

# **LVHB\_K64F\_34933EP-EVB\_Brush**

Example project for Low Voltage H-Bridge SW Driver

**Date:** 25/05/2017

**Revision:** 1.0

# Overview

The purpose of this example project is to demonstrate how to control two DC brushed motor using dual H-Bridge device and the Low Voltage H-Bridge (LVHB) SW Driver. The project contains several cases to show how to use most of the driver functions related to DC brushed motor control.

## Hardware Requirements

Following is required:

- FRDM-K64F (MCU freedom board)
- FRDM-34933-EP-EVB (H-Bridge freedom board)
- Two DC motors (2-7V)
- External Power Source (according to DC motors supply voltage)
- USB Micro B cable

## Setting up Hardware

Target platform for this example is FRDM-K64F and FRDM-34933-EP-EVB. Note that the driver supports also other LVHB devices and other MCUs. MCUs supported by SDK 2.x can be found in a roadmap on the [NXP community](#). For more information about supported devices refer to LVHB SW driver user guide.

In Figure 1 you can see HW connection of FRDM-34933-EP-EVB freedom board with load. Description of HW connection is in Table 1.

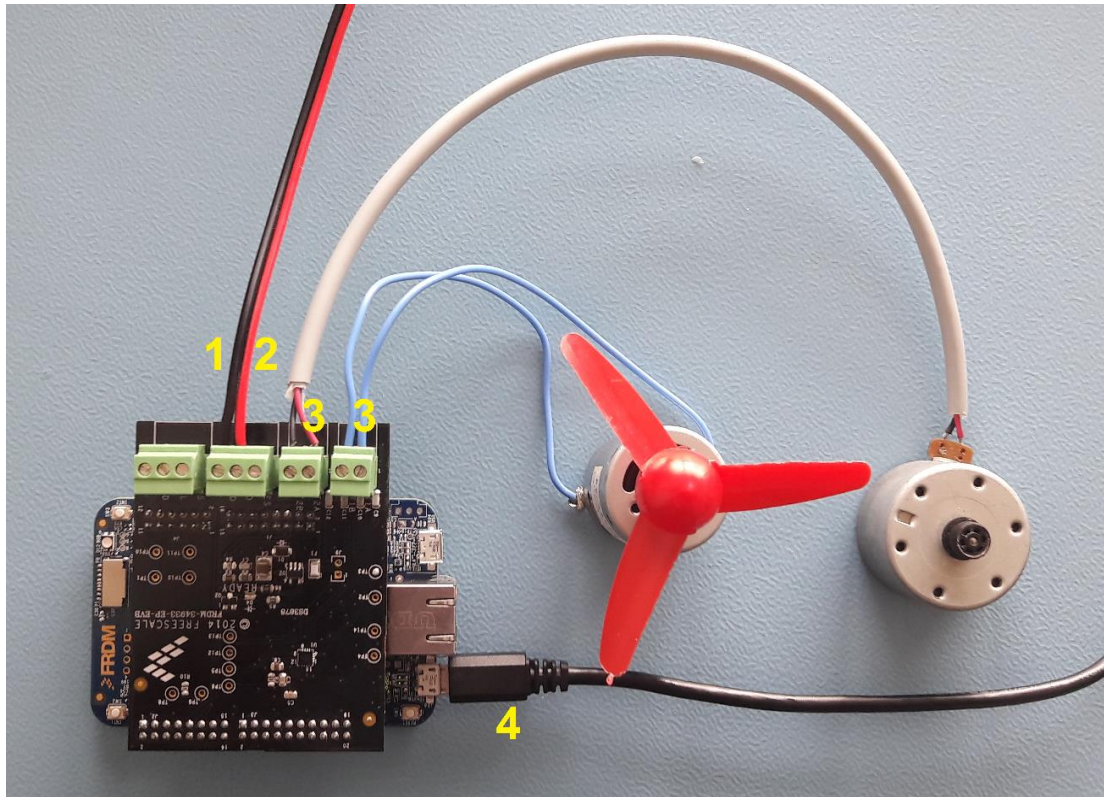


Figure 1. HW connection of FRDM-34933-EP-EVB

Table 1. Legend for HW connection

Label	Description
1.	DC Power supply (GND)
2.	DC Power supply (+)
3.	DC brushed motor connection
4.	USB Micro

## Setting up Software

Make sure that you have installed KDS 3.2.0 or newer.

The application uses debug interface with virtual serial port to print user messages. Check that your debug connection has been set up properly. Type of used debug connection depends on used MCU. FRDM-K64F uses **J-Link**, see Figure 2. Note that number of COM port may differ because of different system resource usage. Baud rate is 115200 Bd.

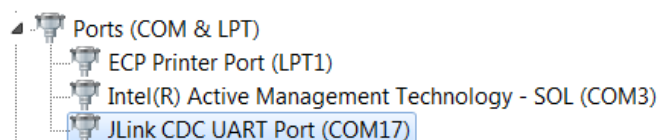


Figure 2. J-Link virtual port

# Description

The purpose of this example project is to show how to control two DC brushed motor using dual H-Bridge device. The project contains several cases to show how to use different driver functions.

The project uses the following peripherals:

- FTM0 – IN1B pin.
- GPIO – IN1A, IN2A and IN2B pins.
- UART0 – Print info to serial COM console.

Pin selection for all mentioned peripherals follows in Table 2 for selected MCU.

**Table 2. Pin selection**

Pin Function	FRDM-K64F
IN1A	PTB9
IN1B	PTA1/FTM0_CH6
IN2A	PTB23
IN2B	PTA2
UART RX	PTB16
UART TX	PTB17

Application uses virtual serial port to print user messages that describe executed test cases. Serial port settings are following:

- Data width: 8 bits
- Baud rate: 115 200 Bd
- Parity: none

In module *main.c* the board hardware is initialized. Then the configuration structure of LVHB driver is filled in. According to the configuration structure utilized timer periphery and GPIO pins are initialized.

Pins are configured as follows: IN1B pin is set as timer output. Other IN pins are controlled by GPIO. It enables speed adjustment of the first motor connected to H-Bridge interface 1 (OUT1A and OUT1B outputs). The second motor connected to interface 2 (OUT2A and OUT2B outputs) can be only turned on/off.

The project consists of several test cases:

1. Setting H-Bridge output to tri-state – the motor connected to H-bridge interface 1 is running when function *LVHB\_SetTriState* is called. You can see that motor slowly stops. Motor brake (H-Bridge outputs to LOW) follows so you can compare immediate and slow stopping. The same procedure is done with the second motor on interface 2.

2. Control of motor connected to interface 1 – PWM duty is increasing to reach value 100% (the highest speed). Then the duty is decreased to 0%. Motor direction is reversed and the procedure described above is repeated. Function *LVHB\_RotateProportional* is used in this test case.
3. Control of motor connected to interface 2 – motor is turned on and then back to off. This is repeated in reversed direction. This test case uses method *LVHB\_RotateFull*.

In *main.c* following set of functions is implemented covering LVHB SW driver functionality:

- *GetDefaultConfig* – Fills the driver configuration by default values.
- *ConfigureGpio* – Configures GPIO for usage with the driver.
- *ConfigureTimer* – Configures timer for usage with the driver.
- *Init* – Initializes the device.
- *RotateProportional* – Spins the motor in desired direction at PWM duty speed.
- *RotateFull* – Spins the motor in desired direction at full speed.
- *SetDirection* – Sets direction of brush motor movement.
- *SetTriState* – Sets output of H-Bridge to tri-state (high impedance).

## Import the Example Project

The following steps show how to import an example project into KDS 3.2.0.

1. In KDS click on the **File / Import**.
2. Choose **General / Existing Projects into Workspace**.
3. Click **Browse to select root directory** with your downloaded example projects.
4. **Select project** named **LVHB\_K64F\_34933EP-EVB\_Brush** and click **Finish** to complete the process.
5. Now the example project should be in your workspace and ready to run.

## Building and Running the Project

In order to build and run the project you need to **build** the project usual way. If the build is successful, **debug and run** the project. This can be accomplished in following steps:

1. Click on the **arrow** next to the **debug icon** and select **Debug Configurations**.
2. **Select** one of the existing configurations with **project name** under **SEGGER J-Link** group or **create** one by double clicking on this group.
3. Apply changes and click on **Debug**.

If you have any questions related to how to work with debug configurations, see **Kinetis Design Studio User's Guide**.