NXP Semiconductors
User's Guide

Document identifier: ACIMSCHVPUG
Rev. 0, 03/2020

ACIM Sensorless Control using HVP-56F83783

Contents

Chapter 1
Introduction

This user’s guide provides a step-by-step guide on how to open, compile, and run AC Induction Machine (ACIM) project using NXP Digital Signal Controller (DSC) MC56F83xxx. It describes the basic compiling steps for CodeWarrior for the supported development platforms mentioned in Supported development boards. It also describes the initialization of the FreeMASTER GUI tool for motor-control applications.

Chapter 2
Supported development boards

Development boards with the MC56F837xx MCU are currently supported for motor-control applications. The development boards and the supported MCU are shown in Table 1. 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
MC56F83783	HVP-56F83783	HVP-MC3PH

Chapter 3
Hardware setup

This section describes the default supported hardware configurations consisting of the HVP-MC3PH power stage, all supported daughter cards, and the default induction motor.

3.1 Default AC induction motor

The default induction motor (for which the application is pre-tuned) is the Elektrim 0.33HP motor. The motor parameters provided by the manufacturer are listed in Table 2:

Table 2. Elektrim 0.33HP motor parameters

Characteristic	Symbol	Value	Units
Nominal voltage	U_{RN}	230/400	V
Nominal frequency	f_{RN}	50	Hz
Nominal current	I_{RN}	1.5/0.85	A
Number of pole pairs	pp	2	—

3.2 Running ACIM application on high-voltage development platform

To run the ACIM 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 from www.nxp.com/hvp or from distributors, and easily build the hardware platform for the target application.

3.2.1 HVP-MC3PH power stage

The NXP High-Voltage Platform is an evaluation and development solution for 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 the *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.

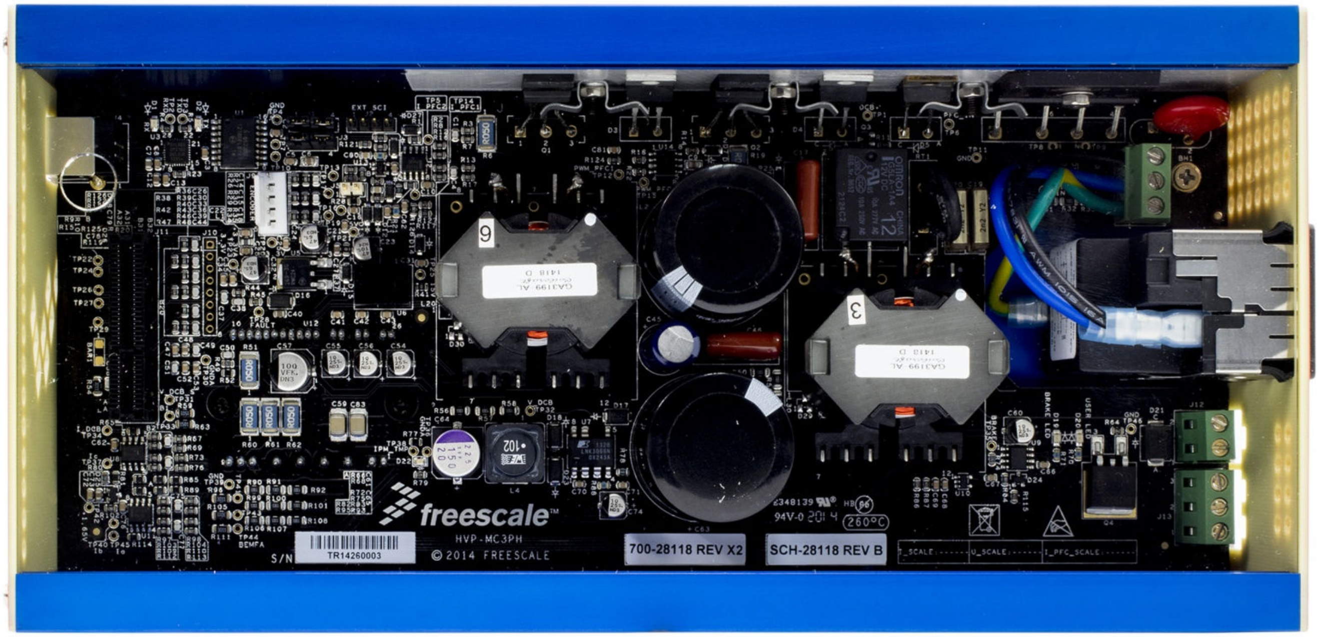


Figure 1. HVP-MC3PH High-Voltage Platform

3.2.2 HVP- 56F83783 daughter card

The HVP-56F83783 MCU daughter card contains the DSC MC56F83783 family MCU 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.

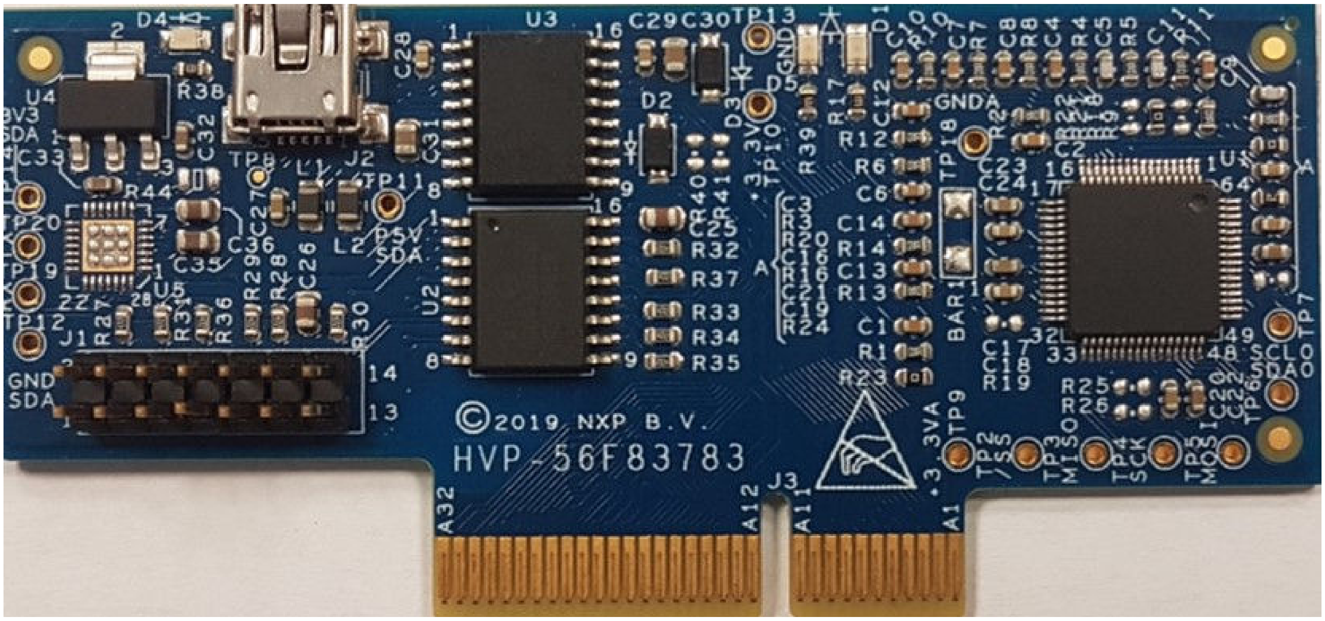


Figure 2. HVP-56F83783 daughter card

3.2.3 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 ACIM motor phase wires to the screw terminals on the board (MOTOR connector J13).
4. Place the protective plastic cover on top of the power stage to ensure safety.
5. Plug the USB cable from the USB host to the OpenSDA micro-USB connector.
6. 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 ACIM 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 the 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 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 microprocessors. The CW IDE supports a wide range of debuggers, such as P&E Micro or USB-TAP.

To open the 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 the CW IDE.
- Click the “File” menu in the top-left corner of the IDE, and select “Import…”:

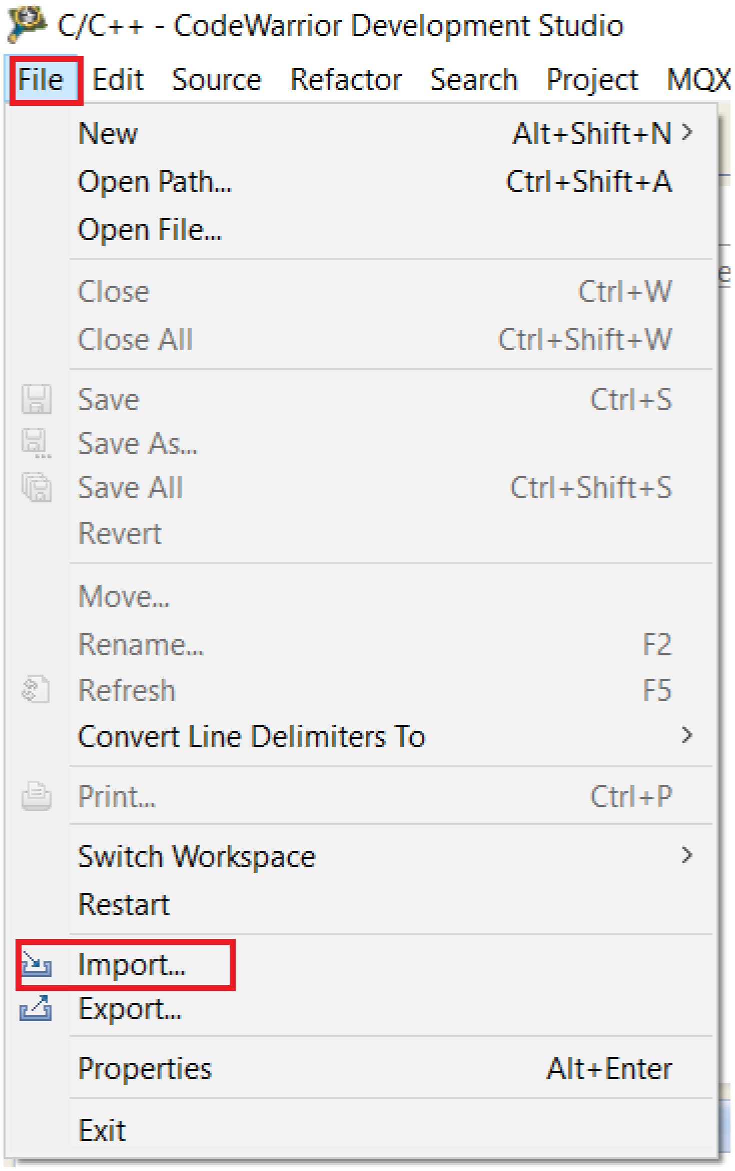


Figure 3. Importing project in CW IDE—first step

- The “Import” window opens. Highlight the “Existing Projects into Workspace” option in the “General” folder, and click the “Next” button. The window shown in Figure 4 appears. Select “General/Existing Projects into Workspace”.

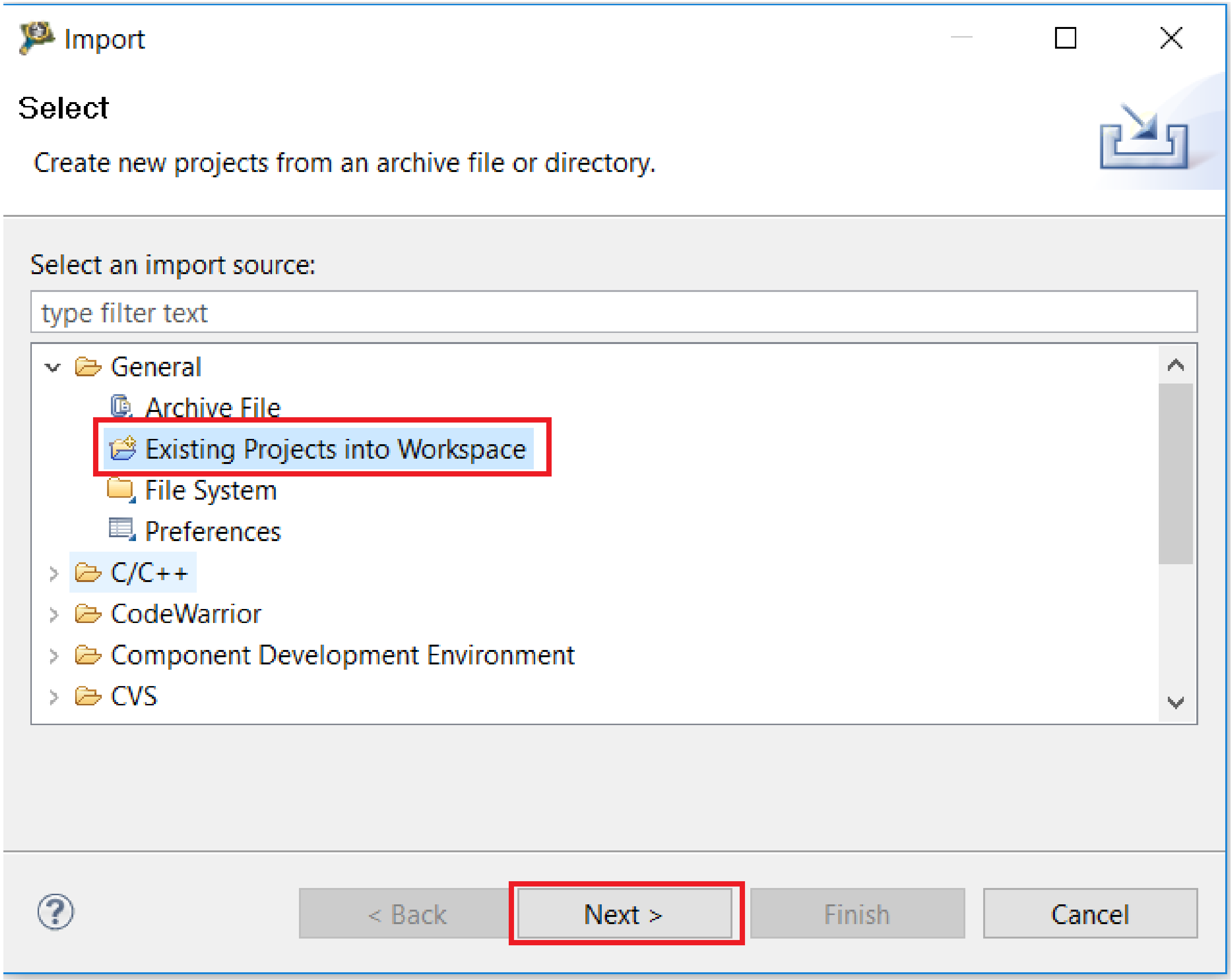
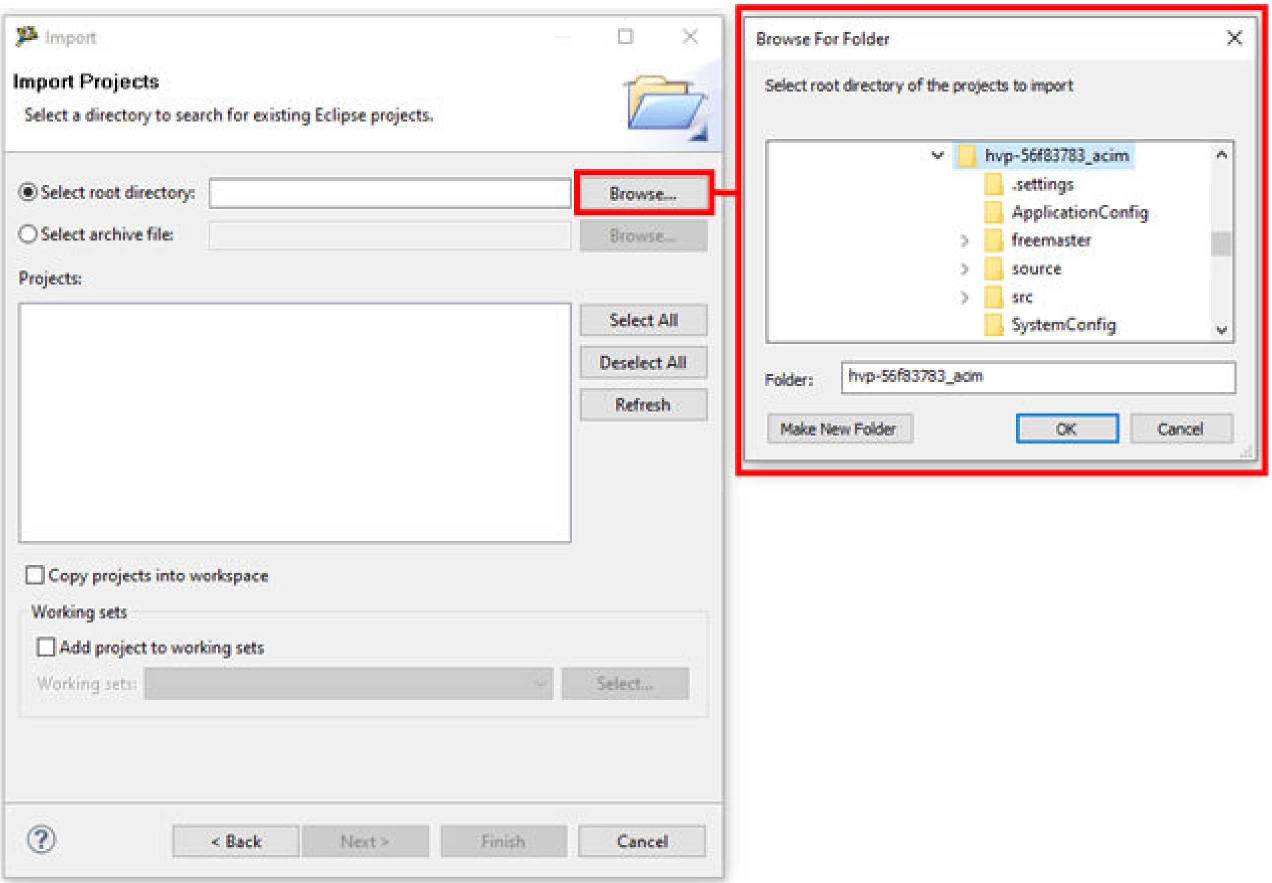


Figure 4. Importing project in CW IDE—second step

- The “Import” window opens (as shown in Figure 5). Click the “Browse” button, and then locate the project. Click the “OK” button.



Import Projects

Select a directory to search for existing Eclipse projects.

● Select root directory:

○ Select archive file:

☐ Copy projects into workspace

☐ Add project to working sets

Browse For Folder

Select root directory of the projects to import

Folder: hvp-56f83783_acim

Figure 5. Importing project in CW IDE—third step

- Confirm the selection by clicking “Finish”, as shown in Figure 6:

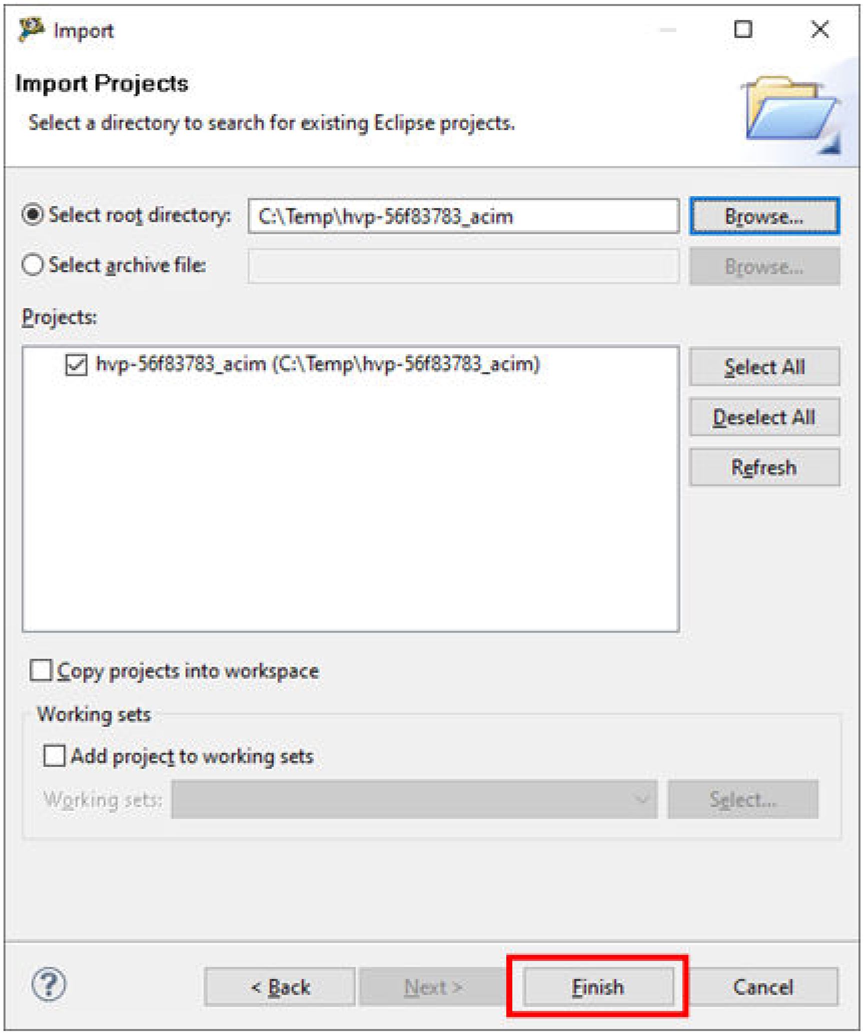

Import
Import Projects
Select a directory to search for existing Eclipse projects.
● Select root directory: C:\Temp\hvp-56f83783_acim
Browse...
○ Select archive file:
Browse...
Projects:
☑ hvp-56f83783_acim (C:\Temp\hvp-56f83783_acim)
Select All
Deselect All
Refresh
☐ Copy projects into workspace
Working sets
☐ Add project to working sets
Working sets:
Select...
< Back
Next >
Finish
Cancel

Figure 6. Importing project in CW IDE—fourth step

The project is now imported in the CW IDE (Figure 7). Point 1 shows the imported project in the “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 in Figure 7) 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).

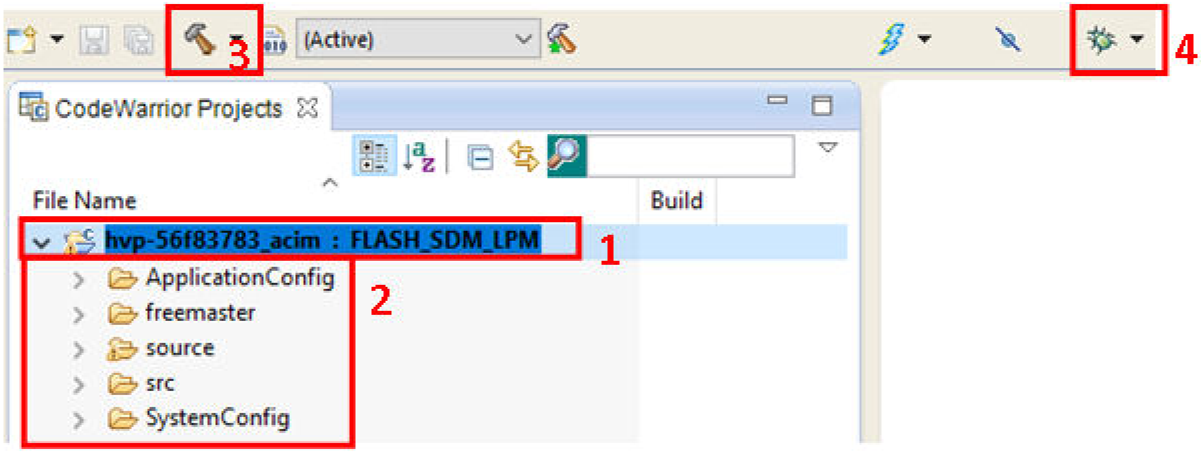
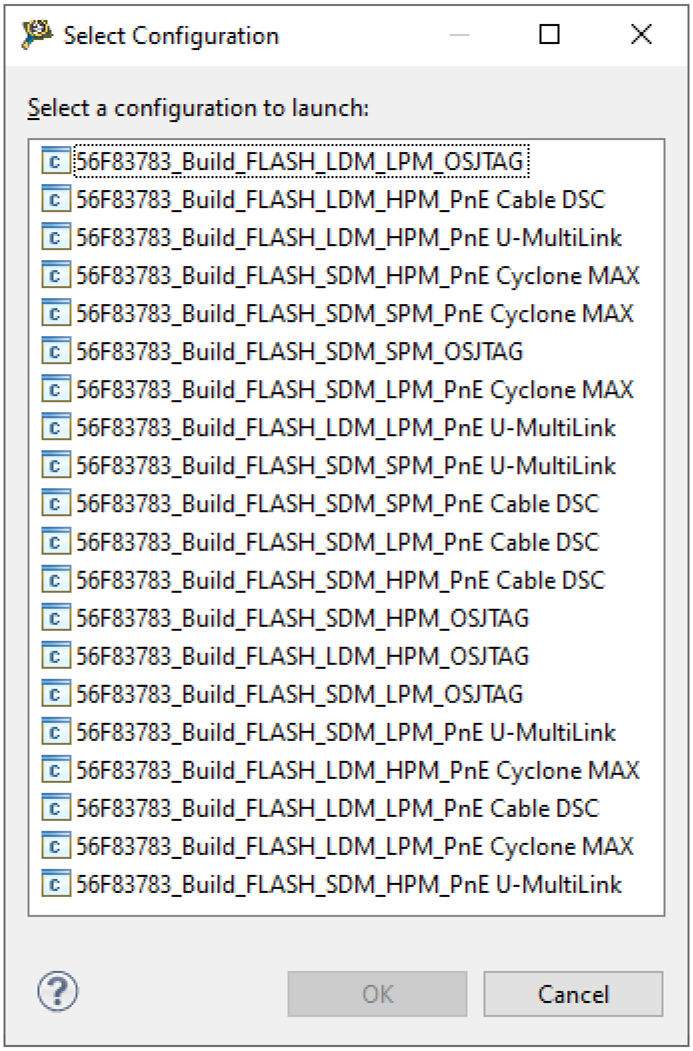


Figure 7. Imported CW IDE project

Click the debug button (Figure 7). The “Select configuration” window opens (Figure 8). Select the correct configuration for the debugger interface which you are using (P&E Micro, USB-TAP, and so on).



Select Configuration

Select a configuration to launch:

- 56F83783_Build_FLASH_LDM_LPM_OSJTAG
- 56F83783_Build_FLASH_LDM_HPM_PnE Cable DSC
- 56F83783_Build_FLASH_LDM_HPM_PnE U-MultiLink
- 56F83783_Build_FLASH_SDM_HPM_PnE Cyclone MAX
- 56F83783_Build_FLASH_SDM_SPM_PnE Cyclone MAX
- 56F83783_Build_FLASH_SDM_SPM_OSJTAG
- 56F83783_Build_FLASH_SDM_LPM_PnE Cyclone MAX
- 56F83783_Build_FLASH_LDM_LPM_PnE U-MultiLink
- 56F83783_Build_FLASH_SDM_SPM_PnE U-MultiLink
- 56F83783_Build_FLASH_SDM_SPM_PnE Cable DSC
- 56F83783_Build_FLASH_SDM_LPM_PnE Cable DSC
- 56F83783_Build_FLASH_SDM_HPM_PnE Cable DSC
- 56F83783_Build_FLASH_SDM_HPM_OSJTAG
- 56F83783_Build_FLASH_LDM_HPM_OSJTAG
- 56F83783_Build_FLASH_SDM_LPM_OSJTAG
- 56F83783_Build_FLASH_SDM_LPM_PnE U-MultiLink
- 56F83783_Build_FLASH_LDM_HPM_PnE Cyclone MAX
- 56F83783_Build_FLASH_LDM_LPM_PnE Cable DSC
- 56F83783_Build_FLASH_LDM_LPM_PnE Cyclone MAX
- 56F83783_Build_FLASH_SDM_HPM_PnE U-MultiLink

Figure 8. Select Configuration window

The CW IDE project supports several configurations that are supported for the particular MCU (MC56F83xxx) and the board (see Table 3). The configurations are derived from theQuickStart stationary configurations. The selection of project configuration is available in drop-down menu (point 1 in Figure 7).

Table 3. CW IDE 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 this function is not used. This case can occur in situations where you implement the function as a supporting function for better usability, but do not use the function for any special purposes for a while.

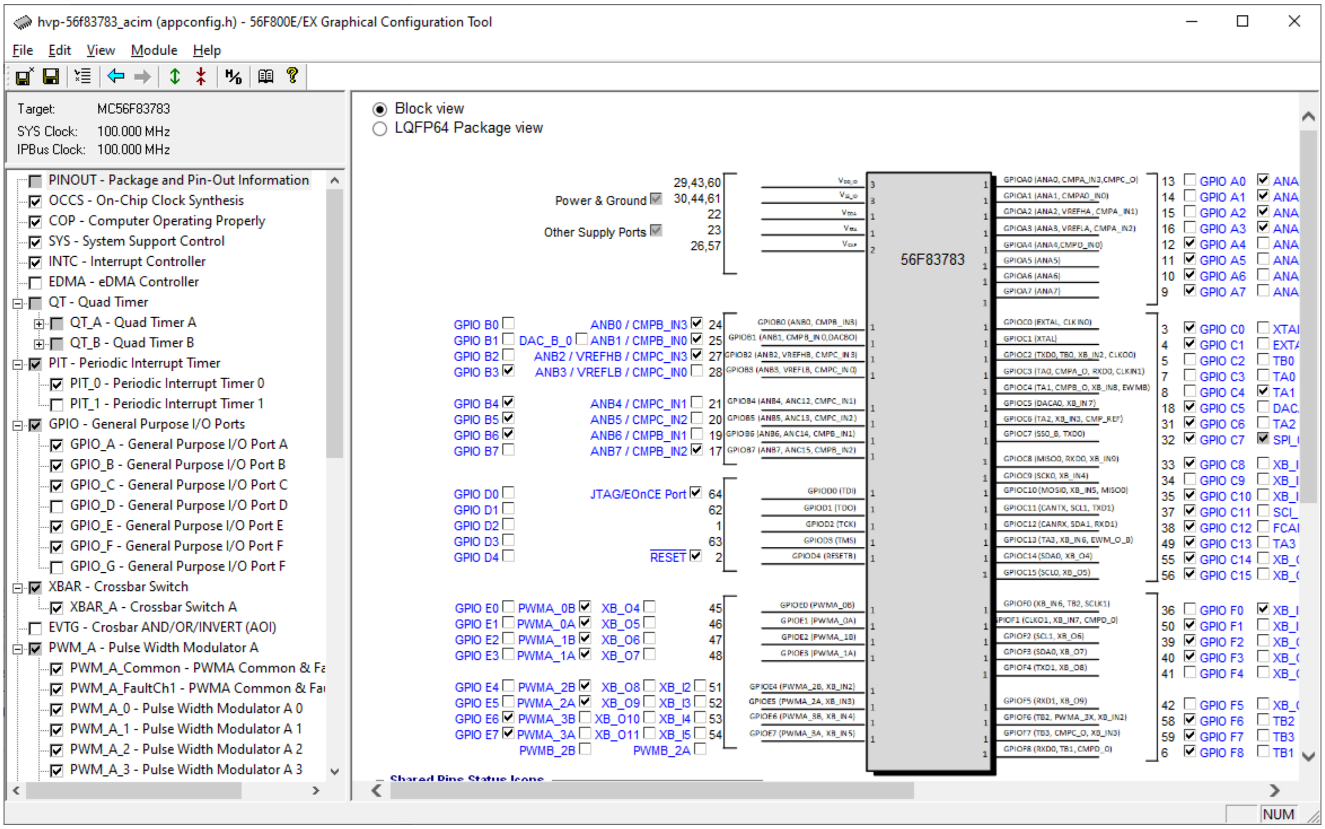
Chapter 6
DSC56800EX QuickStart

The DSC56800EX QuickStart development environment provides fully debugged peripheral drivers, examples, and interfaces, that allow programmers to create their own C application code, independent of the core architecture. This environment is developed to complement the existing development environment for NXP 56F8xxx embedded processors. It provides a software infrastructure that allows the development of 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 QuickStart environment is composed of the following major components:

- Core-system infrastructure
- On-chip drivers with a defined API
- Sample example applications
- Graphical Configuration Tool (GCT); see Figure 9.
- 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 the *DSC56800EX Quick Start User's Guide* (document DSC56800EXQSUG).



Target: MC56F83783
SYS Clock: 100.000 MHz
IPBus Clock: 100.000 MHz

● Block view
○ LQFP64 Package view

- ☐ PINOUT - Package and Pin-Out Information
- ☑ OCCS - On-Chip Clock Synthesis
- ☑ COP - Computer Operating Properly
- ☑ SYS - System Support Control
- ☑ INTC - Interrupt Controller
- ☐ EDMA - eDMA Controller
- ☐ QT - Quad Timer
  - ☐ QT_A - Quad Timer A
  - ☐ QT_B - Quad Timer B
- ☑ PIT - Periodic Interrupt Timer
  - ☑ PIT_0 - Periodic Interrupt Timer 0
  - ☐ PIT_1 - Periodic Interrupt Timer 1
- ☑ GPIO - General Purpose I/O Ports
  - ☑ GPIO_A - General Purpose I/O Port A
  - ☑ GPIO_B - General Purpose I/O Port B
  - ☑ GPIO_C - General Purpose I/O Port C
  - ☐ GPIO_D - General Purpose I/O Port D
  - ☑ GPIO_E - General Purpose I/O Port E
  - ☑ GPIO_F - General Purpose I/O Port F
  - ☐ GPIO_G - General Purpose I/O Port F
- ☑ XBAR - Crossbar Switch
  - ☑ XBAR_A - Crossbar Switch A
- ☐ EVTG - Crosbar AND/OR/INVERT (AOI)
- ☑ PWM_A - Pulse Width Modulator A
  - ☑ PWM_A_Common - PWMA Common & Fa
  - ☑ PWM_A_FaultCh1 - PWMA Common & Fai
  - ☑ PWM_A_0 - Pulse Width Modulator A 0
  - ☑ PWM_A_1 - Pulse Width Modulator A 1
  - ☑ PWM_A_2 - Pulse Width Modulator A 2
  - ☑ PWM_A_3 - Pulse Width Modulator A 3

Figure 9. Graphical configuration tool

6.1 GCT usage MCU peripheral configuration

The MCU clock and peripherals are configured in the GCT. The final configuration is generated and stored in the *appconfig.h* file, which is attached in the CodeWarrior IDE 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. An initialization function exists for each module and it is typically invoked using the “ioctl” INIT call (for example; use the SCI1_INIT ioctl command to initialize the SCI module 1). See the *Quick Start User's Guide* (document DSC56800EXQSUG) for more details.

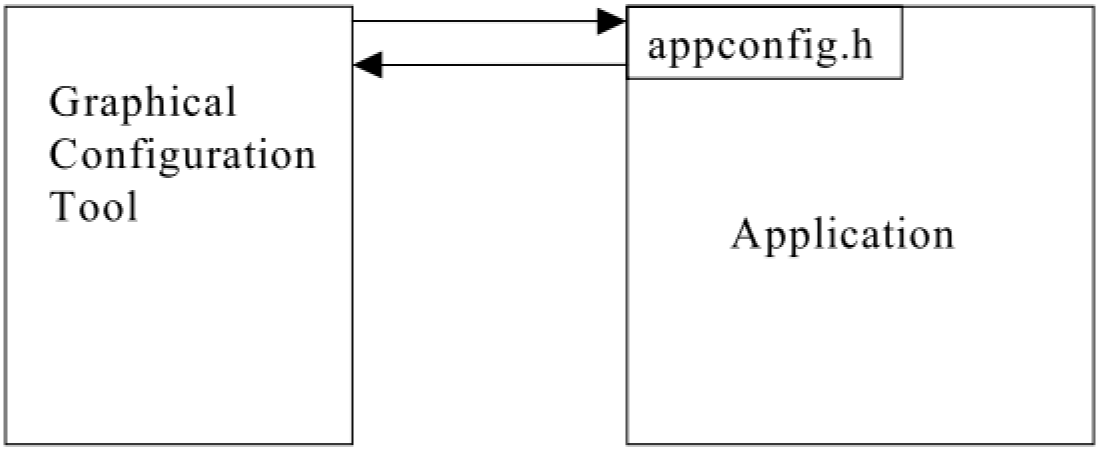


Figure 10. GCT usage

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 CP2102 USB-TO-UART bridge located on the HVP-MC3PH platform (USB connector J4). 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 ACIM 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 the up-to-date version of FreeMASTER at www.nxp.com/freemaster.

Perform these steps to control an PMS motor using FreeMASTER:

1. Open the FreeMASTER file located in */freemaster/* (*acim_ref_sol.pmp*).
    - Click the communication button (the red STOP button in the top left-hand corner, as shown in Figure 11) to establish the communication.

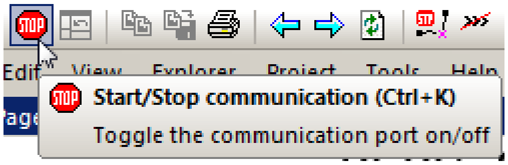
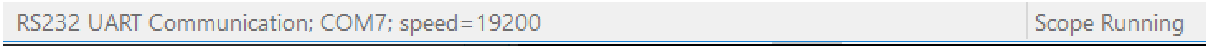

Edit
View
Explorer
Project
Tools
Help
Start/Stop communication (Ctrl+K)
Toggle the communication port on/off

Figure 11. 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 12). Otherwise, the FreeMASTER warning pop-up window appears.



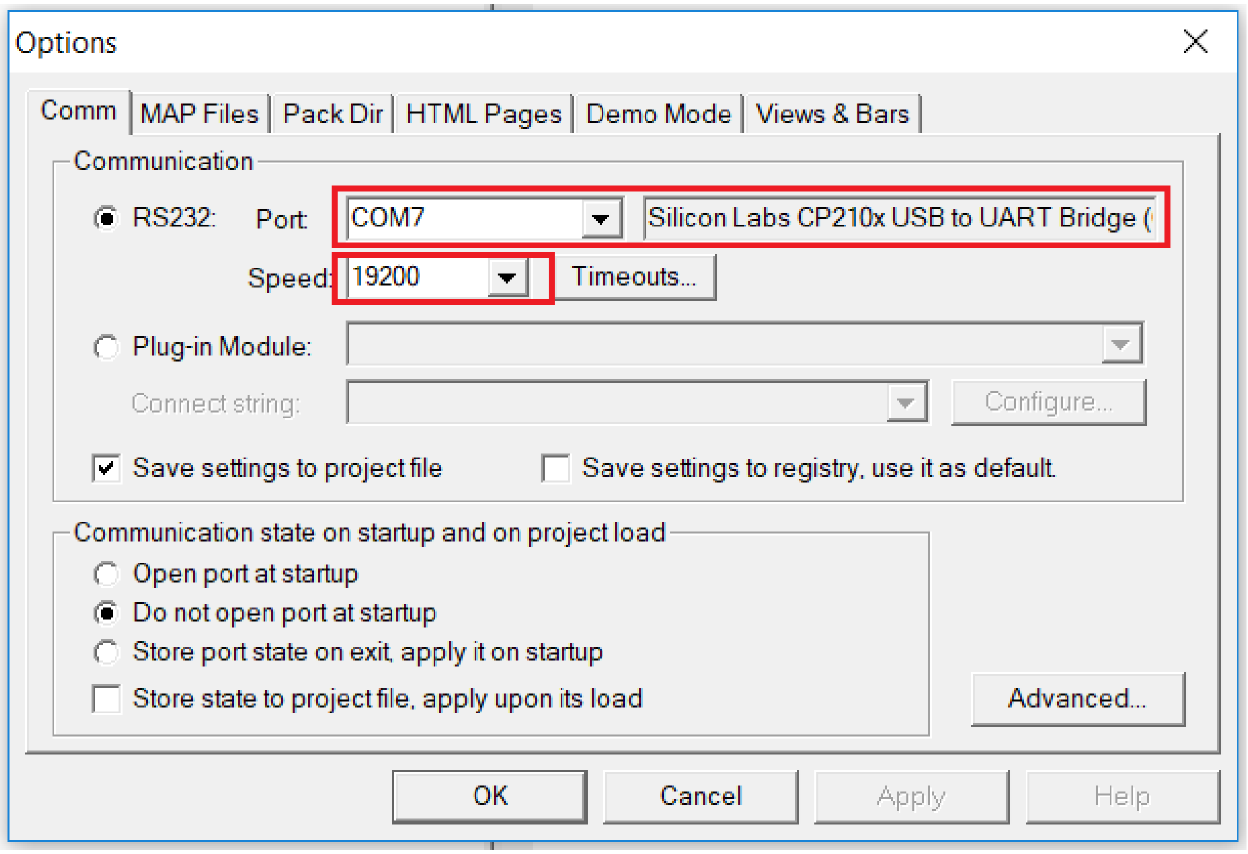
RS232 UART Communication; COM7; speed=19200 | Scope Running

Figure 12. Example FreeMASTER communication established successfully

2. Control the PMS 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.



● RS232: Port: COM7 Silicon Labs CP210x USB to UART Bridge (

Speed: 19200

○ Plug-in Module:

Connect string:

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

○ 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

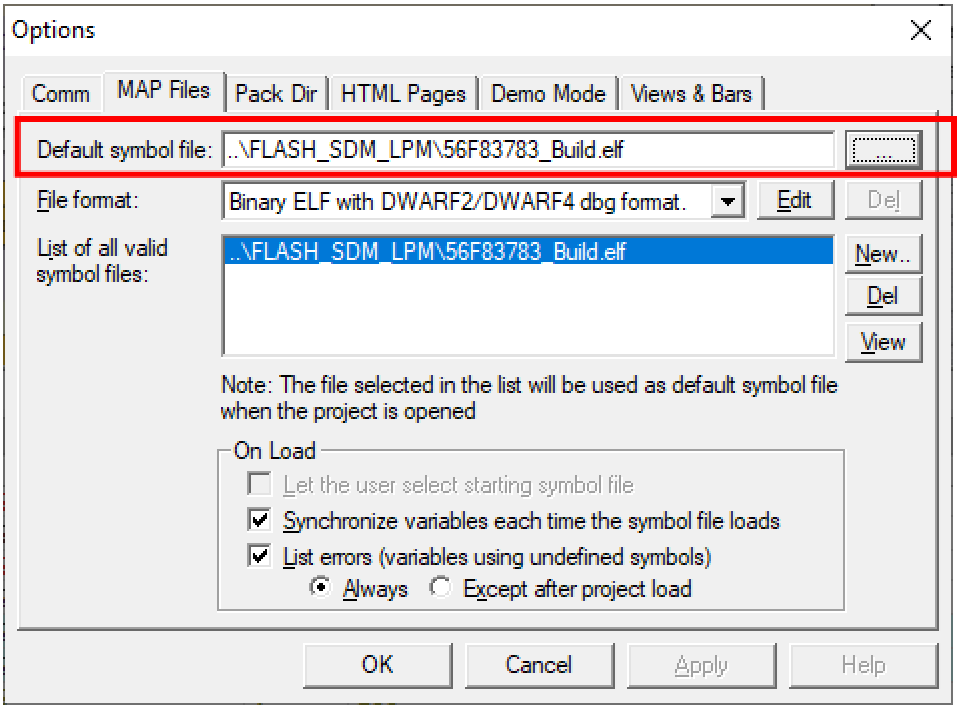
Figure 13. FreeMASTER communication setup window

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 and reopen the FreeMASTER project and make sure that the CP210x drivers are installed on your host PC (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 IDE configuration. The symbol files have the **.elf* extension (*56F83789_Build.elf*).



Options

Default symbol file: ..\FLASH_SDM_LPM\56F83783_Build.elf

File format: Binary ELF with DWARF2/DWARF4 dbg format.

List of all valid symbol files: ..\FLASH_SDM_LPM\56F83783_Build.elf

Note: The file selected in the list will be used as default symbol file when the project is opened

On Load

☐ Let the user select starting symbol file

☑ Synchronize variables each time the symbol file loads

☑ List errors (variables using undefined symbols)

● Always ○ Except after project load

Figure 14. FreeMASTER MAP files tab

7.1.2 Control page – control struct

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

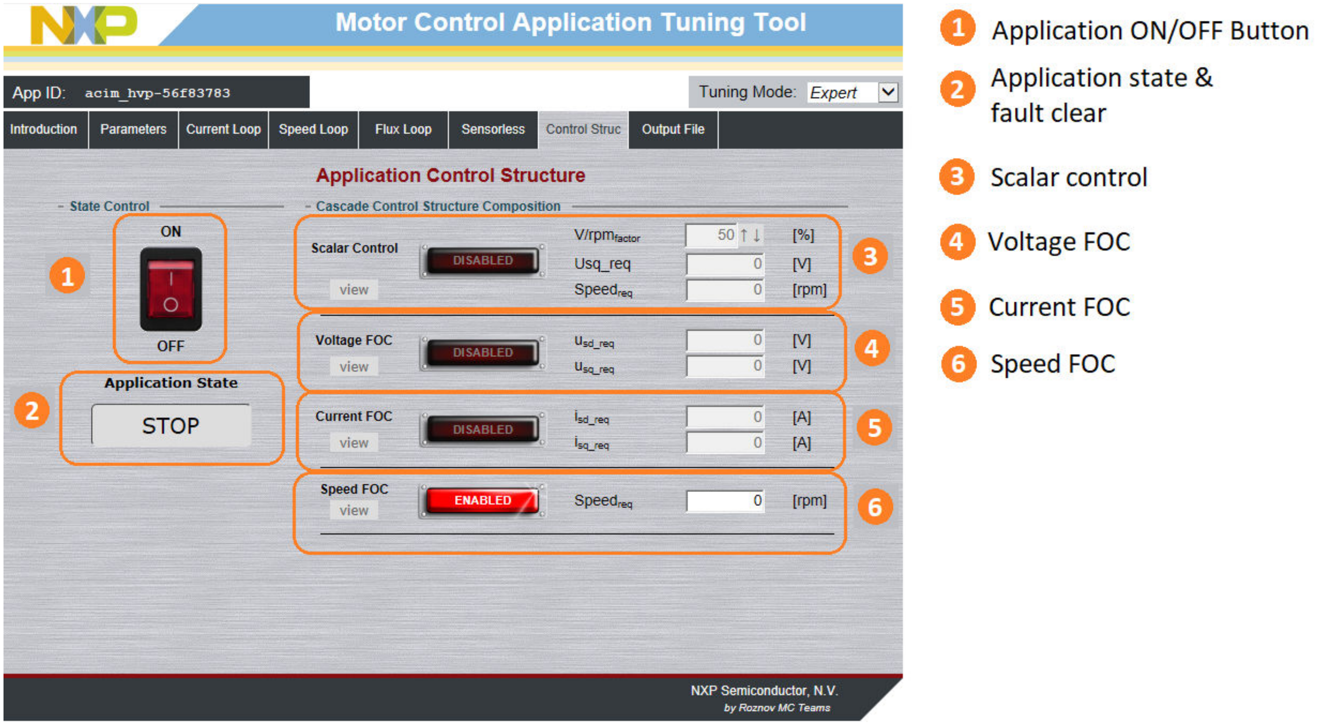


Figure 15. FreeMASTER control struct page

The basic instructions for running 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 15 – point 6).
- In case of a fault, click on the fault notification to clear the fault (Figure 15 – 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 MC56F83783* (document AN12379). The control page is still available in the “App Control” tab in MCAT.

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 the 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 up the required motor speed, as shown in Figure 15.

8.2 Stopping the motor

1. Click the “STOP” button in the FreeMASTER control page (Figure 15).
2. Set the speed to zero to "Speedreq" box (Figure 15 – point 6) or set the "Speed Required" variable to zero.
3. In case of 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 (Figure 15).

Chapter 9
Acronyms and abbreviations

Table 4. Acronyms and abbreviations

Term	Meaning
AC	Alternating Current
ACIM	AC Induction Machine
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 ACIM Field-Oriented Control*(document DRM150)
- *Sensorless ACIM Field-Oriented Control on DSC 56F83xxx* (document AN12379)
- *MC56F83XXX Reference Manual* (document MC56F83XXXRM)
- *Freescale High-Voltage Motor Control Platform User’s Guide* (document HVPMC3PHUG)
- *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: ACIMSCHVPUG