LVHB_KL27Z_VariousEVB_Stepper

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 a stepper motor using dual H-Bridge device and Low Voltage H-Bridge (LVHB) SW Driver. The project contains several cases to show how to use most of driver functions for stepper motor control.

Hardware Requirements

Following is required:

- FRDM-KL27Z (MCU freedom board)
- FRDM-17529EVB, FRDM-17531AEJEVB, FRDM-17531AEPEVB, FRDM-17C724EVB or FRDM-34933EVB (H-Bridge freedom board)
- Stepper motor
- External Power Source (according to used stepper motor)
- USB Mini B cable

Setting up Hardware

Target platform for this example is FRDM-KL27Z as MCU board and FRDM-17529EVB, FRDM-17531AEJEVB, FRDM-17531AEPEVB, FRDM-17C724EVB or FRDM-34933EVB as H-Bridge device board. 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.

Be sure that jumper J26 is open on FRDM-KL27Z board.

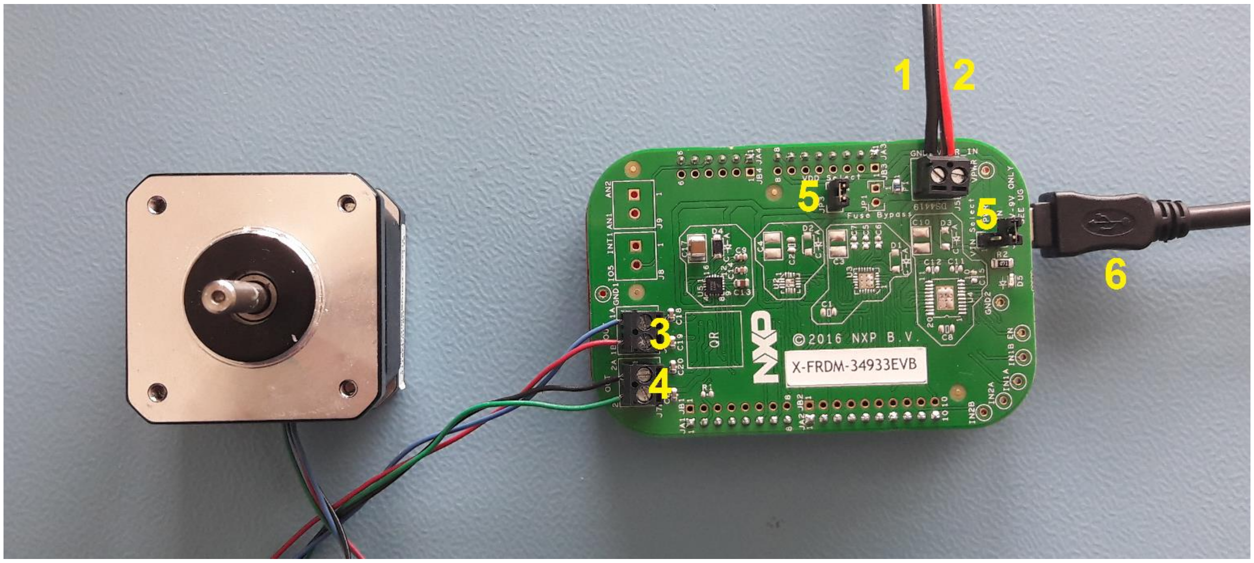

Figure 1. HW connection of FRDM-34933EVB

In Figure 1 you can see HW connection and jumper configuration of FRDM-34933EVB freedom board with load. Jumper JP3 must be short and JP2 open. Description of HW connection is in Table 1. HW connection of other supported H-bridge boards is similar.

Table 1. Legend for HW connection

Label	Description
1.	DC Power supply (GND)
2.	DC Power supply (+)
3.	First winding connection of stepper motor
4.	Other winding connection of stepper motor
5.	Jumper configuration
6.	USB mini

Setting up Software

Make sure that you have installed KDS 3.2.0 or newer.

Description

The purpose of this example project is to show a simple demo intended for demonstration of full-stepping and micro-stepping mode without acceleration ramp. Full-stepping and micro-stepping signals are generated using timer periphery.

The project uses the following peripherals:

- TPM0 – IN1A, IN1B, IN2A and IN2B pins.
- GPIO – EN pin (not used by FRDM-34933EVB board), red onboard LED diode (error signaling).

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

Table 2. Pin selection

Pin Function	FRDM-KL27Z
IN1A	PTE24/TPM0_CH0
IN1B	PTC9/TPM0_CH5
IN2A	PTE31/TPM0_CH4
IN2B	PTA5/TPM0_CH2
EN	PTA2
RED LED	PTB18

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.

There are several test cases, which demonstrate full-step and micro-step mode. Function *LVHB_AlignRotor* is called at the start of the example to align rotor to a full-step position. Function *WaitForCompletion* uses polling to wait for completion of motor movement. *BasicDemo*, *AdvancedDemo* and *SetMicroStepSizeDemo* functions are called in a loop.

1. **Basic demo** (function *BasicDemo*) demonstrates control of motor in full-step and micro-step mode. Firstly, full-stepping and micro-stepping speed is set by functions *LVHB_SetFullStepSpeed* and *LVHB_SetMicroStepSpeed*. Then, micro-step size is set to 2 micro-steps per one full-step by function LVHB_SetMicroStepSize. Function *LVHB_MoveSteps* moves motor in forward direction by 100 full-steps. There is no way to start new movement while motor is running. Function *WaitForCompletion* is used to detect movement completion. Demonstration of micro-stepping follows, function *LVHB_MoveMicroSteps* starts movement and rotor should return back to its initial position.
2. **Continual mode** (function *AdvancedDemo*) shows usage of LVHB driver functions related to continual mode. Full-stepping speed is increased and motor runs in continual mode. Note that it is not possible to change speed while motor is running. The movement is started by function *LVHB_MoveContinual* and continues until function *LVHB_StopContinualMovement* is called. Second part of the function *AdvancedDemo* is similar to the first one, motor speed is only decreased by micro-stepping movement. Note that stepper motor does not stop immediately (i.e. when the *LVHB_StopContinualMovement* is called), it can execute several steps. Therefore, motor does not have to stop at the same position where it started. Finally, stepper position counter is restarted by *LVHB_ResetFullStepPosition* and H-Bridge outputs are set to LOW using function *LVHB_DisableMotor*. Note that motor holds position (coils are powered) when movement is complete and function *LVHB_DisableMotor* is not called.
3. **Micro-step size demo** (function *SetMicroStepSizeDemo*) shows how to change size of micro-step. Micro-step size is set to values: 2, 4, 8, 16 and 32. Motor runs in micro-step mode when the size is adjusted.

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.
- *SetMode* – Sets H-Bridge device mode using enable pin.
- *AlignRotor* – Aligns rotor to the full-step position.
- *MoveSteps* – Moves motor by specified number of full-steps.
- *MoveMicroSteps* – Moves motor by specified number of micro-steps.
- *MoveContinual* – Moves motor continually in full-step mode.
- *MoveMicroContinual* – Moves motor continually in micro-step mode.
- *StopContinualMovement* – Stop continual movement of stepper motor.
- *DisableMotor* – Disables the stepper motor.

- *GetMotorStatus* – Returns status of stepper motor control.
- *SetFullStepSpeed* – Sets the speed of full-step mode.
- *SetMicroStepSpeed* – Sets the speed of micro-step mode.
- *SetMicroStepSize* – Changes the size of micro-step.
- *ResetFullStepPosition* – Sets the counter of full-steps to zero.
- *GetFullStepPosition* – Returns the current full-step position.
- *GetMicroStepPosition* – Returns the current micro-step position.

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_KL27Z_VariousEVB_Stepper** and click ***Finish*** to complete the process.
5. **Find *source* folder** of imported project in Project Explorer window. It contains four main files.
6. Rename either ***main.c_FRDM-17529EVB***, ***main.c_17531AExEVB***, ***main.c_FRDM-17C724EVB*** or ***main.c_FRDM-34933EVB*** to ***main.c*** according to used H-Bridge freedom board.
7. 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***.