NXP Semiconductors
User's Guide

Document Number: BLDCDEMOUG
Rev. 7, 12/2020

BLDC Control Demo User's Guide

1. Introduction

This document provides instructions for running and controlling the Brushless DC (BLDC) sensorless application with the Freedom and High voltage development boards shown in Table 1.

The required software, hardware setup, jumper settings, project arrangement, and user interface is described in the following sections. For more information, see Section 9, “References” or visit www.nxp.com/motorcontrol_bldc.

Contents

2. Supported development boards

There are four supported development boards with two Kinetis KV series motor-control MCUs for motor-control applications and two development boards with Kinetis KE series. The development boards and supported MCUs are shown in Table 1. The Freedom development platform is targeted for low-voltage and low-power applications with BLDC control type. The High-Voltage Platform (HVP) is designed to drive high-voltage (115/220 V) applications with up to 1 kW of power.

Table 1. Supported development boards

—		Platform	
		FRDM	**HVP**
Power stage		FRDM-MC-LVBLDC	HVP-MC3PH
MCU	**KV11**	FRDM-KV11Z	HVP-KV11Z75M
	KV31	FRDM-KV31F	HVP-KV31F120M
	KE15Z	FRDM-KE15Z	—
	KE16Z	FRDM-KE16Z	—

3. Motor Control vs. SDK Peripheral Drivers

The motor-control examples use the SDK peripheral drivers to configure the general peripherals (such as clocks, SPI, SIM, and ports). However, motor control requires critical application timing because most control algorithms run in a 50-µs loop. To optimize the CPU load, most peripheral-hardware features are implemented for the PWM signal generation, analog signal sampling, and synchronization between the PWM and ADC units.

The standard SDK peripheral drivers do not support the configuration and handling of all required features. The motor-control drivers are designed to configure the critical MC peripherals (FTM, ADC, and PDB).

It is highly recommended not to modify the default configuration of the allocated MC peripherals due to a possible application-timing conflict. The particular *mcdrv_< board&MCU >.c* source file contains configuration functions of allocated peripherals.

4. Hardware setup

The BLDC sensorless application runs on High Voltage and Freedom development platforms with a default 24-V Linix motor.

4.1. Linix 45ZWN24-40 motor

The BLDC sensorless application uses the Linix 45ZWN24-40 motor described in Table 2. The motor can be used with the Freedom development platform.

Table 2. Linix 45ZWN24-40 motor parameters

Characteristic	Symbol	Value	Units

Rated Voltage	V_t	24	V
Rated Speed @ V_t	—	4000	RPM
Rated Torque	T	0.0924	Nm
Rated Power	P	40	W
Continuous Current	I_{cs}	2.34	A
Number of Pole Pairs	pp	2	—

Figure 1. Linix motor 45ZWN24-40

The motor has two types of cable connectors. One cable has three wires and it is designated to power the motor. The second cable has five wires and it is designated for the Hall sensors signal sensing. Only the power input wires are needed for the BLDC sensorless application.

4.2. MIGE 60CST-MO1330 motor

The MIGE 60CST-MO1330 motor (described in Table 3) is primarily used for the Permanent Magnet Synchronous Motor (PMSM) sensorless application but you can also use it for the BLDC sensorless application. You can also adapt the application to other motors by defining and changing the motor-related parameters. The motor is connected directly to the high-voltage development board via a flexible cable connected to the 3-wire development board connector.

Table 3. MIGE 60CST-MO1330 motor parameters

Characteristic	Symbol	Value	Units
Rated voltage	V_t	220	V
Rated speed @ V_t	—	3000	RPM
Rated power	P	400	W
Number of pole pairs	Pp	4	—

Figure 2. MIGE motor

4.3.
4.3.
4.3.
4.3.

4.3. Freedom development platform

To run the BLDC application using Freedom, you need these Freedom boards:

- Kinetis KV11Z Freedom board (see FRDM-KV11Z), Kinetis KV31F Freedom board (see FRDM-KV31F), Kinetis KE15Z, or Kinetis KE16Z Freedom board.
- 3-Phase Low-Voltage Power Freedom shield (see FRDM-MC-LVBLDC) including the Linix motor.

4.3.1. FRDM-MC-LVBLDC low-voltage evaluation board

The FRDM-MC-LVBLDC low-voltage evaluation board (in a shield form factor) effectively turns the Freedom development platform into a complete motor-control reference design compatible with the existing Freedom development platforms (FRDM-KV31F, FRDM-KV11Z, FRDM-KE15Z, and FRDM-KE16Z).

The FRDM-MC-LVBLDC board does not require any hardware configuration or jumper settings. It contains no jumpers.

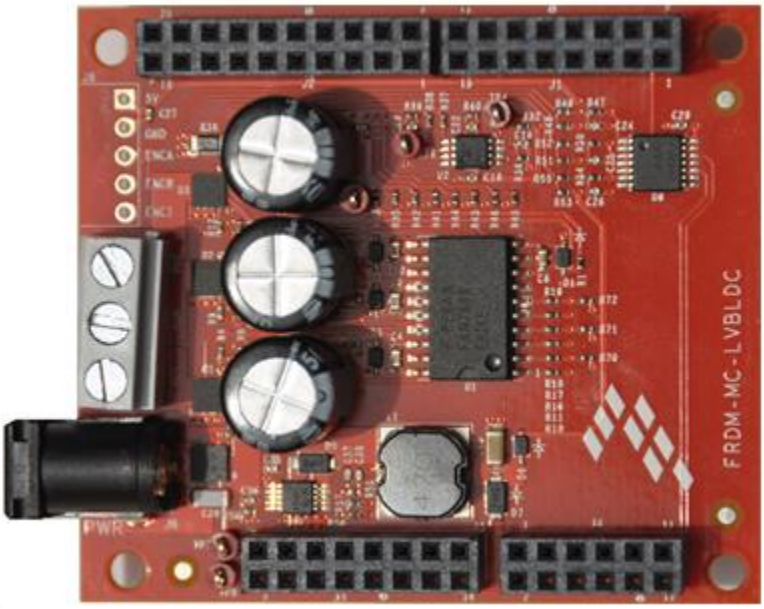

Figure 3. FRDM-MC-LVBLDC

4.3.2. FRDM-KV11Z board

The FRDM-KV11Z board is a low-cost development tool for the Kinetis V series KV1x MCU family built upon the Arm Cortex-M0+ processor. The FRDM-KV11Z board hardware is form-factor compatible with the Arduino R3 pin layout, providing a broad range of expansion board options. The FRDM-KV11Z platform features OpenSDA, the open-source hardware embedded serial and debug adapter running an open-source bootloader.

To begin, configure the jumpers on the FRDM-KV11Z Freedom System module properly. Table 4 lists the specific jumpers and their settings for the FRDM-KV11Z Freedom System module.

Table 4. FRDM-K11Z jumper settings

Jumper	Setting
J10	1-2

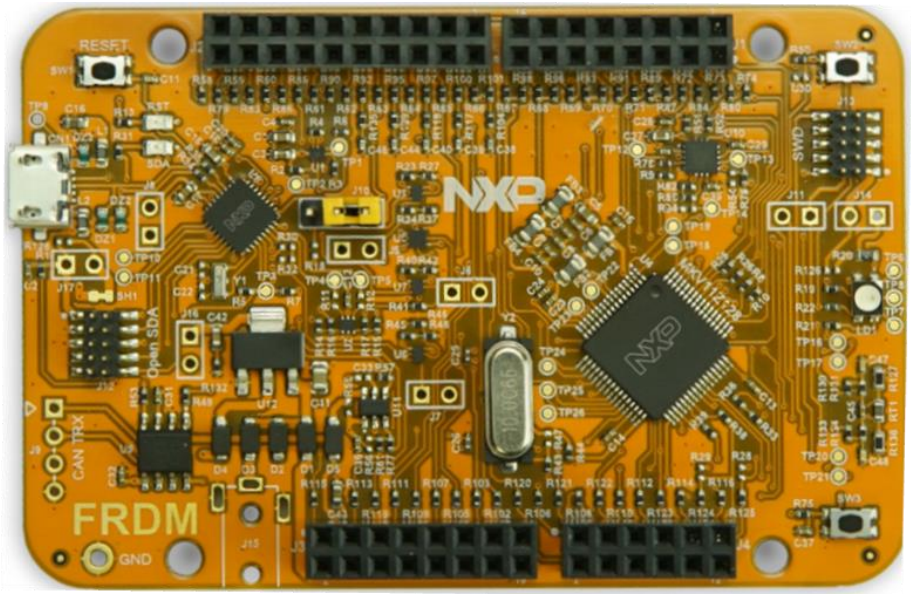

Figure 4. FRDM-KV11Z Freedom development board

4.3.3. FRDM-KV31F board

The FRDM-KV31F board is a low-cost development tool for the Kinetis V series KV3x MCU family built upon the Arm Cortex-M4 processor. The FRDM-KV31F board hardware is form-factor compatible with the Arduino R3 pin layout, providing a broad range of expansion board options, including FRDM-MC-LVPMSM and FRDM-MC-LVBLDC for PMSM and BLDC motor control.

The FRDM-KV31F platform features OpenSDA, the open-source hardware embedded serial and debug adapter running an open-source bootloader. This circuit offers several options for serial communication, flash programming, and run-control debugging.

The FRDM-KV31F board does not require any hardware configuration or jumper settings. It contains no jumpers.

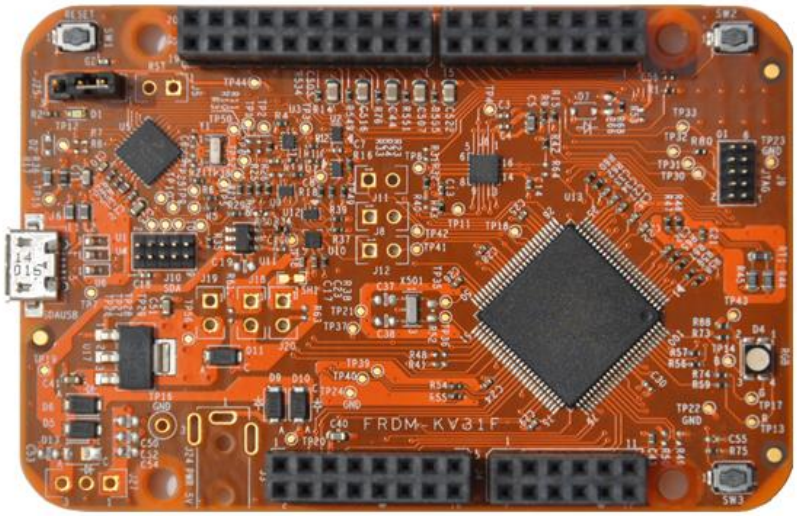

Figure 5. FRDM-KV31F Freedom development board

4.3.4. FRDM-KE15Z board

The FRDM-KE15Z is a low-cost development tool for Kinetis KE1x family of MCUs built around the Arm Cortex-M0+ core. The FRDM-KE15Z hardware is form-factor compatible with the Arduino R3 pin layout, providing a broad range of expansion board options. The FRDM-KE15Z platform features OpenSDA, the NXP open-source hardware embedded serial and debug adapter running an open-source bootloader.

To begin, configure the jumpers on the FRDM-KE15Z Freedom System module properly. Table 5 lists the specific jumpers and their settings for the FRDM-KE15Z Freedom System module.

Table 5. FRDM-KE15Z jumper settings

Jumper	Setting	Jumper	Setting	Jumper	Setting
J7	1-2	J10	1-2	J15	2-3
J8	1-2	J14	1-2		

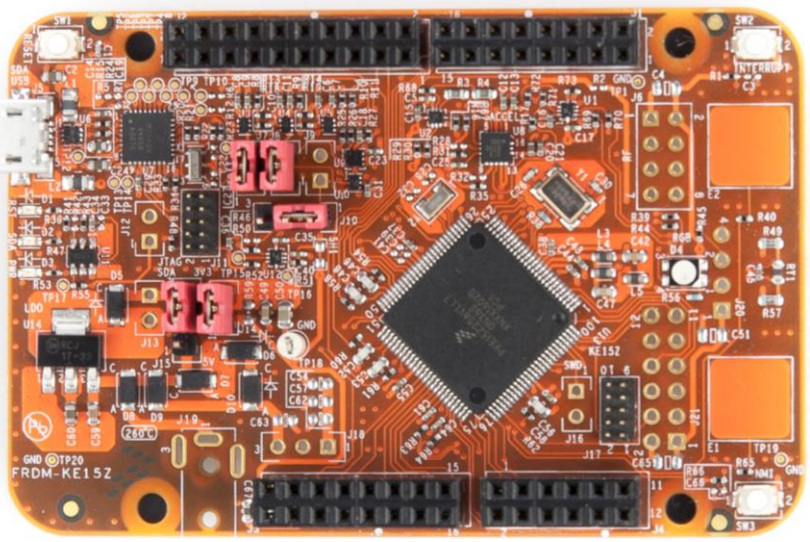

Figure 6. FRDM-KE15Z Freedom development board

4.3.5. FRDM-KE16Z board

The FRDM-KE16Z is a low-cost development tool for Kinetis KE1xZ family of MCUs built around the Arm Cortex-M0+ core. The FRDM-KE16Z hardware is form-factor compatible with the Arduino R3 pin layout, providing a broad range of expansion board options. The FRDM-KE16Z platform features OpenSDA, the NXP open-source hardware embedded serial and debug adapter running an open-source bootloader.

To begin, configure the jumpers and 0-Ω resistors on the FRDM-KE16Z Freedom System module properly. Table 13 lists the specific jumpers and Table 14 lists the 0-Ω resistor position settings for the FRDM-KE16Z Freedom System module.

Table 6. FRDM-KE16Z jumper setting

Jumper	Setting	Jumper	Setting	Jumper	Setting
J7	1-2	J10	1-2	J19	2-3
J8	1-2	J14	1-2	—	—

Table 7. FRDM-KE16Z O-Ω resistor setting

MCU signal	Resistor	Add/remove
PTB4/RGB_GREEN	R24	remove
PTB4/MC_PWM_CT	R25	add
PTB5/RGB_RED	R28	remove
PTB5/MC_PWM_CB	R29	add
PTC0/GES_R0	R110	remove
PTC0/MC_PWM_AT	R111	add
PTC1/GES_R1	R8	remove
PTC1/MC_PWM_AB	R13	add
PTD1/RGB_BLUE	R6	remove
PTD1/MC_PWM_BB	R9	add
PTC1/GES_R1_R6	R107	remove
PTC2/MC_CUR_DCB	R108	add
PTC0/GES_R0	R112	remove
PTB2/MC_VOLT_DCB	R113	add

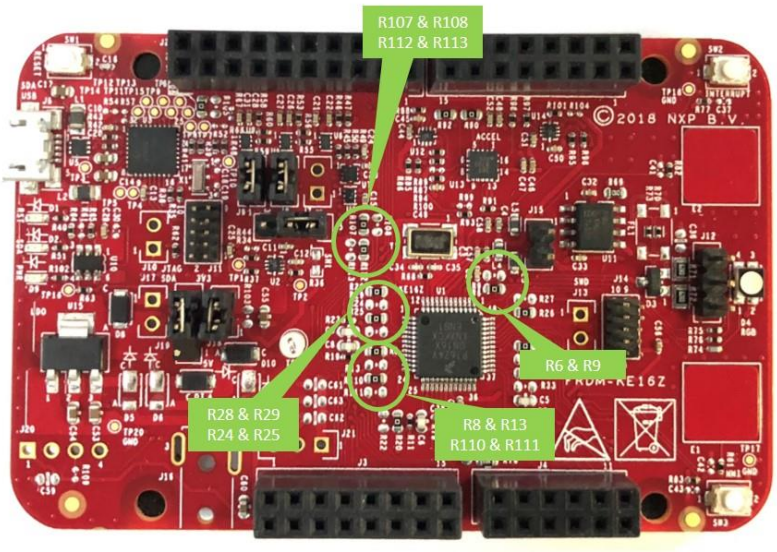

Figure 7. FRDM-KE16Z Freedom development board

4.3.6. Freedom development platform assembly

1. Connect the FRDM-MC-LVBLDC shield on top of the FRDM-KVxxx board.
2. Connect the BLDC motor 3-phase wires into the screw terminals on the board.
3. Plug in the USB cable from the USB host to the OpenSDA micro USB connector.
4. Plug in a 12 V DC power supply to the DC power jack.

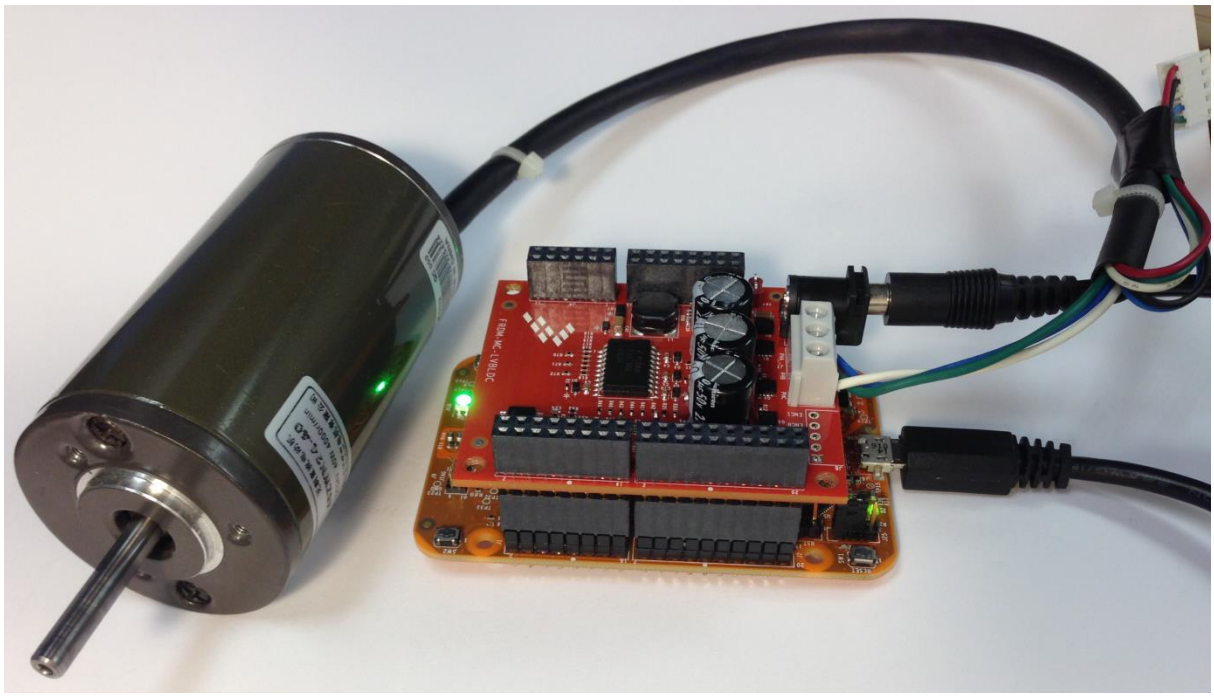

Figure 8. Assembled Freedom system

4.4. High-Voltage Platform

To run the BLDC application within the High-Voltage Platform, you need these components:

- Kinetis KV11F High-Voltage Daughter Board (HVP-KV11Z75) or KV31F High-Voltage Daughter Board (HVP-KV31F512).
- High-Voltage Platform (HVP-MC-3PH) (motor not included).

Order all the modules of the High-Voltage Platform from www.nxp.com (or from distributors) and build the hardware platform for the target application easily.

5. Project file structure

The demo project folder (for example *boards\frdmkv11z\demo_apps\mc_bldc*) contains these folders and files:

- *IAR* folder—contains the configuration files for IAR Embedded Workbench® IDE. If IAR Embedded Workbench for Arm is installed on your computer, open the project using IAR IDE.
- *Project files*—contains the device-specific files. These files specify the peripheral initialization routines, motor definitions, and state machines. The source code contains a lot of comments. The functions of the particular files are explained in this list:
    - *m1_bldc_appconfig.h*—contains the definitions of constants for the application control processes (parameters of the motor and regulators and constants for the BLDC sensorless control-related algorithms).

- *main.c*—contains the basic application initialization (enabling interrupts), the subroutines for accessing the MCU peripherals, and the interrupt service routines.
- *mcdrv.h*—includes the specific *mcdrv_< board&MCU >.h* file into the project.
- *mcdrv_< board&MCU >.c*—contains the motor-control driver peripherals' initialization functions specific to the board and MCU used.
- *mcdrv_< board&MCU >.h*—header file for *mcdrv_<board&MCU>.c*. This file contains the macros for changing the PWM period and ADC channels assigned to the back-EMF voltages and the board voltage and current.
- *pin_mux.c*—contains the board initialization function to configure the pin routing. This file is generated by the Pins tool.
- *pin_mux.h*—header file for *pin_mux.c*.
- *board.c*—common MCUXpresso SDK file containing the initialization of a debug console.
- *board.h*—common MCUXpresso SDK file containing the macros for a specific board pinout.
- *clock_config.c*—contains the MCU clock configuration functions.
- *clock_config.h*—header file for *clock_config.c*.
- *readme.txt*—basic information about the requirements, settings, and the demo.

The motor-control folder *<KSDK_install_folder>\middleware\motor_control\bldc* contains these common source and header files used in all motor-control projects. The folder contains the subfolders common to the entire project in this package:

- *mc_algorithms*—contains the control algorithms used to control the BLDC motor.
- *mc_drivers*—contains the source and header files used to initialize and run motor-control applications.
- *mc_state_machine*—contains the software routines that are executed when the application is in a particular state or state transition.
- *state_machine*—contains the state machine functions for the Fault, Initialization, Stop, and Run states.

Each motor-control project is based on RTCESL (Real-Time Control Embedded Software Library) placed in the *<KSDK_install_folder>\middleware* folder. The library contains the mathematical functions used in the project. It contains the library subfolders for specific cores ("*cm0*", "*cm4*", and "*cm7*"). This subfolder includes the required header files and library files used in the project. The *RTCESL* folder is taken from the RTCESL release 4.5, and it is fully compatible with the official release. The library names are changed for easier use with the available IDEs. See www.nxp.com/rtcesl for more information about RTCESL.

6. User Interface

The application contains the demo application mode to demonstrate the motor rotation. Operate it using the user button. The Freedom boards include a user button associated with the port interrupt (generated whenever one of the buttons is pressed). At the beginning of the ISR, a simple logic executes, and the interrupt flag clears. When you press the SW2 button, the demo mode starts; when you press the same button again, the application stops and transitions back to the STOP state. There is also an LED indication of the current application state. The green continuous LED indicates that the application is in the RUN state, the flashing LED indicates the FAULT state, and the LED off (or red LED) indicates the STOP state.

Control the application using the buttons on the NXP Kinetis V Freedom development boards. Because the HVP platform does not provide any push-buttons to control the application, the demo mode runs automatically after the HVP board is switched on.

6.1. Control button

Pressing the control button switches the demo mode on (or switches the demo mode off if it is currently switched on). Table 8 shows the correct switch button for your development board.

Table 8. Control button assignment

Board	Control button	LED state indication
FRDM-KV11Z	SW2	D4
FRDM-KV31F	SW2	D4
FRDM-KE15Z	SW2	D4
FRDM-KE16Z	SW2	—
HVP-KV11Z75M	—	D20
HVP-KV31F120M	—	D20

6.2. Remote control using FreeMASTER

Remote operation is provided by the FreeMASTER software via the USB interface. FreeMASTER 3.0 is required for optimal operation of the application. The FreeMASTER 3.0 application installation is available for download at www.nxp.com/freemaster.

Perform these steps to control a BLDC motor using FreeMASTER:

1. Open the FreeMASTER project file (*bldc.pmp*) in the *<sdk_package_folder>middleware/motor_control/freemaster* folder.
2. Click the communication button (the green “GO!” button in the top left-hand corner) to establish the communication.

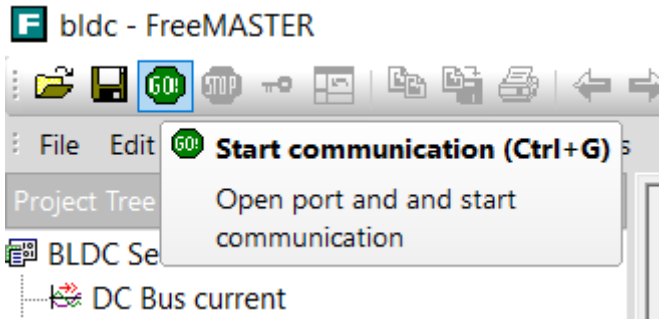

Figure 9. Green "GO!" button in top left hand corner

When the communication is successfully established, the FreeMASTER communication status in the bottom right-hand corner should change from “Not connected” to “RS232 UART Communication; COMxx; speed=115200”. When the communication setup fails, a FreeMASTER warning pop-up window appears.

Figure 10. FreeMASTER communication status when communication is established successfully

3. Control the BLDC motor using the control page.

If the communication setup fails, perform these troubleshooting steps:

1. Go to the “Project->Options->Comm” tab and make sure that “COM_ALL” is set in the “Port” option and the communication speed is set to 19200 bps.

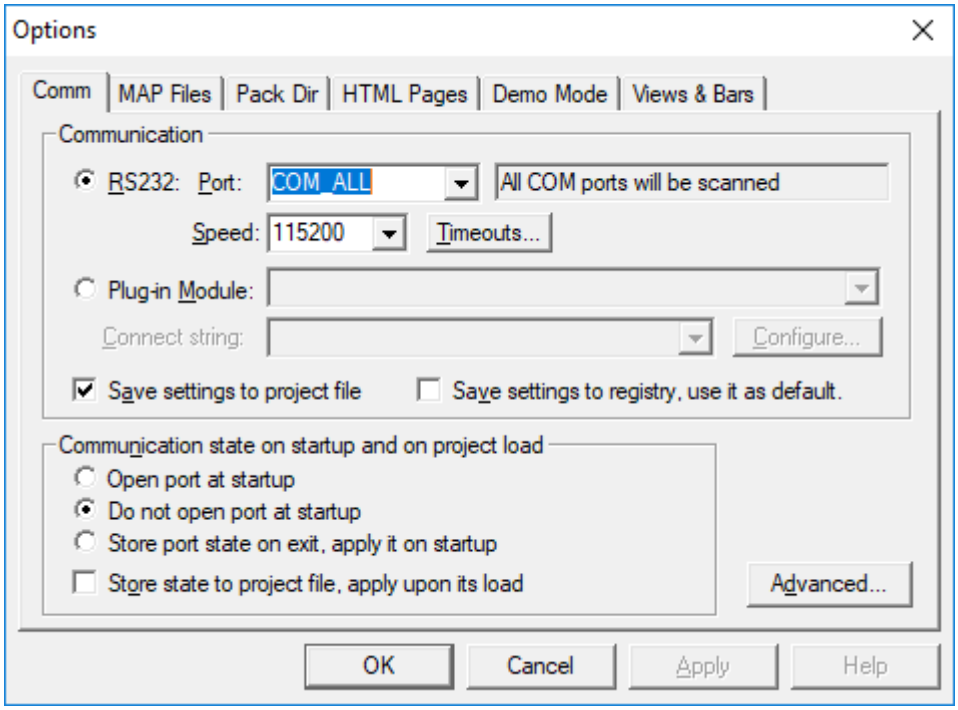

Figure 11. FreeMASTER communication setup window

2. When “OpenSDA-CDC Serial Port” is not displayed in the message window next to the “Port” drop-down menu, unplug and plug in the USB cable and reopen the FreeMASTER project again.

3. Make sure that your development board is supplied by a sufficient power source. Sometimes the USB PC port does not have sufficient power to supply the development board.

6.2.1. FreeMASTER TSA and user variables addition to FreeMASTER watch

By default, all projects use the TSA (Target Side addressing). This means that the information about the variables’ address and size are stored in the MCU flash memory. The TSA feature may be enabled or disabled by the BOARD_FMSTR_USE_TSA macro in the *board.h* file. Only the variables necessary for the MCAT functionality are stored in the TSA and visible in FreeMASTER. If you want to monitor your own variables, provide a symbol file which contains the information about the addresses of all variables in the project to FreeMASTER. The symbol files are generated during the build process to the *boards/demo_apps/mc_bldc/<compiler>/<debug or release>* folder. The symbol files have different extensions for different compilers (IAR generates **.out*, Keil and MCUXpresso generate **.axf*). For more information about the TSA, see *FreeMASTER Serial Communication Driver* (document FMSTERSCIDRVUG).

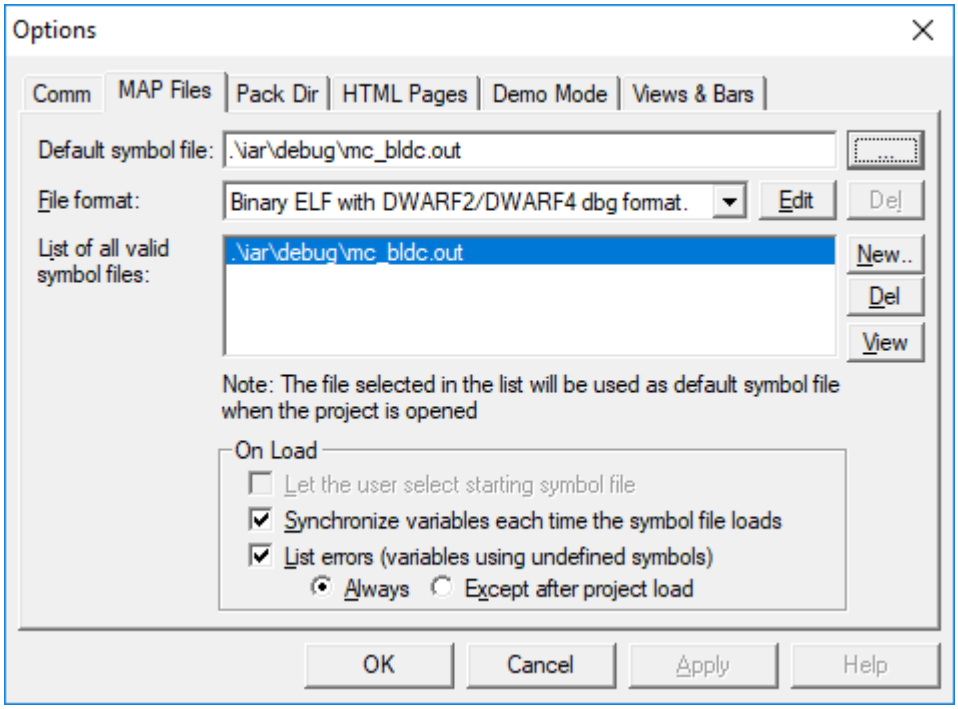

Figure 12. FreeMASTER symbol file selection

6.2.2. Control page

After launching the application and performing all necessary settings, control the BLDC motor using the FreeMASTER control page. The FreeMASTER control page contains:

- Speed gauge—shows the actual and required speeds.

- Speed slider—sets the required speed.
- DC-bus voltage gauge—shows the actual DC-bus voltage.
- DC-bus current gauge—shows the actual DC-bus current.
- DC-bus current limitation slider—sets up the DC-bus current limit.
- Demo mode button—turns the demonstration mode on/off.
- STOP button—stops the whole application.
- Application State Notification—shows the actual application state and faults.

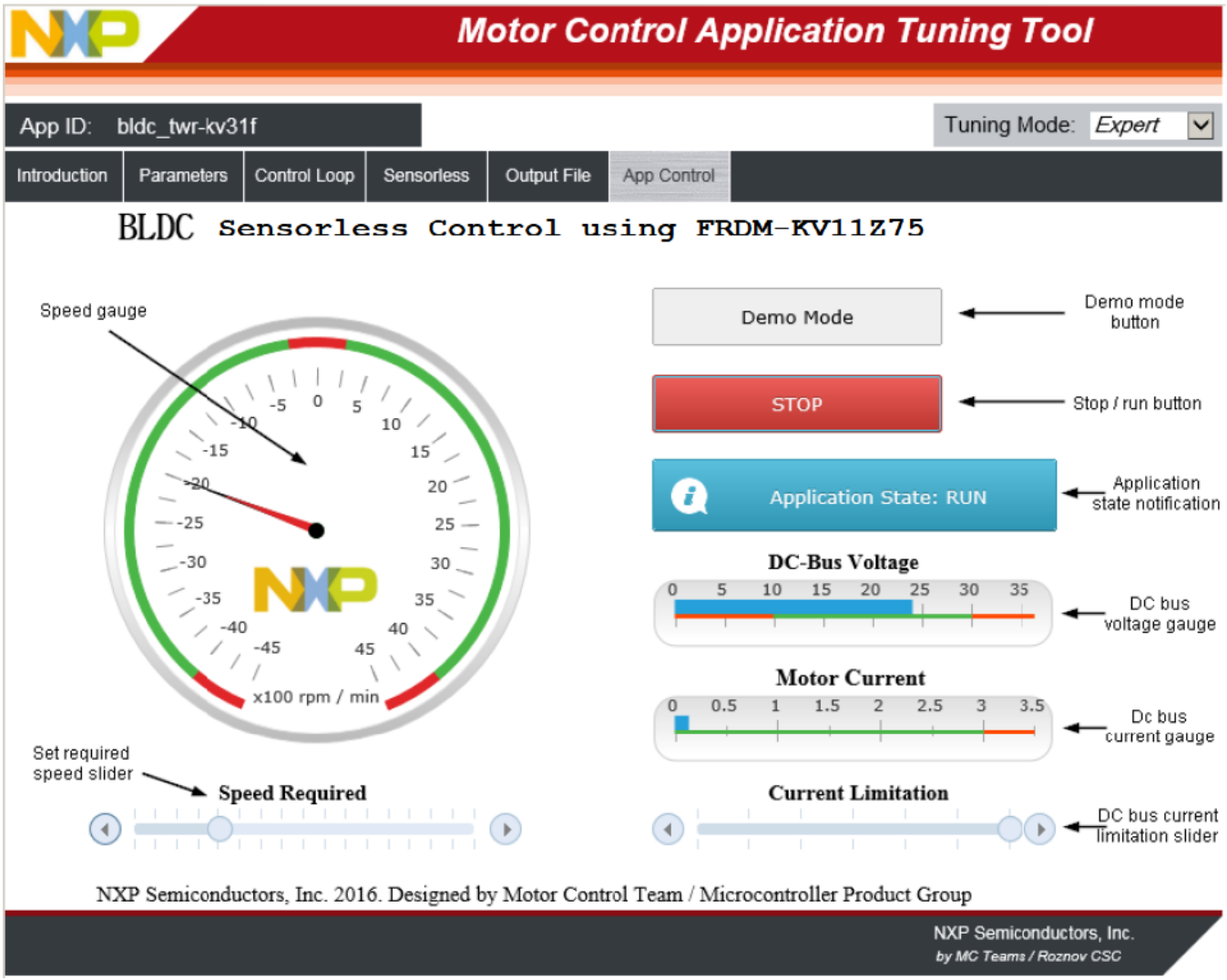

Figure 13. FreeMASTER control page

The MCAT tuning tool plug-in module is in the FreeMASTER GUI. This tool is targeted at motor-control applications, where you can change the motor-control parameters or adjust the start-up and control parameters. Each tab represents one submodule of the embedded-side control, and tunes its parameters. For more information, see *Sensorless BLDC Control on Kinetis KV* (document AN4642).

Here are the basic instructions:

- To start the motor, set up the required speed using the speed slider.

- In case of a fault, click on the fault notification to clear the fault.
- Click the “Demo Mode” button to turn the demonstration mode on/off.
- Click the “STOP” button to stop the motor.

7. Instructions

7.1. Running the motor

1. Assemble your NXP motor control hardware
2. Download the correct project into your target using the OpenSDA debug interface.
3. Open the FreeMASTER project, establish the communication between MCU and PC
4. Set up the required speed of the motor on the speed slider on the FreeMASTER control page.

7.2. Stopping the motor

1. Click on the Stop button on the FreeMASTER control page.
2. Require zero speed with using speed slider.
3. In emergency cases, turn off power supply.

7.3. Clearing a fault

To clear a fault click on the CLEAR FAULT button on the control page.

7.4. Turning on demonstration mode

1. If you use the FreeMASTER control page, click the “Demo” button.
4. If you do not use the FreeMASTER control page push the corresponding button on your development board to turn demonstration mode on/off.

8. Acronyms and abbreviations

Table 9. Acronyms and abbreviations

Term	Meaning
AN	Application Note
BLDC	Brushless DC motor
DRM	Design Reference Manual
MCU	Microcontroller
MSD	Mass Storage Device

9. References

See these documents at www.nxp.com:

1. Embedded Software Libraries User's Guides
2. *3-Phase BLDC Sensorless Motor Control Application* (document DRM144)
3. *Sensorless BLDC Control on Kinetis KV* (document AN5263)
4. *Getting Started with MCUXpresso SDK* (document MCUXSDKGSUG)

10. Revision history

Table 10. Revision history

Revision number	Date	Substantive changes
0	07/2016	Initial release.
1	03/2017	Updated the document with information about FRDM-KV11Z and the MCUXpresso IDE.
2	09/2017	Updated the document with information about the HVP platform.
3	10/2018	Added FRDM-KE16Z and removed KDS.
4	05/2019	Added FreeMASTER control, updated TWR-KV46 and TWR-KV58 board figures, removed KV10 references.
5	03/2020	Deleted the sections about the TWR-KV11 and TWR-KV31 boards. These boards are not supported in the SDK.
6	06/2020	Updated the FreeMASTER project file path.
7	12/2020	Updated Supported devices (Tower removed completely, HVP several boards removed)

How to Reach Us:

Home Page:
www.nxp.com

Web Support:
www.nxp.com/support

Document Number: BLDCDEMOUG
Rev. 7
12/2020

arm