

LVHB_KL25Z_17510EJ-EVB_Brush_FreeMASTER

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 use FreeMASTER application along with Low Voltage H-Bridge (LVHB) SW Driver to control DC brushed motor.

Hardware Requirements

Following is required:

- FRDM-KL25Z (MCU freedom board)
- FRDM-17510-EJ-EVB (H-Bridge freedom board)
- DC Motor (2-15V)
- External Power Source (according to DC motor supply voltage)
- USB Mini B cable

Setting up Hardware

Target platform for this example is FRDM-KL25Z and FRDM-17510-EJ-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-17510-EJ-EVB freedom board with load. Description of HW connection is in Table 1.

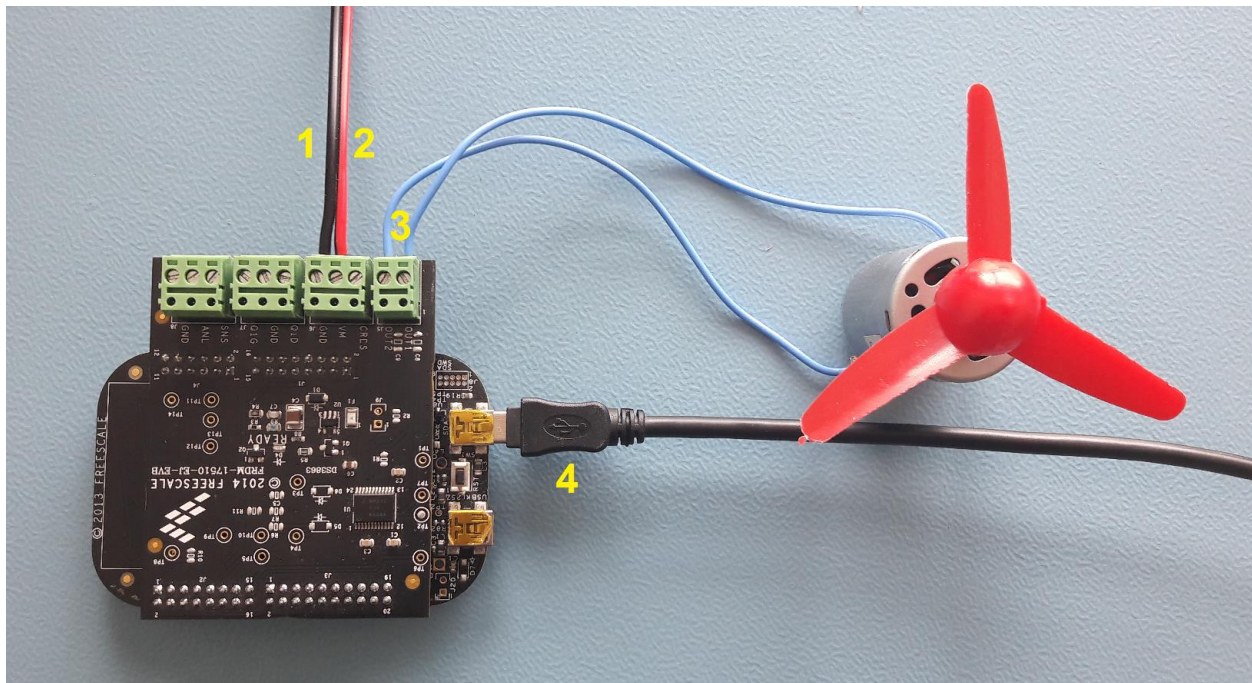


Figure 1. HW connection of FRDM-17510-EJ-EVB

Table 1. Legend for HW connection

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

Setting up Software

Make sure that you have installed KDS 3.2.0 or newer.

The application uses debug interface with virtual serial port for FreeMASTER communication. Check that your debug connection has been set up properly. Type of used debug connection depends on used MCU. FRDM-KL25Z uses **OpenSDA**, see Figure 2. Note that number of COM port may differ because of different system resource usage. Baud rate is 115200 Bd.



Figure 2. OpenSDA virtual port

Description

The demo project uses FreeMASTER tool to control DC brushed motor using low voltage H-Bridge device and LVHB SW driver.

The project uses the following peripherals:

- TPM0 – IN1 pin.
- GPIO – IN2, GIN and EN pins.
- UART0 – Communication with FreeMASTER application.

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

Table 2. Pin selection

Pin Function	FRDM-KL25Z
IN1	PTD4//TPM0_CH4
IN2	PTA12
EN	PTC7
GIN	PTC0
UART RX	PTA1
UART TX	PTA2

The project consists of two parts.

The first part is a FreeMASTER application that runs on PC (see Figure 3). Motor control parameters encompass motor direction and speed in percent (it is proportional to PWM duty). Command Brake causes that H-Bridge outputs are set to LOW. Outputs go floating (high impedance) when you click Freewheel. There are also settings of H-Bridge device such as device mode (normal operational mode or power conserving sleep mode) and control of gate driver output.

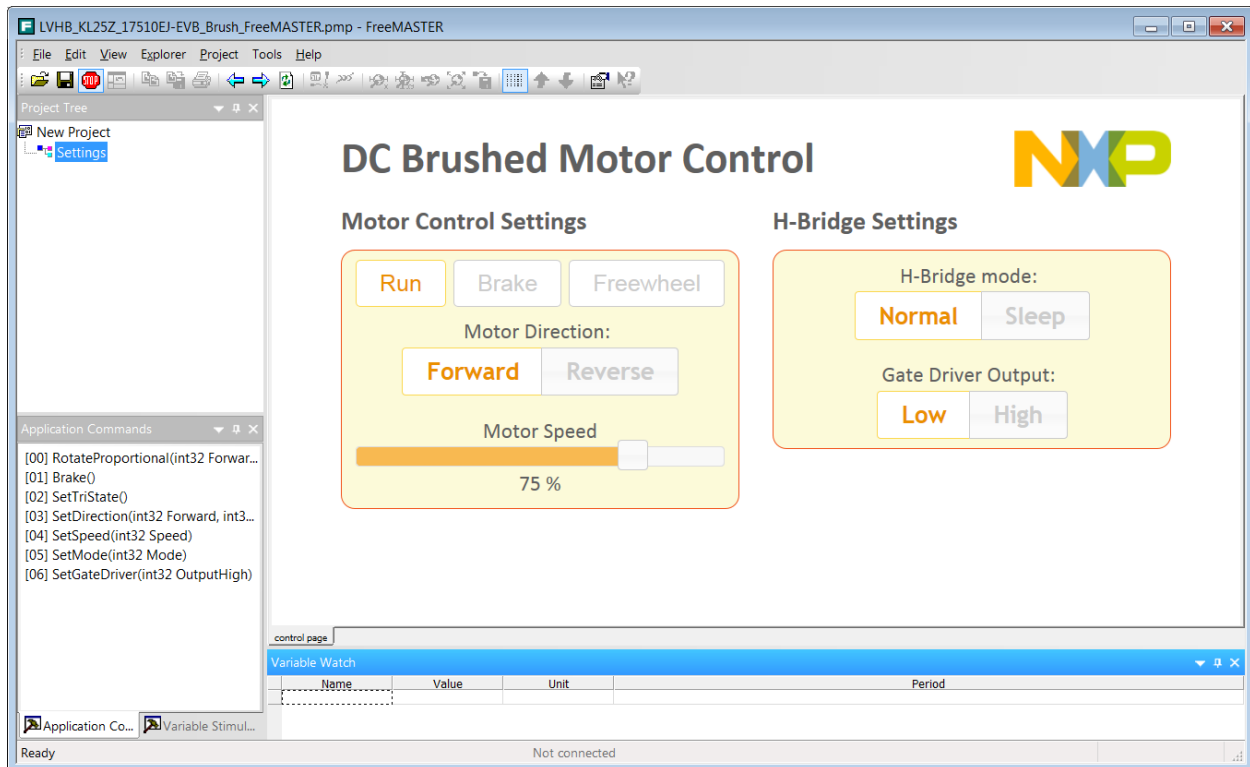


Figure 3. FreeMASTER application

The second part of the demo project is an application which runs on target platform. This application communicates with FreeMASTER and controls H-Bridge device according to FreeMASTER commands.

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.
- *SetDirection* – Sets direction of brush motor movement.
- *SetMode* – Sets H-Bridge device mode using enable pin.
- *SetTriState* – Sets output of H-Bridge to tri-state (high impedance).
- *SetGateDriver* – Sets device gate driver output.

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_KL25Z_17510EJ-EVB_Brush_FreeMASTER** 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 **PEMicro** group or **create** one by double clicking on this group.
3. Pick up proper **debug interface** and **USB port**.
4. 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**.

Follow these steps to run the FreeMASTER application:

1. Launch FreeMASTER application. You can download it from NXP [webpage](#). Install FreeMASTER by clicking downloaded **FMASW.exe** application.
2. Click **File / Open Project...** (see Figure 4).

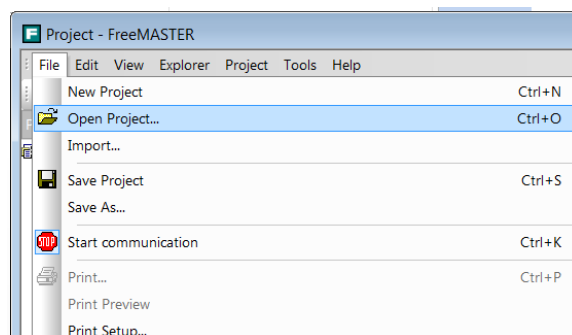


Figure 4. Opening FreeMASTER application

3. Locate FreeMASTER project file (.pmp) in the example project folder: LVHB_SDK_SW\KDS_Examples\LVHB_KL25Z_17510EJ-EVB_Brush_FreeMASTER. Select **LVHB_KL25Z_17510EJ-EVB_Brush_FreeMASTER.pmp** file then click Open (see Figure 5).

- If a dialog saying Missing symbol definition pops up click Continue. It is needed to link .elf file to the FreeMASTER project. Click **Project->Options** and under Map Files tab there is **Default symbol** file input. Select .elf file from Debug directory which is located in this LVHB_KL25Z_17510EJ-EVB_Brush_FreeMASTER project folder. If the .elf file is not present, you have to build the project in KDS.

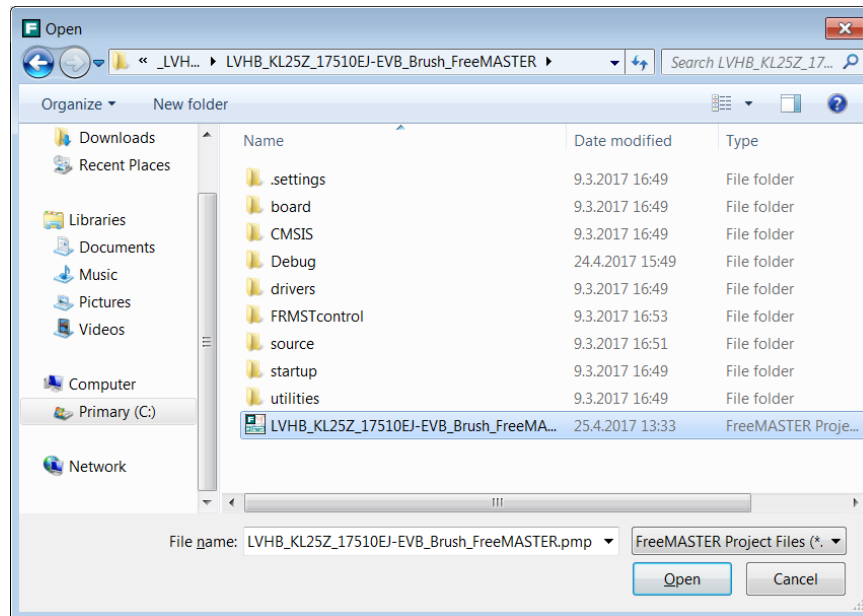


Figure 5. Selection of FreeMASTER project

4. Check if the correct COM port is selected. Click **Project->Options**. Check if the number of the COM port in FreeMASTER corresponds to the number of COM port on your computer (see Figure 6). Also check if the communication speed corresponds to the UART communication speed implemented in the MCU firmware (115 200 bauds).

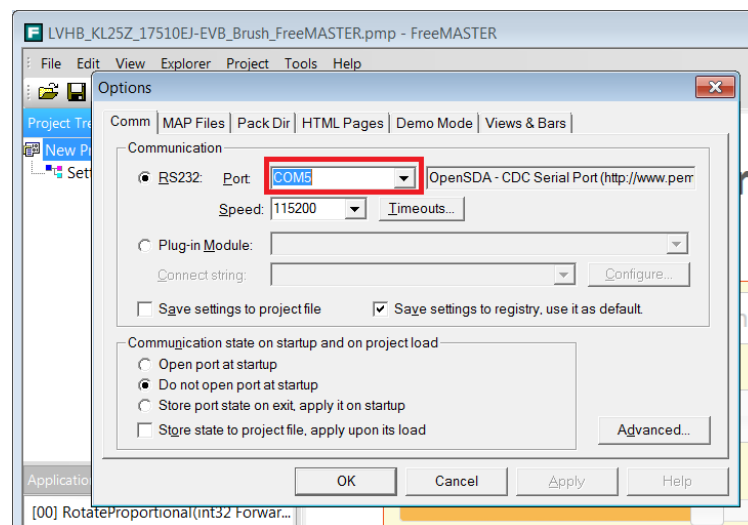


Figure 6. FreeMASTER settings

5. Start FreeMASTER monitoring (click on the red button named **STOP**, see Figure 7).

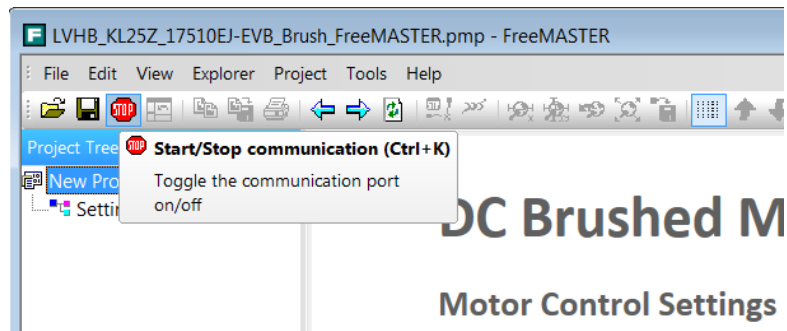


Figure 7. Run FreeMASTER application