 7. LVHBridge Component Location after CodeWarrior Import

The LVHBridge component is ready to use.

6.2.2   Import an Example Project into CodeWarrior

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.**

2. Locate the example in folder: LVHBridge_PEx_SW\CodeWarrior_Examples (see Figure 8, which shows LVH_KL25Z_brush_MC34933 as the imported project). Then click **Finish**.

The project is now in the CodeWarrior workspace where you can build and run it.
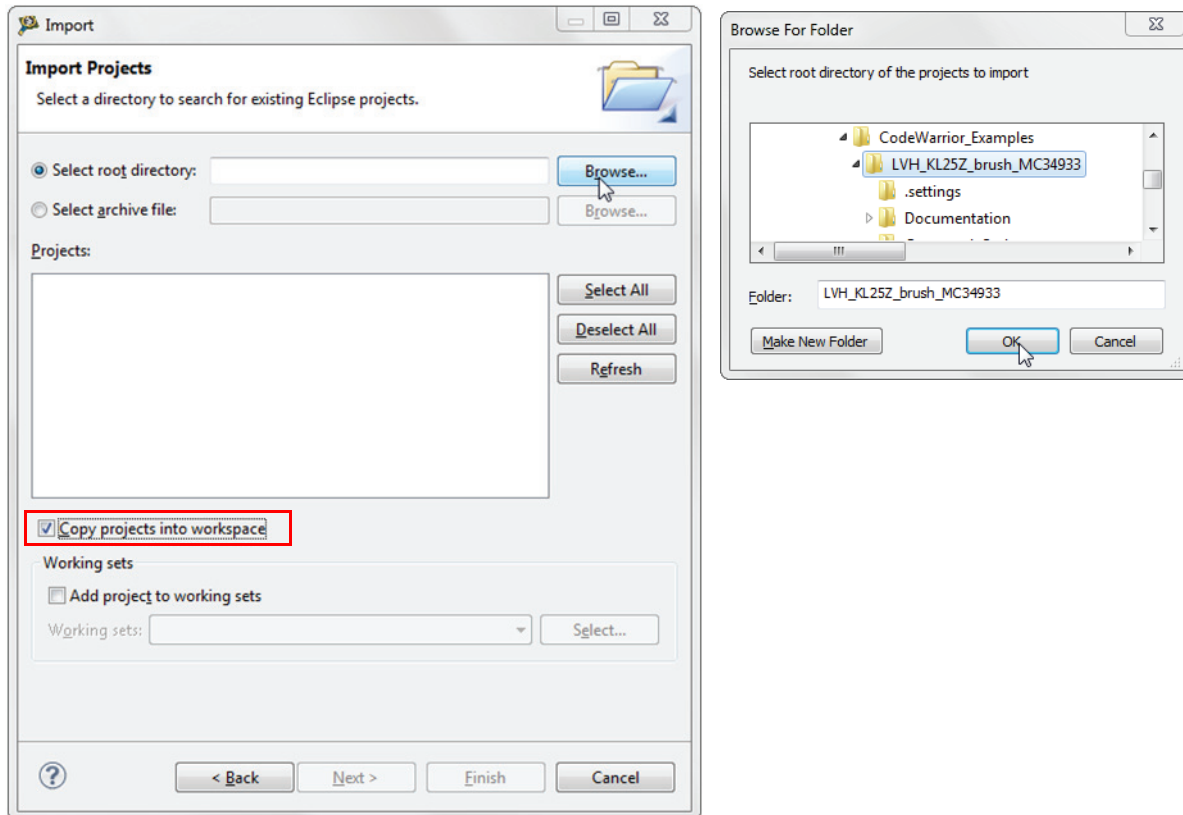
Figure 8. Example Project Import

6.3 Create a New Project with Processor Expert and LVHBridge 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 LVHBridge component. If you do not have the LVHBridge component in the Processor Expert Library, please follow steps in Section 6.2.1.

1. Create and name an MCU Bareboard project (see Figure 9).
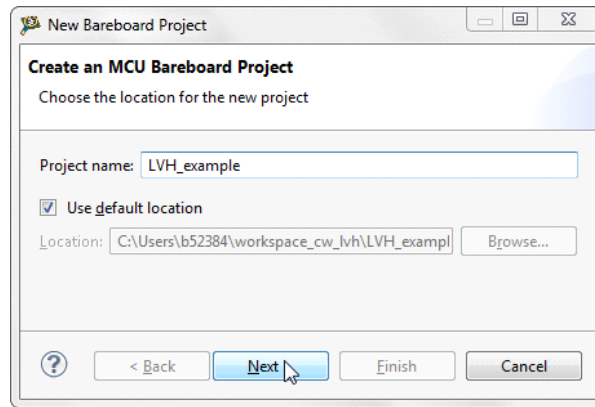
Figure 9. Create an MCU Bare-board Project

2. Choose the MCU class to be used in the freedom MCU board (MKL25Z128 in this example). Then select the connections to be used (see Figure 10).
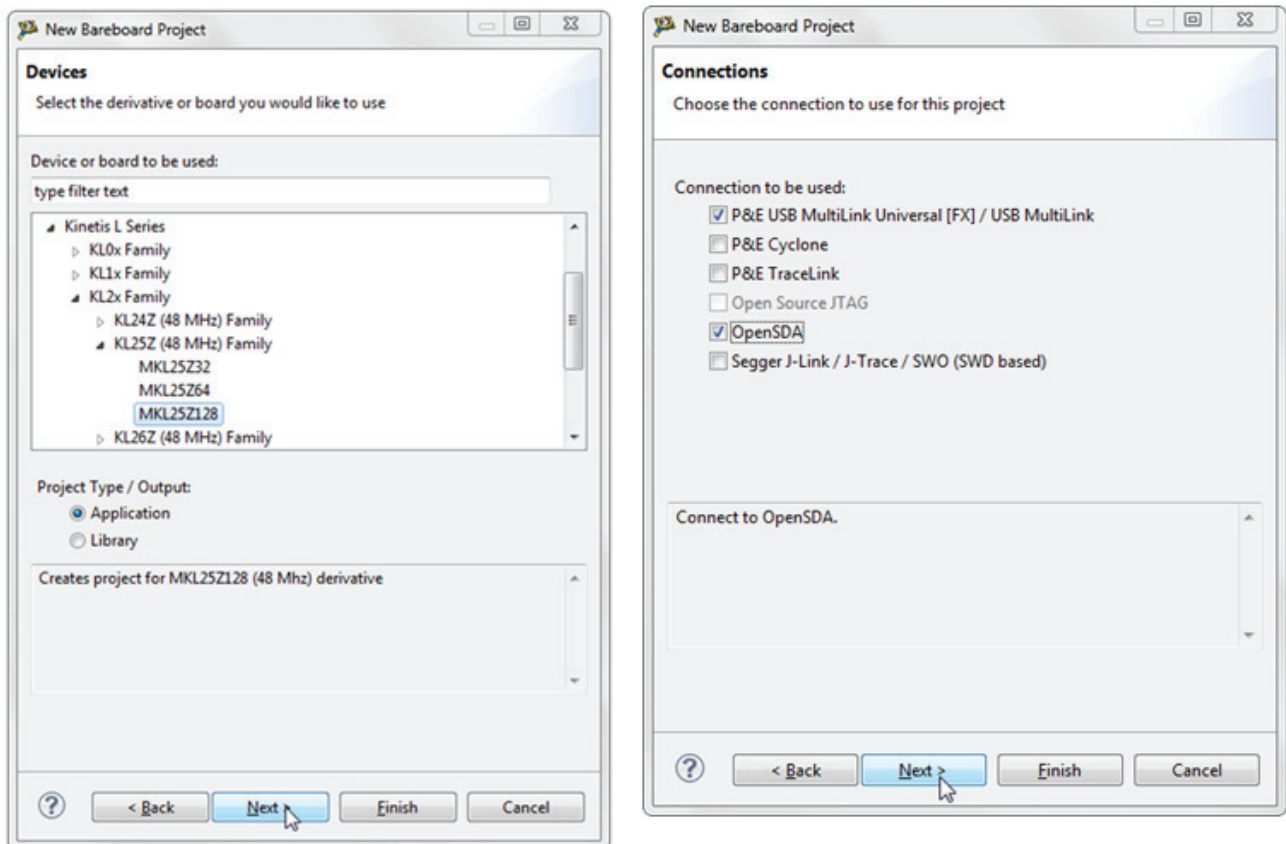
Figure 10. Select the MCU Class and Connections

KTFRDM17529EVUG, Rev. 2.0

3. Select the **Processor Expert** option, and then click **Finish** (see Figure 11).
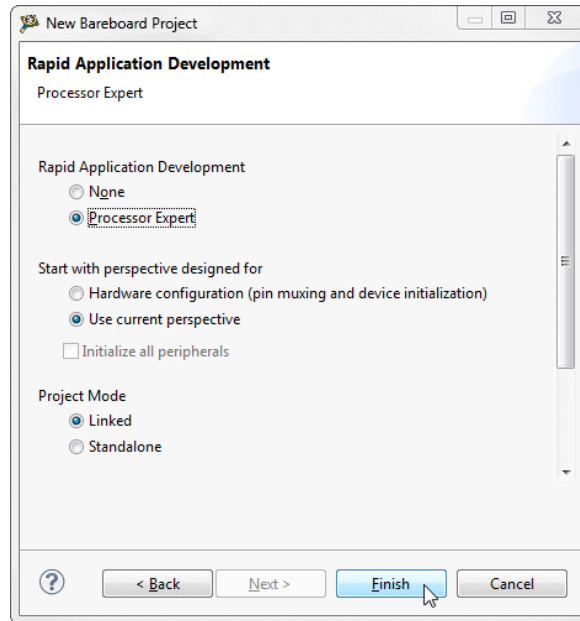
Figure 11. Select the Processor Expert Option

6.3.1   Add LVHBridge Component into the Project

1. Find LVHBridge in the **Components Library** and add it into your project (see Figure 12).
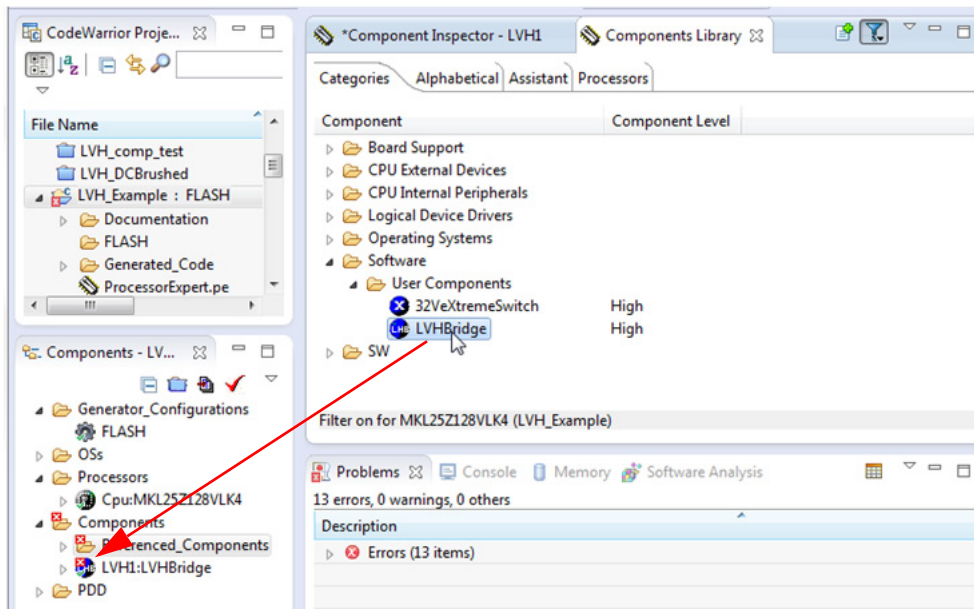
Figure 12. Add the LVHBridge Component to the Project

2. Double click LVHBridge component in the **Components** window (see Figure 13) to show the configuration in the **Component Inspector** view.
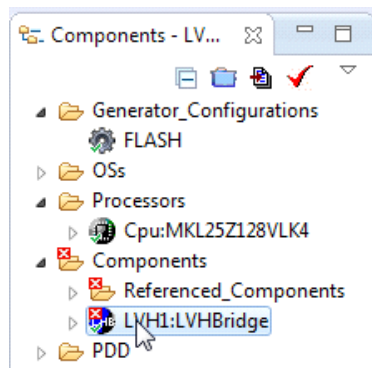
Figure 13. Select the Component

Figure 14. Component Inspector View

6.3.2 General Settings of LVHBridge Component

Component settings in the **Component Inspector** view have a tree structure. H-Bridge Model is on top of the tree.

ActiveMode defines the H-Bridge device operational mode (normal or power-conserving sleep mode), which is controlled by the enabling pin. Selection of the enabling pin is in the **Enable Pins** group. For more information, see your H-Bridge model's data sheet. The mode can be changed later using the C code method **SetMode**.

The **Motor Control** group involves timer settings, H-Bridge device and motor control settings. The **Timer Settings** group contains the **Primary Timer Component** property (the name of a linked **TimerUnit_LDD** component) and the name of the hardware timer being used (defined in the **Primary Timer Device** property). **Secondary Timer** encompasses the properties of an additional timer.

Note that the **Secondary Timer Component** property must use a different **TimerUnit_LDD** component than the **Primary Timer Component** property. The purpose of the primary and secondary timers is to allow the input control pins of an H-Bridge device to be connected to different timers (this applies for some freedom H-Bridge boards and freedom MCUs). But these timers must be synchronized to control a stepper motor. So the primary timer is designed to be the source for the global time base and the secondary timer is synchronized with the primary timer. Please see your MCU's data sheet to find out which timer provides the global time base (GTB) and set the **Primary Timer Device** property accordingly. An example of a timer selection using the FRDM-KL25Z MCU is shown in Figure 15. If you are using a single timer, set the **Secondary Timer Component** to **Disabled**.

⊿ **Timer Settings**	Enabled	
Primary Timer Component	TU1	
Primary Timer Device	TPM1_CNT	TPM1_CNT
⊿ **Secondary Timer**	Enabled	
Secondary Timer Component	TU2	
Secondary Timer Device	TPM0_CNT	TPM0_CNT

Figure 15. Selection of a FRDM-KL25Z MCU Primary and a Secondary Timer Device

H-Bridge 1 MCU Interface and H-Bridge 2 MCU Interface allow you to set H-Bridge control function. The H-Bridge 2 MCU Interface is shown only for dual H-Bridge models (for example MC34933). The DC Brush group is described in Section 6.3.3. The **Input Control Pins** allow you to select the H-Bridge input control pins that utilize the timer's channels or GPIO pins.

⊿ **H-Bridge 1 MCU Interface**		
⊿ **DC brush**		
⊿ **Control Mode**	Speed Control	
PWM Frequency	10 kHz	10.001 kHz
Direction Control	Bidirectional	
Init. Direction	Forward	
⊿ **Input Control Pins**	Two PWM Pins	
Pin for IN1A	PTD4/LLWU_P14/SPI1_PCS0/UART...	
Pin for IN1B	PTA12/TPM1_CH0	
⊿ **H-Bridge 2 MCU Interface**	Enabled	
⊿ **DC brush**		
Control Mode	State Control	
Init. Direction	Forward	
⊿ **Input Control Pins**	Two GPIO Pins	
Pin for IN2A	TSI0_CH5/PTA4/I2C1_SDA/TPM0_...	
Pin for IN2B	PTA5/USB_CLKIN/TPM0_CH2	
Auto Initialization	yes	

Figure 16. LVHBridge Component — General Settings

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 17. Setup of the Component to Control a Brush Motor

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

   a) **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 1 and Interface 2), the **PWM Frequency** property on Interface 2 will be set automatically to the same value as Interface 1 (because Interface 2 uses the same timer.)

   b) **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 **TimerUnit_LDD** component is not required anymore by the LVHBridge 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   Setting up a Project to Control a Stepper Motor

Select the dual H-Bridge model you want to configure and set **Stepper** in the **Motor Control** property. Note that the dual H-Bridge model is required, because a two phase bipolar stepper motor has four inputs.

Figure 18. Component Settings to Control a Stepper Motor

In the **Stepper Motor** group, set the properties that apply to your environment.
- The **Output Control** property defines the control method.
   a) With **PWM** selected the component utilizes four channels of a timer to control the stepper motor. Signal is generated in hardware and micro-step mode is also available.
   b) In **GPIO** mode, GPIO pins are used instead of timer channels and only full-step mode is available (no micro-step mode).
- **Manual Timer setting** property is only visible when you switch the visibility of the component properties to Advanced (see later). It is designed to change the Counter frequency of the linked TimerUnit_LDD component. By default the Counter frequency is set automatically by LVHBridge component. In some cases the frequency value does not have to be set appropriately (user wants to set a different value or there an error has occurred). For more information see Section 6.3.5.
- **Motor Control Mode** allows you to select the Step Mode. Selecting Full-step and Micro-step mode allows you to switch between full-stepping and micro-stepping in C code.
   a) **Full-step Configuration** contains speed and acceleration settings. Code for the acceleration and deceleration ramp is generated when the **Acceleration** property is set to a value greater than zero. Note that acceleration is always the same as deceleration. An example of an acceleration ramp is depicted in Figure 19. The acceleration setting is 400, as shown in Figure 18.
      - Desired motor speed is set to 100 full-steps per second. This value is defined by the **Speed** property in Processor Expert GUI and can be changed in C code.
      - Acceleration and deceleration is set to 400 full-steps per second[2]. This value is defined by the Acceleration property. Note that the motor reaches the speed in 0.25 second (desired_speed / acceleration = 100 / 400 = 0.25).
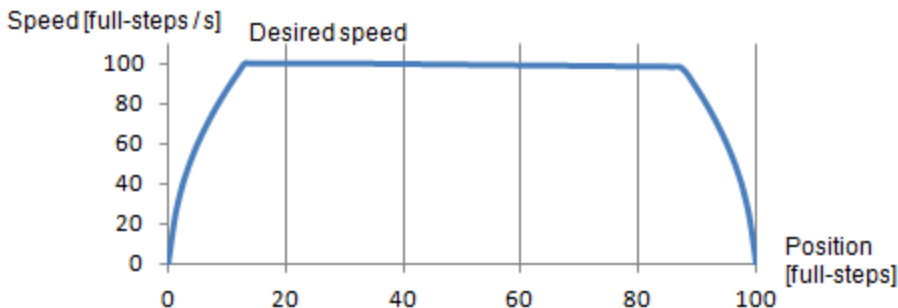
KTFRDM17529EVUG, Rev. 2.0

Figure 19. Acceleration and Deceleration Ramp

    b)   **Micro-step Configuration** settings are similar to those of the Full-step Configuration. **PWM Frequency** is the frequency of the micro-step PWM signal. **Micro-step per Step** is the number of micro-steps per one full-step.

6.3.5 Stepper Motor Speed

This component defines the stepper motor's minimum and maximum speed. These limit values are used by the component methods. Minimum speed in full-step and micro-step modes is one step per second. Maximum speed is 5000 steps per second. There is a specific case in which minimum full-stepping speed is affected by timer input frequency. This applies only when you are using one FTM timer to control the stepper motor. In this case, the **Primary Timer Device** property must use FTM timer values (FTM0_CNT, or FTM1_CNT, etc.). The **Secondary Timer** property must be set to **Disabled**. The Stepper Motor **Output Control** property must be set to **PWM**. Figure 20 illustrates this configuration.
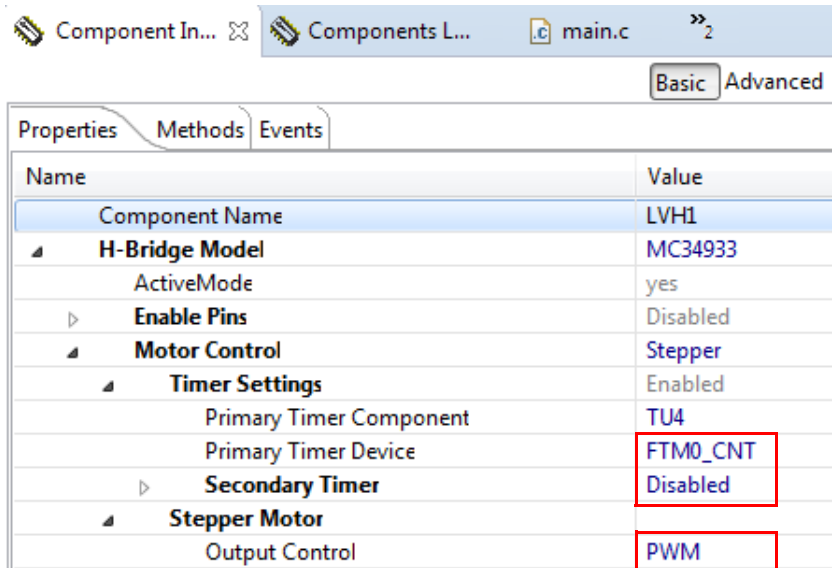

Figure 20. Stepper Mode Configuration Affecting Minimum Full-stepping Speed

Possible values for the timer input frequency (**Counter frequency** property in **TimerUnit_LDD**) are in Table 10. Input frequency values depend on LVHBridge component settings. Note that two frequency values are needed in "Full-step and Micro-step mode" in one case (LVHBridge component switches in runtime between these two values).

Table 10. Minimum and Maximum Timer Input Frequency per Stepper Control Mode

Mode Description	LVHBridge component properties				Primary Timer Input Frequency			Secondary Timer Input Frequency
	Timer Device	Secondary Timer	Output Control	Motor Control Mode	Values	Min	Max	
Full-step mode	TPM	Don't care	PWM	Full-step	1	131 kHz	1 MHz	Any value (user selection)
Full-step and Micro-step mode	TPM	Don't care	PWM	Full-step and Micro-step	1	1.2 MHz	10 MHz	Any value (user selection)
Full-step mode (SW control)	FTM or TPM	Disabled	GPIO	Full-step	1	131 kHz	1 MHz	Secondary timer is not enabled
Full-step mode	FTM	Disabled	PWM	Full-step	1	131 kHz	1 MHz	Secondary timer is not enabled
Full-step mode	FTM	Enabled	PWM	Full-step	1	131 kHz	1 MHz	The same values as for primary timer
Full-step and Micro-step mode	FTM	Disabled	PWM	Full-step and Micro-step	2	1st value for Full-step: 131 kHz / 2nd value for Micro-step:1.2 MHz	1st value for Full-step: 1 MHz / 2nd value for Micro-step:10 MHz	Secondary timer is not enabled
Full-step and Micro-step mode	FTM	Enabled	PWM	Full-step	1	1.2 MHz	10 MHz	The same values as for primary timer

6.3.5.1   Computation of Minimum Full-stepping Speed

The minimum full-stepping speed depends on the timer input frequency only when the **Primary Timer Device** is set to FTM (FTM0_CNT, or FTM1_CNT, etc.), the **Secondary Timer** property is disabled and **Output Control** is set to **PWM**. The Full-step signal is generated by a timer while channels toggle on compare (See Figure 21).
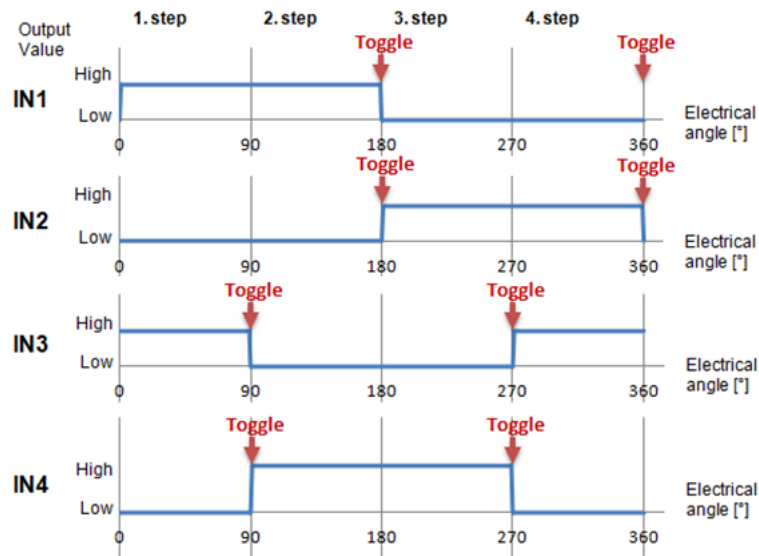
Figure 21. Generating the Full-step Control Signal

The Full-step minimum speed is derived from the input frequency of the timer device (the counter frequency property of the **TimerUnit_LDD** component being used). You can find minimum values for speed in the LVHBridge header file (see constant <component_name>_MIN_FULLSTEP_ SPEED). The formula for calculation of this value is as follows:

$$Speed_{min} = \frac{2 \ X \ Counter\_frequency}{65536} + 1$$

where:

Counter_frequency = input frequency of the timer device

65536 = maximum value of **TimerUnit_LDD** counter (16-bit counter).

Adding 1 ensures that the 16-bit counter does not overflow (which is the point of the formula.)

For example if the Counter frequency is set to 187,500 Hz, the minimum speed is:

$$Speed_{min} = \frac{2 \ X \ Counter\_frequency}{65536} + 1 = \frac{2 \ X \ 187500}{65536} + 1 = 6.72$$

The MCU rounds the value down, so the result is 6 full-steps per second.

6.3.5.2   Setting the Minimum Full-stepping Speed

This section describes how to change the input frequency of the **TimerUnit_LDD** component.

1. Launch Processor Expert and select the LVHBridge component.

2. In the Processor Expert menu bar, set component visibility to **Advanced**.

3. In the Properties tab, find the **Motor Control** -> **Stepper Motor** -> **Manual timer** setting property and set the value to **Enabled**. If you do not see this property, make sure that component visibility is set to Advanced (see Figure 22).

4. Set the **TimerUnit_LDD** frequency

    a)   In the Components view, double click on the **TimerUnit_LDD** component.

    b)   Press the button in the **Counter frequency** field.

    c)   Set the frequency value (187.5 kHz in the illustration). The list of available frequencies depends on the CPU component settings (with an external crystal as the clock source and a core clock of 48 MHz).

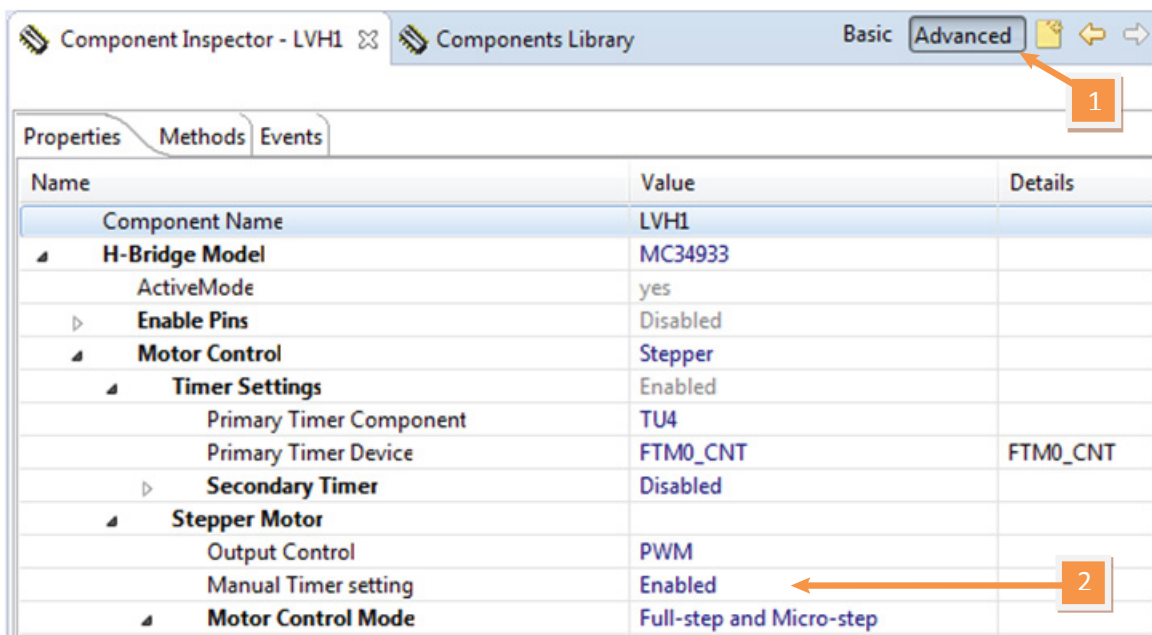   d)   Set the **Allowed Error** value at 10% (see Figure 24).
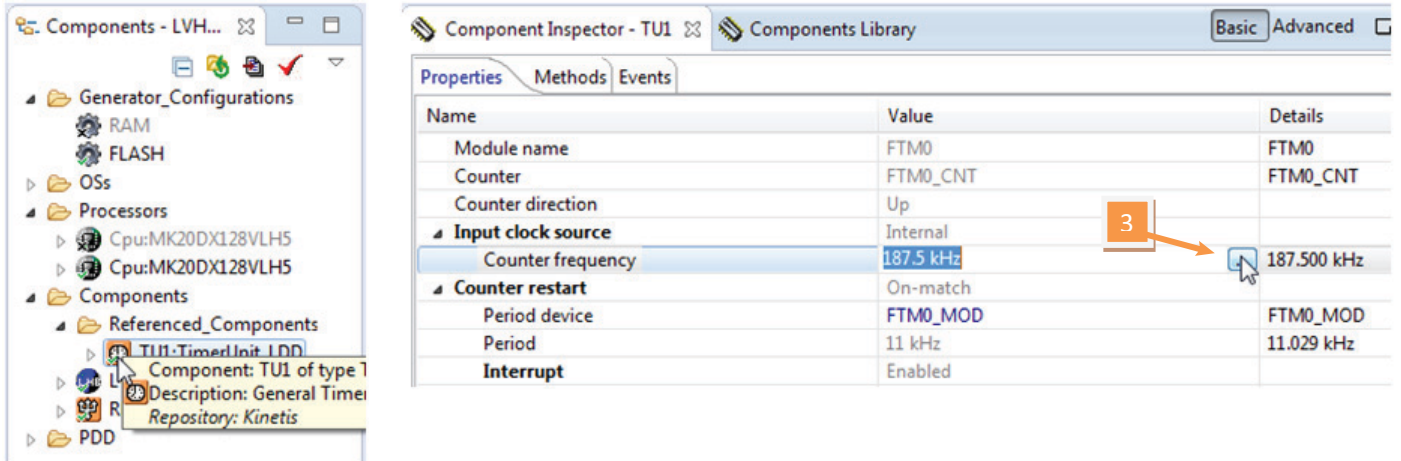

Figure 22. Enabling the Manual Frequency Setting

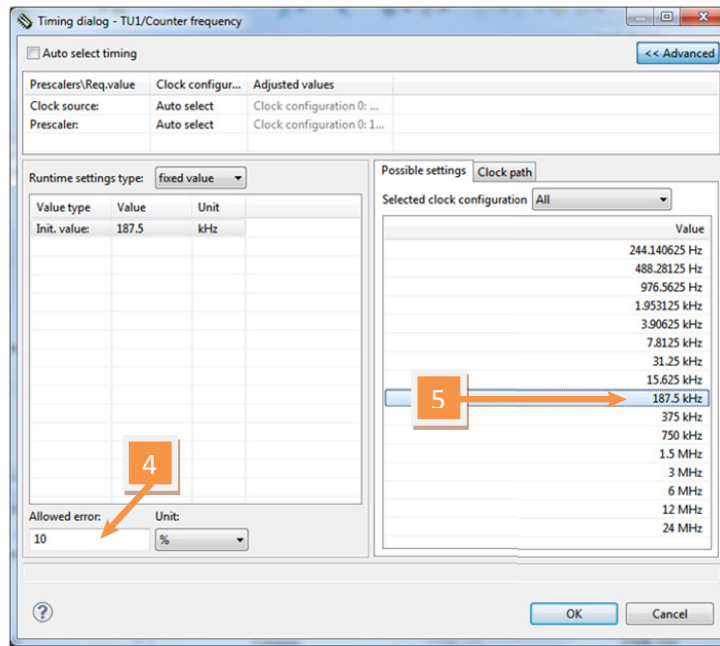**Figure 23. Component TimerUnit_LDD Timing Dialog**

Figure 24. Component TimerUnit_LDD Timing Dialog—Select Input Frequency

6.3.6 Generating Application Code

After configuration, generate the source code by clicking on the icon in the upper right corner of the Components screen.
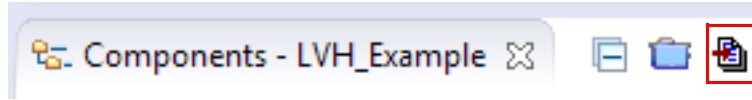

Figure 25. Generating the Source Code

The driver code for the H-Bridge device is generated into the **Generated_Code** folder in the project view. The component only generates application driver code. It does not generate application code.
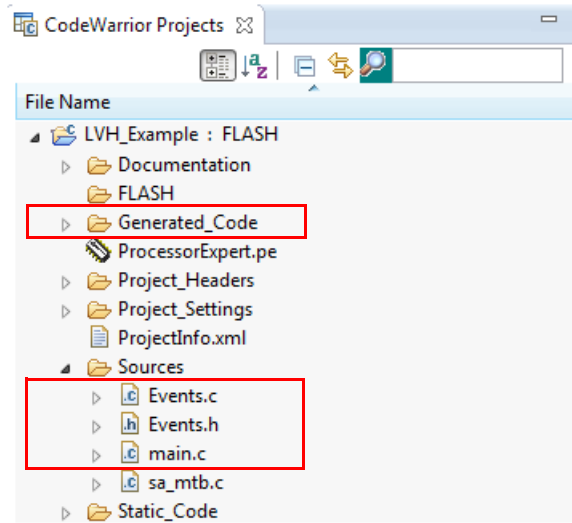

Figure 26. Generated Files

6.3.7 Using the Interface

Application code can be written and tested in the project. For example, you can open the LVHBridge component method list, drag and drop **RotateProportional** to **main.c** (see Figure 27), add any necessary parameters, then compile the program.

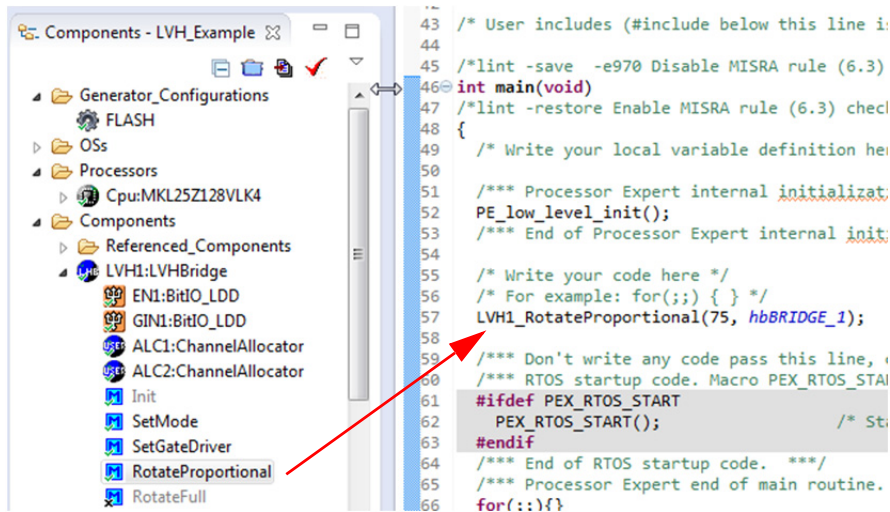
Figure 27. Using the Interface

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 28).
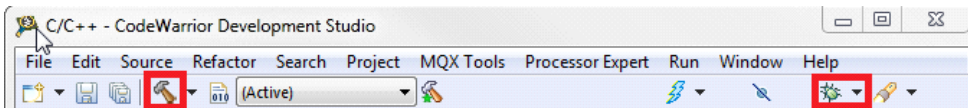

Figure 28. Compile and Download the Application

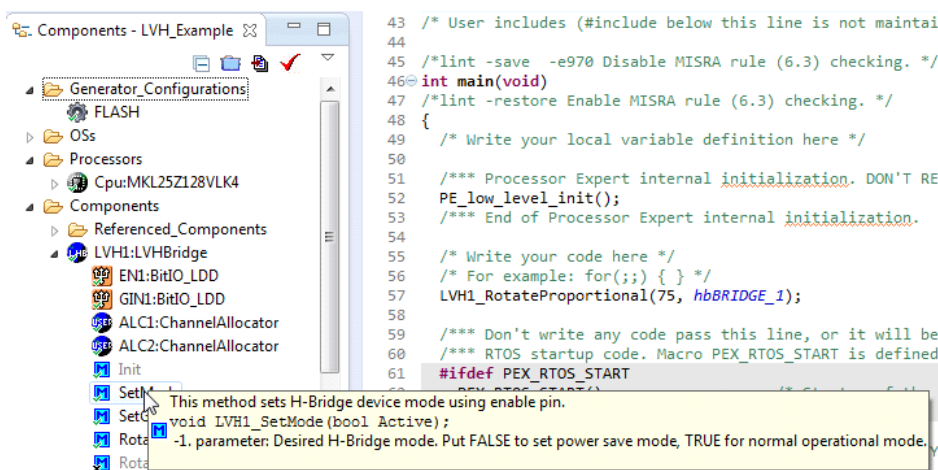A description of each LVHBridge method appears in the pop-up window (see Figure 29).

Figure 29. LVHBridge Method Information

6.4 Stepper Motor Control Application Notes

The LVHBridge component is designed to control a two phase bipolar stepper motor. Because a stepper motor uses electrical commutation to rotate, it requires a dual H-Bridge device. The basic control method is full-stepping which fully powers each coil in sequence. Increased precision is achieved by using the PWM to control coil current (open loop control). This method is called micro-stepping (available in the LVHBridge component.)

In both micro-step and full-step mode you can control motor speed, direction, acceleration and deceleration and the position of the stepper motor.

The following application notes apply to stepper motor control:

- The LVHBridge component was tested with a core clock frequency ranging from 20 MHz (minimum value) to 120 MHz.
- Do not change the settings of the timer device (**TimerUnit_LDD**) linked by the LVHBridge component. The component sets the timer device automatically.
- The acceleration and deceleration ramp of the stepper motor is computed in real-time using integer arithmetic. This solution is based on the article "Generate stepper-motor speed profiles in real time" (Austin, David. 2005.)
- The stepper motor holds its position (coils are powered) after motor movement is completed. Use method **DisableMotor** to set H-Bridge outputs to LOW (coils are not powered).
- Forward motor direction indicates that steps are executed in the order depicted in Figure 30. IN1 through IN4 are the input pins of the H-Bridge device which control H-Bridge outputs. These pins input to the stepper motor. You must connect the stepper motor to output pins OUT1-OUT4 and select control input pins on your MCU in the component settings.
- The FTM or TPM timer device is needed by the stepper control logic.
- The **AlignRotor** method affects the position of the motor. This method executes four full-steps. It is available only when full-step mode is enabled.

6.4.1 Full-step Control Mode

The component uses normal drive mode where two coils are powered at the same time.

As mentioned in Section 6.3.4, you can generate a full-stepping signal either by using four channels of a timer or by using four GPIO pins. The signal generated by the MCU (inputs of H-Bridge device) using four timer channels is shown in Figure 30. The voltage levels applied to the coils of the stepper motor are depicted in Figure 31. Note that the voltage is applied to both coils at the same time.
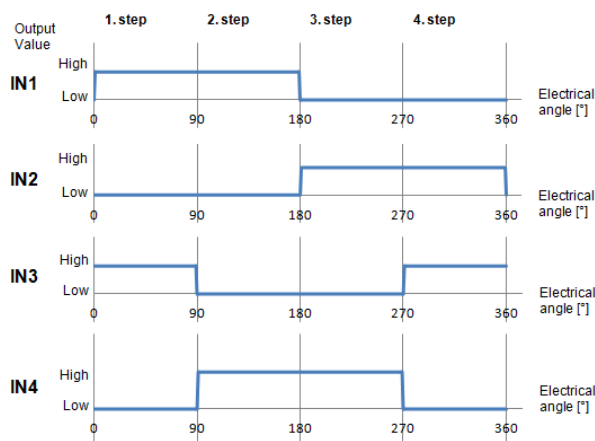

Figure 30. Signals of Logic Input Pins Generated by the MCU in Full-step Mode
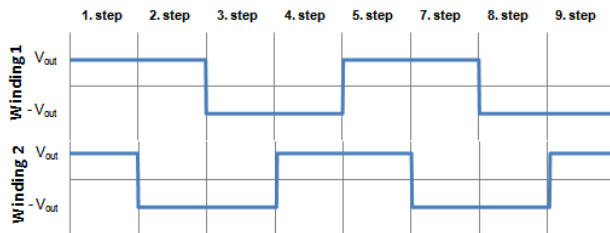
Figure 31. Output of the H-Bridge Device in Full-step Mode

6.4.2   Micro-step Control Mode

Micro-stepping allows for smoother motor movement and increased precision. The current varies in motor windings A and B depending on the micro-step position. A PWM signal is used to reach the desired current value (see the following equations). This method is called sine cosine micro-stepping.

$$I_A = I_{MAX} \times \sin(\theta)$$
$$I_B = I_{MAX} \times \cos(\theta)$$

where:

I_A = the current in winding A

I_B = the current in winding B,

I_{MAX} = the maximum allowable current

θ = the electrical angle

In micro-step mode, a full-step is divided into smaller steps (micro-steps). The LVHBridge component offers 2, 4, 8, 16, and 32 micro-steps per full-step. The micro-step size is defined by the property "Micro-steps per Step" and can be changed later in C code.

Figure 32. Micro-stepping phase diagram

KTFRDM17529EVUG, Rev. 2.0

Table 11. Micro-Step Phase Table[1]
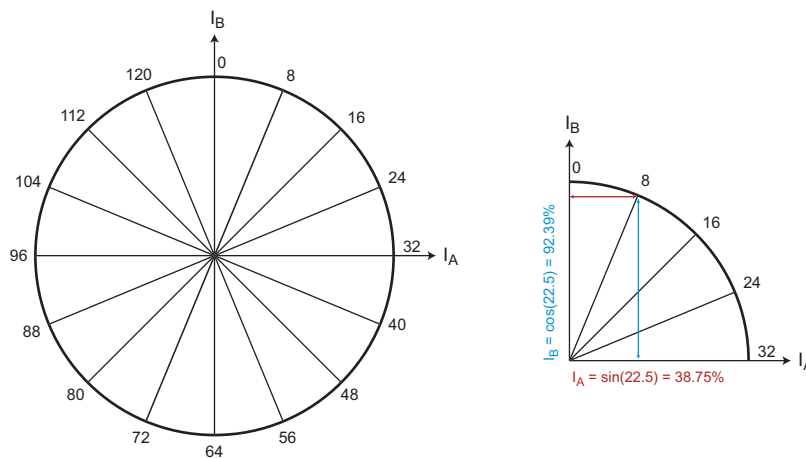
Micro-step size					Angle	I [% of I_{MAX}]	
1/2	1/4	1/8	1/16	1/32		A	B
0	0	0	0	0	0.0	0	100
				1	2.8	4.91	99.88
		1		2	5.6	9.8	99.52
				3	8.4	14.67	98.92
		1	2	4	11.3	19.51	98.08
				5	14.1	24.3	97
			3	6	16.9	29.03	95.69
				7	19.7	33.69	94.15
	1	2	4	8	22.5	38.27	92.39
				9	25.3	42.76	90.4
			5	10	28.1	47.14	88.19
				11	30.9	51.41	85.77
		3	6	12	33.8	55.56	83.15
				13	36.6	59.57	80.32
			7	14	39.4	63.44	77.3
				15	42.2	67.16	74.1
1	2	4	8	16	45	70.71	70.71
				17	47.8	74.1	67.16
			9	18	50.6	77.3	63.44
				19	53.4	80.32	59.57
		5	10	20	56.3	83.15	55.56
				21	59.1	85.77	51.41
			11	22	61.9	88.19	47.14
				23	64.7	90.4	42.76
	3	6	12	24	67.5	92.39	38.27
				25	70.3	94.15	33.69
			13	26	73.1	95.69	29.03
				27	75.9	97	24.3
		7	14	28	78.8	98.08	19.51
				29	81.6	98.92	14.67
			15	30	84.4	99.52	9.8
				31	86.4	99.8	6.3
2	4	8	16	32	90	100	0.00
				33	92.8	99.88	-4.91
			17	34	95.6	99.52	-9.8
				35	98.4	98.92	-14.67
		9	18	36	101.3	98.08	-19.51
				37	104.1	97	-24.3
			19	38	106.9	95.69	-29.03
				39	109.7	94.15	-33.69
	5	10	20	40	112.5	92.39	-38.27
				41	115.3	90.4	-42.76
			21	42	118.1	88.19	-47.14
				43	120.9	85.77	-51.41
		11	22	44	123.8	83.15	-55.56
				45	126.6	80.32	-59.57
			23	46	129.4	77.3	-63.44
				47	132.2	74.1	-67.16
3	6	12	24	48	135	70.71	-70.71
				49	137.8	67.16	-74.1
			25	50	140.6	63.44	-77.3

Micro-step size					Angle	I [% of I_{MAX}]	
1/2	1/4	1/8	1/16	1/32		A	B
4	8	16	32	64	180	0	-100
				65	182.8	-4.91	-99.88
			33	66	185.6	-9.8	-99.52
				67	188.4	-14.67	-98.92
		17	34	68	191.3	-19.51	-98.08
				69	194.1	-24.3	-97
			35	70	196.9	-29.03	-95.69
				71	199.7	-33.69	-94.15
	9	18	36	72	202.5	-38.27	-92.39
				73	205.3	-42.76	-90.4
			37	74	208.1	-47.14	-88.19
				75	210.9	-51.41	-85.77
		19	38	76	213.8	-55.56	-83.15
				77	216.6	-59.57	-80.32
			39	78	219.4	-63.44	-77.3
				79	222.2	-67.16	-74.1
5	10	20	40	80	225	-70.71	-70.71
				81	227.8	-74.1	-67.16
			41	82	230.6	-77.3	-63.44
				83	233.4	-80.32	-59.57
		21	42	84	236.3	-83.15	-55.56
				85	239.1	-85.77	-51.41
			43	86	241.9	-88.19	-47.14
				87	244.7	-90.4	-42.76
	11	22	44	88	247.5	-92.39	-38.27
				89	250.3	-94.15	-33.69
			45	90	253.1	-95.69	-29.03
				91	255.9	-97	-24.3
		23	46	92	258.8	-98.08	-19.51
				93	261.6	-98.92	-14.67
			47	94	264.4	-99.52	-9.8
				95	266.4	-99.8	-6.3
6	12	24	48	96	270	-100	0.00
				97	272.8	-99.88	4.91
			49	98	275.6	-99.52	9.8
				99	278.4	-98.92	14.67
		25	50	100	281.3	-98.08	19.51
				101	284.1	-97	24.3
			51	102	286.9	-95.69	29.03
				103	289.7	-94.15	33.69
	13	26	52	104	292.5	-92.39	38.27
				105	295.3	-90.4	42.76
			53	106	298.1	-88.19	47.14
				107	300.9	-85.77	51.41
		27	54	108	303.8	-83.15	55.56
				109	306.6	-80.32	59.57
			55	110	309.4	-77.3	63.44
				111	312.2	-74.1	67.16
7	14	28	56	112	315	-70.71	70.71
				113	317.8	-67.16	74.1
			57	114	320.6	-63.44	77.3

KTFRDM17529EVUG, Rev. 2.0

Table 11. Micro-Step Phase Table[1] (continued)

1/2	1/4	1/8	1/16	1/32	Angle	A	B	1/2	1/4	1/8	1/16	1/32	Angle	A	B
				51	143.4	59.57	-80.32					115	323.4	-59.57	80.32
		13	26	52	146.3	55.56	-83.15			29	58	116	326.3	-55.56	83.15
				53	149.1	51.41	-85.77					117	329.1	-51.41	85.77
			27	54	151.9	47.14	-88.19				59	118	331.9	-47.14	88.19
				55	154.7	42.76	-90.4					119	334.7	-42.76	90.4
	7	14	28	56	157.5	38.27	-92.39		15	30	60	120	337.5	-38.27	92.39
				57	160.3	33.69	-94.15					121	340.3	-33.69	94.15
			29	58	163.1	29.03	-95.69				61	122	343.1	-29.03	95.69
				59	165.9	24.3	-97					123	345.9	-24.3	97
		15	30	60	168.8	19.51	-98.08			31	62	124	348.8	-19.51	98.08
				61	171.6	14.67	-98.92					125	351.6	-14.67	98.92
			31	62	174.4	9.8	-99.52				63	126	354.4	-9.8	99.52
				63	176.4	6.3	-99.8					127	356.4	-6.3	99.8
4	8	16	32	64	180	0.00	-100	8	16	32	64	128	360	0.00	100

Notes:

1.   Shaded rows indicate one quarter step of the motor

The micro-stepping signal is generated using four timer channels (see Figure 33). Output from logic analyzer in Figure 34 shows the change of PWM duty with respect to the micro-step position. Current values applied to the stepper motor coils are depicted in Figure 35.
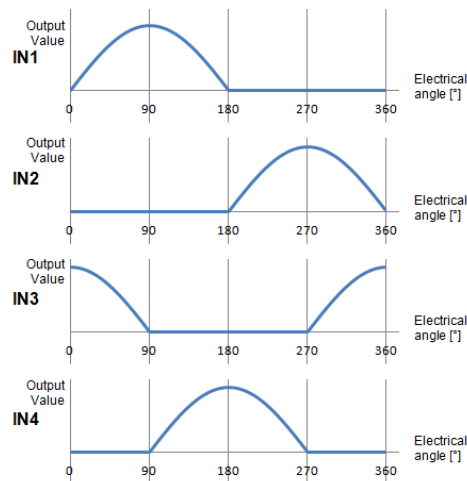

Figure 33. Logic Input Pin Signals Generated by the MCU in Micro-step Mode

Figure 34. Logic Analyzer Output
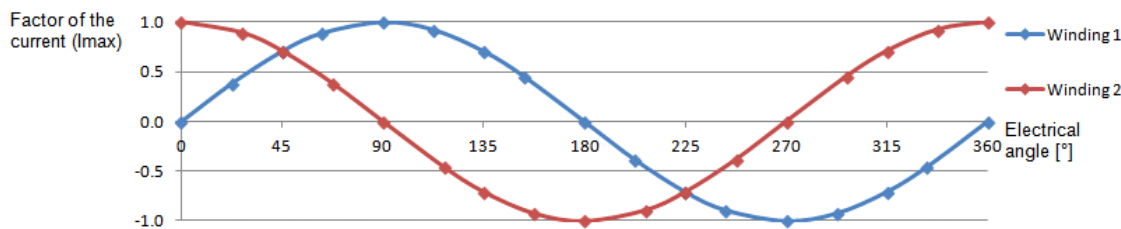
KTFRDM17529EVUG, Rev. 2.0

Figure 35. H-Bridge Device Output in Micro-step Mode

6.5   Frequently Asked Questions

Q: How do I set up the LVHBridge component when two or more components with conflicting values are configured to control brushed motors? (See Figure 36)

Figure 36. Conflict in the Required Values for Components in the Project

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

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 LVHBridge component in your project, the **TimerUnit_LDD** component channels have not been allocated. If this occurs, changing certain LVHBridge properties will force allocation of the channels. If you are configuring a stepper motor (**Motor Control** property set to **Stepper**), try changing the **Output Control** property to **GPIO** and then back to **PWM**. If 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 1 or interface 2.
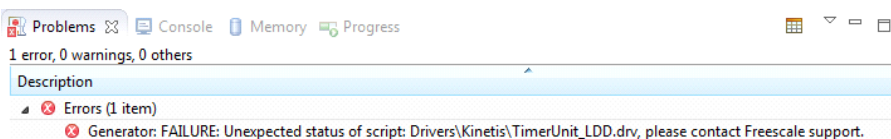

Figure 37. Unexpected Error Related to the LVHBridge TimerUnit_LDD Component

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 configurations (using the CPU **SetClockConfiguration** method), the speed of the stepper 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. LVHBridge may not pick up the new value and continues to use the previous value in its calculations.

Q: What does the error message "The component has no method to enable its event (**OnCounterRestart**)" raised in an LVHBridge TimerUnit_LDD component mean?

A: This occurs only when you add an LVHBridge component to a project and set the **Motor Control** property to **Stepper**. The error will disappear if you change any property of the LVHBridge component.

7   Schematic

Figure 38. Schematic

8   Silkscreen

8.1   Silkscreen Top
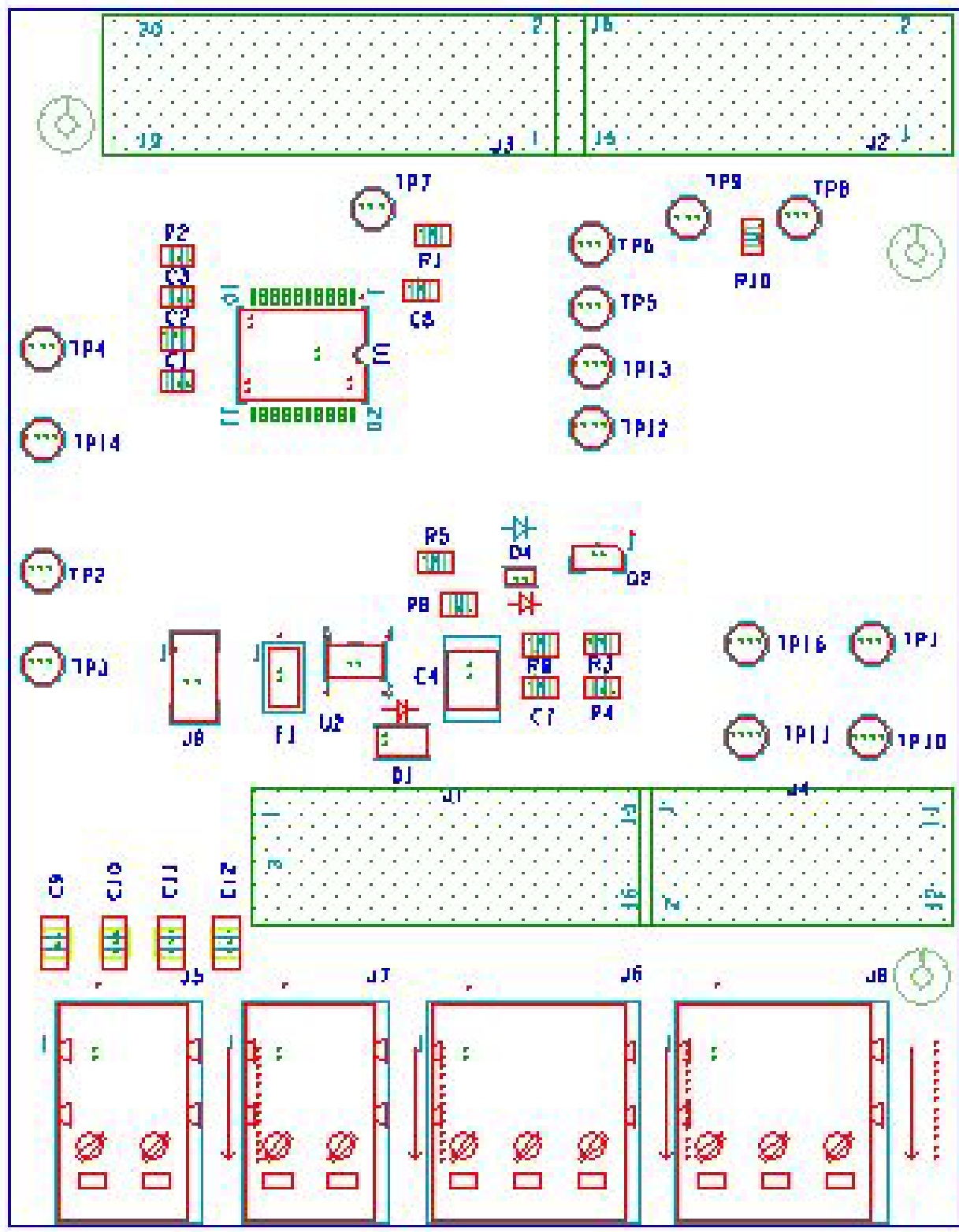
9   Bill of Materials

Table 12. Bill of Materials [2]

Item	Qty	Schematic Label	Value	Description	Part Number	Assy Opt
Active Components						
1	1	U1	TSSOP24	H-Bridge motor driver	MPC17529EV	[3]
Other Components						
2	1	U2	SOT23-5	Linear Reg LDO 1.5-15 V 150 mA 2.5-16 V	MIC5205	[3]
Transistors						
3	2	Q1, Q2	SOT-23	Transistor NMOS 50 V 220 mA	BSS138	
Diodes						
4	1	D1	SOD123	Diode Zener – 6.2 V 0.5 W	MMSZ5234B	
LEDs						
5	1	D4	0603	LED Green Single 20 mA	LG L29K-G2J1-24-Z	
Capacitors						
6	3	C1, C2, C3	0.1 uF	Ceramic 0.1 µF 50 V 10% X7R	0805	
7	1	C4	10 uF	Ceramic 10 µF 35 V 10% X7R	1210	
8	1	C8	2.2 uF	Ceramic 2.2 µF 16 V 10% X7R	0805	
9	1	C7	470 pF	Ceramic 470 pF 50 V 5% COG	0805	
Fuses						
10	1	F1	1.25 A	Fuse Fast 1.25 A 63 V SMT		
Resistors						
11	1	R3	10 k	Metal Film 10 k 1/10 W 1%	0805	
12	1	R5	220	Metal Film 220 OHM 1/8 W 1%	0805	
13	1	R8	9.1 k	Metal Film 9.1 k 1/10 W 1%	0805	
14	1	R9	15 k	Metal Film 15 k 1/8 W 5%	0805	
Connectors						
15	2	J1, J2		HDR 2X8 TH 100MIL CTR TSW-108-07-G-D SAMTEC	HDR 2X8	
16	1	J3		HDR 2X10 TH 100MIL CTR TSW-110-07-S-D SAMTEC	HDR 2X10	
17	1	J4		HDR 2X6 TH 100MIL CTR TSW-106-07-S-D SAMTEC	HDR 2X6	
18	2	J5, J7		SUBASSEMBLY CON 1X3 TB TH 3.81MM SP 201H -- 138L + TERM BLOCK PLUG 3.81MM 2POS 210-80097, 210-80098	TERM BLOCK 1x2	
19	2	J6, J8		SUBASSEMBLY CON 1X3 TB TH 3.81MM SP 201H -- 138L + TERM BLOCK PLUG 3.81MM 3POS210-80099, 211-79220	TERM BLOCK 1x3	

Notes:

2. Freescale does not assume liability, endorse, or warrant components from external manufacturers that 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.
3. **Critical components.** For critical components, it is vital to use the manufacturer listed.

10   References

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

Freescale.com Support Pages	Description	URL
FRDM-17529EV-EVB	Tool Summary Page	www.freescale.com/FRDM-17529EV-EVB
MCP17529	Product Summary Page	www.freescale.com/webapp/sps/site/prod_summary.jsp?code=MPC17529
FRDM-KL25Z	Freescale Development Platform	www.freescale.com/webapp/sps/site/prod_summary.jsp?code=FRDM-KL25Z
	Analog Home Page	freescale.com/analog
	Automotive Home Page	www.freescale.com/automotive
	mbed Home Page	www.mbed.org
Processor Expert	Tool Summary Page	www.freescale.com/LVHBRIDGE-PEXPERT
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

10.1 Support

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

10.2 Warranty

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

11   Revision History

Revision	Date	Description of Changes
1.0	11/2014	• Initial Release
2.0	9/2015	• Added Processor Expert section

Document Number: KTFRDM17529EVUG
Rev. 2.0
9/2015