Document identifier: PMSMSCEVKHVPUG
Rev. 0, 03/2020

PMSM Sensorless Control using MC56F8300-EVK and HVP-56F83783

Contents

Chapter 1
Introduction

This user's guide provides a step-by-step guide on how to open, compile, and run Permanent Magnet Synchronous Motor (PMSM) projects using the NXP MC56F83xxx Digital Signal Controller (DSC). It describes the basic compiling steps for CodeWarrior for supported development platforms mentioned in Supported development boards. It also describes the initialization of the FreeMASTER GUI tool for controlling motor-control applications.

Chapter 2
Supported development boards

There are currently two supported development boards with the MC56F837xx MCU for motor-control applications. The development boards and the supported MCU is shown Table 1. The NXP EVK and the Freedom development platform are targeted for low-voltage and low-power applications with PMSM 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 platforms

MCU	MCU Board	Power Board
MC56F83789	MC56F83000-EVK	FRDM-MC-LVPMSM
MC56F83783	HVP-56F83783	HVP-MC3PH

Chapter 3
Hardware setup

This section describes the default supported hardware configurations consisting of the FRDM-MC-LVPMSM power stage, MCU EVK, and the default PMS motor. The PMSM sensorless application runs on the Freedom development platform with 24-V Linix motors and the High-Voltage Platform with 230-V PMSM motors in the default configuration.

3.1 Linix 45ZWN24-40 motor

The Linix 45ZWN24-40 motor (described in Table 2) is a low-voltage 3-phase motor used in PMSM sensorless applications.

Table 2. Linix 45ZWN24-40 motor parameters

Characteristic	Symbol	Value	Units
Rated voltage	Vt	24	V
Rated speed @ Vt	—	4000	RPM
Rated torque	T	0.0924	Nm
Rated power	P	40	W
Continuous current	Ics	2.34	A
Number of pole pairs	pp	2	—

Figure 1. Linix motor

The motor has two types of connectors (cables). The first cable has three wires and it is used to power the motor. The second cable has five wires and it is used for Hall sensors signal sensing. For the PMSM sensorless application, you need only the power input wires.

3.2 MIGE 60CST-MO1330 motor

The MIGE 60CST-MO1330 motor (described in Table 3) is used in the PMSM 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	Vt	220	V
Rated speed @ Vt	—	3000	RPM
Rated power	P	400	W
Number of pole pairs	Pp	4	—

Figure 2. MIGE motor

3.3 PMSM application on low-voltage development platform

To run the PMSM application using the NXP development platform, you need these boards:

- MC56F83000-EVK board.
- FRDM-MC-LVMTR kit – contains the FRDM-MC-LVPMSM board, 3-phase low-voltage power Freedom shield, and 3-phase Linix motor.

3.3.1 FRDM-MC-LVPMSM

The FRDM-MC-LVPMSM low-voltage evaluation board (in a shield form factor) turns an NXP Freedom development board into a complete motor-control reference design compatible with existing DSC development platforms (MC56F83000-EVK).

The FRDM-MC-LVPMSM board does not require any hardware configurations or jumper settings.

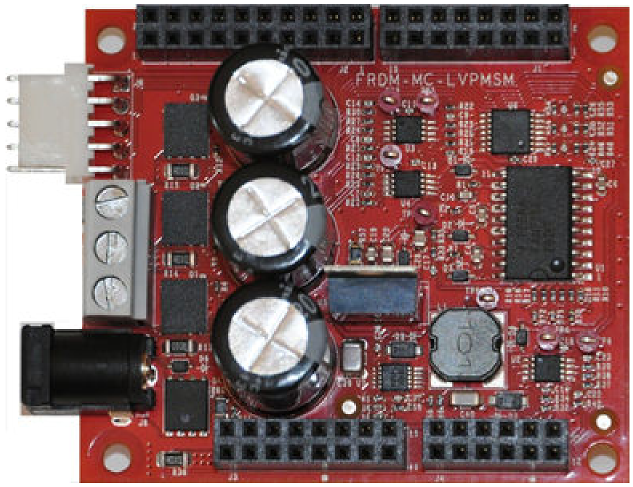

Figure 3. FRDM-MC-LVPMSM

3.3.2 MC56F83000-EVK

The MC56F83xxx family is NXP's performance level of DSCs. It is based on the 32-bit 56800EX DSP core with both the core and BUS frequencies of up to 100 MHz. The MC56F83000-EVK is an ultra-low-cost development platform for the MC56F83xxx DSC family, enabling rapid prototyping and application development.

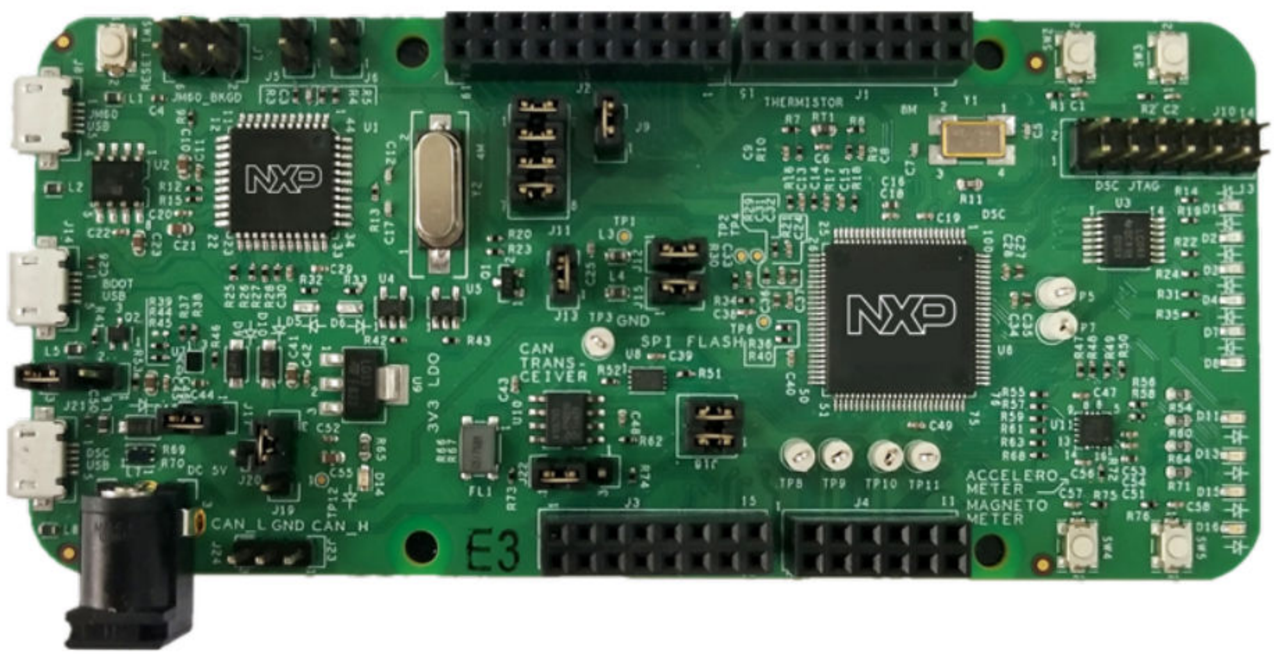

Figure 4. MC 56F83000-EVK MCU board

Table 4. MC56F83000-EVK jumper settings

Jumper	Setting	Jumper	Setting	Jumper	Setting
J5	open	J6	open	J7	open

Table continues on the next page...

Table 4. MC56F83000-EVK jumper settings (continued)

J9	1-2	J10	open	J11	1-2, 3-4, 5-6, 7-8
J12	1-2	J13	1-2	J15	1-2
J16	2-3	J17	1-2	J18	1-2, 3-4
J19	2-3	J20	open	J22	1-2
J23	open	—	—	—	—

3.3.3 Board assembling

1. Connect the FRDM-MC-LVPMSM shield onto the top of the MC56F83000-EVK board (there is only one possible option).
2. Connect the Linix motor 3-phase wires to the screw terminals on the board.
3. Plug the USB cable from the USB host to the J8 micro-USB connector (debugging).
4. Plug the USB cable from the USB host to the Virtual Serial Port J14 micro-USB connector. (FreeMASTER communication)
5. Plug the 24-V DC power supply to the DC Power connector.

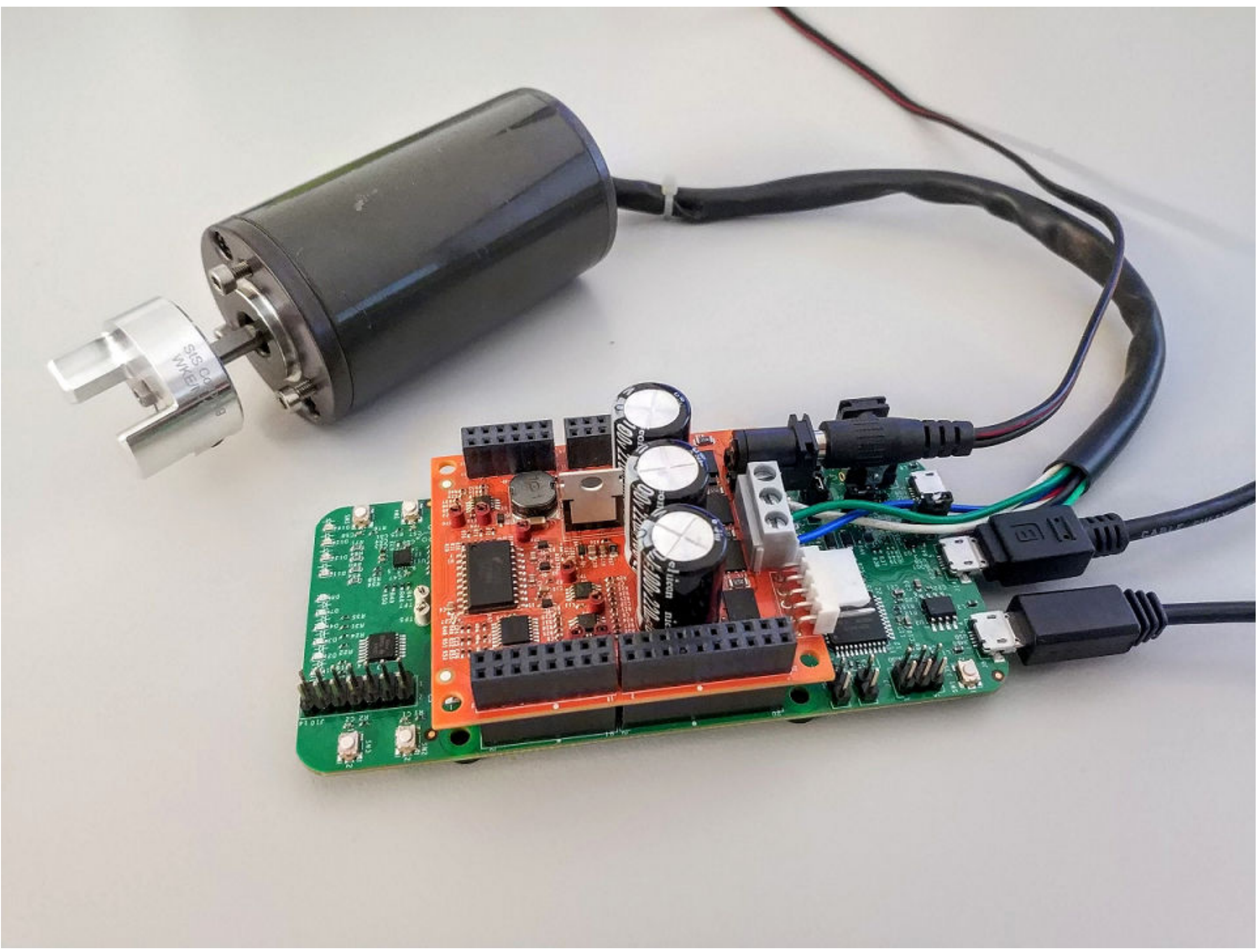

Figure 5. Board assembling

3.4 Running PMSM application on high-voltage development platform

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

- HVP daughter card with a DSC series MCU HVP-56F83783.
- High-Voltage Platform power stage (HVP-MC-3PH) (motor not included).

You can order all modules of the High-Voltage Platform at www.nxp.com/hvp or from distributors and easily build the hardware platform for the target application.

3.4.1 HVP-MC3PH power stage

The NXP High-Voltage Platform is an evaluation and development solution for the Kinetis and DSC series MCUs. This platform enables the development of 3-phase PMSM, BLDC, and ACIM motor-control and power-factor-correction solutions in a safe high-voltage environment. The boards work in the default configuration and you don't have to set any jumpers to run the attached application. See *High-Voltage Motor ControlPlatform User's Guide* (document HVPMC3PHUG).

You don't have to set up the HVP-MC3PH high-voltage development board in any way. The board does not contain any jumpers.

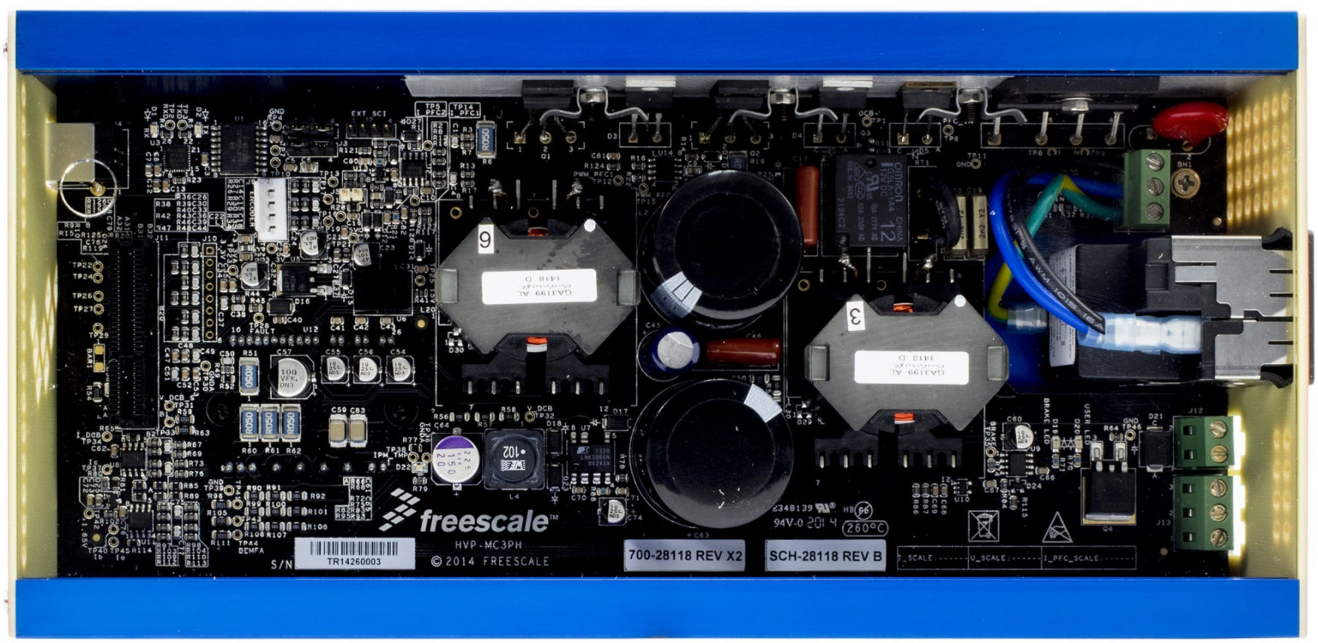

Figure 6. HVP-MC3PH High-Voltage Platform

3.4.2 HVP-56F83783 daughter card

The HVP-56F83783 MCU daughter card contains the MC56F83783 family DSC, built around the 56800/E core running at 100 MHz. This daughter card is developed for use in motor-control applications, together with the High-Voltage Platform power stage. This daughter card features OpenSDA, the NXP open-source hardware embedded serial and debug adapter running an open-source bootloader.

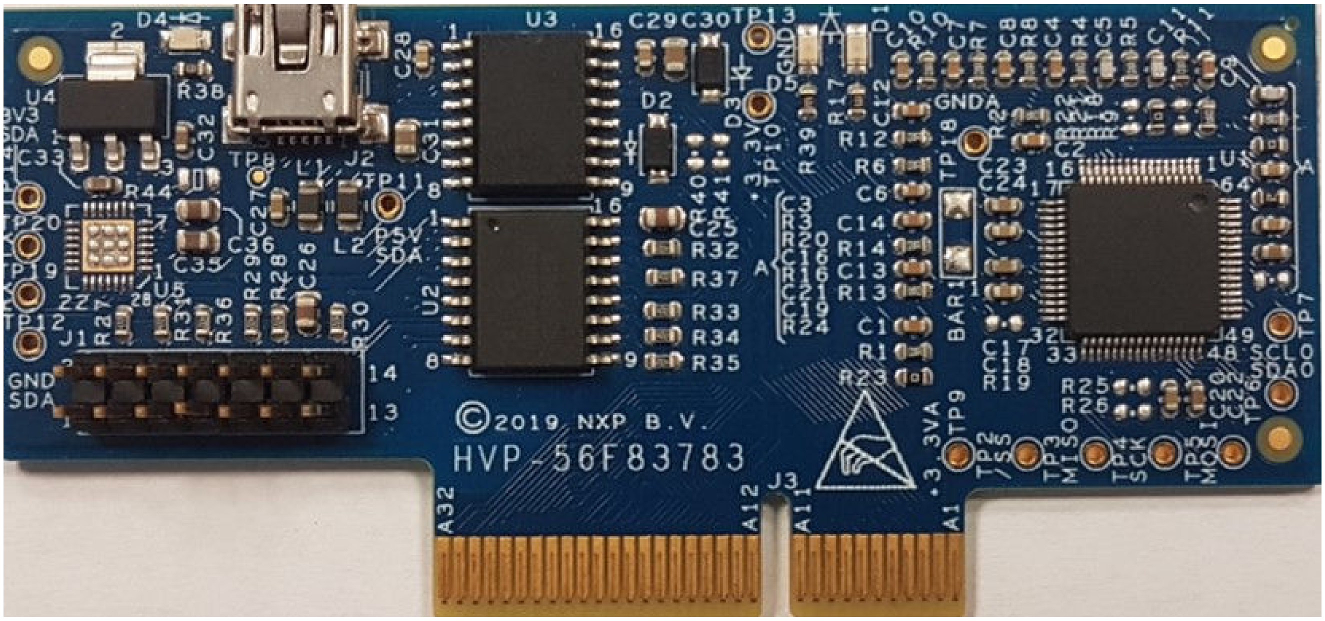

Figure 7. HVP-56F83783 daughter card

3.4.3 High-Voltage Platform assembling

1. Check whether the HVP-MC3PH main board is unplugged from the voltage source.
2. Insert the HVP-56F83783 daughter board to the HVP-MC3PH main board (there is only one possible option).
3. Connect the PMSM motor 3-phase wires into the screw terminals on the board.
4. Plug the USB cable from the USB host to the OpenSDA micro-USB connector.
5. Plug a 230-V power supply to the power connector and switch it on.

Chapter 4
Tools

Install this software on your PC to run and control the PMSM sensorless application:

- CodeWarrior Development Studio IDE (11.1 or higher)
- CodeWarrior for MCU 11.1 Update 1 – update including pack support for MC56F83xxx devices
- DSC56800EX Quick Start 2.7 – initialization and development tool
- FreeMASTER Run-Time Debugging Tool (2.5 or higher)

Chapter 5
Building and debugging application

5.1 CodeWarrior Development Studio IDE

The CodeWarrior embedded software development studio is a complete integrated development environment (IDE) that provides a highly visual and automated framework to accelerate the development of the most complex embedded applications. The CodeWarrior Development Studio IDE (CW) is an IDE tool that can be used to develop and test software for Kinetis®, ColdFire®, S12Z, HCS12, MPC5xx, MPC56xx, MobileGT®, and 56800/E MPUs. CW supports a wide range of debuggers, such as P&E Micro or USB-TAP.

To open a solution project, run the CW IDE from the default installation path or from installed programs, and then perform these steps:

- Select a workspace and run CW.
- Click the “File” menu in the top-left corner of the IDE, and select “Import…”:

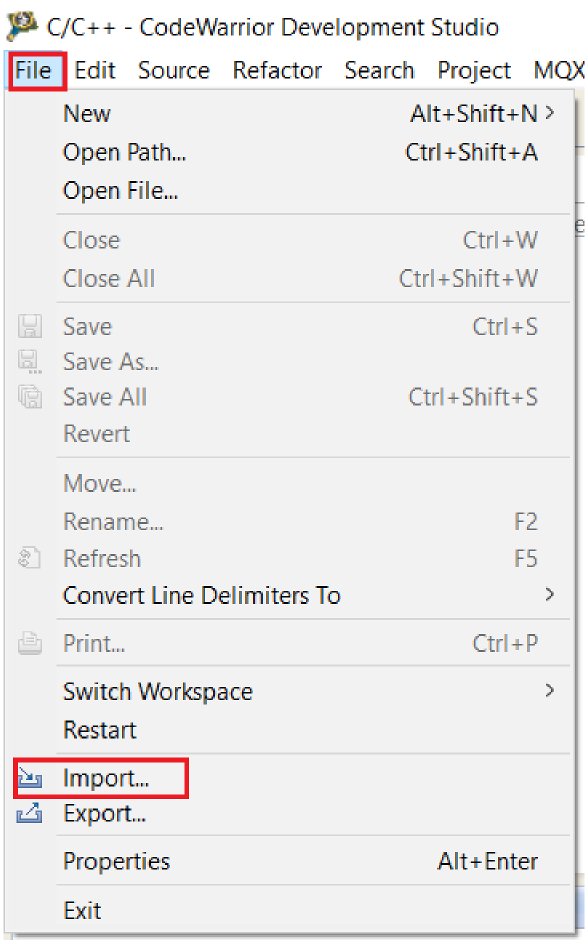

Figure 8. Importing project in CW—first step

- The “Import” window opens. Highlight “Existing Projects into Workspace” in the “General” folder, and click the “Next” button. The window shown in Figure 9 appears. Select “General/Existing Projects into Workspace”:

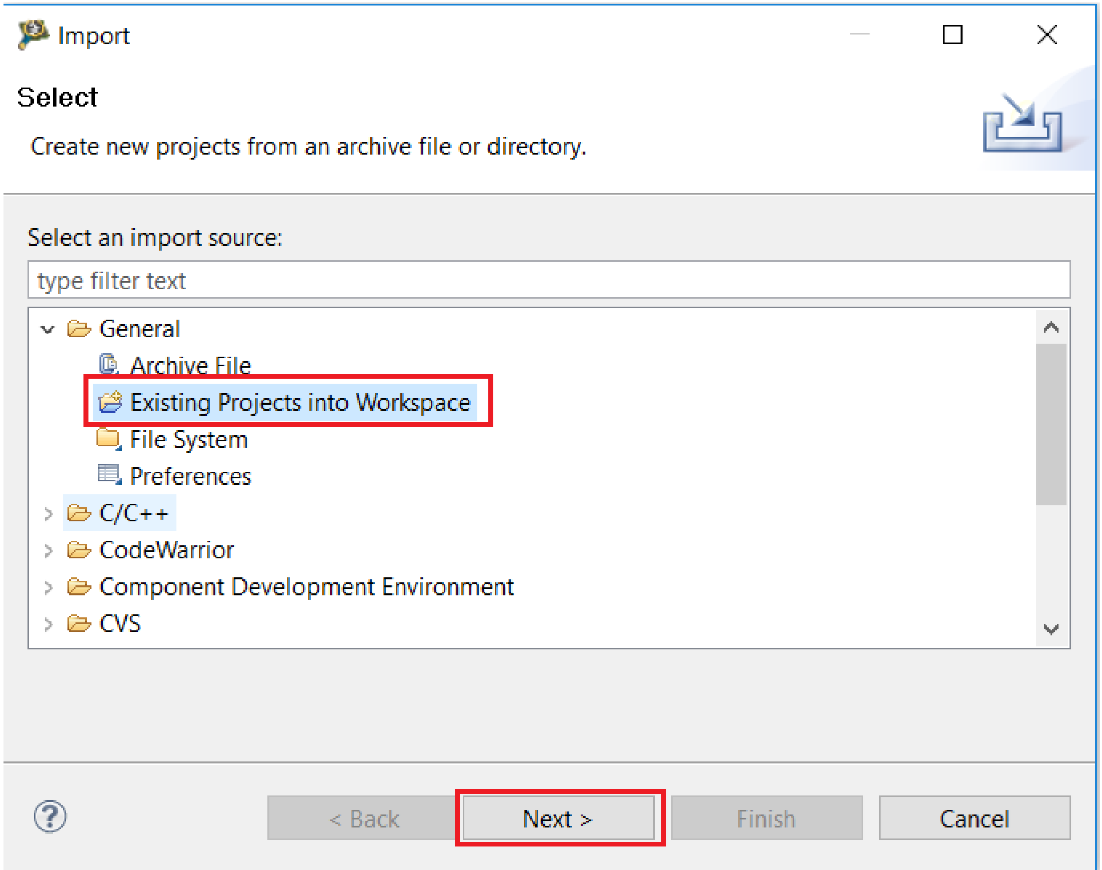

Figure 9. Importing project in CW—second step

- The “Import” window opens. Click the “Browse” button and locate the project. Click the “OK” button:

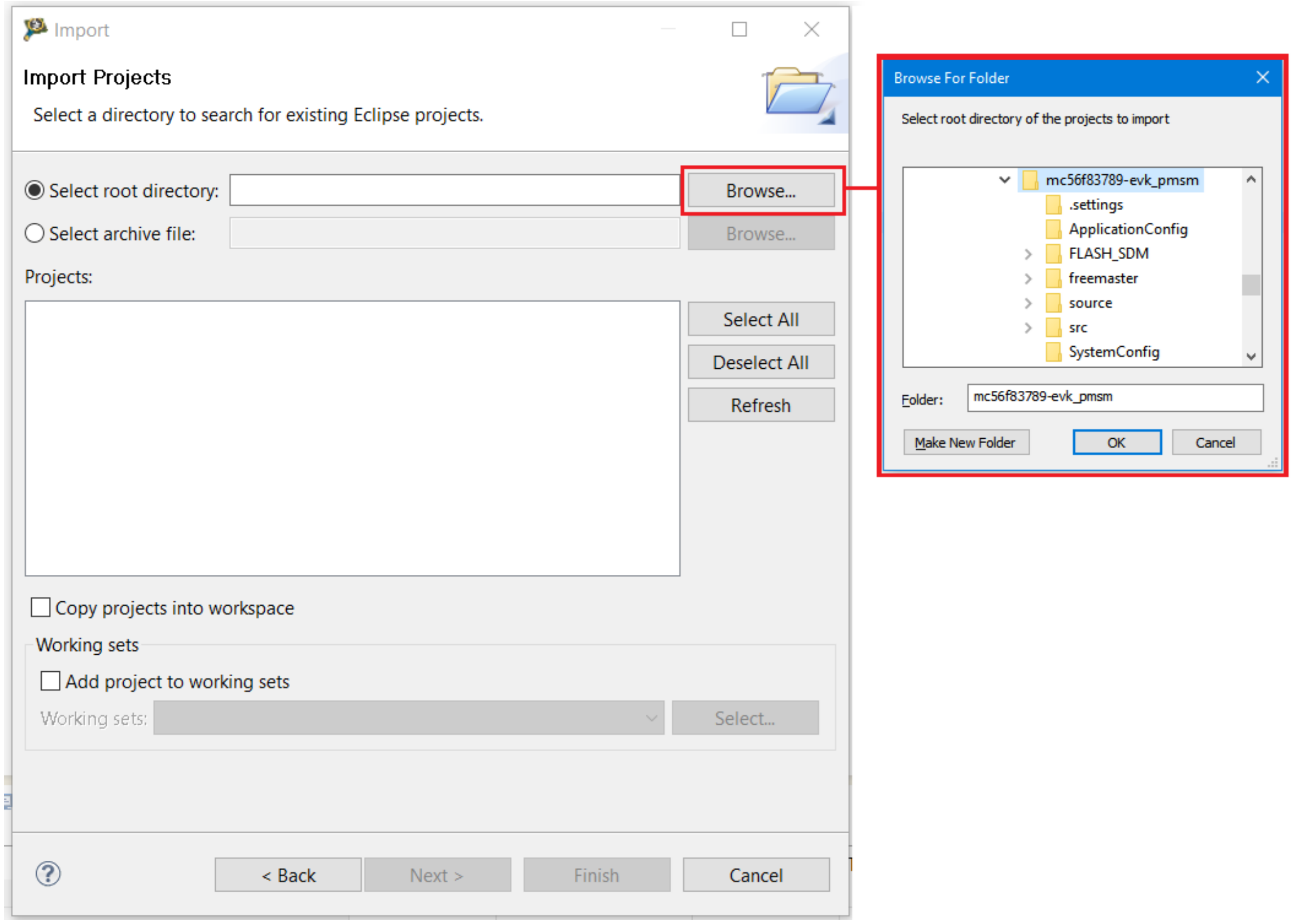

Figure 10. Importing project in CW—third step

- Confirm the selection by clicking “Finish”, as shown in Figure 11:

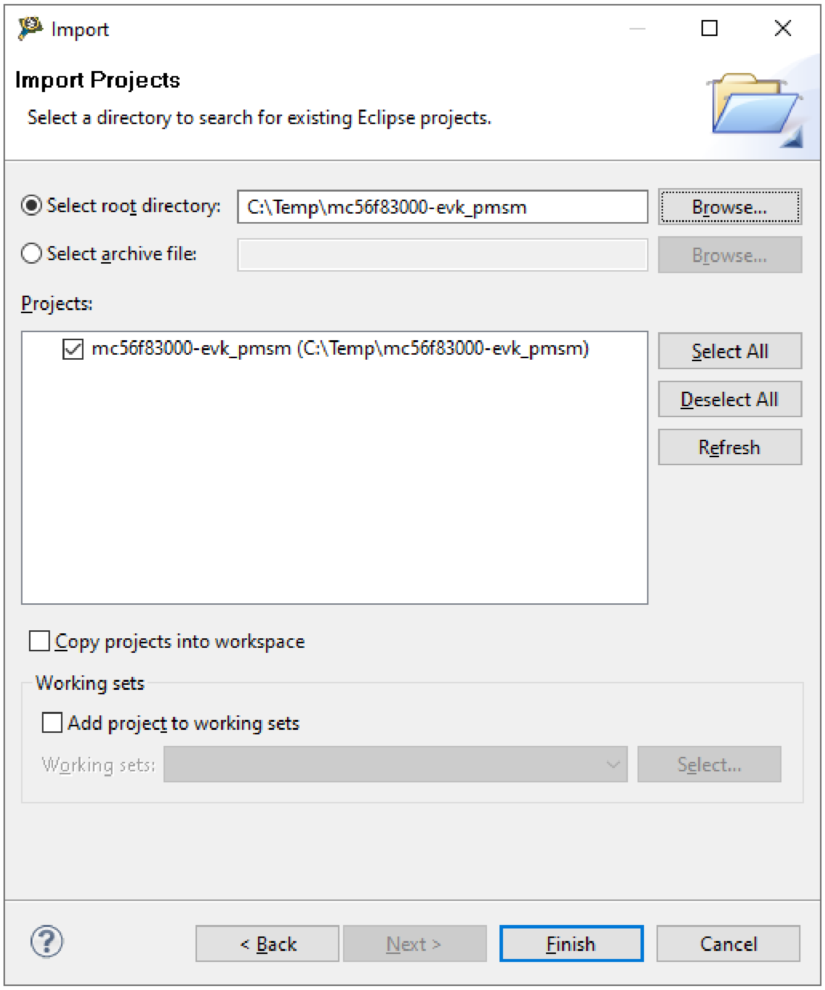

Figure 11. Importing project in CW—fourth step

The project is now imported in the CodeWarrior Development Studio (Figure 12). Point 1 shows the imported project in "Project Explorer" and point 2 shows the source code of this project. Build the project by clicking the "build" icon (point 3). When the project is compiled, use the debugger (point 4) to load the software into the MCU. Use the predefined debugger (P&E Micro—OpenSDA) or choose a different debugger from the menu. In the top list menu, select "Run-> Debug Configuration" to define a different type of debugger or change the conditions of debugging (such as the optimization level).

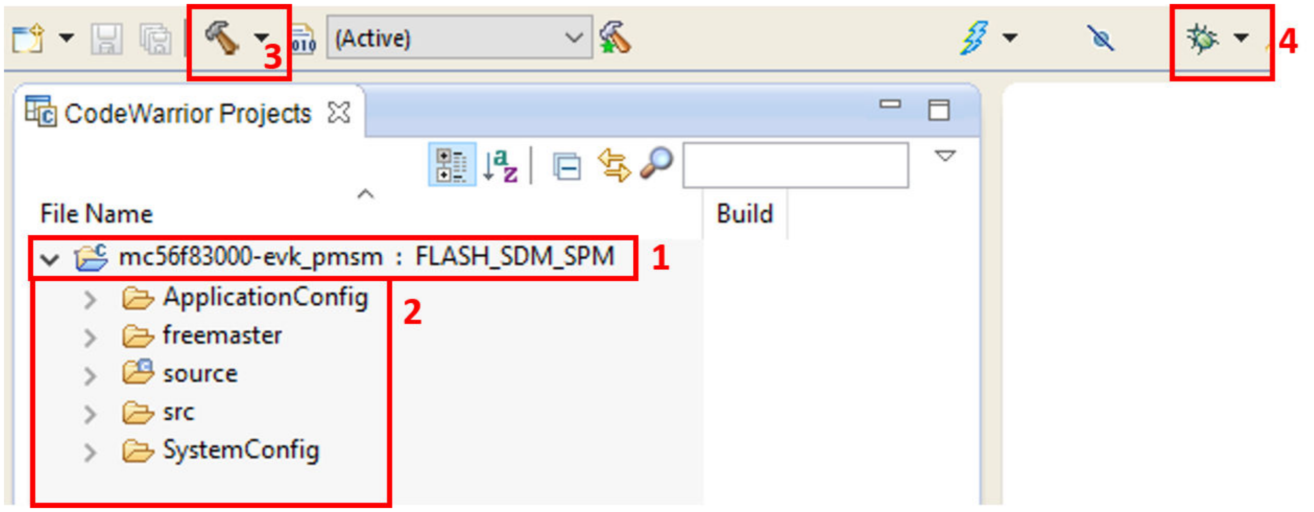

Figure 12. Imported CW project

- Click the debug button (point 4). The “Select configuration” window opens (Figure 13). Select the correct configuration for the debugger interface that you are using (P&E Micro, USB-TAP, and others).

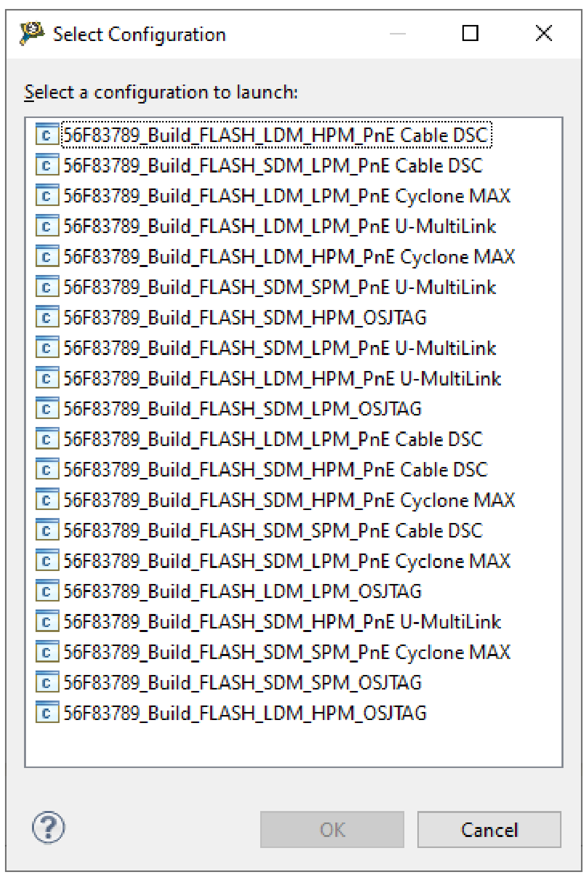

Figure 13. Select Configuration window

The CW project supports several configurations that are supported for the particular MCU (MC56F83xxx) and the board (see Table 5). The configurations are derived from the QuickStart stationary configurations. The selection of project configuration is available in the drop-down menu (point 1 in Figure 12).

Table 5. CodeWarrior project configuration

Configuration	Description
FLASH_SDM_SPM	Flash short data model, short program model
FLASH_SDM_LPM	Flash short data model, long program model
FLASH_SDM_HPM	Flash short data model, huge program model
FLASH_LDM_LPM	Flash long data model, long program model
FLASH_LDM_HPM	Flash long data model, huge program model

5.2 Compiler warnings

Warnings are diagnostic messages that report constructions that are not inherently erroneous and warn about potential runtime, logic, and performance errors. In some cases, warnings can be suspended, and these warnings do not show during the compiling process. One of such special cases is the “unused function” warning, where the function is implemented in the source code with its body, but the function is not used. This case can occur when you implement the function as a supporting function for better usability, but do not use the function for any special purposes for a while.

Chapter 6
DSC56800EX QuickStart

The DSC56800EX_Quick_Start development environment provides fully debugged peripheral drivers, examples, and interfaces that enablke you to create your own C application code, independent of the core architecture. This environment is developed to complement the existing development environment for the NXP 56F8xxx embedded processors. It provides a software infrastructure that enables you to develop efficient, ready-to-use, high-level software applications that are fully portable and reusable between different core architectures. The maximum portability is achieved for devices with comparable on-chip peripheral modules.

The DSC56800EX_Quick_Start environment is composed of these major components:

- Core-system infrastructure
- On-chip drivers with a defined API
- Sample example applications
- Graphical Configuration Tool (GCT) – see Figure 14
- FreeMASTER software support

Download an up-to-date version of the DSC Quick Start tool (2.7 or higher). To install and integrate the tool into the CodeWarrior V11.1 IDE, see DSC56800EX Quick Start User Manual (document DSC56800EXQSUG).

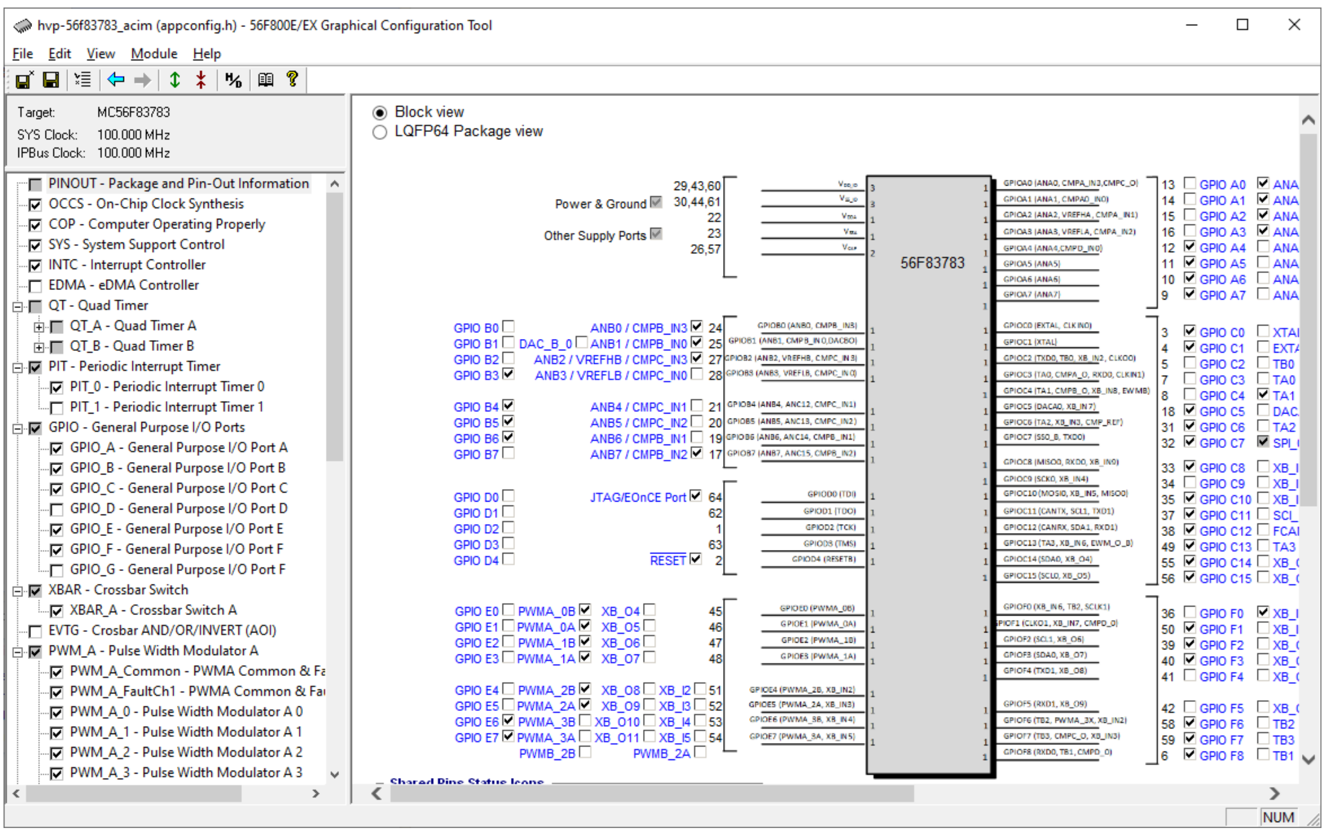

Figure 14. Graphical Configuration Tool

6.1 MCU peripheral configuration

The MCU clock and peripherals are configured in GCT. The final configuration is generated and stored in the *appconfig.h* file, which is attached in the CodeWarrior project.

The *appconfig.h* file is included in the application source code and the register initialization values are used in the “init” functions to physically configure the peripheral module. Initialization functions exist for each module and they are typically invoked using the “ioctl” INIT call (for example; use the SCI1_INIT "ioctl" command to initialize SCI module 1). See the quick start user manual for more details.

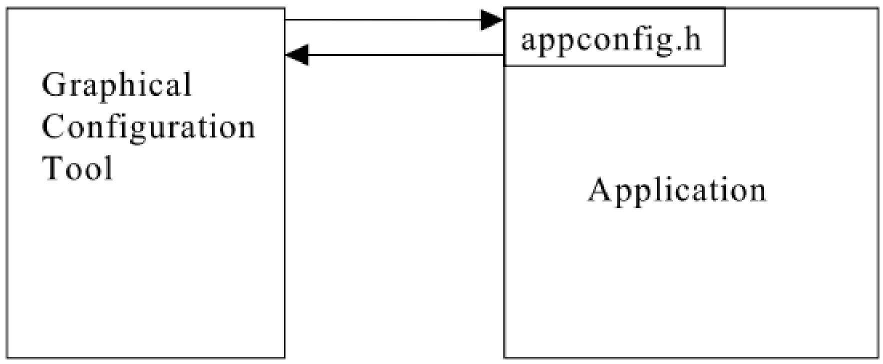

Figure 15. GCT usage

Chapter 7
User interface - FreeMASTER

FreeMASTER consists of two parts: the PC application used for variable visualization, and the set of software drivers running in the embedded application. The data is transferred between the PC and the embedded application via the serial interface. This interface is provided by the Silicon Labs CP210x USB-TO-UART bridge located on the FRDM-56F83000 platform (BOOT USB connector J14) . To run FreeMASTER including the MCAT tool, double-click the **.pmp* file located in the *//freemaster/* folder. FreeMASTER starts and the environment is created automatically, as defined in the **.pmp* file. This application enables you to tailor the PMSM sensorless application demonstration for any induction motor.

7.1 Remote control using FreeMASTER

The remote operation is provided by FreeMASTER via the USB interface. FreeMASTER 2.5 (or higher) is required for the application to operate properly. Download an up-to-date version of FreeMASTER at www.nxp.com/freemaster.

Perform these steps to control a PMSM motor using FreeMASTER:

1. Open the FreeMASTER file located in the */freemaster/* folder (*pmsm_ref_sol.pmp*).
   - Click the communication button (the red STOP button in the top left-hand corner, as shown in Figure 16) to establish the communication.

Figure 16. Red STOP button placed in top left-hand corner

   - If the communication is established successfully, the FreeMASTER communication status in the bottom right-hand corner changes from “Not connected” to “RS232 UART Communication; COMxx; speed=19200” (see Figure 17). Otherwise, the FreeMASTER warning pop-up window appears.

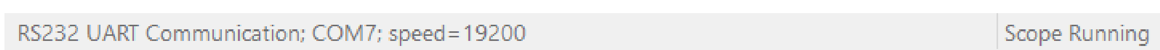

Figure 17. Example FreeMASTER communication established successfully

2. Control the PMSM motor using the control page or MCAT .

If the communication is not established successfully, perform these steps:

1. Go to the “Project→Options→Comm” tab and make sure that “Silicon Labs CP210x” is set in the “Port” option and the communication speed is set to 19200 bit/s.

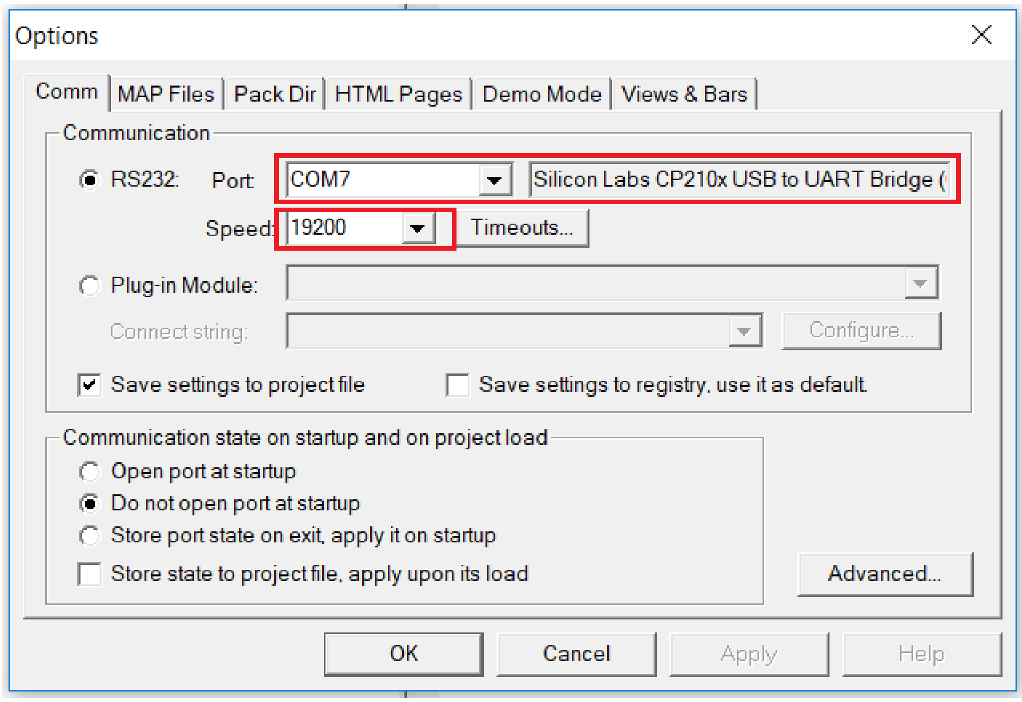

Figure 18. FreeMASTER communication setup window

2. If "Silicon Labs CP210x USB to UART Bridge" is not printed out in the message box next to the "Port" dropdown menu, unplug and then plug in the USB cable, reopen the FreeMASTER project, and make sure that the CP210x drivers are installed on your host PC. https://www.silabs.com/documents/public/software/CP210x_Windows_Drivers.zip
3. Supply your development board from a sufficient energy source. Sometimes the PC USB port is not sufficient to supply the development board.

7.1.1 Symbol file selection

The symbol table can be loaded directly from the embedded application executable, if it is the standard ELF debugging format. For other cases, the text MAP file generated by the linker may be loaded and parsed for symbol information.

The symbol file is generated during the build process to the */FLASH_xxx_xxx/* folder. The folder specific name depends on the active CodeWarrior configuration. The symbol file has the **.elf* extension (*56F83789_Build.elf*).

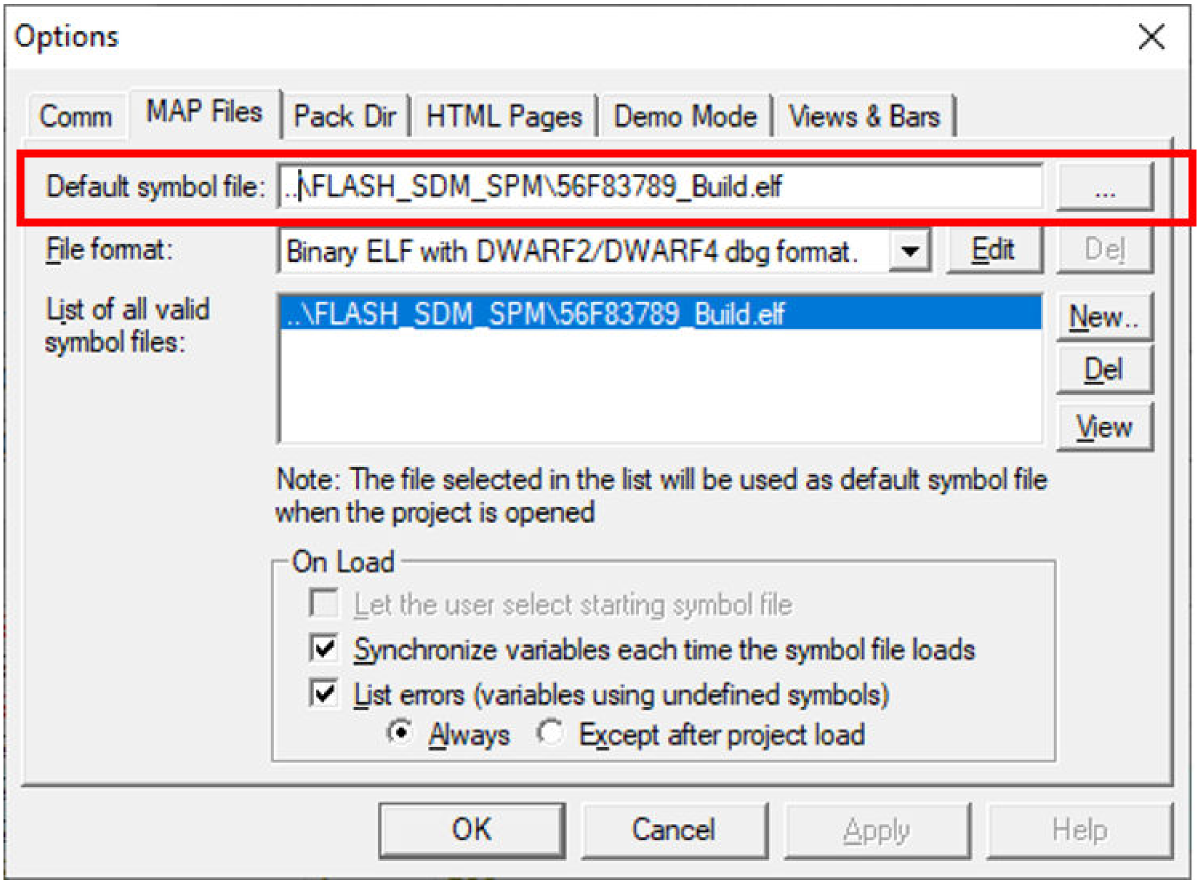

Figure 19. FreeMASTER MAP Files tab

7.1.2 Control page – control struct

After launching the application and performing all necessary settings, control the PMSM motor using the FreeMASTER control struct page.

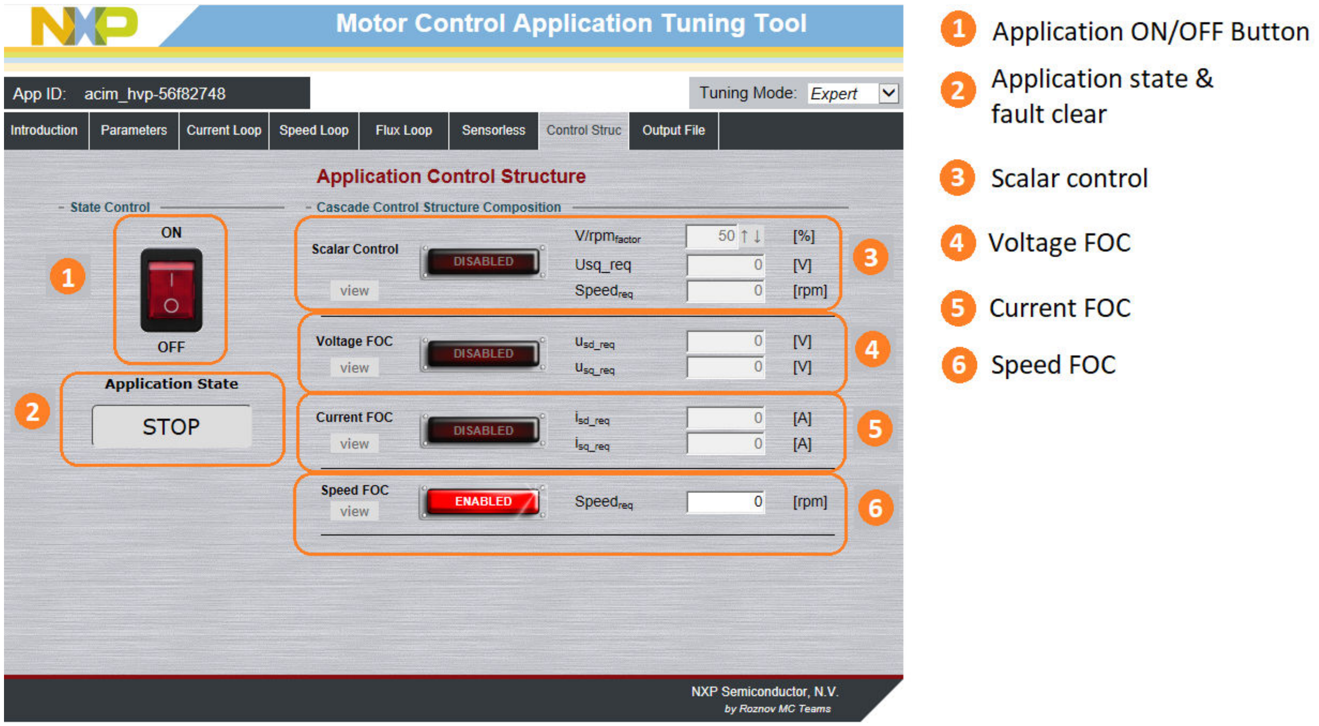

Figure 20. FreeMASTER control struct page

The basic instructions to run a motor are as follows:

- Click the “ON/OFF” button to start the application.
- To start the motor, set up the required speed in the "Speedreq" box (Figure 20 – point 6).
- In case of a fault, click the fault notification to clear the fault (Figure 20 – point 2).
- Click the “ON/OFF” button to stop the motor.

The MCAT tuning tool is available. This tool is targeted at motor-control applications, where you can change the motor-control parameters. Each tab represents one submodule of the embedded-side control and tunes its parameters. For more information, see *Sensorless ACIM Field-Oriented Control on MC56F287xx* (document ANxxxx). The control page is still available in the “App Control” tab in MCAT.

7.1.3 Control page – app control

The application can be controlled also in the "App Control" page.

The basic instructions for running a motor are as follows:

- Click the “RUN” button to start the application.
- To start or stop the DEMO MODE, click the “Demo Mode On/Off” button.
- For manual control, click only the “RUN” button and set the required speed on the slider. You can also set the current limitation on the second slider.

Click the “STOP” button to stop the motor.

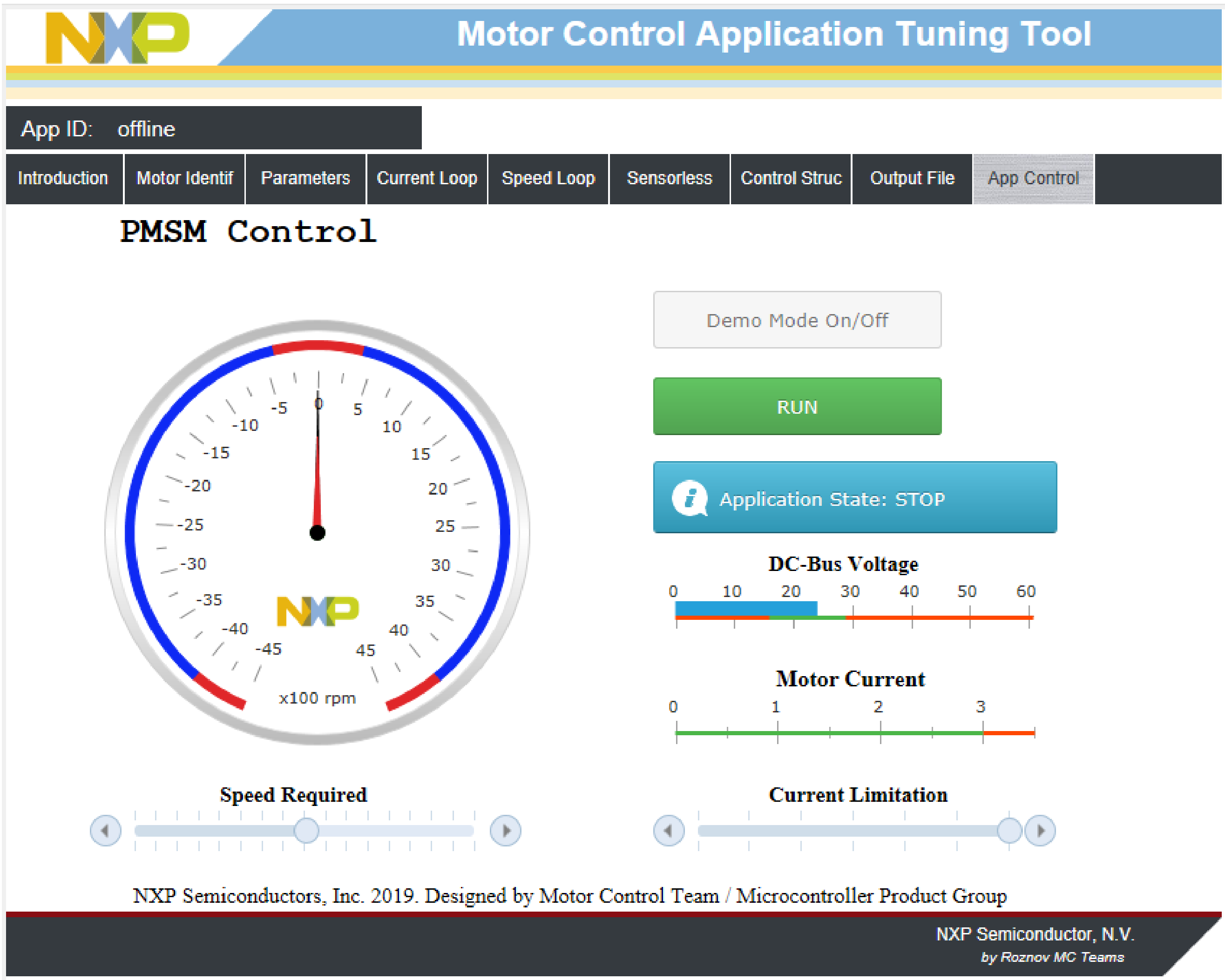

Figure 21. FreeMASTER App Control page

Chapter 8
Performing basic tasks

8.1 Running the motor

1. Assembly the NXP hardware according to the instructions in Hardware setup.
2. Flash the correct project into the target device via the debug interface, as described in Building and debugging application.
3. Open the FreeMASTER project and establish the communication between the MCU and the PC according to the instructions in Remote control using FreeMASTER.
4. Set the required motor speed.

8.2 Stopping the motor

1. Click the “STOP” button in the FreeMASTER control page (Figure 16).
2. Set the speed in the "Speedreq" box to zero or set the Speed Required variable to zero.
3. In case of an emergency, turn off the power supply.

8.3 Clearing the fault

To clear the fault, remove the fault source (for example under-voltage) and click the fault notification in the control page.

Chapter 9
Acronyms and abbreviations

Table 6. Acronyms and abbreviations

Term	Meaning
AC	Alternating Current
PMSM	Permanent Magnet Synchronous Motor
AN	Application Note
DRM	Design Reference Manual
FOC	Field-Oriented Control
MCAT	Motor Control Application Tuning tool
MCU	Microcontroller
MSD	Mass Storage Device

Chapter 10 References

The following documents are available at www.nxp.com:

- *Sensorless PMSM Control Design*(document DRM148)
- *Sensorless PMSM Field-Oriented Control on DSC 56F837xx* (document AN12745)
- *MC56F83XXX Reference Manual* (document MC56F83XXXRM)
- *Freescale High-Voltage Motor Control Platform User's Guide* (document THVPMC3PHUG)
- *DSC56800EX Quick Start User's Guide* (document DSC56800EXQSUG)

How To Reach Us

Home Page:

nxp.com

Web Support:

nxp.com/support

For more information, please visit: http://www.nxp.com

For sales office addresses, please send an email to: salesaddresses@nxp.com

Date of release: 03/2020

Document identifier: PMSMSCEVKHVPUG