Freescale Semiconductor
User Guide

Document Number: PMSMUG
Rev. 0, 06/2013

PMSM Field-Oriented Control Using MC56F84789 DSC With Encoders

Demo Guide

by: Pavel Rech

1 Introduction

The application described in this user guide enables the execution of the field-oriented control (vector control) of 3-phase permanent magnet synchronous motor (PMSM) drives. Vector control allows high dynamic behavior because the motor torque and field excitation (the motor magnetic flux) are controlled independently and separately. But this can be achieved only when the motor excitation is precisely synchronized with the instantaneous rotor position. To execute the vector control algorithm successfully, the shaft position information must be known. The incremental encoder is used as the rotor position sensor in this application.

The demo application is based on the Freescale Digital Signal Controller (DSC), MC56F84789 MCU tower board. The hardware is built on the Freescale Tower rapid prototyping system. This includes the MCU tower board with MC56F84789 DSC and the Low-Voltage Three-Phase Motor Control module inserted in the tower system. Communication with the FreeMASTER debug tool is done using the SCI-to-USB converter placed on the MCU board. The tower system and PMSM are mounted on the Plexiglas.

Contents

The demo application consists of the following components.

- Plexiglas including:
    - The tower system, which consists of:
        - MC56F84789 tower board
        - Low-Voltage 3-Phase Motor Control module
    - 3-Phase permanent magnet synchronous motor
- Power switching supply source
- Power supply cable
- Demo guide
- Handout
- CD with FreeMASTER

It is possible to control the demo using two buttons on the MCU tower board (the Reset button is the third one, which is not considered now), or via the graphical interface in the FreeMASTER software installed (See Figure 2).

The demo concept is shown in Figure 1.

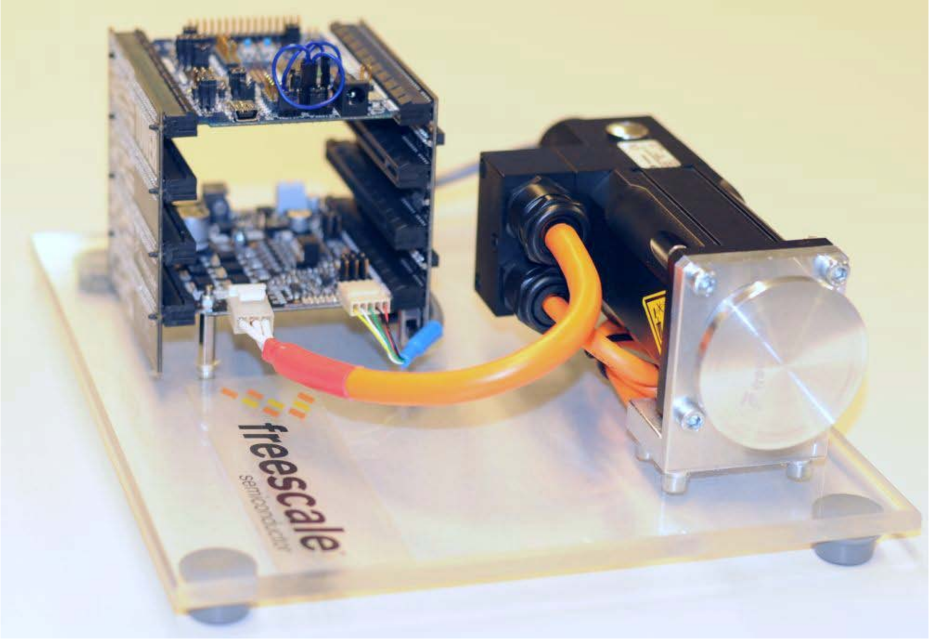
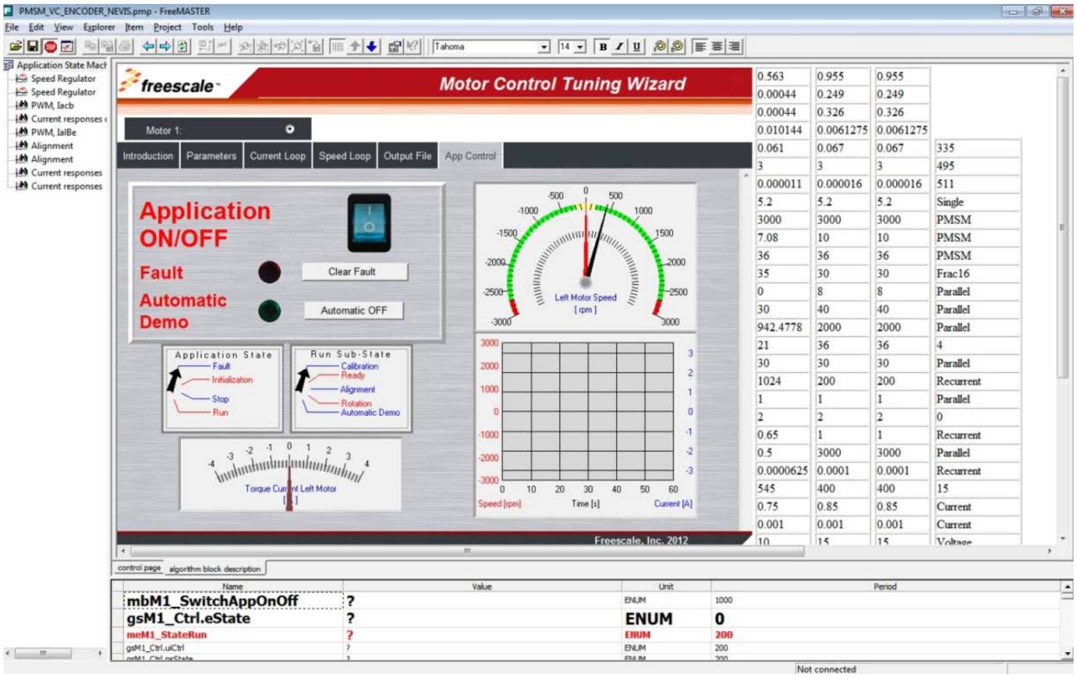


Figure 1. PMSM field-oriented control demo with encoder



0.563	0.955	0.955	
0.00044	0.249	0.249	
0.00044	0.326	0.326	
0.010144	0.0061275	0.0061275	
0.061	0.067	0.067	335
3	3	3	495
0.000011	0.000016	0.000016	511
5.2	5.2	5.2	Single
3000	3000	3000	PMSM
7.08	10	10	PMSM
36	36	36	PMSM
35	30	30	Frac16
0	8	8	Parallel
30	40	40	Parallel
942.4778	2000	2000	Parallel
21	36	36	4
30	30	30	Parallel
1024	200	200	Recurrent
1	1	1	Parallel
2	2	2	0
0.65	1	1	Recurrent
0.5	3000	3000	Parallel
0.0000625	0.0001	0.0001	Recurrent
545	400	400	15
0.75	0.85	0.85	Current
0.001	0.001	0.001	Current
10	15	15	Voltage

Name	Value	Unit	Period
mbM1_SwitchAppOnOff	?	ENUM	1000
gsM1_Ctrl.eState	?	ENUM	0
meM1_StateRun	?	ENUM	200
gsM1_Ctrl.uiCtrl	?	ENUM	200

Figure 2. Graphical interface in FreeMASTER

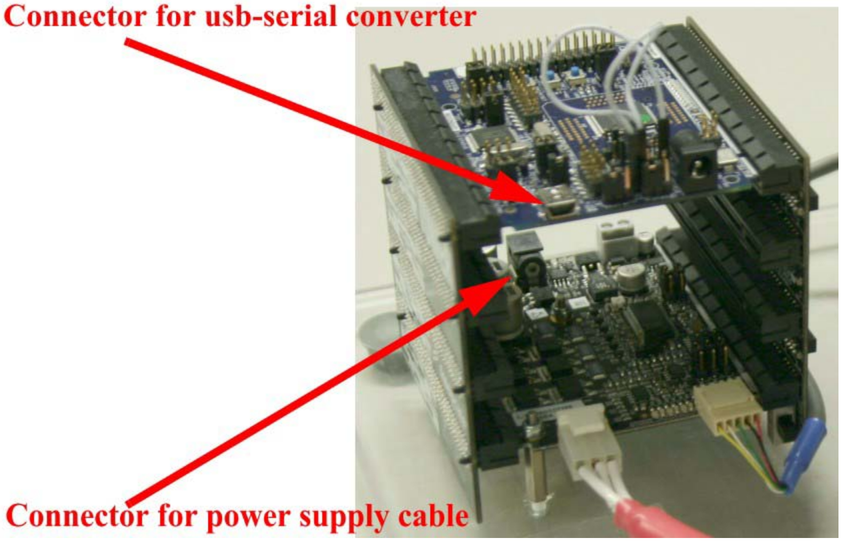
2 Demo initialization

The demo is packaged in an aluminum suitcase. The steps required to successfully initialize the demo are given in the following subsection. For a manual operation, only the first two steps are needed, but for controlling the demo through the FreeMASTER tool, all the steps need to be followed.

2.1 Initialization step by step

1. Open the case and remove all the demo support components from the suitcase (such as power supply cable, Demo Guide, handout, and FreeMASTER CD). All the other elements are mounted on the Plexiglas. This is fixed inside the suitcase by a foam material. The user must be careful while unpacking the demo hardware from the suitcase, because if the demo temperature is low, the cold parts could be bedewed. If this has happened, wait until the demo temperature reaches a normal value, and then plug the power supply cable into a wall socket. Keep all the foam elements, as they will be needed again for packing the demo.

2. Turn on the demo by plugging the mains power supply cable into a wall socket. The low-voltage connector of the switching power supply is plugged in the female connector at the back side of the Low-Voltage 3-Phase Motor Control module (see Figure 3). After the demo is turned on, several LEDs will be lighted and the control algorithm will start the initialization. Now, the demo is ready to start.

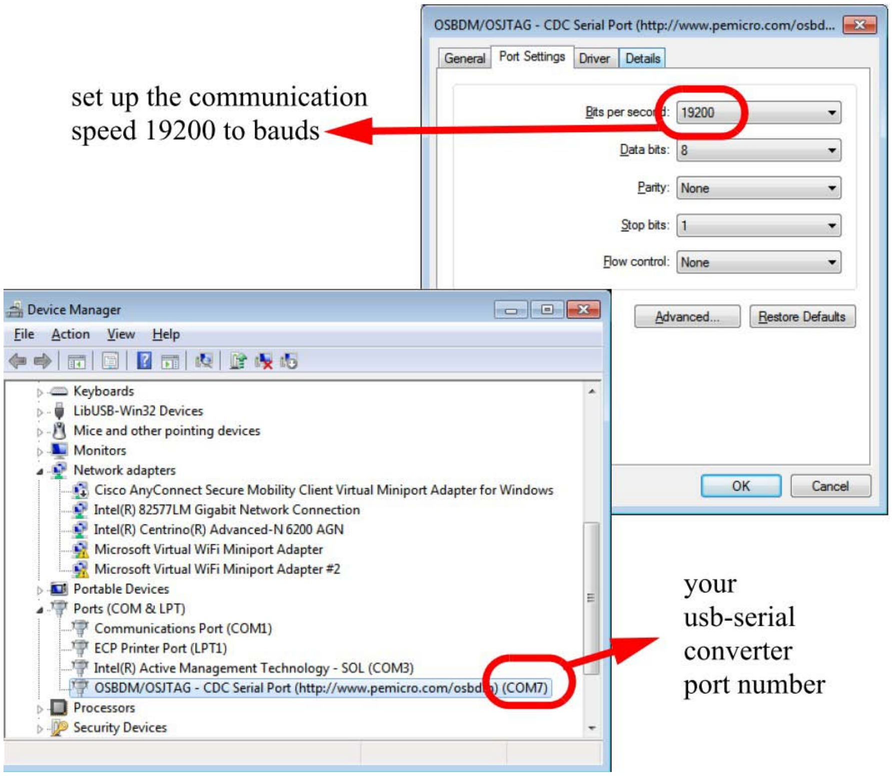


Connector for usb-serial converter

Connector for power supply cable

Figure 3. Connecting power source and USB cable

3. Press the SW2 button to start the calibration process and the alignment. During the alignment, the rotor can turn a maximum of about 60 degrees. When SW2 is pressed for the first time (first step), the control algorithm regulates the speed but a zero speed is required, so the motor will not turn. Each successive press of the same SW2 button will double the required speed of the motor. The button SW1 will decrease the required speed similarly, and eventually, the last press of SW1 (last step) will turn off the demo. For more detail about manually controlling the demo by the buttons, see Manual hardware operation.
4. If it is required to control the demo using the graphical interface in FreeMASTER, continue with the following steps. For control and visualization of quantities, it is necessary to connect the demo to a PC or notebook via a USB cable.
5. Plug in the USB to mini-USB cable into the USB port on the PC or notebook and wait until the converter driver installation finishes.



set up the communication speed 19200 to bauds

Bits per second: 19200

Data bits: 8

Parity: None

Stop bits: 1

Flow control: None

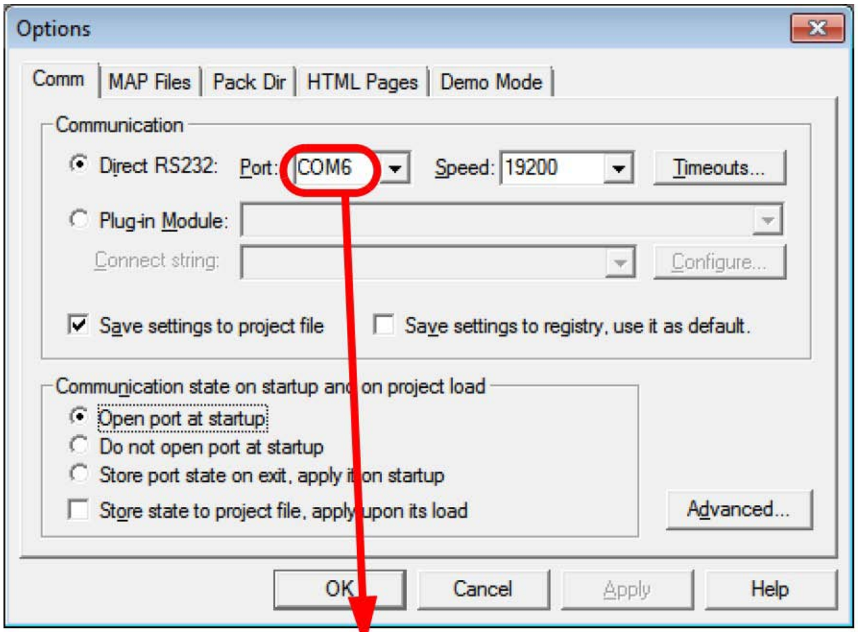
- Keyboards
- LibUSB-Win32 Devices
- Mice and other pointing devices
- Monitors
- Network adapters
  - Cisco AnyConnect Secure Mobility Client Virtual Miniport Adapter for Windows
  - Intel(R) 82577LM Gigabit Network Connection
  - Intel(R) Centrino(R) Advanced-N 6200 AGN
  - Microsoft Virtual WiFi Miniport Adapter
  - Microsoft Virtual WiFi Miniport Adapter #2
- Portable Devices
- Ports (COM & LPT)
  - Communications Port (COM1)
  - ECP Printer Port (LPT1)
  - Intel(R) Active Management Technology - SOL (COM3)
  - OSBDM/OSJTAG - CDC Serial Port (http://www.pemicro.com/osbdm) (COM7)
- Processors
- Security Devices

your usb-serial converter port number

Figure 4. USB-to-serial port setting

6. If the FreeMASTER application has not been installed on your PC or notebook, install it as per the instructions given in FreeMASTER installation.
   1. Open the FreeMASTER graphical interface file from the CD location: *\FreeMASTER\PMSM_VC_ENCODER_NEVIS.pmp*.
   2. If the message “Could not open communication port (Error 0x0004005: Unspecified error)!” appears, click the OK and Continue buttons. Thereafter, continue with the following steps.

7. Check whether the port number of the USB-to-serial converter and the port number set in FreeMASTER are the same.
   1. Go to Control Panel and open the Device Manager.
   2. Find the “Ports (COM & LPT)” group, expand this group, and click OSBDM/OSJTAG -CDC Serial Port (www.pemicro.com/osbdm)(COMX)”, where “X” is the port number selected by the operating system.
   3. A new window opens. Select the Port Setting tab and edit all the values according to Figure 4. Remember your port number.
8. On the FreeMASTER graphical interface, choose Project > Options. A window as shown in Figure 5, appears. All settings must be in accordance with this figure, except for item “Port”, where you must select the same port number as in the USB-to-serial converter.
9. Click OK and restart FreeMASTER.
10. If no error message appears, it is possible to control the demo from FreeMASTER. For more details on the graphical user interface, see FreeMASTER User Interface.



Communication

● Direct RS232: Port: COM6 Speed: 19200

○ 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

select the same the port number as the usb-serial converter port number

Figure 5. FreeMASTER port setting

3 Manual hardware operation

For manual operation, there are two buttons, SW1 and SW2.

- The SW2 button first starts the control algorithm, and then increases the speed.
- The SW1 button decreases the speed, turns off the application, clears a flag, or turns the application to the autonomous demo mode.

3.1 Manual operation principle

The required predefined motor speed values are provided in the following table. The predefined values are almost the double of those in the previous step, except for the zero value in the first step.

Table 1. Predefined required speed

Buttons	**SW1: Next step, SW2: Previous step**								
Step	0	1	2	3	4	5	6	7	8
Motor speed (rpm)	Stop state	0	47	94	187	375	750	1500	3000

3.2 Operation example

The manual operation can begin after the power is supplied to the demo application. The demo can be turned on by pressing button SW2, after which a zero motor speed is required. Each successive press of the same button will increase the required speed by two times. The required speed can be decreased by pressing SW1 until a zero speed is achieved, and the application will be stopped.

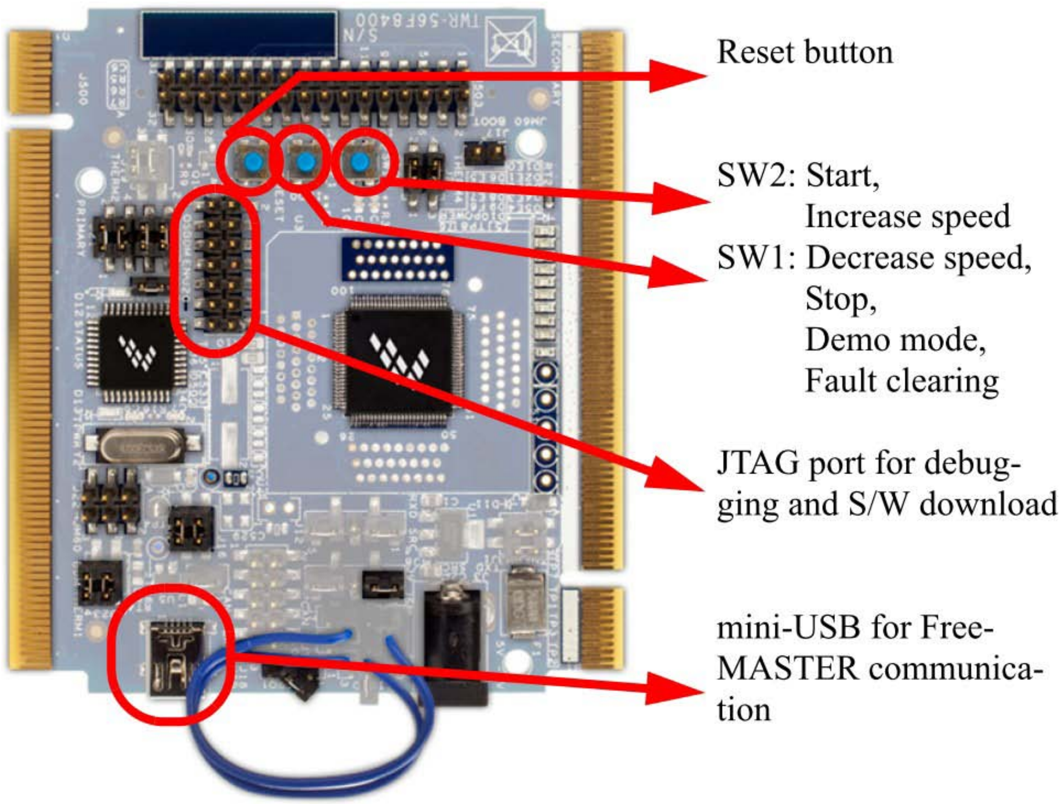


Figure 6. Buttons on the MC56F94789 tower board

3.3 Special button functions

This demo application is based on the principle of using more functions on a single button. The reason for this approach is that there are just a few buttons on the MCU tower board, and it is not possible to change the function of the Reset button.

If a fault occurs, it is possible to clear this machine state using the SW1 button. After this button is pressed, the fault is cleared and if there is no fault measured, the application will jump to the Stop state.

In the Stop state, there are two possibilities.

- The first is to run the motor again by pressing the SW2 button.
- The second possibility is that the autonomous demo mode can be turned on by pressing SW1 during the Stop state. Each even press of the button SW1 will turn off the autonomous demo mode, and each odd press of the same button will turn on the autonomous demo mode.

4 FreeMASTER operation

FreeMASTER is a real-time debug monitor and data visualization tool. The graphical interface in FreeMASTER is a more comfortable way to control the demo application. In this way, the user obtains more possibilities and feedback over controlling the motor. Moreover, the actual speed values are measured and can be visualized in waveforms.

4.1 FreeMASTER installation

The FreeMASTER installation file is found on the FreeMASTER CD:

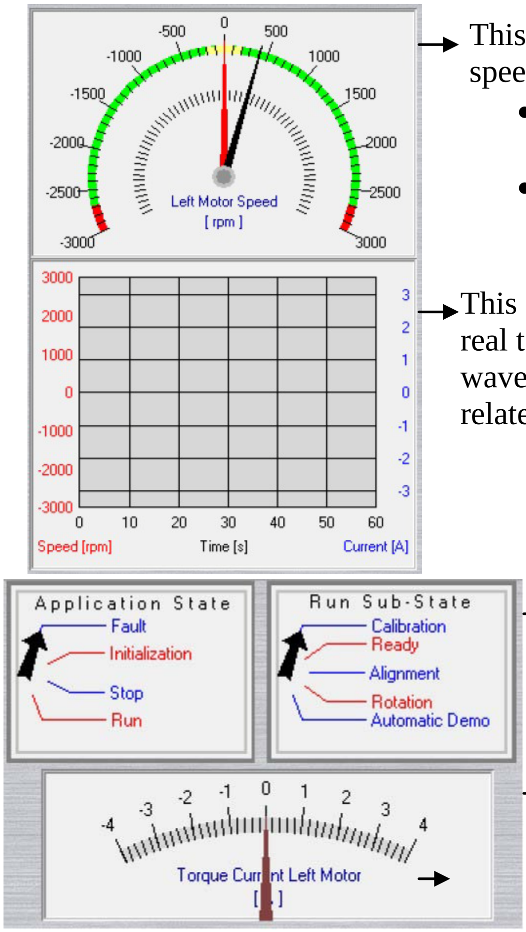
\Install\FreeMASTER\fmaster13-16.exe, or the most recent version can be found on **FreeMASTER.**

After clicking the installation file, a common user guide appears and takes you through the whole installation process.

4.2 FreeMASTER user interface

For the demonstration purpose, the graphical user interface was created for controlling the demo. This interface is based on an html page and includes ActiveX components such as gauges, knobs, and graphs. Most of these elements are interactive, which means that it is possible to control them by clicking or dragging the mouse. The principle of such control is that the ActiveX component calls the applicative function which changes the FreeMASTER variable, connected to the memory address of the corresponding control variable of the MCU through serial communication. The user interface in FreeMASTER is described in Figure 7 and Figure 9.

The next very useful function of FreeMASTER is the real-time monitoring. This feature is based on individual sampling to determine variables and to chart up to eight variables in the oscilloscope or recorder view. This application uses a high-speed recorder for visualization of the target quantities. There are many predefined recorder settings for various measurements on the left-hand side of FreeMASTER: the transient or steady state of an arbitrary motor, speed and sector measurement, current measurement in various systems of coordinates, and so on. An example of such a recorder measurement can be seen in Figure 8. The lower part of the FreeMASTER window contains the observed variables. Some of these variables can be changed directly in FreeMASTER, while some can be changed using ActiveX components.



This speed gauge assigns the required motor speed. There are two needles:

- Black needle can be controlled by dragging or clicking mouse.
- Red needle indicates the actual mechanical rotor speed only.

This graph shows the actual mechanical speed and real torque current (Q current component). The red waveform corresponds to speed, and the blue one relates to torque current.

This knob indicates the application state and the run substate, both of which can be changed.

This current gauge corresponds to the current component which influences the machine torque. This component has a limited current value of 3A.

Figure 7. Active-X components

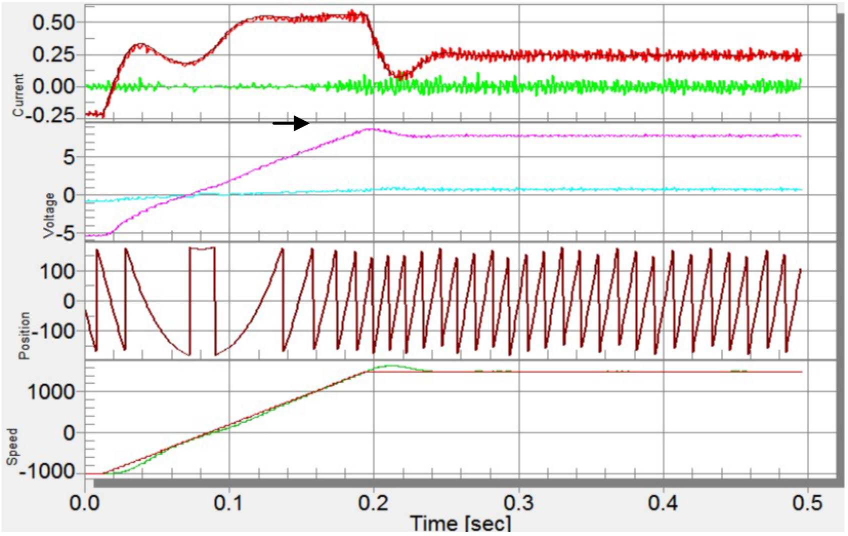


Figure 8. Example of using the recorder

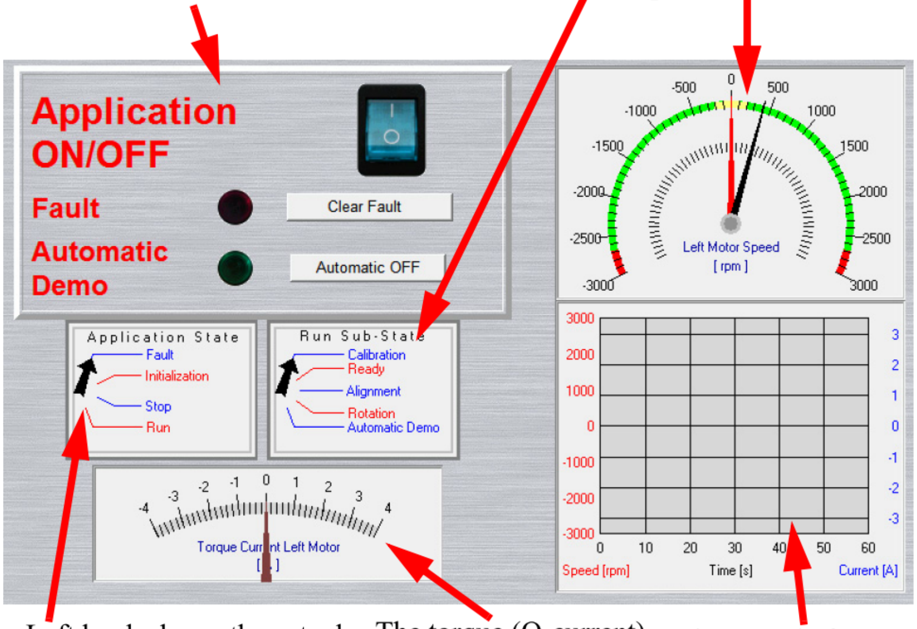
4.3 FreeMASTER operation

This figure shows a simple layout of FreeMASTER graphical user interface.

Buttons and indicators for application control. The largest buttons turns on/off the application. Below are the fault and autonomous demo mode indicators. Middle buttons on the right side clear the fault, and the bottom button enables or disables the autonomous demo mode.

Right knob shows the run sub/states.

Required and measured speeds



Left knob shows the actual application state

The torque (Q-current) current gauge

The record of motor speed and torque current

Figure 9. FreeMASTER graphical user interface

A different layout of the ActiveX components suitable for a wide screen is shown in this figure.

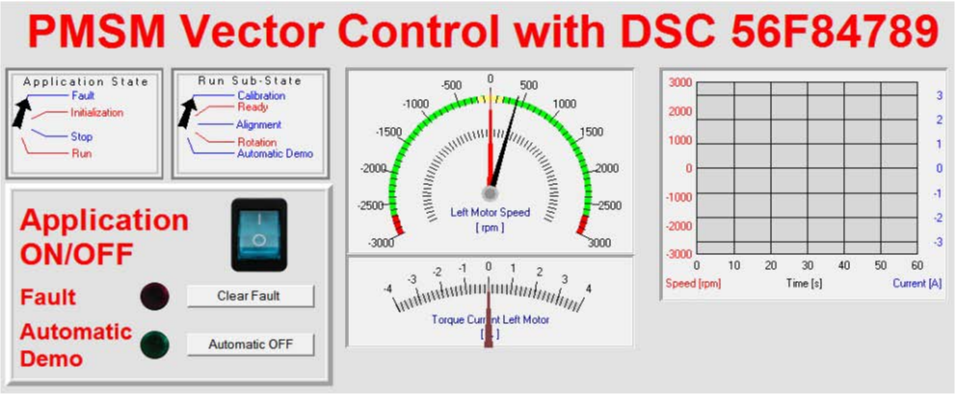
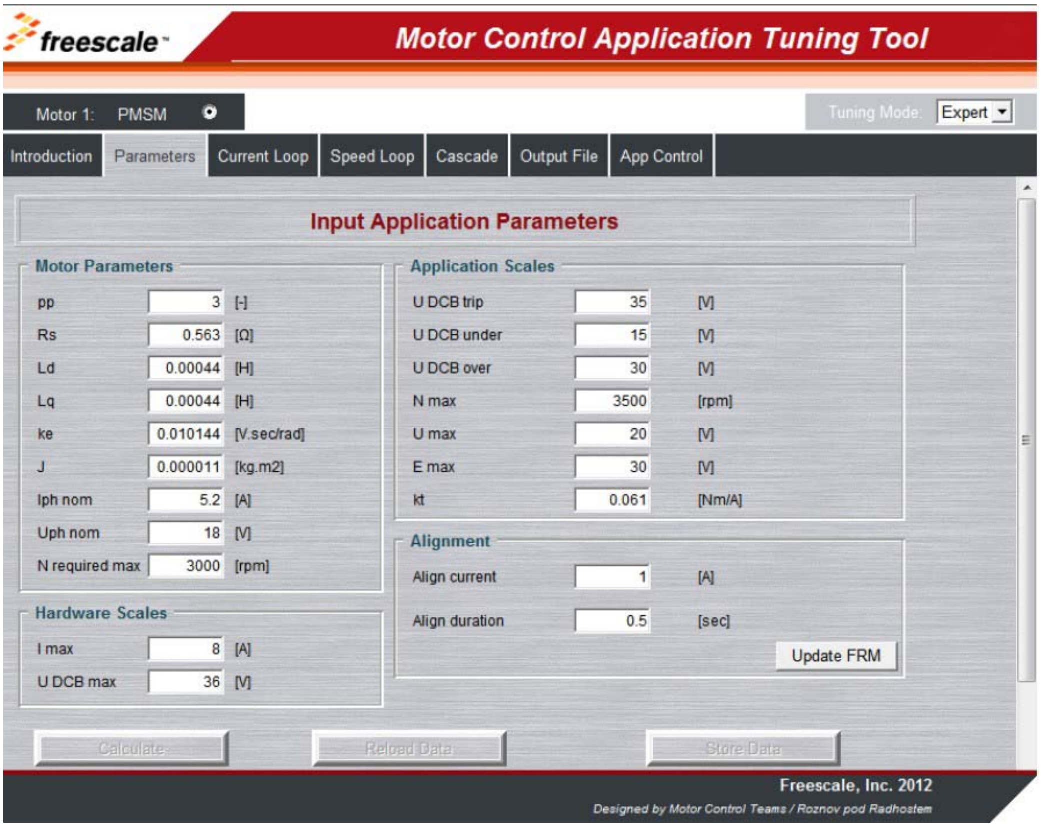


Figure 10. FreeMASTER graphical user interface for wide screen

5 Motor Control Application Tuning (MCAT) tool

The Motor Control Application Tuning (MCAT) tool is an upper-level motor control tool which enables easy parameter configuration of the control loops, motors, or observers. Integrating the MCAT tool with FreeMASTER can set up and tune the application constants without any motor control knowledge. After the drive has been tuned, the user can save all constants to the application header file.

5.1 Input Application Parameters



Motor 1: PMSM

Tuning Mode: Expert

Motor Parameters

pp	3	[-]
Rs	0.563	[Ω]
Ld	0.00044	[H]
Lq	0.00044	[H]
ke	0.010144	[V.sec/rad]
J	0.000011	[kg.m2]
Iph nom	5.2	[A]
Uph nom	18	[V]
N required max	3000	[rpm]

Hardware Scales

I max	8	[A]
U DCB max	36	[V]

Application Scales

U DCB trip	35	[V]
U DCB under	15	[V]
U DCB over	30	[V]
N max	3500	[rpm]
U max	20	[V]
E max	30	[V]
kt	0.061	[Nm/A]

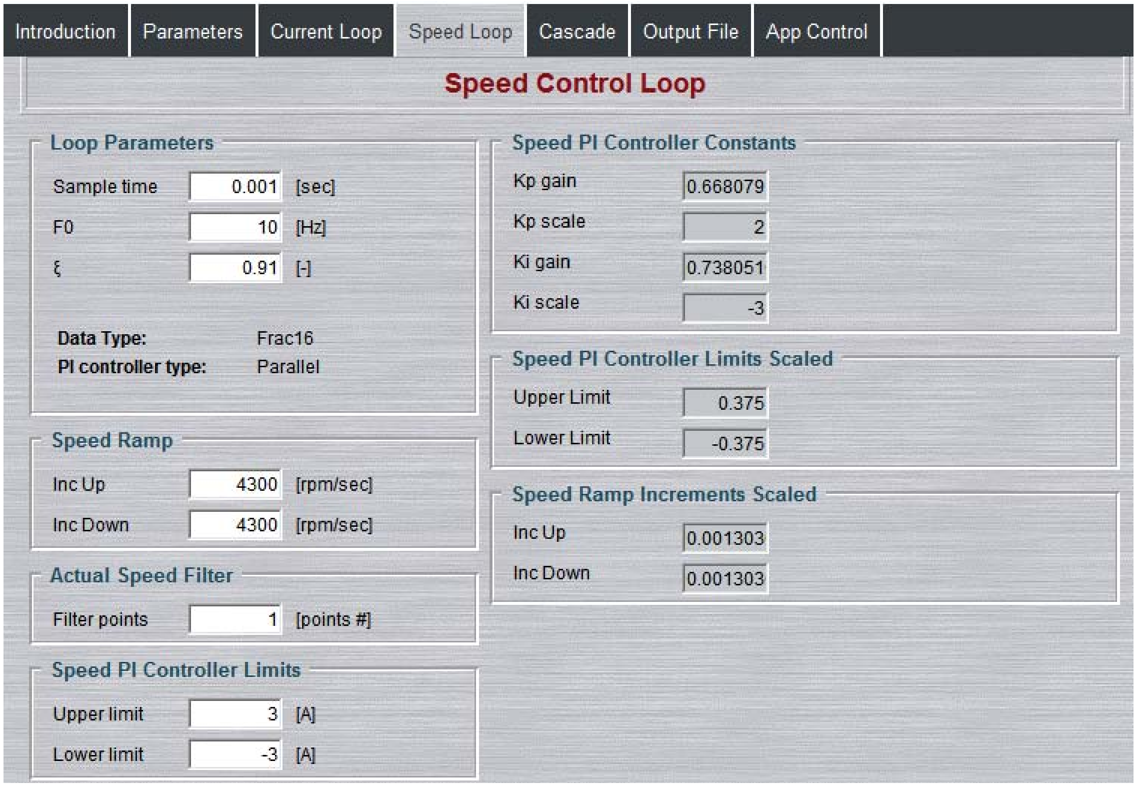
Alignment

Align current	1	[A]
Align duration	0.5	[sec]

Figure 11. Input application parameters

Input parameters can be divided into three groups: Motor Parameters, Application Scales, and Hardware scales. The motor parameters can be found in the motor datasheet. The application and hardware scales depend on the power stage used. The buttons Reload Data and Store Data are used to load and store written data respectively.

- The two tabs, Speed Loop and Control Loop set up the parameters of the current and speed loops. The user should write the parameters on the left side. The tool will calculate the current controller parameters according to the loop parameters after pressing the Calculate button. Then it is possible to write these values directly into the application variables by clicking the Update FRM button. Figure 12 and Figure 13 show an example of a tool setting.



Speed Control Loop

Loop Parameters

Sample time	0.001	[sec]
F0	10	[Hz]
ξ	0.91	[-]

Data Type: Frac16
PI controller type: Parallel

Speed Ramp

Inc Up	4300	[rpm/sec]
Inc Down	4300	[rpm/sec]

Actual Speed Filter

Filter points	1	[points #]

Speed PI Controller Limits

Upper limit	3	[A]
Lower limit	-3	[A]

Speed PI Controller Constants

Kp gain	0.668079
Kp scale	2
Ki gain	0.738051
Ki scale	-3

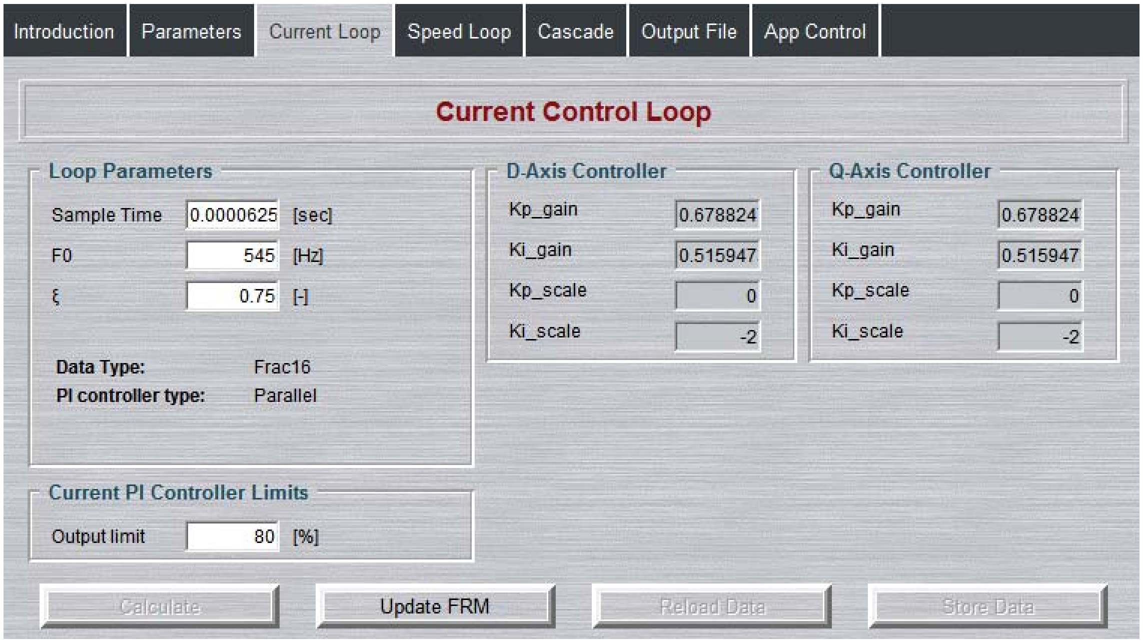
Speed PI Controller Limits Scaled

Upper Limit	0.375
Lower Limit	-0.375

Speed Ramp Increments Scaled

Inc Up	0.001303
Inc Down	0.001303

Figure 12. Speed loop parameters



Current Control Loop

Loop Parameters

Sample Time	0.0000625	[sec]
F0	545	[Hz]
ξ	0.75	[-]

Data Type: Frac16
PI controller type: Parallel

Current PI Controller Limits

Output limit	80	[%]

D-Axis Controller

Kp_gain	0.678824
Ki_gain	0.515947
Kp_scale	0
Ki_scale	-2

Q-Axis Controller

Kp_gain	0.678824
Ki_gain	0.515947
Kp_scale	0
Ki_scale	-2

Figure 13. Current loop parameters

- The Cascade tab helps user to tune the application (see Figure 14). The user can select one from following tuning methods:
  - Scalar Control—no closed loop, no feedback information

- o Voltage FOC Control—no closed loop, rotor position is measured.
- o Current FOC Control—current closed loop, rotor position is measured.
- o Speed FOC Control—current and speed closed loops, rotor position and speed are measured.

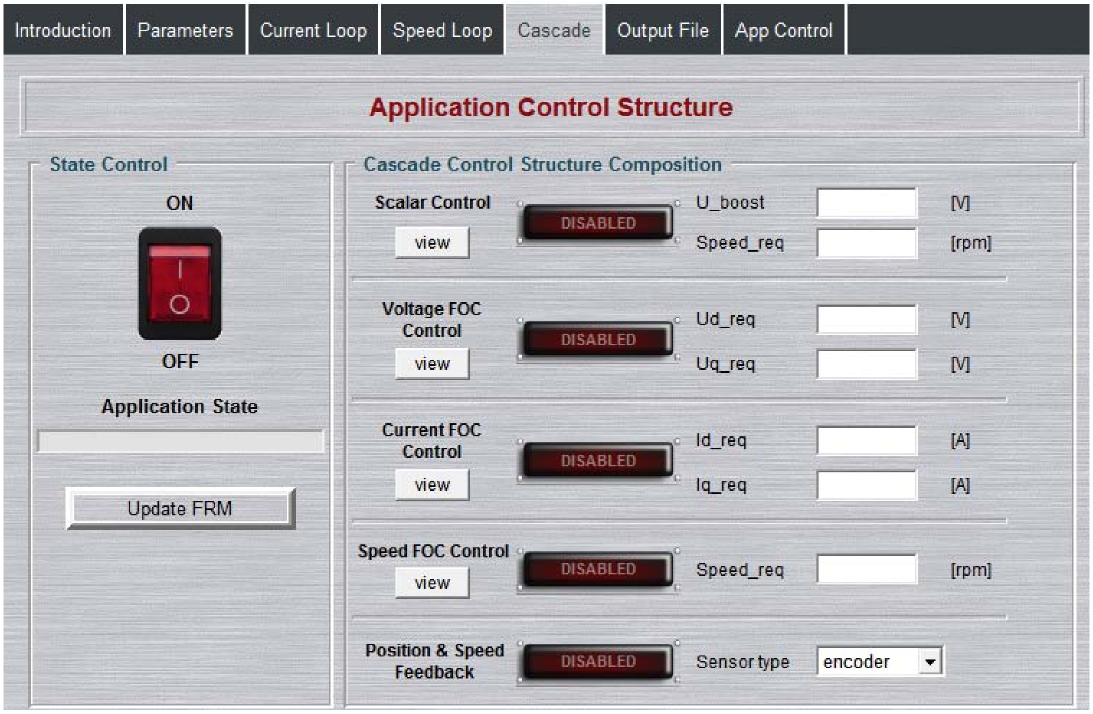


Figure 14. Cascade tab

- The Output File tab displays the output header files which are used for writing the application constants directly to the MCU flash memory after compiling and downloading the code.
- The App control tab can be used for controlling the application via the graphical environment.

6 References

MC56F8458X Advance Information Data Sheet, available at **freescale.com**

7 Revision history

Revision number	Date	Substantive changes
0	06/2013	Initial release

How to Reach Us

Home Page:
freescale.com

Web Support:
freescale.com/support

Document Number: PMSMUG
Rev. 0, 06/2013
28 June, 2013