NXP Semiconductors
User's Guide

Document Number: ESCSU16UG
Rev. 0, 03/2017

ESC–MC9S08SU16 User’s Guide

1. Introduction

The NXP ESC-MC9S08SU16 reference board is a set of hardware tools for a drone electronic speed control (ESC). It is an ideal tool for the rapid prototyping of microcontroller-based applications.

The ESC-MC9S08SU16 supports a power supply voltage range from 4.5 V to 18 V. It features a MC9S08SU16, a device boasting a maximum operating frequency of 40 MHz, up to 16 KB Flash, 768 Bytes Ram size, and numerous analog and digital peripherals. The GDU is the highlight of this MCU. It integrates the pre-driver and decreases the bill of material (BOM) cost.

The ESC-MC9S08SU16 includes a debug adapter known as background debug mode (BDM). This circuit offers serial communications, flash programming, and run-control debugging.

The software development tool for ESC–MC9S08SU16 is Codewarrior10.7. It supports the user in programming and debugging.

This application demonstrates low-power 3-phase brushless DC (BLDC) motor drive software. It is focused on a simple and easy to understand control approach to BLDC in a time-critical application. This includes a sensorless BLDC drive which uses back electromotive force(BEMF) sensing for position recognition. It serves as an example of a BLDC motor control system for drone ESC applications.

Contents

2. Application platform

The following figure shows the entire application platform.

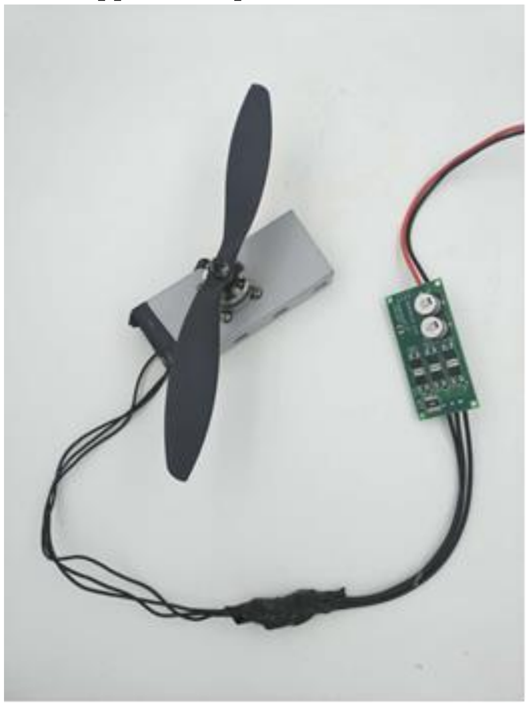


Figure 1. Application platform

3. Warnings

CAUTION

The reference board features power components that can reach temperatures hot enough to cause burns. To facilitate safe operation, the input power range depends on the motor's rated voltage, which should come from a DC power supply whose limited current is not greater than 15 A.

Keep the following points in mind while following the instructions in this guide:

- Before moving scope probes, making connections, and so on, power off the reference board.
- It is advisable to use a heavy shield to cover the motor with blade during operation in power electronics lab.
- Do not connect a USB cable to the demo while power is applied to the reference board. Connecting a USB cable to the reference board will cause damage to the MC9S08SU16 MCU and other systems.
- Check and confirm that the blade is firmly fixed and screwed down to the base of the motor. If it is not, it may cause damage to users or to laboratory equipment.

3.1. Set-up guide

The hardware board is shown in the following two figures:

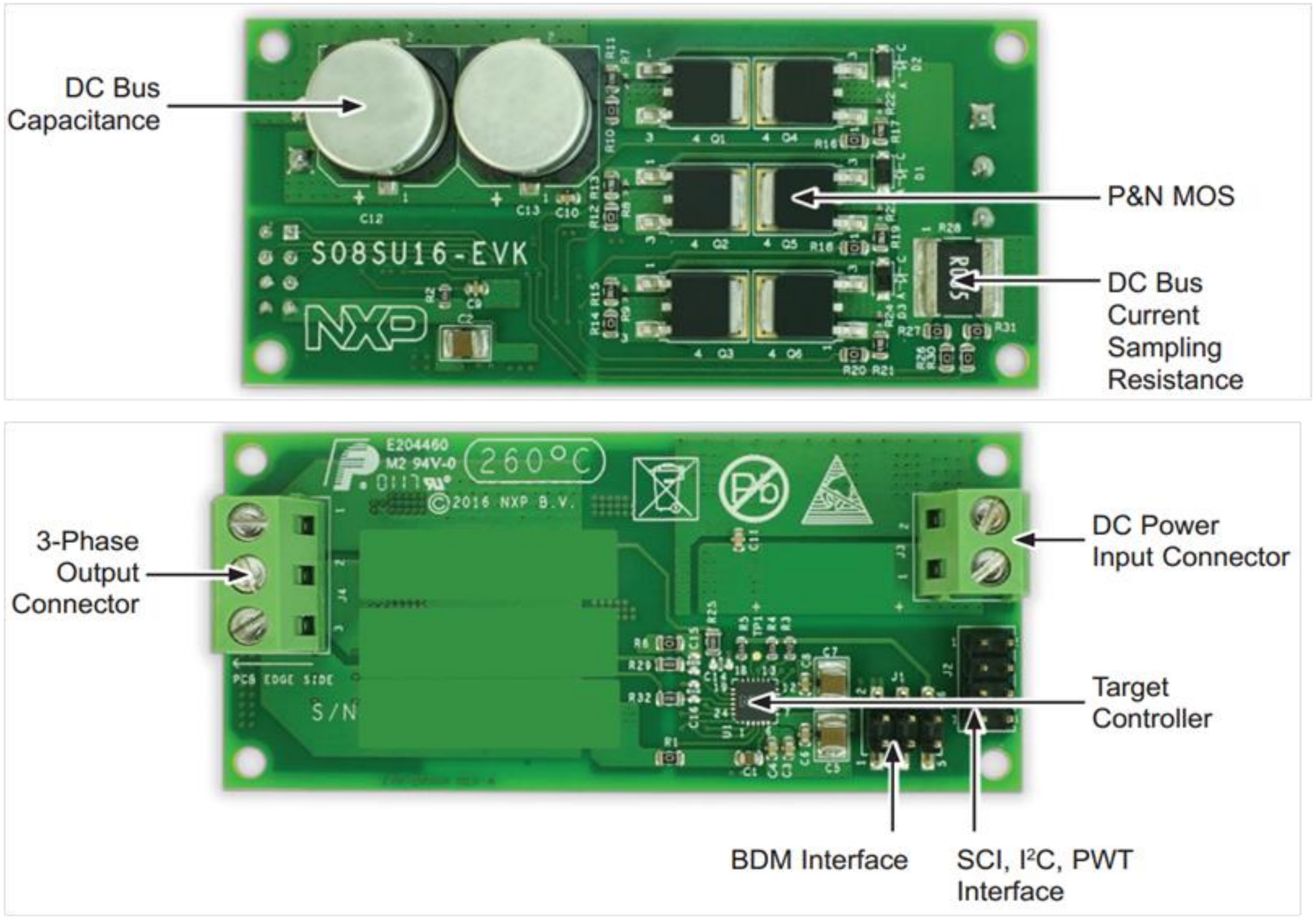


Figure 2. Hardware board

- Connect the motor three connector to the reference board.
- To debug the demo via FreeMASTER, connect a USB Multilink universal between the hardware board and the USB port of your computer.
- Connect a ribbon cable between the target debug header (J43, BDM interface) and PORT C on the multilink. Confirm the pin number of the target debug header. Incorrect connection may damage both the target MCU as well as the multilink.
- Install FreeMASTER2.0.
- Install Codewarrior10.7 development studio.

NOTE

Codewarrior10.7 installation is advisable. If you have the source code of this project, you can debug it on Codewarrior10.7. If you do not have the source code and only have the .abs file, you can directly program it to the target.

- Open the project using Codewarrior10.7.
- Open the FreeMASTER project in ...\FMSTR\SU16-E_Drone.pmp.
- Connect the 14 V supply voltage to reference board ().
- Setup the communication tab in FreeMASTER project options, as shown in the following three figures.

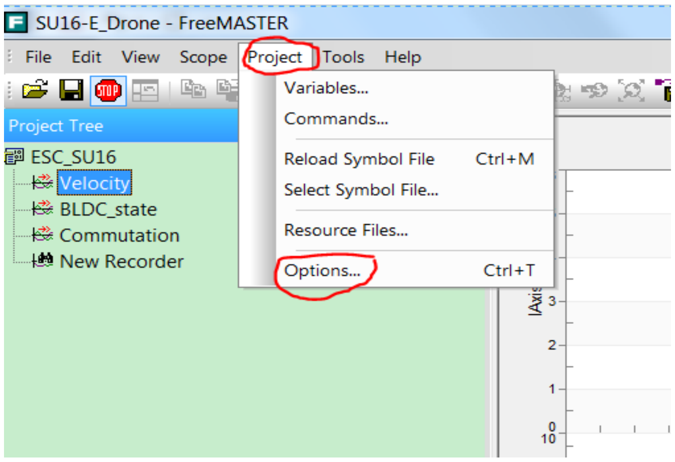
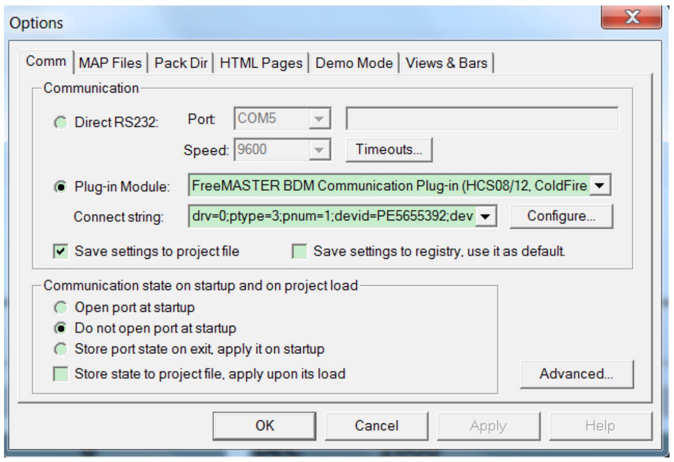


Figure 3. FreeMASTER options


Options

Comm | MAP Files | Pack Dir | HTML Pages | Demo Mode | Views & Bars

Communication

- ○ Direct RS232: Port: COM5 Speed: 9600 Timeouts...
- [✓ ● Plug-in Module:] FreeMASTER BDM Communication Plug-in (HCS08/12, ColdFire
- Connect string: drv=0;ptype=3;pnum=1;devid=PE5655392;dev Configure...
- [✓ Save settings to project file] ☐ Save settings to registry, use it as default

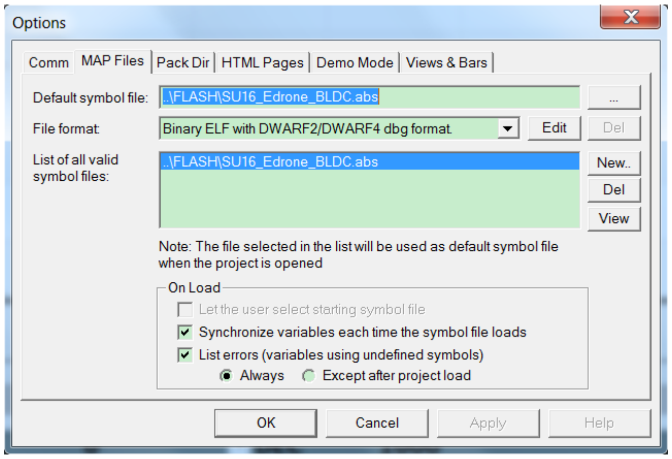
Communication state on startup and on project load

- ○ 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

Advanced...

OK Cancel Apply Help

Figure 4. FreeMASTER Comm configuration


Options

Comm | MAP Files | Pack Dir | HTML Pages | Demo Mode | Views & Bars

Default symbol file: ..\FLASH\SU16_Edrone_BLDC.abs

File format: Binary ELF with DWARF2/DWARF4 dbg format

List of all valid symbol files: ..\FLASH\SU16_Edrone_BLDC.abs

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
- [x] Synchronize variables each time the symbol file loads
- [x] List errors (variables using undefined symbols)
  - ● Always ○ Except after project load

Figure 5. FreeMASTER MAP Files configuration

- Start communication in FreeMASTER, as shown in the red icon in the following figure:

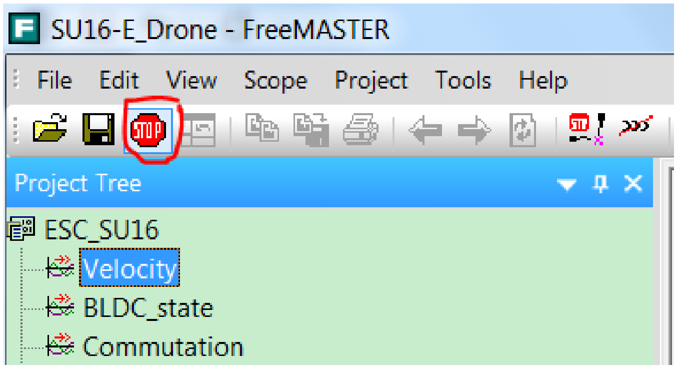


Figure 6. FreeMASTER communication start

- There are two methods to control this application. One is using FreeMASTER to set the duty command, the other is using the knob switch, which generates the pulse width signal to set the duty command. To use FreeMASTER to control the target, the macro “PPM_SPEED_REGULATION”, which is in the app_init.h file, should be removed. Otherwise it should be reserved.

NOTE

The item cannot be modified through FreeMASTER. If the user wants to change the control mode, it must be modified in the source code.

- When using FreeMASTER to set the duty command, the variable “uw16DutyCycleReq”, which is located on the variable watch tab of FreeMASTER, can be modified in the range of 300 to 2400.
- When using the knob switch to generate the pulse width to control the application, JP20 (pin 2, PWT0IN0) and JP19 (pin 1, GND) are the interface for the external switch signal.
- When using the external switch to drive the application and cable connection is complete, the throttle calibration function is integrated in this application. For further details, refer to Section 6 “Throttle calibration”.
- If the motor does not run, check the status of the application in FreeMASTER. If the overcurrent

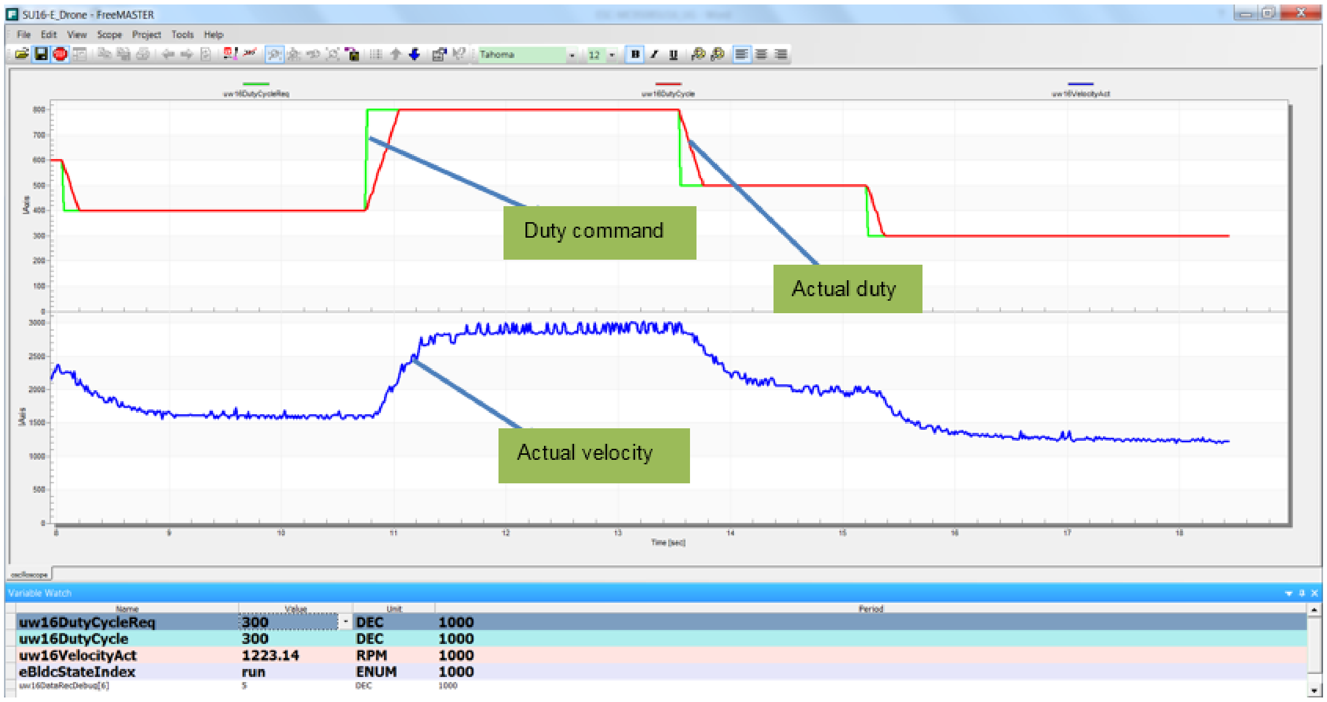
fault has occurred, check the overcurrent limit. If the overvoltage fault has occurred, check the supply voltage. If overcurrent limit and supply voltage is normal, check the hardware component on the board.

4. FreeMASTER software installation

FreeMASTER software can be installed via the NXP FreeMASTER web page, under the "downloads" section.

5. FreeMASTER control

After launching the application and performing all settings described above, click the scope: Velocity item in the project tree window, as shown in the following figure.



Name	Value	Unit	Period
uw16DutyCycleReq	300	DEC	1000
uw16DutyCycle	300	DEC	1000
uw16VelocityAct	1223.14	RPM	1000
eBldcStateIndex	run	ENUM	1000
uw16DataRecDebug[6]	5	DEC	1000

Figure 7. FreeMASTER user interface

Description of variables:

- **uw16DutyCycleReq:** this variable serves for entering the required duty command of the motor. You can modify this variable from 300 to 2400. Any other settings will be ignored.
- **uw16DutyCycle:** this variable is the actual duty cycle.
- **uw16VelocityAct:** this variable is the actual motor speed.
- **eBldcStateIndex:** this variable is the enumeration-type variable, and manifest the current state machine.

6. Throttle calibration

If the user wants to use an external switch to generate the pulse width signal to set the duty command, the period of external signal should not be greater than 0.4 s to avoid PWT0 count overflow. The calibration process is as follows:

1. Ensure that the knob switch is set to the maximum throttle range before the reference board is powered on. After the reference board is powered on, a sound will be emitted after a certain time. The period between power on and the sound being emitted is flexible. If the frequency of the external signal is 50 Hz, the period before the sound is about 4 s. The user can modify the variable “uw16SoundWaitTime” in the source code to select appropriate time before sound according to the external signal frequency, the default value of this variable is 200. The user also can modify the variable “uw16SoundTime” to select the appropriate sound period. The default value is 50. When the sound finishes, the high throttle calibration is complete.
2. Next, move the knob switch to the minimum throttle stage. The duty of the minimum throttle stage should be less than the maximum duty which has already been calibrated as above. A sound will then be emitted by the motor after a certain time just like the first step.
3. When the sound stops, the low throttle calibration is complete and the throttle value is stored in EEPROM.

When the throttle calibration is complete, the user can regulate speed freely using the external switch.

7. Revision History

Table 1. Revision history

Revision number	Date	Substantive changes
0	03/2017	Initial release

Document Number: ESCSU16UG
Rev. 0
03/2017