LVHB_KL27Z_34933EVB_Stepper_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 a stepper motor.

Hardware Requirements

Following is required:

- FRDM-KL27Z (MCU freedom board)
- 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 and FRDM-34933EVB. 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 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.

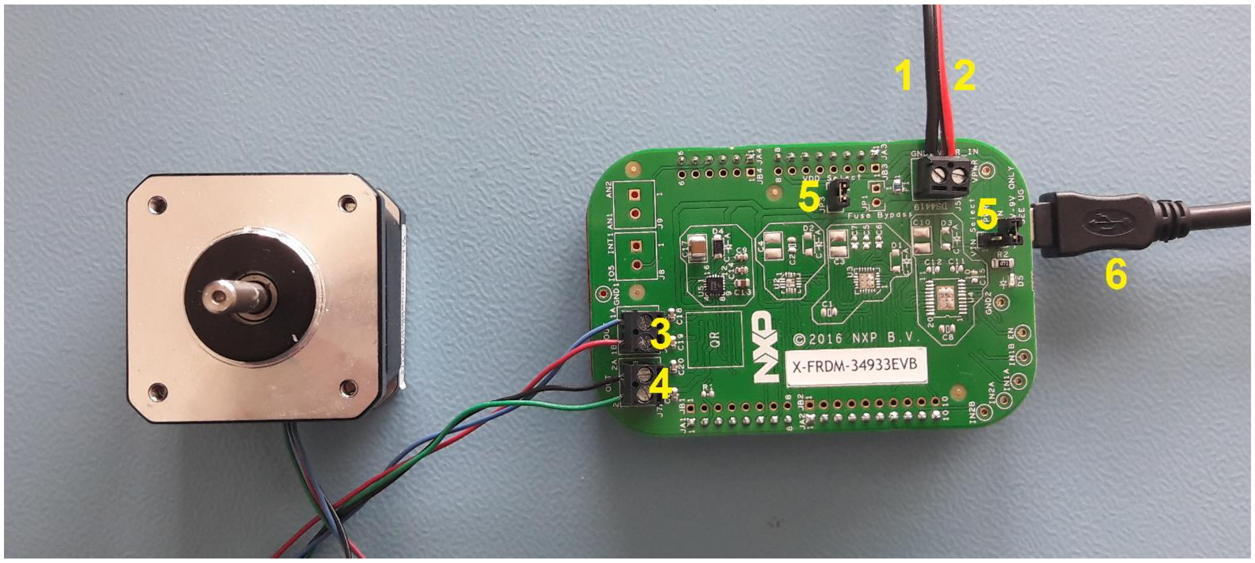


Figure 1. HW connection of FRDM-34933EVB

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.

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-KL27Z uses **OpenSDA**, see Figure 2. Note that number of COM port may differ because of different system resource usage. Baud rate is 115200 Bd.

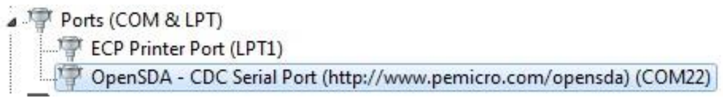

Ports (COM & LPT)
ECP Printer Port (LPT1)
OpenSDA - CDC Serial Port (http://www.pemicro.com/opensda) (COM22)

Figure 2. OpenSDA virtual port

Description

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

The project uses the following peripherals:

- TPM0 – IN1A, IN1B, IN2A and IN2B pins.
- LPUART0 – Communication with FreeMASTER application.

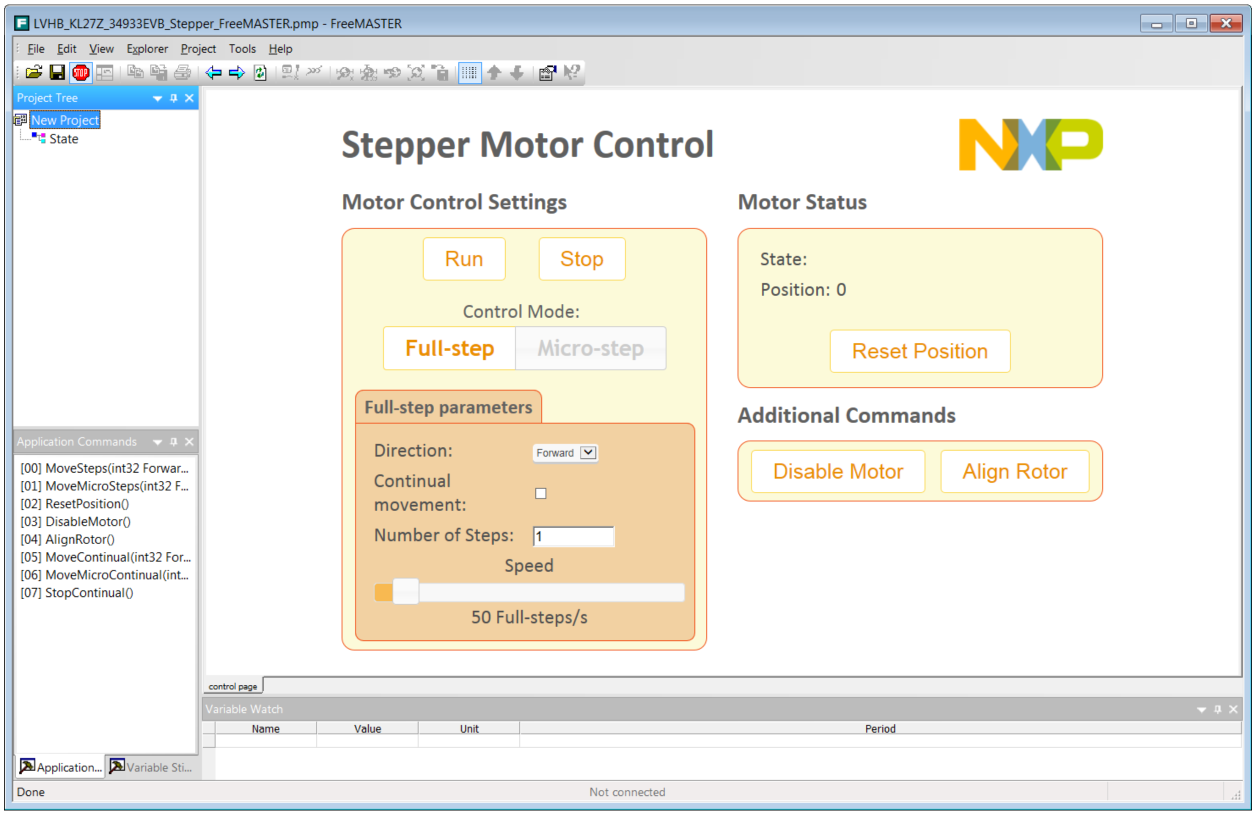
Pin selection for all mentioned peripherals follows in Table 2 for selected MCU.

Table 2. Pin selection

Pin Function	FRDM-KL27Z
IN1A	PTE24/TPM0_CH0
IN1B	PTC9/TPM0_CH5
IN2A	PTE31/TPM0_CH4
IN2B	PTA5/TPM0_CH2
LPUART RX	PTA1
LPUART TX	PTA2

The project consists of two parts.

The first part is a FreeMASTER application that runs on PC (see Figure 3). Full-step and micro-step control functions are available. Full-step settings encompass motor direction, continual movement (see later), number of steps to be executed and motor speed. When you check continual movement option the motor runs (after you click “Run” button) unless you click “Stop”. Note that number of steps is not changeable, because this parameter is meaningless in continual movement mode. Micro-stepping parameters are the same as in full-step mode with one exception. Additional parameter “Micro-steps per step” defines size of micro-step (up to 32 micro-steps per full-step are supported). Motor status panel consists of motor state (running or ready to run) and current motor position. Position is specified in number of full-steps and micro-steps taken from initial position. Value of position can be positive (forward direction) or negative number (reverse direction). You can reset the position to zero using Reset Position command. There is also command Disable Motor to set H-Bridge outputs to LOW.



[00] MoveSteps(int32 Forwar...
[01] MoveMicroSteps(int32 F...
[02] ResetPosition()
[03] DisableMotor()
[04] AlignRotor()
[05] MoveContinual(int32 For...
[06] MoveMicroContinual(int...
[07] StopContinual()

Direction: Forward
Continual movement: ☐
Number of Steps: 1
50 Full-steps/s

State:
Position: 0

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 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 is initialized. Following set of functions is implemented covering LVHB SW driver functionality:

- *GetDefaultConfig* – Fills the driver configuration by default values.

- *ConfigureTimer* – Configures timer for usage with the driver.
- *Init* – Initializes the device.
- *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.
- *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_34933EVB_Stepper_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 **FMASTERSW.exe** application.

2. Click **File / Open Project…** (see Figure 4).

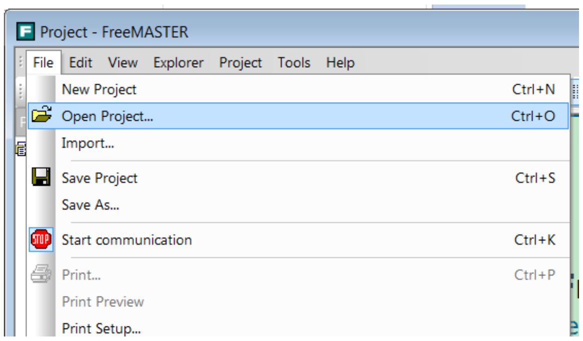
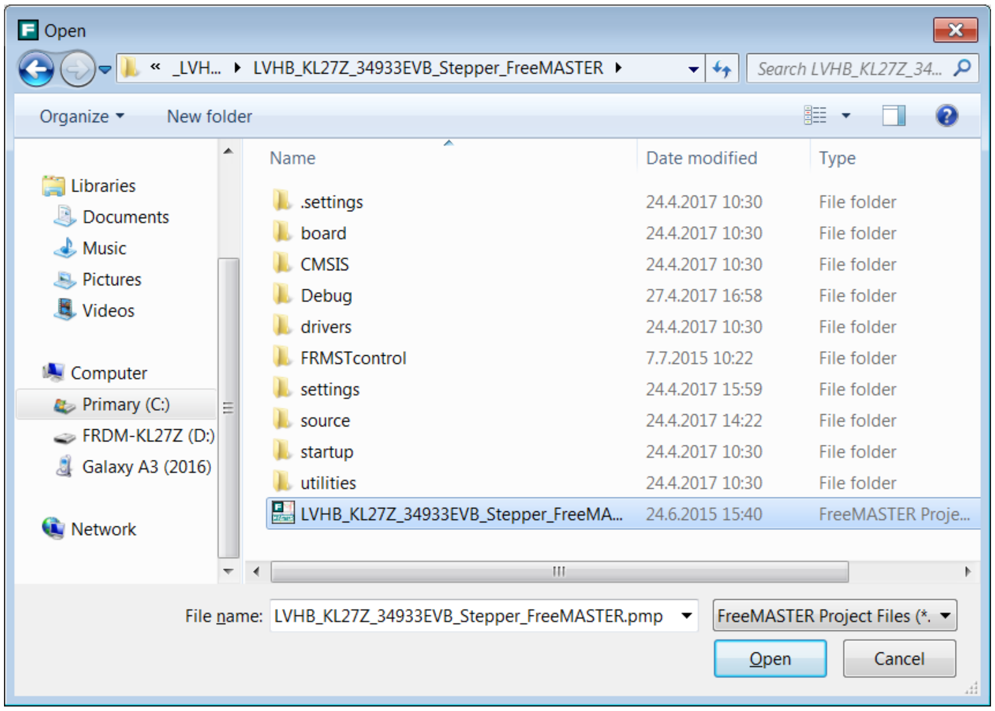


Figure 4. Opening FreeMASTER application

3. Locate FreeMASTER project file (.pmp) in the example project folder: LVHB_SDK_SW\KDS_Examples\LVHB_KL27Z_34933EVB_Stepper_FreeMASTER. Select **LVHB_KL27Z_34933EVB_Stepper_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 located in this LVHB_KL27Z_34933EVB_Stepper_FreeMASTER project folder. If the .elf file is not present, you have to build the project in KDS.



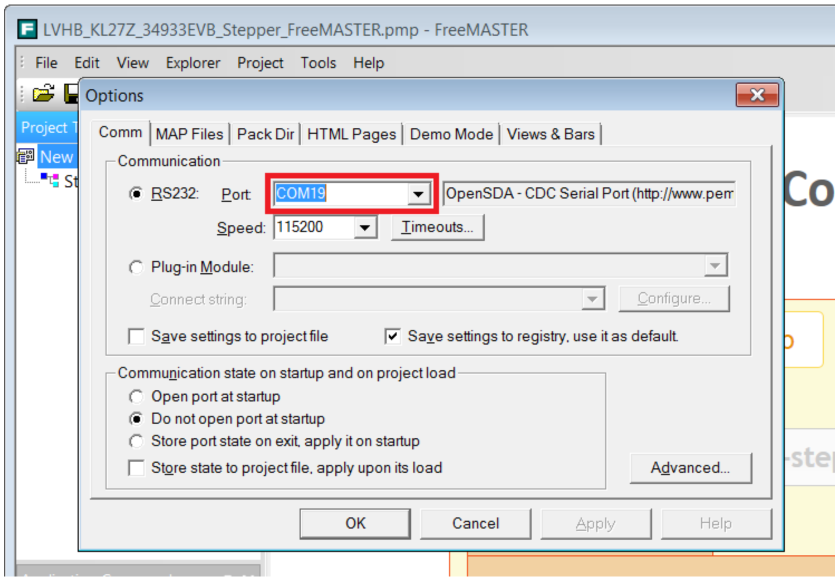
Name	Date modified	Type
.settings	24.4.2017 10:30	File folder
board	24.4.2017 10:30	File folder
CMSIS	24.4.2017 10:30	File folder
Debug	27.4.2017 16:58	File folder
drivers	24.4.2017 10:30	File folder
FRMSTcontrol	7.7.2015 10:22	File folder
settings	24.4.2017 15:59	File folder
source	24.4.2017 14:22	File folder
startup	24.4.2017 10:30	File folder
utilities	24.4.2017 10:30	File folder
LVHB_KL27Z_34933EVB_Stepper_FreeMA...	24.6.2015 15:40	FreeMASTER Proje...

File name: LVHB_KL27Z_34933EVB_Stepper_FreeMASTER.pmp — FreeMASTER Project Files (*.

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 LPUART communication speed implemented in the MCU firmware (115 200 bauds).


LVHB_KL27Z_34933EVB_Stepper_FreeMASTER.pmp - FreeMASTER
File Edit View Explorer Project Tools Help
Options
Comm | MAP Files | Pack Dir | HTML Pages | Demo Mode | Views & Bars
OK
Cancel
Apply
Help
Timeouts...
Configure...
Advanced...

Communication

[● RS232:] Port: COM19 OpenSDA - CDC Serial Port (http://www.pem

Speed: 115200

○ Plug-in Module:

Connect string:

☐ Save settings to project file ☑ Save settings to registry, use it as default

Communication state on startup and on project load

○ Open port at startup

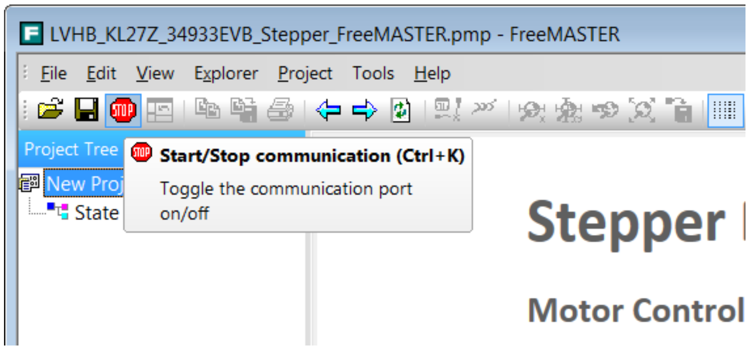
● Do not open port at startup

○ Store port state on exit, apply it on startup

☐ Store state to project file, apply upon its load

Figure 6. FreeMASTER settings

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


LVHB_KL27Z_34933EVB_Stepper_FreeMASTER.pmp - FreeMASTER
File Edit View Explorer Project Tools Help
STOP
Project Tree
New Proj
State
Stepper
Motor Control

Start/Stop communication (Ctrl+K)

Toggle the communication port on/off

Figure 7. Run FreeMASTER application