Motor Control with Freescale Microcontrollers

Eduardo Viramontes – Applications Engineer
1. Motor Control & Power Market Trends
2. Target Applications
3. Freescale Motor Control Solutions
4. Motor Control Basics
5. Motor control portfolio
6. BLDC Control Basics
7. Commutation
8. Back EMF Basics
9. Hardware Overview
10. BLDC – Recommended Application
11. Why BLDC with MP16?
12. Controlling a BLDC motor with sensors
13. BLDC with sensors control code on the S08MP16
14. Mini hands-on: Find out how the sensors behave (Lab 0)
15. Electronic motor commutation
16. Hands-on: Run the BLDC with sensors demo (Lab1)
17. Practical sensorless motor control: How is BEMF used?
18. Measure BEMF
19. Open loop startup on a sensorless application
20. Hands-on: Run the BLDC without sensors demo (Lab 2)
21. Summary
Objectives

By the end of this session, you should be able to

• Understand the principles of Motor Control
• Know the newest Control solutions provided by Freescale including
  ▪ MP16
  ▪ DSC portfolio
  ▪ PPC
  ▪ Kinetis
• Know how to start writing motor control applications either from scratch or from Freescale reference designs.
## Motor Control and Power Conversion Market Trends

<table>
<thead>
<tr>
<th>Motor Control</th>
<th>Power Conversion</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Reduce power consumption</strong></td>
<td><strong>Increase Conversion Efficiency</strong></td>
</tr>
<tr>
<td>Intelligent motor control improves efficiency by 30 percent or more</td>
<td>Cost-effective soft switching techniques</td>
</tr>
<tr>
<td><strong>Reduce system and development cost</strong></td>
<td><strong>High Power Density</strong></td>
</tr>
<tr>
<td>More on-chip peripherals to reduce component count</td>
<td>Compact size: high watt per cubic inch</td>
</tr>
<tr>
<td><strong>Reuse software, hardware and tools across platforms</strong></td>
<td><strong>High Intelligence Control</strong></td>
</tr>
<tr>
<td>Ease software migration across wide performance range</td>
<td>Digital Controlled Power conversion</td>
</tr>
<tr>
<td><strong>Cost-effective safety, reliability and security</strong></td>
<td><strong>Low Cost</strong></td>
</tr>
<tr>
<td>On-chip safety and security protection</td>
<td>System monitoring and protection with less components usage</td>
</tr>
</tbody>
</table>

---

*Smart Appliance*

*Renewable Energies*

*Digital Power*
## Motor Control – Freescale Alignment with Trends

<table>
<thead>
<tr>
<th>Market Trend</th>
<th>Freescale Alignment with Customer Requirements</th>
</tr>
</thead>
<tbody>
<tr>
<td>Reduce power consumption</td>
<td>• Vector and sensorless control technology designed into every motor control processor</td>
</tr>
</tbody>
</table>
| Reduce system and development cost| • 8- and 16-bit MCUs start at <$0.70  
• 16-bit DSCs for ACIM and PMSM solutions  
• 32-bit Power Architecture® MCUs for standard/premium drives                                               |
| Reuse software, hardware and tools across platforms | • Rich tools, training, reference designs and libraries  
• Devices are ruggedized with long life and reliability  
• Industrial products ship 10+ years, with high quality and expert customer support                             |
| Cost-effective safety, reliability and security | • Secure SRAM, on-chip data fusing to protect against IP cloning  
• Hardware encryption to protect against network data hacking  
• Watchdog and ECC protection against soft errors  
• Certified IEC software modules                                                                          |
Motor Control Target Applications

- **Pumps and fans**
  - pool pumps, factory systems

- **HVAC**
  - heating fans, air-conditioners

- **Industrial drives**
  - Manufacturing assembly, robotics, wind turbines, printing presses

- **Appliances**
  - washers, dryers, power tools

- **Medical**
  - scanners, pumps, diagnostic and therapy

- **Automotive Motor Control Trends**
  - Fuel/Water pumps, HVAC Fan Control, Window lift
  - Increasing adoption of electric and hybrid vehicles continues to drive BLDC demand
Industrial Motor Control Solutions

Freescale Motor Control Processors
Typical Motor Control MCU Peripheral Functions

► Timer:
  • PWM signals < 20Khz
  • Dead time insertion
  • Commutation (mask-out)
  • ADC triggering
  • Fault control

► ADC
  • Measure current

► Delay block
  • Set ADC measurement at specific times

► Position decoder
  • Quadrature decoder inputs if not sensorless
Many Different Motor Types …

- DC Motor
- Brushless DC Motor (BLDC)
- Stepper Motor (full step)
- Stepper Motor (half step)
- AC Induction Motor (ACIM)
- Permanent Magnet Synchronous Motor (PMSM)
- Switched Reluctance Motor
Understanding the Motor Basics

DC Motor Torque Establishment

DC Motor Principle

- The stator of a Permanent Magnet DC Motor is composed of two or more permanent magnet pole pieces

- The rotor is composed of windings which are connected to a mechanical commutator. In this case the rotor has three pole pairs

Brush DC motor control is simple:

Apply voltage → Commutation occurs mechanically
Simple Model of a DC Motor

DC Motor Equivalent Circuit

Applied Voltage

Back EMF

$E = K_e \Phi \omega$

Resistor

Inductor

$R$

$L$

$i_d$

Dynamic Motor speed:

$$\omega = \frac{V_d - I_d \times R - L \frac{dI_d}{dt}}{K_e \times \Phi}$$

- Speed is increased by increasing the voltage
- Torque is controlled by controlling the current
- Direction is determined by the direction of the current
PWM Control of DC Motor

► Same motor can have different control configurations
  (i.e. Simple switch vs. H-Bridge)

► Same control configuration can have different operating modes
  (i.e. Bipolar PWM vs. Unipolar PWM, independent vs. complementary)

► Different modes have advantages and disadvantages
Simple Speed Control On A Brush DC Motor

1. Measure speed of the motor
1. Measure speed of the motor

2. Compare the measured speed with the desired speed and generate an error signal
1. Measure speed of the motor

2. Compare the measured speed with the desired speed and generate an error signal

3. Amplify the error signal to generate a correction voltage
Simple Speed Control On A Brush DC Motor

1. Measure speed of the motor

2. Compare the measured speed with the desired speed and generate an error signal

3. Amplify the error signal to generate a correction voltage

4. Modulate the correction voltage onto the motor terminals

BUT…. THIS DOES NOT LIMIT CURRENT!!
Current Control On A Brush DC Motor

1. Measure the current of the motor

NOTE.... THIS IS ALSO A TORQUE CONTROLLER!!
1. Measure the current of the motor

2. Compare the measured current with the desired current and generate an error signal

NOTE.... THIS IS ALSO A TORQUE CONTROLLER!!
Current Control On A Brush DC Motor

1. Measure the current of the motor

2. Compare the measured current with the desired current and generate an error signal

3. Amplify the error signal to generate a correction voltage

NOTE.... THIS IS ALSO A TORQUE CONTROLLER!!
1. Measure the current of the motor

2. Compare the measured current with the desired current and generate an error signal

3. Amplify the error signal to generate a correction voltage

4. Modulate the correction voltage onto the motor terminals

NOTE…. THIS IS ALSO A TORQUE CONTROLLER!!

Compare the measured current with the desired current and generate an error signal.

Measure the current of the motor.

Amplify the error signal to generate a correction voltage.

Modulate the correction voltage onto the motor terminals.
Controlling Speed and Current

1. Measure speed from the motor shaft
1. Measure speed from the motor shaft

2. Compare the measured speed with the desired speed and generate an error signal
1. Measure speed from the motor shaft

2. Compare the measured speed with the desired speed and generate an error signal

3. Amplify the error signal to generate a correction to desired current
1. Measure speed from the motor shaft

2. Compare the measured speed with the desired speed and generate an error signal

3. Amplify the error signal to generate a correction to desired current

4. Input desired current into the torque controller
1. Measure speed from the motor shaft
2. Compare the measured speed with the desired speed and generate an error signal
3. Amplify the error signal to generate a correction to desired current
4. Input desired current into the torque controller
5. Modulate the correction voltage from the torque controller onto the motor terminals
Simple DC Motor Control from the MCU’s Perspective

![Diagram of DC Motor Control System]

- **Torque Controller**
- **PID Controller**
- **PWM**
- **ADC**
- **Event Timer**
- **Speed Feedback**
- **Current Feedback**

Components:
- **Motor**
- **Armature**
- **Encoder**
- **User Interface** (e.g., ADC or SCI)

Feedback Loop:
- Speed Command → PID → Torque Controller → PWM → Armature
- Current Command → PID → Torque Controller

Integration with User Interface:
- Communication through ADC or SCI

Additional Notes:
- Speed Feedback
- Current Feedback

Trademarks:
- Motorola, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc.
- BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MEX, Platform-in-a-Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink, and VortiQa are trademarks of Freescale Semiconductor, Inc.
- © 2010 Freescale Semiconductor, Inc.
Motor Control Roadmap

Hi-end – DSP, 32-bit

Low-end to 8-bit, Low-end 32-bit

3-phase sensored (V/Hz, Slip)

Mid Range – 3-phase sensorless (vector control) 16-bit DSP

Low-end to 8-bit, Low-end 32-bit

3-phase sensored (V/Hz, Slip)

Ultra Low End – Low End 8Bit

MCU / MPU

8bit
16bit
32bit

Kinetis

MPC5604P

56F82xx
56F802x/3x
56F801x
56F800x
51AC
S08MP16
S08AC
S08SH
S08SF
S08QD4

MC33991
Dual Gauge Driver

MM908E626
Stepper Motor Driver w/ LIN

MPC17C724
0.4 Amp Dual H-Bridge Motor

MC33879
Octal Serial Switch with Open Load Detect Current Disable

MC33999
16-Output Switch with SPI and PWM Control

MC33937
Three Phase Field Effect Transistor Pre-driver

Analog Portfolio

Freescale, the Freescale logo, AllWec, C-5, Code/TEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeHIVE, BeesStack, CoreNet, the Energy Efficient Solutions logo, Flexis, M/C,, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VoHGs are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
MC9S08MP16

Enabling Safe, Accurate and Inexpensive BLDC Motor Control

http://www.freescale.com/S08MP16
**Core / Temp / Package**
- Industrial Version: 50MHz (25MHz bus), -40 to 105°C
- Automotive Version: 40MHz (20MHz bus), -40 to 125°C
- 2.7V to 5.5V operating range

**Memory**
- 16KB Flash / 1KB RAM
- 12KB Flash / 512B RAM

**Features**
- 2x FlexTimers (6ch + 2ch) – automatic fault protection
- 3 Analog Comparators – h/w sample trigger from PWM module allowing comparison at any point in cycle
- 2x Programmable Delay Blocks (PDB)
- 12-ch 12-bit ADC – 3.5 uS conversion, h/w trigger from PWM module allowing conversion at any point in cycle
- Programmable Gain Amplifier (PGA)
- 8-bit Modulo Timer Module (MTIM)
- LIN SCI, SPI, IIC
- 3x 5-bit DAC used as a 32 tap voltage reference
- RTC
- Software Programmable Internal Clock Source
- 3x low power modes & peripheral CLK gating
- Power Management Controller (PMC)
- KBI
- POR / LVI – supports 4 interrupt priority levels
- Background Debug Mode Interface/ICE

**System Protection**
- Cyclic Redundancy Check Generator (CRC)
- Watchdog Timer with Independent Clock Source
The S08MP16 8-bit MCU delivers safe, accurate, and inexpensive Brushless DC motor control for a wide range of Industrial and Automotive applications.

<table>
<thead>
<tr>
<th>Safe Motor Control</th>
<th>Rich Analog Integration</th>
<th>Broad Development Support</th>
</tr>
</thead>
<tbody>
<tr>
<td>Safe motor operation and increased system integrity for safety-critical applications.</td>
<td>Rich on-chip peripheral integration minimizes BOM costs and delivers precise system control.</td>
<td>Extensive off-the-shelf development resources and complementary analog product solutions.</td>
</tr>
</tbody>
</table>

- **Over-current protection:** Analog Comparators in conjunction with PWM FlexTimer fault inputs, provide fast and accurate over-current shutdown protection by driving PWMs to a safe state.

- **Reduced system cost:** integrated Programmable Gain Amplifier (PGA) and Analog Comparators minimise external component count. Low cost small footprint SOIC & LQFP packages with automotive-qualified high temperature option.

- **Dedicated application support:** Reference Designs, Application Notes, and Software Libraries covering all motor control topologies. Regional Motor Control Centers of Excellence provide application support from concept to delivery.

- **Enhanced system integrity for safety-critical applications implementing IEC60730:** Independently Clocked COP & Cyclic Redundancy Check Engine provide clock failure protection & memory content validation.

- **Precise motor control:** FlexTimer provides dead-time insertion in hardware – more accurate than software insertion and reduces CPU bandwidth usage. PGA allows high resolution ADC readings over a wide range of motor loads and speeds.

- **Reduced time-to-market:** Free CodeWarrior™ IDE up to 32KB with Processor Expert and Freemaster Real-Time Control and Monitoring Tool. One-stop-shop motor control system solutions including MC33937 3-ph FET Pre-Driver, and MC33880/87 & MC33926/32 H-bridges.
Freescale Digital Signal Controller- 56800E family
What is Digital Signal Controller

• Specialized microprocessor whose architecture contains a core engine capable of competitively performing both microcontroller and digital signal processor functionalities
• Core processing capability applicable to many types of system solutions
• Common basic features:
  > MAC, single instruction cycle allowing several memory accesses, address generation units, algorithms for efficient looping
• Specialized Low cost, high performance on-board interfaces utilized in implementing embedded control applications:
  > PWM; multifunction timer; high speed ADCs; DACs; Comparators; SCIs (UART); SPIs; CANs and I2Cs, etc.
• Embedded nonvolatile memory:
  > Flash memory, ROM or EEPROM
• Easy to use development tools
Traditional Microcontroller

- Designed for Controller Code
- Compact Code Size
- Easy to Program
- Inefficient Signal Processing

Traditional DSP Engine

- Designed for DSP Processing
- Designed for Matrix Operations
- Complex Programming
- Less Suitable for Control

- Instructions Optimized for Controller Code, DSP, Matrix Operations
- Compact Assembly and “C” Compiled Code Size
- Easy to Program
- Additional MIPS Headroom and extended addressing space
<table>
<thead>
<tr>
<th>Flash Size</th>
<th>Low Power</th>
<th>Large Capacity</th>
<th>High Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;16kB</td>
<td>MC56F801x 32MHz, HR PWM, ADC</td>
<td>MC56F800x V. Low Cost, HR PWM</td>
<td>MC56F822x 60MHz, UHR PWM UHS ADC</td>
</tr>
<tr>
<td>&lt;32kB</td>
<td>MC56F802x 32MHz, HR PWM, ADC, DAC</td>
<td>MC56Fxxx 40MHz Ultra Low Cost</td>
<td>MC56F822x 60MHz, UHR PWM UHS ADC</td>
</tr>
<tr>
<td>&lt;64kB</td>
<td>MC56F803x 32MHz, HR PWM, CAN, ADC, DAC</td>
<td>MC56F814x 40MHz</td>
<td>MC56F832x 60MHz</td>
</tr>
<tr>
<td>&lt;144kB</td>
<td></td>
<td>MC56F833x 60MHz</td>
<td>MC56F84xx 32 Bit Core, 100MHz, DMA, UHS ADC</td>
</tr>
<tr>
<td>&lt;280kB</td>
<td></td>
<td>MC56F815x 40MHz</td>
<td>MC56F84xx 32 Bit Core, 100MHz, DMA, UHS ADC</td>
</tr>
<tr>
<td>&lt;500kB</td>
<td></td>
<td>MC56F815x 40MHz</td>
<td>MC56F84xx 32 Bit Core, 100MHz, DMA, UHS ADC</td>
</tr>
<tr>
<td>MC56F84xx 32 Bit Core, 100MHz, DMA, UHS ADC</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F836x 60MHz</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F835x 60MHz</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F834x 60MHz</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F803x 32MHz, HR PWM, CAN, ADC, DAC</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F833x 60MHz</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F834x 60MHz</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
<tr>
<td>MC56F832x 60MHz</td>
<td>2011</td>
<td>2011</td>
<td>2010</td>
</tr>
</tbody>
</table>

**Production** - Available NOW

**Execution** - Specification Frozen, in design

**Proposal** - Specification Subject to Change
Freescale Digital Signal Controller- 56800E family
Freescale Single Chip Solution Strategy

- **High Speed DSP Core**
  - CPU:56800E
  - Flash Memory
  - EEPROM
  - RAM

- **Clock System**
  - PLL
  - HS_CLK
  - Sys_CLK
  - Sys_Bus

- **Serial Peripherals**
  - CANs
  - I2Cs
  - SPIs
  - SCIs
  - PWMs
  - Multi Function Timers
  - Tick Timers (PIT)

- **Control Peripherals**
  - Cross Bar
  - Interconnection

- **System**
  - COP
  - SIM
  - LVI
  - POR
  - 12bit ADC
  - ACMPs
  - DACs

- **Vbus**
  - 3.3V On-Chip Regulator

- **Single Power Supply**
- Advanced DSP Core
- Scaleable product portfolio with full enablement
- Internal inter-module Connections
## 56F8000 Series Feature Summary

<table>
<thead>
<tr>
<th>Feature</th>
<th>56F8002</th>
<th>56F8006</th>
<th>56F8011</th>
<th>56F8013</th>
<th>56F8014</th>
<th>56F8023/33</th>
<th>56F8025/35</th>
<th>56F8036</th>
<th>56F8027/37</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Performance</strong></td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
<td>32MHz/MIPs</td>
</tr>
<tr>
<td><strong>Temperature Range (V)</strong></td>
<td>-40°C~105°C</td>
<td>-40°C~105°C</td>
<td>-40°C~125°C</td>
<td>-40°C~125°C</td>
<td>-40°C~125°C</td>
<td>-40°C~125°C</td>
<td>-40°C~125°C</td>
<td>-40°C~125°C</td>
<td>-40°C~125°C</td>
</tr>
<tr>
<td><strong>Voltage Range</strong></td>
<td>1.8V - 3.6V On-Chip</td>
<td>1.8V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
<td>3.0V - 3.6V On-Chip</td>
</tr>
<tr>
<td><strong>Voltage Regulator</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Program/Data Flash</strong></td>
<td>12KB</td>
<td>16KB</td>
<td>12KB</td>
<td>16KB</td>
<td>16KB</td>
<td>32KB/64KB</td>
<td>64KB</td>
<td>32KB/64KB</td>
<td>64KB/64KB</td>
</tr>
<tr>
<td><strong>Program/Data RAM</strong></td>
<td>2KB</td>
<td>2KB</td>
<td>2KB</td>
<td>4KB</td>
<td>4KB</td>
<td>4KB/8KB</td>
<td>8KB</td>
<td>4KB/8KB</td>
<td>8KB/8KB</td>
</tr>
<tr>
<td><strong>OnChip Relaxation Osc</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>PLL</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>COP (Watchdog)</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>PWM (96Mhz Clock)</strong></td>
<td>1 x 6ch</td>
<td>1 x 6ch</td>
<td>1 x 6ch</td>
<td>1 x 6ch</td>
<td>1 x 5ch</td>
<td>1 x 6ch</td>
<td>1 x 6ch</td>
<td>1 x 6ch</td>
<td>1 x 6ch</td>
</tr>
<tr>
<td><strong>PWM Fault Inputs</strong></td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
</tr>
<tr>
<td><strong>12-bit ADCs</strong></td>
<td>2 x 8ch</td>
<td>2 x 12ch</td>
<td>2 x 3ch</td>
<td>2 x 3ch</td>
<td>2 x 4ch</td>
<td>2 x 3ch</td>
<td>2 x 4ch</td>
<td>2 x 5ch</td>
<td>2 x 8ch</td>
</tr>
<tr>
<td><strong>12-bit DACs</strong></td>
<td>0</td>
<td>0</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>2</td>
<td>2</td>
<td>2 (Pinned out)</td>
<td>2</td>
</tr>
<tr>
<td><strong>Analog Comparator</strong></td>
<td>3</td>
<td>3</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>2</td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>Prog Gain Amp</strong></td>
<td>2</td>
<td>2</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td><strong>16-bit Timers</strong></td>
<td>3</td>
<td>3</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td><strong>Prog. Interval Timers</strong></td>
<td>1 (RTC)</td>
<td>1 (RTC)</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>1</td>
<td>3</td>
<td>3</td>
<td>3</td>
</tr>
<tr>
<td><strong>GPIO (max) (+/-8mA)</strong></td>
<td>23</td>
<td>40</td>
<td>26*</td>
<td>26*</td>
<td>26*</td>
<td>26*</td>
<td>35*</td>
<td>39*</td>
<td>53*</td>
</tr>
<tr>
<td><strong>IIC</strong></td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>SCI (UART) / LIN Slave</strong></td>
<td>1 - SCI</td>
<td>1 - SCI</td>
<td>1 - SCI</td>
<td>1 - SCI</td>
<td>1 - SCI</td>
<td>1 - QSCI</td>
<td>1 - QSCI</td>
<td>1 - QSCI</td>
<td>1 - QSCI</td>
</tr>
<tr>
<td><strong>SPI (Synchronous)</strong></td>
<td>1 - SPI</td>
<td>1 - SPI</td>
<td>1 - SPI</td>
<td>1 - SPI</td>
<td>1 - SPI</td>
<td>1 - QSPI</td>
<td>1 - QSPI</td>
<td>1 - QSPI</td>
<td>1 - QSPI</td>
</tr>
<tr>
<td><strong>CAN</strong></td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>MSCAN</td>
</tr>
<tr>
<td><strong>JTAG/EOnCE</strong></td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Power Consumption</strong></td>
<td>IDD = 45.6mA; IDDA = 4.5mA</td>
<td>IDD = 42mA; IDDA = 13.5mA</td>
<td>IDD = 48mA; IDDA = 18.8mA</td>
<td>IDD = 48mA; IDDA = 18.8mA</td>
<td>32LQFP (.8p)</td>
<td>32LQFP</td>
<td>32LQFP</td>
<td>32LQFP</td>
<td>32LQFP</td>
</tr>
</tbody>
</table>

* 5V tolerance I/O

Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, CorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Ultra Low cost 56F8000 member: 56F8002, 56F8006

- 32 MHz/32 MIPS 56800E Core
- 1.8-3.6V Operation
- 12K - 16K Bytes Program FLASH with Flash security
- 2K Bytes Program/Data RAM
- Tunable Internal Relaxation Oscillator and 32KHz clock
- Phase Locked Loop (PLL)
- Up to 96 MHz Peripherals – Timers, PWM & Hi-SCI
- 6 Output PWM Module with 4 Programmable Fault Inputs
  - Programmable Dead timer insertion
  - Programmable PWM generation for Power supply apps
  - Multiple PWM Frequency outputs
- Two Programmable Gain Amplifiers with x2, x4, x8, x16 gains (Clocked in order to cancel input offset)
- Two 12-bit ADCs with up to 24 Inputs, 2.5us Per conversion
- Programmable Delay Block provides precise control of ADC/PGA sample times relative to PWM reload cycles
- Three High Speed Analog Comparators
- 2 multiple function Programmable Timers
- Computer Operating Properly Timer
- One Periodic Interval Timer (PIT)
- 1 High Speed Serial Communication Interface (Hi-SCI)
- 1 Serial Peripheral Interface (SPI)
- I2C Communications Interface
- Up to 40 GPIOs – Versatile pin usage
- JTAG/EOnCE™ Debug Port
- Lead Free “Green” Packages
- Industrial temp: -40C – 105C

Freescale, the Freescale logo, AliVic, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MXC, Platform in a Package, Processor Expert, CorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.

Freescale Semiconductor Confidential and Proprietary Information
Cost Effective 56F8000 Solutions
56F8011/56F8013/56F8014

- 32 MIPS Performance
- 12K - 16K Bytes Program FLASH
- 4K Bytes Program/Data RAM
- Tunable Internal Relaxation Oscillator
- Software Programmable Phase Locked Loop
- Up to 96 MHz Peripherals – Timers and PWMS
- Up to 6-Output PWM Module with up to 4 Programmable Fault Inputs
- Selectable PWM frequency for each complementary PWM signal pair
- Two 12-bit ADCs with up to 8 Inputs, 1.125us conversion rate
- Synchronization between PWM and ADC
- Four 16-bit General Purpose Programmable Timers
- Computer Operating Properly Timer
- Serial Ports: SCI, SPI, I2C
- Up to 26 GPIOs – Versatile pin usage
- Low Power Consumption – 59mA Max and .026mA Min
- JTAG/EOnCE™ Debug Port
- Industrial & Automotive temp
- Cost Effective
56F8000 Family Expansion
56F8023/56F8025/56F8036/56F8037 Features

- 32 MHz/32 MIPS 56800E Core
- 3.0-3.6V Operation
- 32K-64K Bytes Program FLASH
- 4K-8K Bytes Program/Data RAM
- Flash security
- Tunable Internal Relaxation Oscillator
- Software Programmable Phase Locked Loop
- Up to 96 MHz Peripherals – Timers and PWMs
- 6 Output PWM Module with 4 Programmable Fault Inputs
- Selectable PWM frequency for each complementary PWM signal pair
- Two 12-bit ADCs with up to 16 Inputs, 1.125us conversion rate
- Up to Two 12-bit Digital to Analog Converters
- Two Analog Comparators
- Synchronization between PWM and ADC
- 4 or 8 16-bit General Purpose Programmable Timers
- 1 or 3 Programmable Interval Timers (PIT)
- Computer Operating Properly Timer
- 2-Queued Serial Communications Interface
- 2-Queued Serial Peripheral Interface
- Optional MSCAN
- I²C Communications Interface
- Up to 53 GPIOs – Versatile pin usage
- JTAG/EOnCE™ Debug Port
- Industrial & Automotive temp

Package 32 LQFP, 44LQFP, 48LQFP, 64LQFP

Freescale, the Freescale logo, AltiVec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MWC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VonIQs are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Anguilla Black
56F824x / 56F825x Features

- 60 MHz/60MIPS From 56800E Core
- Up to 60 MHz Peripherals
  - 8 Output PWM Module
    - 520ps PWM duty cycle resolution
  - 2 x12-bit ADCs with total of 16 Inputs
    - 500ns conversion rate
    - Built-in PGA - 1x, 2x, 4x, gains
  - 1 x 12-bit Digital to Analog Converter
  - 3 x 5bit Digital to Analog Converters
  - 3 Analog Comparators
  - 8 x16-bit Enhanced Multifunction Programmable QTimers
  - Cyclic Redundancy Check Generator (CRC)
  - 5v tolerant up to 54 GPIOs
  - Inter Module Cross-Bar
Motor Control Use Case

16-bit ADC & PGA:
- Measures 3 phase bridge current and voltage

FlexMemory:
- Saving motor calibration data
- Remote update bootloader

Timers:
- Drives various motor types including stepper, BLDC, and PMAC motors with sensor or sensorless algorithms
- Built-in quadrature decoder detects motor speed

DSP hardware:
- Accelerates motor control calculations

DMA:
- Off loads CPU from repetitive data transfers

16-bit ADC & PGA:
- Measures 3 phase bridge current and voltage

Analog Comparator:
- Detects back EMF
- Monitors over current

Programmable delay block:
- Schedules delayed ADC conversions relative to Timer triggers

I^2C, UART, SPI, CAN:
- Communicates with HMI processor
FlexTimer (FTM) Features

► 16-bit counter with prescaler divide-by 1, 2, 4, 8, 16, 32, 64, or 128

► Each channel can be configured for input capture, output compare, edge-aligned PWM mode, or center-aligned PWM mode

► Each pair of channels can be combined to generate a PWM signal with independent control of both edges of PWM signal

► Each pair can operate as complementary outputs with deadtime insertion

► Dual edge capture for pulse and period width measurement

► Quadrature decoder with input filters for relative position counting

► Global Time Base mode shares single time base across multiple FTM instances
ACIM/PMSM motor control

1) 6 Channels FTM control a 3 phase bridge.
2) Quadrature Decoder output, which is mounted on motor shaft, is two 90° out of phase pulse string.
3) FTM1 is used as quadrature decoder to measure the motor shaft speed.
Power Architecture for Motor Control:
MPC5604P
MPC5604P

Power Architecture® Core
- Up to 64 MHz e200 zen0h core, 32-bit Power Architecture Book E CPU with Harvard architecture
- VLE instruction set encoding for code size footprint reduction

On-Chip Memory Options

<table>
<thead>
<tr>
<th>Device</th>
<th>MPC5602P</th>
<th>MPC5603P</th>
<th>MPC5604P</th>
</tr>
</thead>
<tbody>
<tr>
<td>Program Flash with ECC</td>
<td>256K</td>
<td>384K</td>
<td>512K</td>
</tr>
<tr>
<td>DataFlash® with ECC</td>
<td>64K</td>
<td>64K</td>
<td>64K</td>
</tr>
<tr>
<td>SRAM with ECC</td>
<td>24K</td>
<td>32K</td>
<td>40K</td>
</tr>
<tr>
<td>FlexRay</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Interrupt Controller</td>
<td>100ch</td>
<td>100ch</td>
<td>144ch</td>
</tr>
</tbody>
</table>

I/O Peripherals
- 1 x High speed FlexCAN with 32 Message Buffers (MB)
- 1 x Safety port (can be used as additional FlexCAN – 32 MB)
- 1 x FlexRay Controller - Dual Channel with 32 MB
- 2 x LinFlex
- 4 x DSPI (4 independent chip selects each)
- 1 x FlexPWM (4 channels with 4 fault inputs)
- 2 x eTimer (6 channels incl. quad decode)
- 2 x ADC - 2 x 12 ch. (4 shared channels)
  - 10-bit, conversion time 700 nsec (2 x 6 ch., shared on 100-pin package)
- 1 x CTU triggering unit: 32 input channels, 8 events, 24 ADC cmds.
- 1 x Fault collection unit

System
- 2 x PLL (one FM-PLL, one for FlexRay™)
- Crossbar switch architecture for concurrent access to peripherals
- 16-ch. eDMA
- 16 MHz internal RC OSC
- Junction temperature sensor
- Non-Maskable Interrupt
- Programmable Watchdog
Electric Motor Control Peripherals

Cross Triggering Unit
- Allows mcTIM, PWM, ATD to be synchronized
- Automatic ADC & eTimer acquisitions
- No CPU intervention during the control cycle

FlexPWM
- Based on DSC PWM
- Optimized for 3ph motor control
- One “extra“ pair of PWM integrated
- Includes dead time insertion, fault channels, center/edge alignment, Distortion correction, …
- Register protections
- Double buffered registers
- eDMA supported
- 2 x BUS frequency → high resolution

Timer Module:
- DSC based
- Six Ch IC/OC
- Double buffered registers for detecting two edges in a row
- eDMA supported
- Integrated quad decoder support
- 2 x BUS frequency → high resolution

2x ADC
- Up to 24 independent and 4 shared channels
- 10-bit
- 700 nsec conversion time
- Limit checking & zero crossing detect
Motor Control PWM Peripheral Module

Main Features
- 4 Sub-modules, each with complementary PWM generation, Isense IC/OC and fault input
- 16 bits of resolution for center, edge aligned, and asymmetrical PWMs
- PWM outputs can operate as complimentary pairs or independent channels
- Independent control of both edges of each PWM output
- Independently programmable PWM output polarity
- Separate dead time for rising and falling edges
- Each complementary pair can operate with its own PWM frequency and deadtime values
- All outputs can be programmed to change simultaneously via a "Force Out" event
- Double buffered PWM registers
  - Integral reload rates from 1 to 16
  - Half cycle reload capability

Safety Features
- Write protection for critical registers
- Fault inputs can be assigned to control multiple PWM outputs
- Programmable filters for fault inputs

- Permanent magnet synchronous motor (PMSM, PMAC)
- Brushless DC motor (BLDC)
- AC induction motor (ACIM)
- Switched reluctance motor (SRM)
- Variable reluctance motor (VRM)
- Stepper motors
- DC/DC converters
Summary

   Freescale offers technology for every motor control application

► Energy efficient motor control
   Vector and sensorless control technology in motor control processors
   8- and 16-bit
   16-bit DSCs for ACIM and PMSM solutions
   32-bit Power Architecture® MCUs for standard and premium drives

► Strong technical support
   Rich tools, training, reference designs, libraries
   Devices are ruggedized with long life and reliability
   Industrial products ship 10+ years with high quality and expert support

► Cost-effective safety and security on-chip
   Protect against IP cloning, network data hacking and soft errors
Freescale:

• Motor Control Homepage – www.freescale.com/motorcontrol
• 8-bit Microcontrollers – www.freescale.com/8bit
• 16-bit DSC – www.freescale.com/dsc
• 32-bit Power Architecture® Processors – www.freescale.com/powerpc
• Analog Products – www.freescale.com/analog
• Industrial Segment – www.freescale.com/industrial
Sensorless PMSM Motor Control Using MC56F80xx

- MC56F80xx digital signal controller
- 3-phase AC/BLDC High Voltage Power Stage Board
- 1-phase line input 110/230VAC @ 50/60Hz
- Appliance PM motor
- Initial rotor position detection
- Full torque at motor start-up
- Field weakening
- Application based on C-callable library functions (GFLIB, GDFLIB, MCLIB, ACLIB)
- Current control loop execution time: 55us
- Speed control loop with Field weakening execution time: 17us
- Flash: ~ 6KB, RAM ~ 1.5KB
Low Cost BLDC Motor Control Demo Board

- Brushless motor, Maxon EC-200187, 6W 9V
- Motor interface connector
- Input power connector
- Daughter card connector for connecting the 56F8013 demonstration board
- LED power indicator
- Motor bus voltage sense logic
- Motor bus current sense logic
- Back EMF phase voltage sense logic
- Zero-crossing logic
- Hall-effect/zero-crossing selector
- 3-phase H-bridge power stage
- Power regulation logic

(Optional) Five on-board real-time user debugging LEDs

Order Number: APMOTOR56F8000
Three-phase brushless DC motor sensorless drive

Designed to fit into fan, pump and compressor applications

Using MC56F8013 32 MIPS hybrid controller

Available for two power stages and two motors

Input power supply voltage +12 Vdc for power stages

Control technique incorporates:

- Sensorless, trapezodial control of 3-phase brushless DC motor with back-EMF sensing
- Using A/D converter zero-cross sensing for sensorless control
- Speed and current closed loop with PI controller

Speed range: 200 – 2000 and 500 – 5000 RPM (depending on the motor used)

Manual interface (run/stop switch, up/down pushbuttons)
High Speed Application

- 3-Phase BLDC Drive Using Variable DC Link Six-Step Inverter
- Application Note Number: DRM078
- Speed can exceed 10Krpm
MC9S08MP16: Enablement

- DEMO9S08MP16 (generic demo board)
- Demos
  - Sensorless BLDC Motor Control using ADC approximation
- Reference Designs
  - Sensorless BLDC Motor control using Comparators
  - 3-Ph ACIM V/Hz Drive with PFC
  - Industrial HID Lamp with PFC
- Software Libraries
  - S08 Math and Embedded
- Application Notes
  - S08MP16 comparators for BLDC sensorless motor control
  - Using MP16 peripheral modules (FTM, Delay block, ADC)
  - high speed BLDC sensorless control using ADC approximation
  - BLDC Motor Control using Hall Sensors
  - Using FlexTimer in DC/BLDC Motor Control Application
BLDC Control Basics
Commutation
Sensorless BLDC Motor Control Theory

► Six Step BLDC Motor Control

- Voltage applied on only two phases
- It creates 6 flux vectors
- Phases are powered based on rotor position
- The process is called *Commutation*

Power Stage

Phases voltage
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
</tr>
</tbody>
</table>

Animation part 1/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
</tr>
</tbody>
</table>
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_DCB</td>
<td>-V_DCB</td>
<td>NC</td>
</tr>
<tr>
<td>+V_DCB</td>
<td>NC</td>
<td>-V_DCB</td>
</tr>
<tr>
<td>NC</td>
<td>+V_DCB</td>
<td>-V_DCB</td>
</tr>
<tr>
<td>-V_DCB</td>
<td>+V_DCB</td>
<td>NC</td>
</tr>
<tr>
<td>-V_DCB</td>
<td>NC</td>
<td>+V_DCB</td>
</tr>
<tr>
<td>NC</td>
<td>-V_DCB</td>
<td>+V_DCB</td>
</tr>
</tbody>
</table>

Animation part 3/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+V_{DCB}$</td>
<td>$-V_{DCB}$</td>
<td>NC</td>
</tr>
<tr>
<td>$+V_{DCB}$</td>
<td>NC</td>
<td>$-V_{DCB}$</td>
</tr>
<tr>
<td>NC</td>
<td>$+V_{DCB}$</td>
<td>$-V_{DCB}$</td>
</tr>
<tr>
<td>$-V_{DCB}$</td>
<td>$+V_{DCB}$</td>
<td>NC</td>
</tr>
<tr>
<td>$-V_{DCB}$</td>
<td>NC</td>
<td>$+V_{DCB}$</td>
</tr>
<tr>
<td>NC</td>
<td>$-V_{DCB}$</td>
<td>$+V_{DCB}$</td>
</tr>
</tbody>
</table>

Animation part 4/6
### Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
</tr>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
<tr>
<td>NC</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
<tr>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
</tr>
<tr>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
<tr>
<td>NC</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

Animation part 5/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
</tr>
</tbody>
</table>

Animation part 6/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
</tr>
</tbody>
</table>

Animation part 5/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
</tr>
</tbody>
</table>

Animation part 4/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
</tr>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
<tr>
<td>NC</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
<tr>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
</tr>
<tr>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
<tr>
<td>NC</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
</tr>
</tbody>
</table>

Animation part 3/6
## Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>$+V_{DCB}$</td>
<td>$-V_{DCB}$</td>
<td>NC</td>
</tr>
<tr>
<td>$+V_{DCB}$</td>
<td>NC</td>
<td>$-V_{DCB}$</td>
</tr>
<tr>
<td>NC</td>
<td>$+V_{DCB}$</td>
<td>$-V_{DCB}$</td>
</tr>
<tr>
<td>$-V_{DCB}$</td>
<td>$+V_{DCB}$</td>
<td>NC</td>
</tr>
<tr>
<td>$-V_{DCB}$</td>
<td>NC</td>
<td>$+V_{DCB}$</td>
</tr>
<tr>
<td>NC</td>
<td>$-V_{DCB}$</td>
<td>$+V_{DCB}$</td>
</tr>
</tbody>
</table>

Animation part 2/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
</tr>
</tbody>
</table>

Animation part 1/6
Combining block provides flexibility without breaking compatibility
Added functionality is optional within the architecture
Back EMF Basics
Sensorless BLDC Motor Control Theory

- Trapezoidal Back-EMF
Sensorless BLDC Motor Control Theory

Sensorless Back-EMF zero cross algorithm

- Sensing voltage on disconnected phase
  - When sensed voltage crosses half of DC bus voltage, the rotor is in middle between two commutations
  - The detection of this zero crossing allows detection of rotor position
Sensorless BLDC Motor Control Theory

Sensorless Back-EMF zero crossing - detail
9S08MP16 ADC Structure

- PGA Synch
- Channel Select
- Control Register
- 12-bit ADC Converter
- Data Reg.
- Compare Logic
- Compare Value Reg.
- Selectable gain: 1, 2, 3, 4, 6, 8, 9, 12, 16, 18, 24, 32
- Interrupt when:
  - Less-than
  - Greater than, or
  - Equal-to
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
- AN0
- AN1
- AN11
9S08MP16 ADC Synchronization Features

Programmable Delay Block (PDB1)

- RTC
- HSCMP1
- HSCMP2
- HSCMP3
- FTM1
- FTM2

Overflow
C1OUT
C2OUT
C3OUT
Init
Trigger
Init
Trigger
Init
Trigger
Init
Trigger

PGA

Delay based on PGA config

TM

PGAEN

OR

ADHWT

ADC module

TriggerA

TriggerB

In0

In1

In2

In3

In4

In5

Trigger

SW Trig

Delay

Based on PGA config

Trigger

PGA EN
Comparator Window Sampling

Phase A and Phase B are powered
Phase C is unpowered and used to detect Back EMF
Vc = Back EMF

Current Increase Case When PWM is on

Back EMF
Zero Crossing

Current Decay Case When PWM is off
Hardware Overview
BLDC – Recommended Application: DRM117
DRM117: 3-phase Sensorless BLDC Motor Control using MC9S08MP16.

- 3-phase trapezoidal BLDC motor control with 6-step commutation (60, 120 degree control).
- MP16 in-built high-speed comparator (HSCMP) detects the back-EMF voltage zero-crossing.
- Three sensorless synchronized commutation control algorithms incorporating:
  - Commutation instant calculated directly from the period between two back-EMF zero-crossings — Direct Drive.
  - Commutation period synchronized with the back-EMF zero-crossing using a closed-loop according to a phase error — Synchronized PLL.
  - Constant commutation period forced with the motor voltage controlled in a closed-loop according to a phase error — Forced PLL.
- Controlled acceleration and deceleration.
- Bidirectional rotation.
- Both motor and generator modes. (4 Quadrant operation)
- Two PWM techniques possible: Unipolar or Bipolar.
- Software over-voltage and under-voltage protection.
- Hardware over-current protection.
- FreeMASTER control interface (start/stop, speed setup).
- FreeMASTER software monitor.
  - FreeMASTER software graphical control page (required speed, actual motor speed, start/stop status, DC-Bus voltage level, motor current, system status).
  - FreeMASTER software speed scope.
3-Phase Sensorless BLDC Using MC9S08MP16

Application Code

Design Reference Manual

Control Page

Demo Hardware
Why BLDC with MP16?
Motor Types

- DC Motor
- Permanent Magnet Synchronous Motor (PMSM)
- Brushless DC Motor (BLDC)
## Brushed and Brushless Motors Comparison

<table>
<thead>
<tr>
<th>Feature</th>
<th>Brushed DC motor</th>
<th>BLDC Motor</th>
</tr>
</thead>
<tbody>
<tr>
<td>Commutation</td>
<td>+ Brushed commutation</td>
<td>- Electronic commutation</td>
</tr>
<tr>
<td>Maintenance</td>
<td>- Periodic maintenance is required</td>
<td>+ Less required due to absence of brushless</td>
</tr>
<tr>
<td>Noise/EMI</td>
<td>- Higher</td>
<td>+ Lower</td>
</tr>
<tr>
<td>Life</td>
<td>- Shorter</td>
<td>+ Longer</td>
</tr>
<tr>
<td>Speed/Torque</td>
<td>- Moderately Flat. Higher speeds produces higher friction and this reduces torque.</td>
<td>+ Flat</td>
</tr>
<tr>
<td>Speed Range</td>
<td>- Lower – Mechanical limitations by the brushes</td>
<td>+ Higher – No mechanical limitation</td>
</tr>
<tr>
<td>Acceleration</td>
<td>- Lower</td>
<td>+ Higher (Permanent Magnet rotor has less inertia)</td>
</tr>
<tr>
<td>Torque/Size Ratio</td>
<td>- Lower</td>
<td>+ Higher</td>
</tr>
<tr>
<td>Building Cost</td>
<td>+ Lower</td>
<td>- Higher – Permanent magnets</td>
</tr>
<tr>
<td>Control</td>
<td>+ Simple</td>
<td>- Complex and expensive</td>
</tr>
<tr>
<td>Control Requirements</td>
<td>+ A controller is required only when variable speed is desired</td>
<td>- A controller is always required</td>
</tr>
</tbody>
</table>

---

Freescale, the Freescale logo, AllWec, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MBC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
## Six step control versus sinusoidal control

<table>
<thead>
<tr>
<th>Six step control</th>
<th>Sinusoidal control</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ Simple PWM generation</td>
<td>– More complex PWM generation (sinewave has to be generated)</td>
</tr>
<tr>
<td>+ Simple MCU with the right peripherals can be used.</td>
<td>- DSP is required for calculations (Clark, Park, BEMF estimator)</td>
</tr>
<tr>
<td>– Ripple in the torque (stator flux jumps by 60°)</td>
<td>+ Smooth torque (stator flux rotates fluently)</td>
</tr>
<tr>
<td>– A little noise operation (due to ripple in the torque)</td>
<td>+ Very quiet</td>
</tr>
<tr>
<td>+ Simple sensor</td>
<td>– Requires sensor with high resolution</td>
</tr>
<tr>
<td>+ Direct measurement of BEMF Voltage</td>
<td>– BEMF estimation with observer</td>
</tr>
</tbody>
</table>
Why MP16?

► Compared with DSPs / DSCs
  • Easier to understand if you have experience with 8-bits.
  • Easy to port applications made for 8-bits MCU.
  • Cheaper.
  • Cheaper.

► Compared with regular 8-bits MCUs
  • Automatic Dead-time insertion.
    ▪ Regular MCUs must use “software” PWM for dead-time insertion.
  • External or Internal Fault input.
    ▪ Hardware turns off PWM outputs much faster.
  • LOAD_OK bit for PWM reload all channels at same time.
    ▪ Avoids possible current glitches / short-circuit.
  • Better syncronization with PDB.
    ▪ Much better control, lower CPU overhead.
What do we know so far?
Summary

► BLDC motors provide a more **reliable** and **efficient** control than Brushed DC motors.

► Freescale’s MC9S08MP16 MCU enables **low-cost** BLDC motor control:
  - Motor control **optimized** features: HSCMP, FTM, PDB.
  - Peripheral **integration**: lower system cost: PGA.
  - Lower **price** point than DSPs.

► Freescale provides a rich **enablement** set:
  - Reference **design**, with software, hardware and full documentation.
  - **FreeMaster**, **CodeWarrior** and development board.
  - Application notes, Data Sheets, Reference Manuals.
Controlling a BLDC Motor with Sensors
Sensor Example: Hall Effect Sensor

- A transducer that varies its output voltage in response to changes in magnetic field
- Used for proximity switching, positioning, speed detection and current sensing applications
- In motor control applications they are used as on/off switches

Every time a magnetic field is sensed, a change in voltage can be detected
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Hall Sensors</th>
<th>A</th>
<th>B</th>
<th>C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>NC</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Animation part 1/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
<th>Hall Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
<td>1 0 0</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
<td>1 1 0</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>0 1 0</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
<td>0 1 1</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
<td>0 0 1</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>1 0 1</td>
</tr>
</tbody>
</table>

Animation part 2/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
<th>Hall Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
<td>1 0 0</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
<td>1 1 0</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>0 1 0</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
<td>0 1 1</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
<td>0 0 1</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>1 0 1</td>
</tr>
</tbody>
</table>

Animation part 3/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
<th>Hall Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
<td>1 0 0</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
<td>1 1 0</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>0 1 0</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
<td>0 1 1</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
<td>0 0 1</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>1 0 1</td>
</tr>
</tbody>
</table>

Animation part 4/6
### Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
<th>Hall Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>NC</td>
<td>A  0 0</td>
</tr>
<tr>
<td>+V_{DCB}</td>
<td>NC</td>
<td>-V_{DCB}</td>
<td>1 1 0</td>
</tr>
<tr>
<td>NC</td>
<td>+V_{DCB}</td>
<td>-V_{DCB}</td>
<td>0 1 0</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>NC</td>
<td>0 1 1</td>
</tr>
<tr>
<td>-V_{DCB}</td>
<td>NC</td>
<td>+V_{DCB}</td>
<td>0 0 1</td>
</tr>
<tr>
<td>NC</td>
<td>-V_{DCB}</td>
<td>+V_{DCB}</td>
<td>1 0 1</td>
</tr>
</tbody>
</table>

Animation part 5/6
Control of 3-Phase Inverter

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
<th>Hall Sensors</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>1 0 0</td>
</tr>
<tr>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>1 1 0</td>
</tr>
<tr>
<td>NC</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>0 1 0</td>
</tr>
<tr>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>NC</td>
<td>0 1 1</td>
</tr>
<tr>
<td>NC</td>
<td>-V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>+V&lt;sub&gt;DCB&lt;/sub&gt;</td>
<td>0 0 1</td>
</tr>
</tbody>
</table>

Animation part 6/6

Freescale, the Freescale logo, and all other trademarks are the properties of Freescale Semiconductor Inc. Reg. U.S. Pat. & Tm. Off. BeekIT, BeekStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MII, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VoHIDs are trademarks of Freescale Semiconductor Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Mini Hands-on: How Sensors Behave (Lab 0)
1. Open *CodeWarrior for MCU version 6.3*
2. Click on *File > Open*
3. Browse for the *Labs* folder
4. Open *lab0_bldctraining.mcp*
5. Click *F5* or the *debug* symbol
6. Wait for the debug prompt to open, then click *Ok* to flash the device.
7. Make sure the encoder/sensor connector is properly plugged
8. Wait for the programming to finish and click the *Run* button (or *F5*) when finished
9. Observe the LED in the board
10. Rotate the motor manually and see how the LED change with each step of the motor
Electronic Motor Commutation
PWM Modes used for BLDC Motor Operation

**Independent Mode**
- Simple implementation
- Masking of PWM channel required

**Complementary Mode**
- Required sophisticated PWM support
- Mask and Swap of PWM channel required
- Allows energy recuperation
Quadrants of Operation

First Quadrant
positive speed-positive torque
forward-accelerating

Second Quadrant
negative speed-positive torque
reverse-braking

Third Quadrant
negative speed-negative torque
reverse-accelerating

Fourth Quadrant
positive speed-negative torque
forward-braking

I
II
III
IV

Torque
Speed
Unipolar BLDC Commutation

- No energy recuperation
- Simple implementation
Unipolar BLDC Commutation (3 Complementary PWM pairs)

- Allows energy recuperation
- *Bottom* is opposite of *Top*
- Requires sophisticated PWM support

![Diagram of Unipolar BLDC Commutation](image)
Bipolar BLDC Commutation (6 independent PWMs)

- No energy recuperation
- Simple implementation when 6 PWM available

Diagram showing the commutation cycles for A, B, and C phases.
Bipolar BLDC Commutation (3 Complementary PWM pairs)

- Allows energy recuperation
- *Bottom* is opposite of *Top*
- Requires sophisticated PWM support

![Diagram of Bipolar BLDC Commutation]

commutation commutation commutation commutation

commutation 120° 60° commutation commutation

A\text{TOP} A\text{BOT} B\text{TOP} B\text{BOT} C\text{TOP} C\text{BOT}

A - Off B - Off C - Off

Swap

Freescale, the Freescale logo, AllWec, C-5, Code/TEST, CodeWarrior, ColdFire, C-Ware, mobilEGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MWC, Platform in a Package, Processor Expert, QorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Hands-on: Run the BLDC with Sensors Demo (Lab 1)
Core / Temp / Package

**Industrial Version:**
- 50MHz (25MHz bus), -40 to 105°C, 28SOIC, 32LQFP, 48LQFP

**Automotive Version:**
- 40MHz (20MHz bus), -40 to 125°C, 48LQFP
- 2.7V to 5.5V operating range

Memory

- 16KB Flash / 1KB RAM
- 12KB Flash / 512B RAM

Features

- 2x FlexTimers (6ch + 2ch) – automatic fault protection, supports up to 50MHz CLK, selectable input capture, output compare, edge- or center-aligned PWM; dead time insertion
- 3 Analog Comparators – h/w sample trigger from PWM module allowing comparison at any point in cycle
- 2x Programmable Delay Blocks (PDB)
- 12-ch 12-bit ADC – 3.5 uS conversion, h/w trigger from PWM module allowing conversion at any point in cycle
- Programmable Gain Amplifier (PGA)
- 8-bit Modulo Timer Module (MTIM)
- LIN SCI, SPI, IIC
- 3x 5-bit DAC used as a 32 tap voltage reference
- RTC with periodical timer interval interrupt
- Software Programmable Internal Clock Source
- 3x low power modes & peripheral CLK gating
- Power Management Controller (PMC)
- 3x 8-bit KBI
- POR / LVI – supports 4 interrupt priority levels
- Background Debug Mode Interface/ICE

System Protection

- Cyclic Redundancy Check Generator (CRC)
- Watchdog Timer with Independent Clock Source

---

**MC9S08MP16/12**

SPI | S08 50MHz CPU | 13ch, 12-bit ADC & Temp Sensor
---|---|---
SCI | 16KB Flash | 6ch+2ch FlexTimer with PWM functions
---|---|---
IIC | 1KB RAM | Programmable Gain Amplifier
---|---|---
MTIM | CRC Generator | Programmable Delay Blocks (x2)
---|---|---
DAC | ICS | 3x High Speed Analog Comparators
---|---|---
RTC | COP | BDM/ICE
---|---|---

<table>
<thead>
<tr>
<th>Device</th>
<th>Package</th>
<th>RAM</th>
<th>ADC</th>
<th>PGA</th>
<th>IIC</th>
<th>I/O</th>
</tr>
</thead>
<tbody>
<tr>
<td>MC9S08MP16</td>
<td>48LQFP</td>
<td>1KB</td>
<td>13ch</td>
<td>1</td>
<td>1</td>
<td>40</td>
</tr>
<tr>
<td>MC9S08MP16</td>
<td>32LQFP</td>
<td>1KB</td>
<td>12ch</td>
<td>1</td>
<td>1</td>
<td>25</td>
</tr>
<tr>
<td>MC9S08MP16</td>
<td>28SOIC</td>
<td>1KB</td>
<td>8ch</td>
<td>1</td>
<td>1</td>
<td>22</td>
</tr>
<tr>
<td>MC9S08MP12</td>
<td>28SOIC</td>
<td>512B</td>
<td>8ch</td>
<td>-</td>
<td>-</td>
<td>22</td>
</tr>
<tr>
<td>S9S08M16E2MLF</td>
<td>48LQFP</td>
<td>1KB</td>
<td>13ch</td>
<td>1</td>
<td>1</td>
<td>40</td>
</tr>
</tbody>
</table>

**50K # resale price**

---

Freescale, the Freescale logo, AllWorx, C-5, CodeTEST, CodeWarrior, ColdFire, C-Ware, mobileGT, PowerQUICC, StarCore, and Symphony are trademarks of Freescale Semiconductor, Inc., Reg. U.S. Pat. & Tm. Off. BeeKit, BeeStack, CoreNet, the Energy Efficient Solutions logo, Flexis, MCK, Platform-in-a-Package, Processor Expert, CorIQ, QUICC Engine, SMARTMOS, TurboLink and VortiQa are trademarks of Freescale Semiconductor, Inc. All other product or service names are the property of their respective owners. © 2010 Freescale Semiconductor, Inc.
Run the Demo

1. Open *CodeWarrior for MCU version 6.3*
2. Click *File > Open*
3. Browse for the *Labs* folder
4. Open *lab1_bldctraining.mcp*
5. Click *F5* or the *debug* symbol
6. Wait for the debug prompt to open and click *ok* to flash the device.
7. Make sure the encoder/sensor connector and the motor connector are properly plugged
8. Wait for the programming to finish and click the *run* button (or F5) when finished
1. Open CodeWarrior for MCU version 6.3
2. Click File > Open
3. Browse for the Labs folder
4. Open `lab1_bldctraining.mcp`
5. Click F5 or the debug symbol
6. Wait for the debug prompt to open and click ok to flash the device.
7. Make sure the encoder/sensor connector and the motor connector are properly plugged.
8. Wait for the programming to finish and click the `run` button (or F5) when finished.

Before going forward make sure no obstacles are near the motor shaft … specially your badges. We do not want you tangled up in there!!!
Up: On
Center: On
Down: Stop

Up increase speed/down decrease speed until stop and then invert rotation
Hands-on: The BLDC Sensorless Demo
3-Phase BLDC/PMSM Low-Voltage Motor Control Drive

DC Bus Voltage & Current Sensing

3 Phase Inverter

3 Phase Voltages

3 Phase BLDC Motor

Vdcb, Idcb, Vphasea,b,c, PWM1..6

Over-current FAULT

ADC Module

HSCMP2
ZC Comparator

PDB2
ZC to PWM Synchronization

FTM2
PWM 3pps Generator

PDB1
Synchronization

ADC to PWM Synchronization

PDB2
Zero Crossing Period & Position Recognition

FTM1
Commutation And PWM Control

FTM

SPI
MC33927

MCF51AG 128

GPIO Module

SCI Module

Application Control

Freemaster

USB to COM Convertor

I/O Ports

Superior System Application Monitoring and Control

DC Bus Voltage

24V DC Power Input

PWM Duty cycle

Zero-crossing Period

Desired speed

Actual speed

Speed PI Controller

Torque PI Controller

MTIM
Time Base

On Board Programming

BDM

MC9S08MP16

FTM

Simillar Peripheral to MCF51AG 128
Back-EMF Zero-Crossing Sensing Circuit

½ $U_{DCB}$ reference

- Sampling Window Generator
- Positive MUX
- Negative MUX
- Phase Selection According to PWM Sector
- PWM Sync
- ZC Sampling Window
- Sampling
- $U_{dcb}$
- $U_{dcb}/2$
- Phase a
- Phase b
- Phase c
Practical Sensorless Motor Control: How is BEMF Used?
Back EMF in a Single Loop of Wire
Back EMF in a Multi-turn Winding

Uniform winding density

Source: Electric Drives, an Integrative Approach, by Ned Mohan, University of Minn. Printing Services, 2000
Sensorless Back-EMF Zero-Cross Algorithm

- Sensing voltage on disconnected phase
  - When sensed voltage crosses half of DC bus voltage, the rotor is in middle between two commutations
  - The detection of this zero crossing allows to detect rotor position
Sensorless BLDC Motor Control Theory

Detail: Sensorless Back-EMF Zero-Crossing
BEMF Detection States

BLDC Commutation
Zero Crossing Detection Process

CMT

Zero Crossing Detected

90 120 150 180 210 240 270
Software Implementation

![Diagram of software implementation process]

- BLDC Commutation Current Recirculation
  - BLDC Committed
  - ZC Sensing Prepare (phase selection)
  - Done

- Motor Commutation overtook ZC
  - timeZC = timeCMT
  - Done

- ZC Detection Process (Phase BEMF of ZC expected level search)
  - ZC Get
  - ZC Detected (saves timeZC)
  - Done

- Current Recirculation Done (periodToff timeout)
Open Loop Startup on a Sensorless Application
BEMF Facts

► Must detect BEMF to spin the motor in sensorless mode.

► In order to generate BEMF, the rotor must be spinning.

► The faster the rotor is the higher is the BEMF voltage.

► So .... how to start the motor from zero speed?

► Classic Chicken or the egg question

Open Loop Startup is the answer!
Rotor Alignment

► For the required **higher** initial torque, the rotor position **must** be known to apply the best next commutation.
► Unsure and without sense of the rotor position, **rotor alignment** is required
► One of the simplest techniques is to **force** a known position.
► The rotor position is stabilized by applying PWM signals to **only two** motor phases (no commutation).

*(The current controller keeps the current within predefined limits.)*

<table>
<thead>
<tr>
<th>Phase A</th>
<th>Phase B</th>
<th>Phase C</th>
</tr>
</thead>
<tbody>
<tr>
<td>+V(_{DCB})</td>
<td>-V(_{DCB})</td>
<td>NC</td>
</tr>
<tr>
<td>+V(_{DCB})</td>
<td>NC</td>
<td>-V(_{DCB})</td>
</tr>
<tr>
<td>NC</td>
<td>+V(_{DCB})</td>
<td>-V(_{DCB})</td>
</tr>
<tr>
<td>-V(_{DCB})</td>
<td>+V(_{DCB})</td>
<td>NC</td>
</tr>
<tr>
<td>-V(_{DCB})</td>
<td>NC</td>
<td>+V(_{DCB})</td>
</tr>
<tr>
<td>NC</td>
<td>-V(_{DCB})</td>
<td>+V(_{DCB})</td>
</tr>
</tbody>
</table>

Animation part 1/2
For the required higher initial torque, the rotor position must be known to apply the best next commutation.

Unsure and without sense of the rotor position, rotor alignment is required.

One of the simplest techniques is to force a known position.

The rotor position is stabilized by applying PWM signals to only two motor phases (no commutation).

(The current controller keeps the current within predefined limits.)
Openloop Startup

- Commutation starts the **open-loop** without rotor position feedback.
- Period is controlled by a **linear velocity ramp**.
- Rotor and stator flux need to be in an approximately **90 degree relation** to maintain synchronization.
- Torque is lower and not constant.
- Needs to be a **short state** at a very low speed where the back-EMF is too small so the zero-crossing cannot be reliably detected.
- so the zero-crossing cannot be reliably detected.
Hands-on: Sensorless App
Run the Lab!

► Please refer to the **Lab Guide** for instructions on how to proceed.

► For this first lab, please follow only the **first 6 steps** on the Lab Part1 session.
Run the Lab!

► Please refer to the Lab Guide for instructions on how to proceed.

► For this first lab, please follow only the first 6 steps on the Lab Part 1 session.

WAIT!!!

Before going forward make sure no obstacles are near the motor shaft … specially your badges. We do not want you tangled up in there!!!
FreeMASTER software provides:

- Debugging, diagnostic and demonstration tool for the development of algorithms and applications
- Very useful for tuning the application
- RS-232 serial port, JTAG, BDM or USB communication with PC
Control page is a graphical user interface (GUI) for the 3-phase sensorless BLDC control.

Actions supported:
- Setting the required speed
- Switch running motor on/off

Control page displays:
- Actual and required speed
- DC-bus current and voltage
- Application (fault) status

**BLDC Sensorless Application using MC9S08MP16**

- DC Undervoltage
- DC Overvoltage
- DC Overcurrent
- Commutation Error
- DC Current Limiting

Application Mode
- RUN
- ALIGNMENT

Commutation Mode

RESET
Please refer to the Lab Guide for instructions on how to proceed.

For this lab, follow the remaining steps on Lab Part 1 from 8 to 18.

Before going forward make sure no obstacles are near the motor shaft … specially your badges. We do not want you tangled up in there!!!
Run the Lab!

► Please refer to the **Lab Guide** for instructions on how to proceed.

► For this lab, follow the remaining steps on Lab Part1 from 8 to 18.
Acceleration Comments

► Did you observe any unexpected behavior?

► How would you improve it?

► What other underlying effects does this behavior have?
DC Bus Voltage
Speed Controller with Current (Torque) Limitation

- Speed is controlled using the zero-crossing period feedback provided.
- The outer current regulator limits the motor current.
  
  *This provides the torque limitation in order to limit the maximal motor current.*

- The speed regulator controls the 3-phase power stage PWM.
Add a Speed Ramp

- We need to add a Speed Ramp before the PI controller.

```c
/* function using the acceleration Ramp */
velocityRampAct = ECLIB_Ramp16(rampAccelCL, velocityDesired, velocityRampAct);
```
Run the Lab!

► Please refer to the **Lab Guide** for instructions on how to proceed.

► For this lab, follow the **Lab Parts 2 and 3**.
Run the Lab!

► Please reffer to the Lab Guide for instructions on how to proceed.

► For this lab, follow the Lab Parts 2 and 3.

Before going forward make sure no obstacles are near the motor shaft … specially your badges. We do not want you tangled up in there!!!
Constant Speed Control
Run - Direct Commutation

- CMT time depends on the time difference between the last 2 Zero-crossings.

- In case Zero-cross is not detected properly, the last value is used.

- Very good technique if speed change is required very often.

- The Speed controller uses the PWM duty-cycle to control.

- Zero-crossing time will be automatically changed.

- Any small error in measurement can cause changes in motor speed.
Forced PLL uses a **constant Commutation time** (Tcmt).

As we know the speed required, we know the **Tbemf required**.

The **difference** between the desired Tbemf and the measured Tbemf is calculated.

This difference is then used to control the **PWM duty-cycle**, that will change back the Zero-crossing time.

If the Tbemf difference is **too high**, the control might have problems, so the application changes to **Run-Direct** mode.

This mode is much **more stable** on constant speed applications.
Run the Lab!

► Please refer to the **Lab Guide** for instructions on how to proceed.

► For this lab, follow the **Lab Part 4**.
Run the Lab!

► Please refer to the **Lab Guide** for instructions on how to proceed.

► For this lab, follow the **Lab Part 4**.
Summary
### Control comparison table: with Sensors versus Sensorless

<table>
<thead>
<tr>
<th>Feature</th>
<th>With Sensors</th>
<th>Sensorless</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection issues</td>
<td>- Possible (add connectors and cables to the system)</td>
<td>+ No sensor connections.</td>
</tr>
<tr>
<td>Low Speed</td>
<td>+ Good low speed operation, position feedback from Zero</td>
<td>- Openloop startup needed, no feedback on lower speeds.</td>
</tr>
<tr>
<td>Maximum Speed</td>
<td>+ Can get to higher speed ranges</td>
<td>- Maximum speed limited due to BEMF reading time.</td>
</tr>
<tr>
<td>Control</td>
<td>+ Easier</td>
<td>- More complex (BEMF, Openloop startup)</td>
</tr>
<tr>
<td>Torque control</td>
<td>+ Good on all operation speeds</td>
<td>- Lower on Openloop startup</td>
</tr>
<tr>
<td>Cost</td>
<td>- Higher (sensors, cables, connectors)</td>
<td>+ Lower</td>
</tr>
</tbody>
</table>
Summary

► BLDC motor control schemes go from simple and able to be handled by simple generic microcontrollers to complex PWM switching mechanisms. Although higher-end microcontrollers may seem better suited for these tasks, a complex timer with a simpler CPU is enough to handle most of the more complicated motor control features, leaving the level of mathematical control to the CPU.

► With an adequate motor control-enabled timer, PWM switching schemes are equally simple to integrate, it is just a matter of choosing the right scheme for the application.

► Acceleration ramps allow motor control applications a simple way to maintain DC bus current low during start-up. Ramps can be easily tuned to provide the best start-up time for the motor depending on the amount of expected initial torque.
Freescale Product Longevity Program

► The embedded market needs long-term product support
► Freescale has a longstanding track record of providing long-term production support for our products
► Freescale offers a formal product longevity program for the market segments we serve
  * For the automotive and medical segments, Freescale will make a broad range of program devices available for a minimum of 15 years
  * For all other market segments in which Freescale participates, Freescale will make a broad range of devices available for a minimum of 10 years
  * Life cycles begin at the time of launch
► A list of participating Freescale products is available at: www.freescale.com/productlongevity