Freescale Semiconductor
Application Note

AN1999
Rev. 0, 11/2004

56F8300 Hybrid Controller Used in Control of Electro-Mechanical Brake

This application outlines the basics of EMB design based on actuator control of a Permanent Magnet Synchronous Motor with a 56F8300 device

Peter Balazovic

Note: The PC master software referenced in this document is also known as Free Master software.

1. Introduction

This application note describes the electro-mechanical braking (EMB) system and the design basics of its control based on Freescale's hybrid 56F8300 controller family of devices.

Electro-mechanical braking systems (see [9], **References**), also referred to as brake by-wire, replace conventional hydraulic braking systems with a completely "dry" electrical component system. This occurs by replacing conventional actuators with electric motor-driven units. This move to electronic control eliminates many of the manufacturing, maintenance, and environmental concerns associated with hydraulic systems.

As in electro-hydraulic braking (EHB), EMB is designed to improve connectivity with other vehicle systems, thus enabling simpler integration of higher-level functions, such as traction control and vehicle stability control. This integration may vary from embedding the function within the EMB system, as with ABS, to interfacing to these additional systems via communication links.

Another advantage of both EHB and EMB systems is the elimination of the large vacuum booster found in conventional systems. Along with reducing the dilemma of working with an increasingly tighter space in the engine bay, this elimination

CONTENTS

helps simplify production of right- and left-hand drive vehicle variants. An increase in flexibility for the placement of components is also provided by EMB systems, compared to those of EHB, with the total elimination of the hydraulic system.

The assumed EMB complex control requires a dedicated high-performance embedded controller. Freescale Semiconductors provides dedicated motor control devices based on a 56800E core, which combines DSP power and parallelism with MCU-like programming simplicity. It is targeted for demanding automotive applications like EMB systems. The 56F8300 family includes a number of devices with differnet peripheral and memory sets.

2. Freescale Hybrid Controllers and Automotive ICs

Freescale products serve as the foundation for many automotive applications which are ideally suited for the rugged automotive enviroment. Continuing Freescale's legacy of best-in-class automotive MCUs, Freescale Semiconductor provides a full family of 0.25µ, 16-bit products based on the powerful 56800E CPU core. The high-performance Flash series of 56F8300 hybrid controllers combines the 56800E hybrid core with Flash memory, motor control peripherals, and built-in safety features, targeted specifically at automotive applications to provide 60 MIPS of performance over the full -40°C to 125°C temperature range. For more insight, please see [10] and [11], **References**.

2.1 56F8300 Hybrid Controller

The 56F8300 devices are members of the 16-bit 56800E core-based controller family. Each device's processing power, combining the functionality of a microcontroller with a flexible set of peripherals, creates an extremely cost-effective solution on a single chip. Because of its low cost, configuration flexibility, and compact program code, a 56F8300 device is well-suited for many actuator applications in the automotive industry.

The 56800E core is based on a Harvard-style architecture consisting of three execution units operating in parallel, allowing as many as six operations per instruction cycle. The MCU-style programming model and optimized instruction set allow straightforward generation of efficient, compact 16-bit control code. The instruction set is also highly efficient for C/C++ compilers, enabling rapid development of optimized control applications.

A typical member of the 56F8300 series provides the following peripheral blocks:

- Pulse Width Modulator unit, each with six PWM outputs, three current sense inputs, and fault inputs, fault-tolerant design with deadtime insertion; supports both center- and edge-aligned modes
- 12-bit Analog-to-Digital Convertor (ADC), supporting two simultaneous conversions with dual 4-pin multiplexed inputs; ADC can be synchronized by PWM modules
- Quadrature decoder, dedicated to the quadrature encoder signals with a maximum counting resolution of 4x input signal
- Dedicated general-purpose Quad Timers containing four identical counter/timer groups
- CAN 2.0 A/B module with 2-pin ports used to transmit and receive
- Serial Communication Interface (SCI) allowing asynchronous serial communications with peripheral devices and other controllers
- Serial Peripheral Interface (SPI), with configurable 4-pin port (or four additional GPIO lines)
- Computer Operating Properly (COP)/watchdog timer
- Dedicated external interrupt pins

- General purpose I/O (GPIO) pins, able to multiplex GPIO pins with other peripherals
- External reset pin for hardware reset
- JTAG/On-Chip Emulation (OnCE) for unobtrusive, processor speed-independent debugging
- Software-programmable, Phase Lock Loop-based frequency synthesizer for the core clock

The **Analog-to-Digital Converter** (ADC) consists of two separate and complete ADCs, each with its own sample and hold circuits. A common digital control module configures and controls the functioning of both ADCs. ADC features include:

- 12-bit resolution
- Maximum ADC clock frequency is 5MHz with a 200ns period
- Sampling rate up to 1.66 million samples per second
- Single conversion time of 8.5 ADC clock cycles (8.5 x 200ns = 1.7µs)
- Additional conversion time of 6 ADC clock cycles (6 x 200ns = 1.2µs)
- Eight conversions in 26.5 ADC clock cycles (26.5 x 200ns = 5.3µs) using Simultaneous mode
- ADC conversions can be synchronized by both the PWM and the TMR
- Simultaneous or sequential sampling with additional text
- Ability to simultaneously sample and hold two inputs
- Ability to sequentially scan and store up to eight measurements
- Internally multiplex to select two of eight inputs
- Power savings modes allow the automatic shutdown/startup of all or part of the ADC
- Built-in calibration using on-chip input voltage network
- Optional interrupts at the end of a scan, if an out-of-range limit is exceeded (either high or low), or at zero crossing
- Optional sample correction by subtracting a pre-programmed offset value
- Signed or unsigned result
- Single-ended or differential inputs for all input pins with support for an arbitrary mix of input types

The **Pulse Width Modulation Unit** (PWM) incorporates a PWM generator, enabling the generation of control signals for the motor power stage. The module has the following features:

- Six PWM signals, which may be used:
  - — Independently
  - — In complementary pairs
  - — In a mix of independent and complementary pairs
- Features of complementary channel operation
  - — Dead time insertion
  - — Separate top and bottom pulse width correction via current status inputs or software
  - — Asymmetric PWM output within the center-align operation
  - — Separate top and bottom polarity control
- Edge- or center-aligned PWM signals
- 15 bits of resolution
- Half-cycle reload capability

- Integral reload rates from 1 to 16
- Individual software-controlled PWM output
- Programmable fault protection
- Polarity control
- 10/12 mA current source/sink capability on PWM pins
- Write-protected registers

The powerful timing module called the Quad Timer (QT) contains four identical counter/timer groups, and each 16-bit counter/timer group features:

- Ability to count up and down
- Counters will cascade
- Count modulo can be programmed
- For external clocks, the maximum count rate equals the peripheral clock/2
- For internal clocks, the maximum count rate equals the peripheral clock
- Will count once or repeatedly
- Counters can be preloaded
- Counters can share available input pins
- Each counter has a separate prescaler for
- Each counter has capture and compare capability

2.2 Intelligent High-Current Single-Silicon Switch

The 33982 is a self-protected silicon 2.0mΩ high-side switch used to replace electro-mechanical relays, fuses, and discrete devices in power management applications. The 33982 is designed for harsh environments, and includes self-recovery features. The device is suitable for loads with high in-rush current, as well as motors and all types of resistive and inductive loads.

Programming, control, and diagnostics are implemented via the Serial Peripheral Interface (SPI). A dedicated parallel input is available for alternate and Pulse Width Modulation (PWM) control of the output. SPI programmable fault-trip thresholds allow the device to be adjusted for optimal performance in the application.

- Single 2.0mW maximum high-side switch with parallel input or SPI control
- 6.0V to 27.0V operating voltage with standby currents < 5.0μA
- An external voltage input pin supplies power to the SPI circuit; if V_{DD} is lost, an internal supply provides power to a portion of the logic, ensuring the device’s limited
- SPI control of:

  - Overcurrent limit
  - Overcurrent fault blanking time
  - Output-off open load detection
  - Output on/off control
  - Watchdog time-out
  - Slew rates
  - Fault status reporting
- SPI status reporting of:
  - Overcurrent
  - Open and shorted loads
  - Overtemperature
  - Undervoltage and overvoltage shutdown
  - Fail-safe pin status
  - Program status
- Enhanced 16V reverse-polarity protection of the power supply

2.3 System-Basis Chip with High-Speed CAN Transceiver

The system-basis chip is a dedicated monolithic integrated circuit combining various functions frequently used by automotive ECM units, and provides different voltage level regulators, high voltage inputs, communication physical interface, etc. This chip incorporates:

- V1: Low-drop voltage regulator, current limitation, overtemperature detection, monitoring and reset function, and a total current capability of 200mA
- V2: Tracking function of $V_{DD}1$ regulator;control circuitry for an external bipolar ballast transistor, for high flexibility in the choice of peripheral voltage and current supply
- Four operational modes:
  - Normal
  - Stand-by
  - Stop
  - Sleep
- Low stand-by current consumption in stop and sleep modes
- High speed 1MBaud CAN physical interface
- Four external high voltage wake-up inputs, associated with HS1 Vbat switch
- 150mA output current capability for HS1 Vbat switch, allowing the driving of external switches, pull-up resistors or relays
- Vsup failure detection
- Nominal DC operating voltage from 5.5V to 27V; extended range down to 4.5V
- 40V maximum transient voltage
- Programmable software time-out and window watchdog
- Safe mode with separate outputs for Watchdog time out and Reset

- Wake-up capabilities (four wake-up inputs, programmable cyclic sense, forced wake-up, CAN interface, SPI, and stop mode overcurrent)
- Interface with MCU through the SPI

2.4 3-Phase Power MOSFET Pre-driver

The 3-Phase MOSFET pre-driver is an FET pre-driver for 3-phase motor control and similar applications. The Integrated Circuit (IC) uses Freescale's SMARTMOS technology. and contains three high-side FET pre-drivers and three low-side pre-drivers. Three bootstrap capacitors provide gate charge to the high-side FETs. The IC interfaces to a 5V or 3V MCU via six direct input control signals and an SPI port for device set-up. It is parametrically specified over a 6V to 58V supply range and over a temperature range from -40°C to +125°C.

- 42V battery capable
- Designed to operate from 6V to 58V
- 1.5A gate drive capability with protection
- Protection against reverse charge injection from Cgd and Cgs of FETs
- Includes a simple DC/DC converter for optional reduction of power dissipation
- Includes a simple charge pump to support full FET drive at low battery voltages
- Full static capability with an internal charge pump for continuous FET drive
- Crossover current protection programmable via SPI port
- Simultaneous output capability enabled via safe SPI command
- 3.3V-compatible
- Amplifier for ground current sensing
- Desaturation detection

3. Braking Systems Overview

The need for better fuel economy, simplified system assembly, more environmentally friendly systems, ease of vehicle maneuverability, and improved safety systems has resulted in new types of braking systems.

The centerpiece of the current braking systems is a hydraulic assembly under the hood of the vehicle that brings together the electronic control unit, wheel pressure modulators, pressure reservoir, and electric pump. The interaction of mechanics and electronics is key to the success of the braking system. The microcomputer, software, sensors, valves, and electric pump work together to form the basis of the system.

3.1 Electro-Hydraulic Brakes

Compared to the operation of conventional braking systems, by depressing the brake pedal with the Electro-Hydraulic Braking System (EHB), the appropriate command is transmitted electronically to the electronic controller of the hydraulic unit. This determines the optimum braking pressure and actuates the brake calipers hydraulically (see **Figure 3-1**).

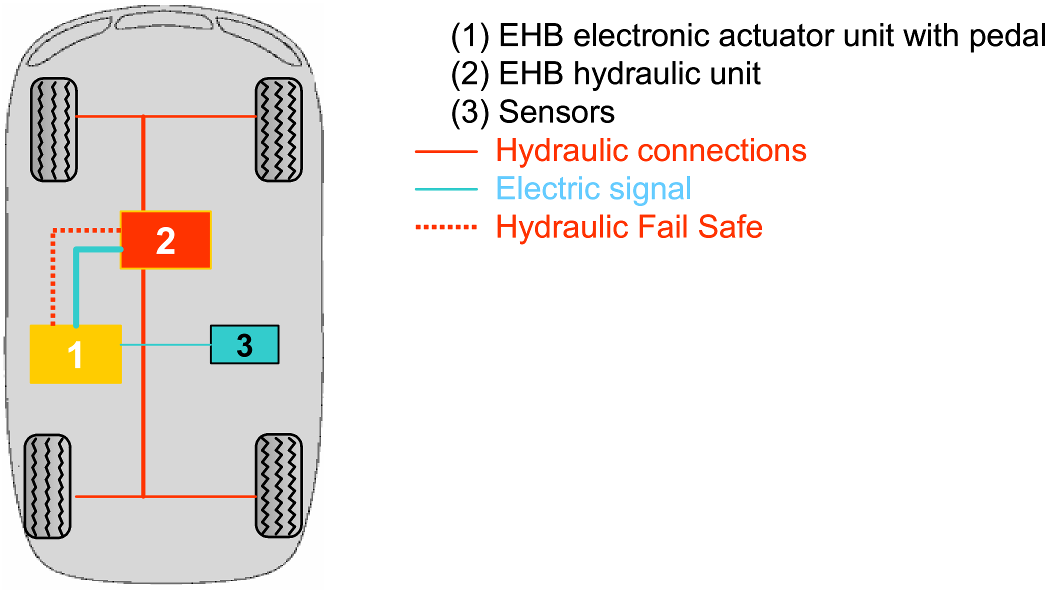

Figure 3-1. Electro-Hydraulic Braking System

3.2 Hybrid Braking

The Hybrid Braking System is the solution approach for vehicles without a fully dimensioned on-board network for dry Electro-Mechanical Brakes (EMB) on all four wheels. There can be integrated electrical parking brake functionality at the rear wheels, eliminating the need for long hydraulic lines and hand-braking cables leading toward the rear axle.

Since EMB wheel brake modules with integrated electrical parking brake function are already installed at the rear wheels, there is no longer any need for long hydraulic lines and hand-braking cables leading toward the rear axle. The front axle is operated hydraulically, as with conventional braking systems. This results in a two-circuit system with two hydraulic wheel brakes on the front axle, and two electro-mechanical wheel brake modules at the rear axle.

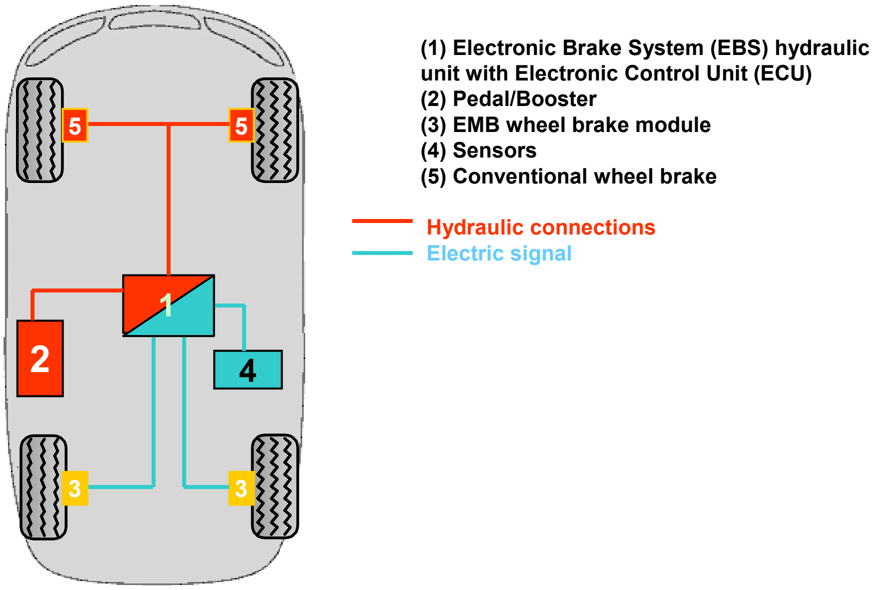

Figure 3-2. Hybrid Braking System

3.3 Electro-Mechanical Brake

The Electro-Mechanical Braking Systems represents a complete change in requirements from the previous hydraulic and electro-hydraulic braking systems. The EMB processing components must be networked using high-reliability bus protocols that ensure comprehensive fault tolerance as a major aspect of system design. The use of electric brake actuators means additional requirements, including motor control operation within a 42-volt power system and high temperatures, and a high density of electronic components.

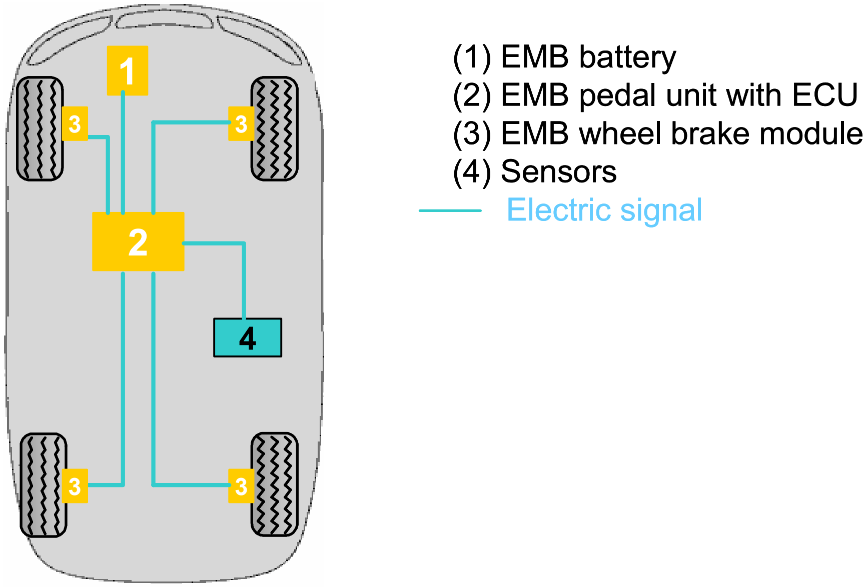

Figure 3-3. Electro-Mechanical Braking System

The EMB involves in pure brake-by-wire technology, which dispenses with brake fluids and hydraulic lines entirely. The braking force is generated directly at each wheel by high-performance electric motors, controlled by an ECU, and executed by signals from an electronic pedal module. The EMB includes all brake and stability functions, such as the Anti-lock Braking System (ABS), Electronic Brake Distribution (EBD), Traction Control System (TCS), Electronic Stability Program (ESP), Brake Assist (BA), and Adaptive Cruise Control (ACC). It is virtually noiseless, even in ABS mode.

Advantages of the EMB:

- Shorter stopping distances and optimized stability
- More comfort and safety due to adjustable pedals
- No pedal vibration in ABS mode
- Virtually silent
- Environmentally friendly with no brake fluid
- Improved crash worthiness
- Saves space and uses fewer parts
- Simple assembly
- Capable of realizing all the required braking and stability functions, such as ABS, EBD, TCS, ESP, BA, ACC, etc.
- Can easily be networked with future traffic management systems
- Additional functions, such as an electric parking brake, can easily be integrated

In addition to supporting existing communication standards, such as CAN, EMB systems require the implementation of deterministic, time-triggered communications, such as those available with FlexRay, to assist in providing the required system fault tolerance. The EMB nodes do not need to be individually fault-tolerant, but they help to provide a fail-safe operation, and rely on a high level of fault detection by the electronic components.

These system requirements must be met using high-end components at very competitive prices, to replace the established, cost-effective technology while maintaining strict adherence to the automotive qualification. Delivering the large current required to stop a full-sized SUV may cause limited adoption at first, so the first implementation will be on small car platforms.

Freescale has vast experience in developing many of the specific aspects required for the implementation of EMB systems. Freescale has a strong background in fault-tolerant communications from previous development of fail-safe microcontrollers; braking-specific modules such as the wheel speed timer; a dedicated motor control lab, and a software center that develops drivers, tools, and operating systems. Freescale is also a core team member in the FlexRay consortium and has been instrumental in the development of this protocol. With this solid foundation, Freescale has the knowledge to develop the right solutions in partnership with its customers.

4. Elements of an Electronic Braking System

An advanced braking system, such as a brake wheel node, generally contains the following elements:

a. Sensors
b. Electric motor actuator
c. Electronic control module (ECM)
d. Gear-reduction mechanism

A brake wheel node is illustrated in **Figure 4-1**.

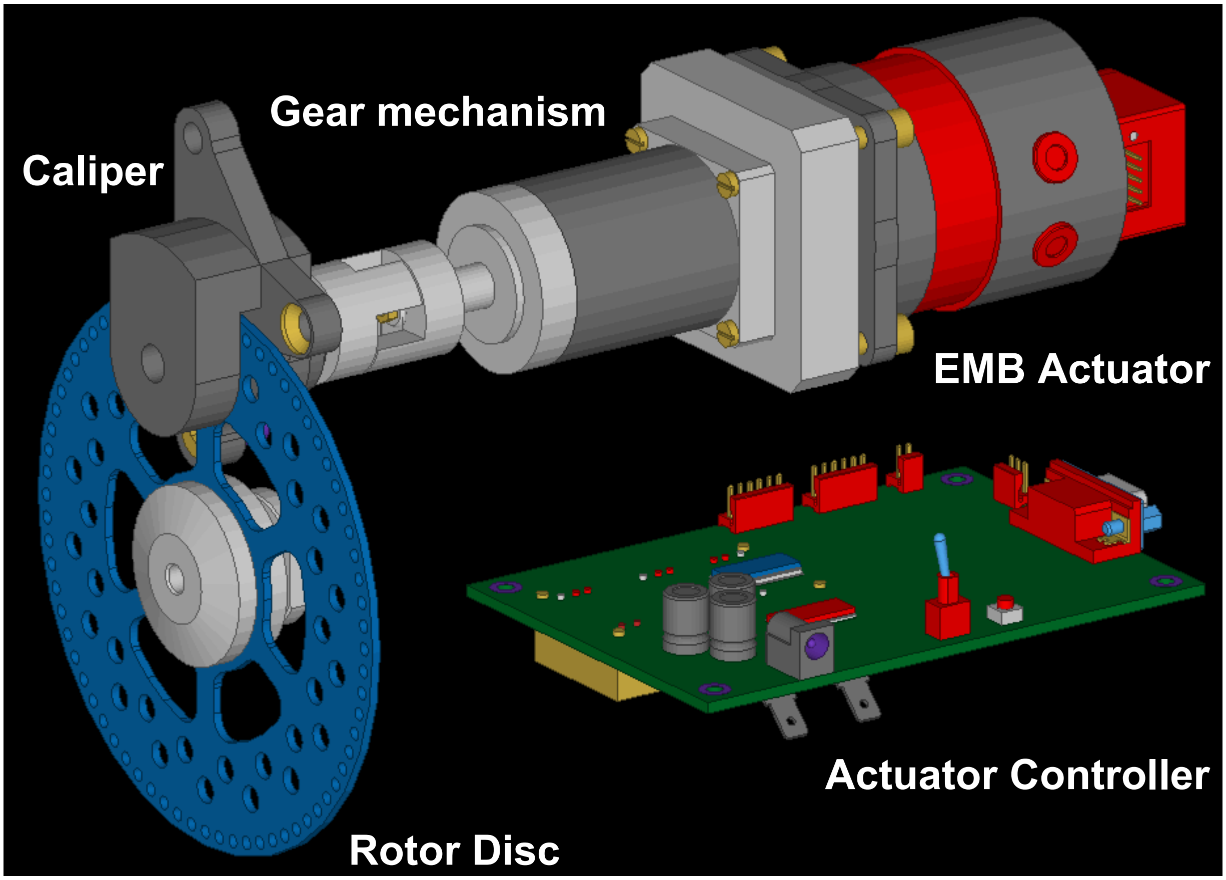

Figure 4-1. Simplified Mechanical Structure of Wheel Brake Node

4.1 Electric Motor Actuator

All electro-mechanically powered braking systems require an electric motor to convert electrical energy into mechanical energy. Ideally, this could be a rotating electrical machine where the stator is a standard 3-phase stator, like that of an induction motor, and the rotor has mounted permanent magnets (see Figure 4-2). The size and weight of the electric motor actuator are extremely important in automotive brake systems, so there is a need to minimize the package size and actuator weight.

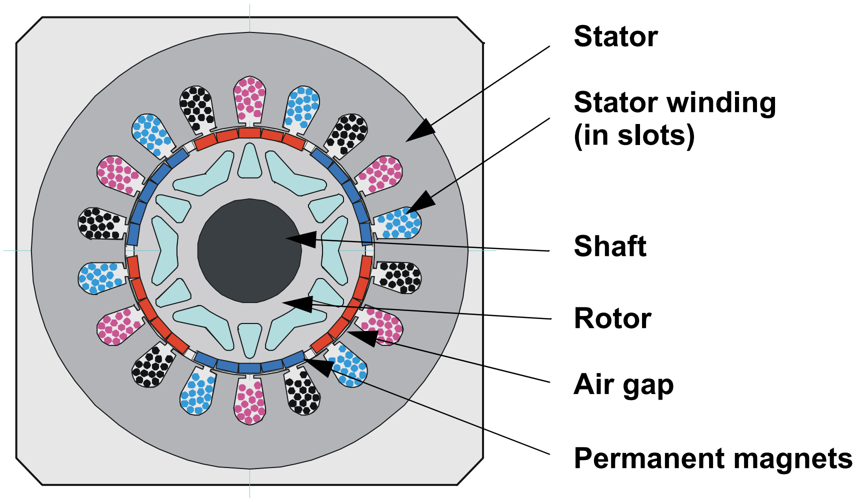

Figure 4-2. Cross Section of a Permanent Magnet AC Machine

Permanent Magnet AC (PMAC) machines provide automotive actuator designers with a unique set of features and capabilities. There are two principal classes of permanent magnet AC machines: the first, Permanent Magnet Synchronous Motors, (PMSM) are sinusoidally excited; the second, brushless DC (BLDC) motors, are trapezoidally excited machines. The primary construction difference is that while stator windings of trapezoidal PMAC machines are concentrated into a narrow-phase pole, the windings of a sinusoidal machine are typically distributed over multiple slots in order to approximate a sinusoidal distribution. These differences in construction are also reflected in their corresponding motion characteristics as well, so the first type of PMAC provides sinusoidal Back-Electromotive Force (back-EMF) generation, and the second type provides trapezoidal back-EMF.

PMSM machines enjoy unique advantages of unsurpassed efficiency and power density characteristics, which are primarily responsible for their wide appeal. On the other hand, a PMSM machine is synchronous, which certainly requires accompanying power electronics, but it also provides the basis for achieving high-quality actuator control. The torque ripple associated with sinusoidal PMAC PMSM machines is generally less than that developed in trapezoidal machines, one reason why sinusoidal motors are preferred in a high-performance motion control application such as electro-mechanical braking. This application note targets the PMSM only.

4.2 Electronic Control Module

Control of the electric motor actuator is performed by the Electronic Control Module (ECM). Increased microcontroller processing power, combined with a digital signal processing capability and a flexible set of peripherals (motor-control related), all provided on single chip, make a 56F8300 device very well suited for use as the EMB processing core. Such a core allows incorporation of intelligent and advanced braking features that can dramatically enhance driver comfort and safety.

Figure 4-3 illustrates the functional block diagram of the ECM for an electro-mechanical braking system. The main function of the ECM is to receive the sensor signals from the brake system and vehicle, process these signals, and provide appropriate voltage vectors so the motor actuator can obtain the desired torque response.

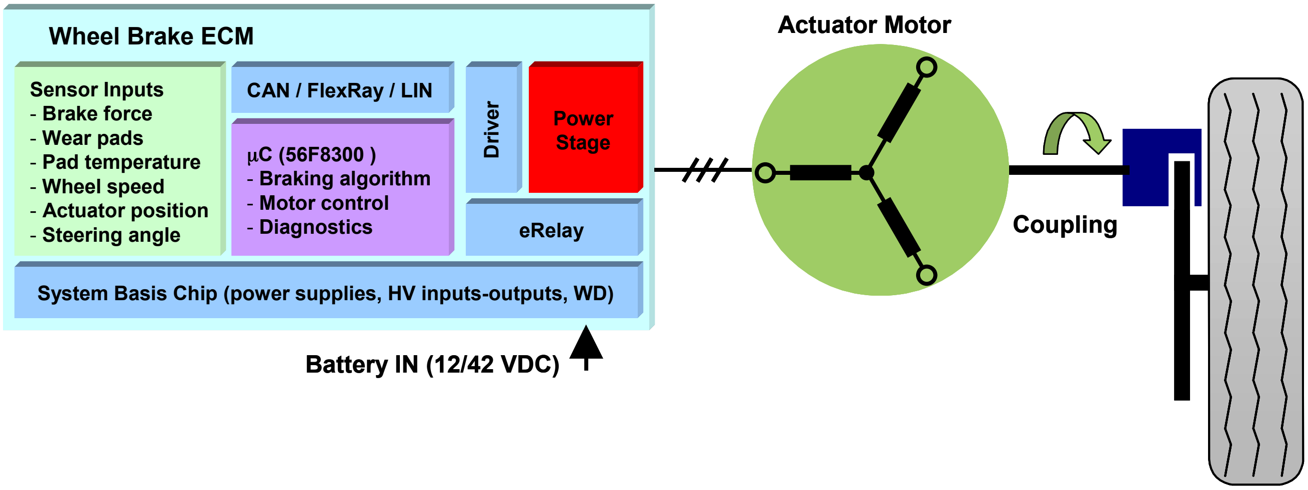

Figure 4-3. Functional Block Diagram of ECM

The ECM controller is partitioned into several blocks, including:

a. Sensor input processing
b. Microcontroller
c. Power electronics and input relay
d. System-basis chip
e. Communication

4.2.1 Sensor Input Processing

Sensor inputs to the ECM unit might come from brake force measurement, brake pads, wheel speed, or the electric motor position. These inputting signals are conditioned and appropriately filtered to be accommodated by the Analog-to-Digital converter at the microcontroller level.

In modern automobiles, the communication and wireless functions make electromagnetic emissions very critical. The circuit design must take into consideration the effect of electromagnetic noise in the vehicle. The circuit should be protected against electrostatic discharge. The protection of these sensitive ECM units from temperature, shocks, and random vibrations under adverse road condition must be considered.

4.2.2 Microcontroller

A 56F8300 device is a member of the 16-bit 56800E core-based controller family. Its processing power, combined with the functionality of a microcontroller and a flexible set of peripherals, creates an extremely cost-effective solution on a single chip. Because of its low cost, configuration flexibility, and compact program code, a 56F8300 device is well suited for intensive automotive applications.

The 56800E core is based on a Harvard-style architecture consisting of three execution units operating in parallel, allowing as many as six operations per instruction cycle. The MCU-style programming model and optimized instruction set allow straightforward generation of efficient, compact 16-bit control code. The instruction set is also highly efficient for C/C++ compilers, enabling rapid development of optimized control applications.

4.2.3 Power Electronics and Input Relay

A power inverter is used to apply the desired voltage across the motor phases. The DC-to-AC inverter topology most widely used today for exciting sinusoidal and trapezoidal PMAC machines is the 3-phase full bridge inverter. Due to lower automotive voltages, power MOSFETs are commonly used as power electronic switches.

Advances in power electronic control have made it possible to control electric motors able to provide significant torque with minimum loss in the power unit. The automotive power MOSFETs can provide hundreds of amperes of current, thus making it possible to use them even for electric braking in larger vehicles. **Figure 4-4** shows a 3-phase inverter used for a PMSM machine.

A power MOSFET driver device is used to amplify digital PWM signals from the microcontroller. The current and voltage sensors are used to sense the phase currents and voltage, thus providing feedback for the motor control part of the electro-mechanical brake software. Either a Hall-effect sensor or a shunt resistor is used to sense the electric motor actuator current. The DC-link capacitors are used to filter DCBus voltage. Either mechanical or electronic relays may be used as disconnecting devices to switch off the electronic system from the vehicle battery bus.

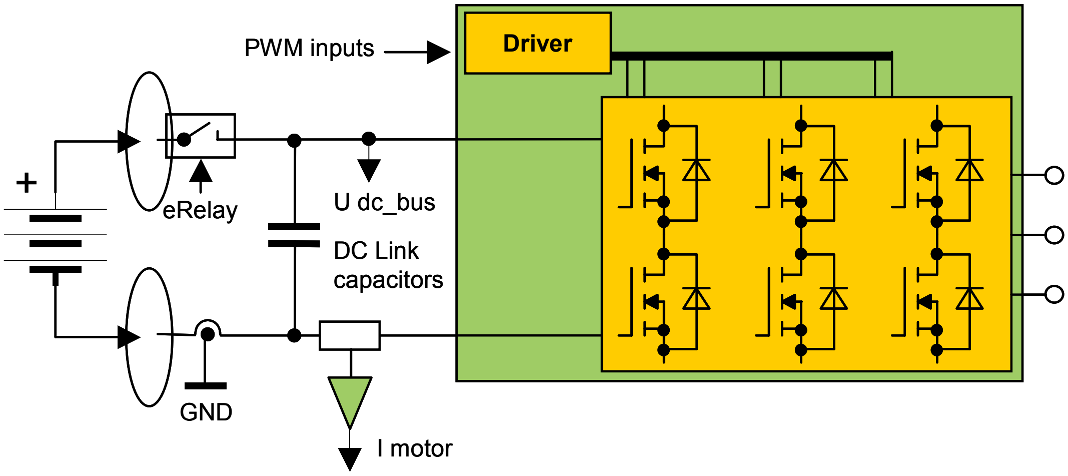

Figure 4-4. 3-Phase DC-to-AC Inverter

The electronic relay (eRelay) is a self-protected silicon switch with very low on-state resistance, e.g., 2mΩ. It is configured as a high-side switch used to replace electro-mechanical relays, fuses, and discrete devices in power management applications. This silicon device is designed for harsh environments, and it includes self-recovery features. The device is suitable for loads with high in-rush current, as well as motors and all types of resistive and inductive loads. The different implemented features of this device, such as diagnostics, control, etc. are programable and accessible via Serial Peripheral Interface (SPI).

For maximum device protection, the device has several programmable overcurrent low and high detection levels. If the load current level ever reaches the selected overcurrent low detect level, and the overcurrent condition exceeds the programmed overcurrent time period, the device will latch the output off.

4.2.4 System-Basis Chip

The system-basis chip is a dedicated monolithic integrated circuit, combining various functions frequently used by automotive ECM units, providing different voltage level regulators, high voltage inputs, communication physical interfaces, etc.

This device is supplied from an automobile battery line through its input power pin, which is designed for sustaining standard automotive voltage conditions such as load dump. When the supply voltage falls significantly, this dedicated system-basis chip is able to detect it and store the information into the SPI register bit. The device usually incorporates a battery-powered early warning function, which can provide the maskable interrupt to the microcontroller for evaluation. The device includes voltage regulator capabilities to provide different voltage levels for on-board power supplies.

A dedicated device such as the SBC has several modes of operation, including Stand by, Normal, Stop, and Sleep Modes. Special modes and configurations are possible for debugging and programming the microcontroller's Flash memory.

There are several wake-up capabilities available to this device when it is in Sleep or Stop Mode. When wake-up has occurred, the wake event is stored in internal device registers, and the voltage regulator is activated so that the microcontroller is powered back on to bring the ECM unit into fully active operation. The wake-up lines are dedicated to sensing external event states and, on any change, will wake up the microcontroller. The device can even wake up from a CAN message if CAN wake-up has been enabled.

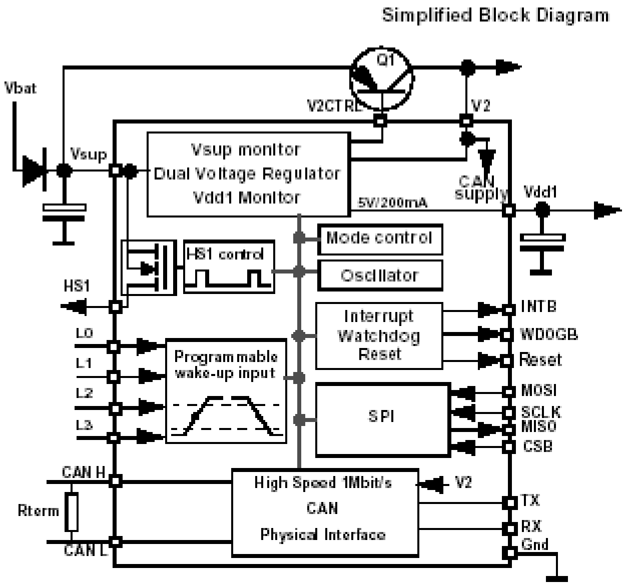

Figure 4-5. Generic Block Diagram of a System-Basis Chip

4.3 Sensors

There are individual sensors which are definitely needed to operate electro-mechanical braking systems, but the ECM unit could further benefit from additional new in-car technologies, which allow receipt of information from the engine, transmission, steering angle, and other subsystem information, via in-car communication lines.

4.3.1 Actuator Position Sensing

Since the PMSM machines specifically discussed here are synchronous machines, maximum torque can only be produced when excitation is precisely synchronized with the instantaneous rotor position. The most straightforward method of providing the required rotor position information is to mount an absolute angular position sensor on the PMSM rotor shaft. Alternatively, the accurate position can be derived indirectly from the motor's voltage and current waveforms using more sophisticated sensorless control algorithms.

Encoder

Whenever mechanical rotary motions of electric machines must be monitored, the encoder is the most important interface between the mechanics and the controller unit. Encoders transform rotary movements of the actuator rotor into a sequence of electrical pulses. A rotary encoder can differentiate a number of discrete positions per revolution. The number of segments determines the resolution of the movement, and thus the accuracy of the position; this number is called its points per revolution. The speed of an encoder is in counts-per-second. Many incremental encoders also have a feature called the index pulse. In rotary encoders, an index pulse occurs once per encoder revolution. It is used to establish an absolute mechanical reference position within one encoder count of the 360° encoder rotation.

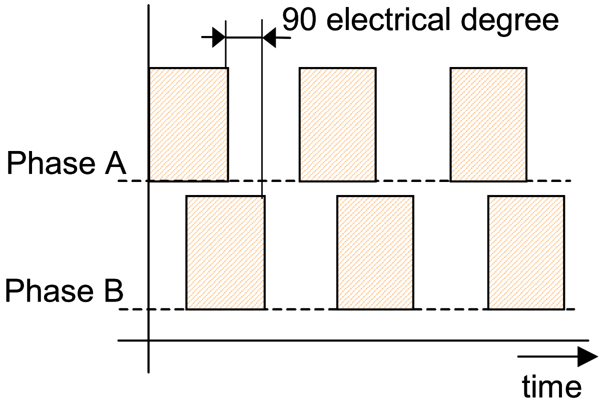

Figure 4-6. Quadrature Encoder Signals

Quadrature encoders are a special type of incremental encoder with at least two output signals, generally called Phase A and Phase B. As seen in **Figure 4-6**, Phase B is offset 90 degrees from Phase A. The addition of a second channel provides direction information in the feedback signal. This signal, leading or lagging by 90 electrical degrees, guarantees the exact determination of the direction of rotation at all times. The ability to detect direction is critical if the encoder rotation stops on a pulse edge. Without this ability to decode direction, the counter may count each transition through the rising edge of the signal and lose the position. Another benefit of the quadrature signal scheme is the ability to electronically multiply the counts during one encoder cycle. In the Times-1 Mode, all counts are generated on the rising edges of Phase A. In the Times-2 Mode, both the rising and falling edges of Phase A are used to generate counts. In the Times-4 Mode, the rising and falling edges of Phase A and Phase B are used to generate counts. This increases the resolution by a factor of four. For encoders with sine wave output, the channels may be interpolated for very high resolution.

Resolver

Resolver position sensors resemble small motors and are essentially rotary transformers, so the coefficient of the coupling between rotor and stator varies with shaft angle. A resolver is based on the concept of encoding the shaft angle into sine and cosine signals.

The majority of resolvers used today are referred to as hollow shaft resolvers. They transfer energy from stator to rotor by means of an auxiliary rotary transformer (see Figure 4-7). The resolver rotor is directly mounted on the motor shaft and the resolver stator is fixed to the motor shield. Note that these resolver parts must be concentrically fastened with the longitudinal axis of the motor shaft. Ordinarily, a concentricity of resolver rotor against stator of up to 0.05 mm is needed; otherwise, resolver parameters might deteriorate. Thanks to the absence of bearings and brushes, the life-cycle of hollow shaft resolvers is practically unlimited.

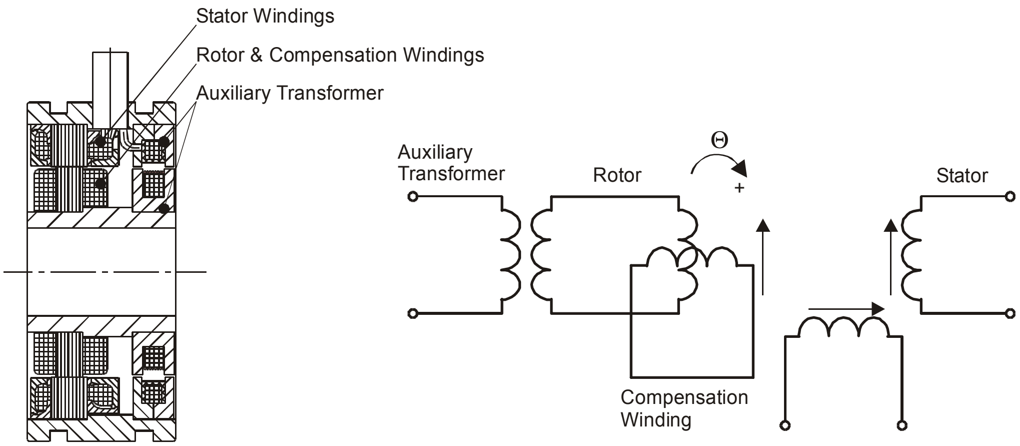

Figure 4-7. Block Scheme of Hollow Shaft Resolver

Hall-Effect Sensor:

This is perhaps the simplest electronic shaft position transducer used for generating commutation pulses synchronized with rotor position. For 3-phase PMAC motors, three Hall-effect switches are spaced at 60 or 120 electrical and mounted on the stator frame.

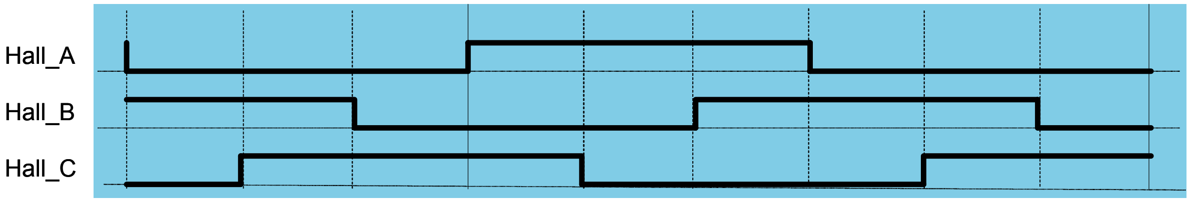

Figure 4-8. Output of Hall-Effect Switches

Either a separate trigger magnet with the correct pole spacing is mounted on the shaft in close proximity to the Hall-effect switches, or the Hall-effect switches can be mounted close to the rotor magnets, where they are energized by leakage flux at the appropriate rotor position. Another option is to have a sensor arrangement consisting of a segmented disk (eight segments) and three Hall sensors mounted 120° from each other. The segmented disc is mounted on the motor shaft to precisely match the rotor commutation positions. The number of rotor poles defines the ratio between the mechanical revolution and the electrical period.

Absolute Position Encoder

Digital information on the rotor position can be obtained by special arrangement of a high resolution digital encoder providing relative position information in combination with a low-resolution commutation sensor. This commutation sensor is attached to the digital encoder and provides synchronization to obtain an absolute rotor position. This sensor allows use of the commutation information for both PMAC machines (brushless DC motors as well as sinusoidal PMAC). The Hall-effect sensor that generates the logic waves used so commonly in trapezoidal PMAC machines (BLDC) works on a similar principle.

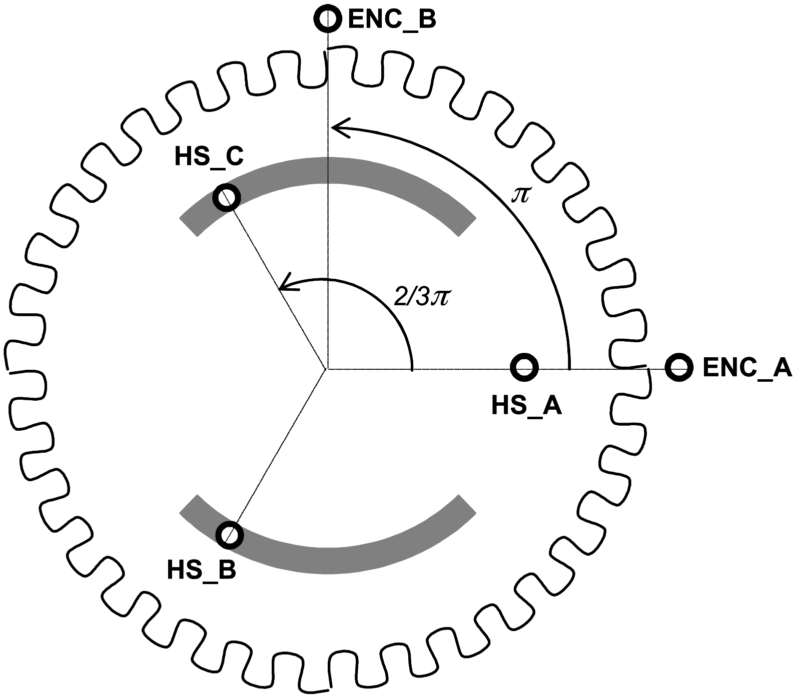

Figure 4-9. Combination of Telative Encoder and Absolute Commutation Sensor

There are two signals (ENC_A, ENC_B) provided by the digital encoder sensor. These digital outputs create two high stream pulse trains, which are intentionally shifted by 90° electrical from each other. Since the digital sensor provides relative position information only, it's necessary to correctly initialize its associated position counter, then perform synchronization consecutively. Initialization information is provided by other sensor signals (HS_A, HS_B, HS_C). The absolute zero position of the rotor is aligned to these sensor signals (HS_A, HS_B, HS_C).

By combining these two approaches in position information generation, and with dedicated peripherals for the 56F8300 microcontroller, it is possible to achieve high resolution of the rotor angular position and know the absolute position.

4.3.2 Wheel Speed

In most cases, there are a number of different speed sensor types available. The selection may be determined by the familiarity of the system designer with a sensor, though on other hand, the output from one sensor can be used for several applications (ABS, ASR, EMB, etc.), and the individual requirements of each application may eventually determine the sensor type to be used.

In wheel speed measurement, speed sensors are attached to all four wheels so that the slip differential between the wheels can be measured. This information can be distributed into applications such as ABS, ASR, four wheel steering, and electro-mechanical braking.

Due to their simplicity and proven reliability, variable reluctance wheel speed sensors are typically used in ABS applications. In conjuction with the exciter rings, this type of sensor produces a sinusoidal output that is directly proportional in frequency and amplitude to the angular velocity of the sensed wheel. A limitation of this measurement technology is the very low speed output, which tends to be too low to be sensed reliably by the control unit, even given the electrically noisy environment typical of vehicles. This can result in errors and cumulative inaccuracies, but normally the antilock function is inhibited at very low speed. Sharing information about wheel speed with an electro-mechanical braking system can be unreliable. A variety of active sensor types, including Hall-effect and magneto-resistive, can be used in an EMB application to provide a very low speed-sensing capability.

5. EMB controller

It is assumed for this application note that a desired braking force is being requested by the vehicle dynamics controller and the focus is on delivery of this command from the wheel ECM unit. The control problem for electro-mechanical brake actuators is introduced in the following sections. An EMB controller is required to drive the actuator and appropriately respond to brake commands from the vehicle dynamics controller.

5.1 Actuator control

Although basic volt-per-hertz control can be used in some particular cases, the majority of PMSM actuator applications incorporate closed-loop control of the motor phase currents in order to achieve high performance torque control. The motor actuator of an electro-mechanical braking system generally requires only torque control, but maintaining accurate information about the PMSM rotor's absolute position is equally important.

Vector Control is an elegant method of controlling the PM synchronous motor, where field-oriented theory is used to control space vectors of magnetic flux, current, and voltage. It is possible to set up the coordinate system to decompose the vectors into electro-magnetic field generation and torque generation parts. The structure of the motor controller (Vector Control controller) is then almost the same as for a separately excited DC motor, which simplifies the control of a PM synchronous motor. This Vector Control technique was developed especially to achieve an excellent dynamic performance of PM synchronous motors.

As explained in **Figure 5-1**, it has been decided to use a widely used current control with an inner position closed loop. In this method, the decomposition of the field generation and the torque generation parts of the stator current allows separate control of the magnetic flux and the torque. In order to do so, it's necessary to set up a rotary coordinate system connected to the rotor magnetic field, generally called a "d-q reference coordinate system". Very high CPU performance is needed to perform the transformation between rotary to stationary coordinate systems. Therefore, a Freescale 56F8300 hybrid controller, accompanied by dedicated motor control peripherals, is well-suited to vector control algorithms. All transformations needed for vector control will be briefly described in the following sections.

5.1.1 Mathematical Model of Actuator Motor

For a description of the Permanent Magnet Synchronous Motor (PMSM), a symmetrical 3-phase smooth-air-gap machine with sinusoidally distributed windings is considered. The voltage equations of stator in the instantaneous form can be expressed as:

$$u_{SA} = R_S i_{SA} + \frac{d}{dt}\psi_{SA} \qquad \text{EQ. 5-1}$$

$$u_{SB} = R_S i_{SB} + \frac{d}{dt}\psi_{SB} \qquad \text{EQ. 5-2}$$

$$u_{SC} = R_S i_{SC} + \frac{d}{dt}\psi_{SC} \qquad \text{EQ. 5-3}$$

where:

u_{SA}, u_{SB} and u_{SC} are the instantaneous values of stator voltages

i_{SA}, i_{SB} and i_{SC} are the instantaneous values of stator currents

ψ_{SA}, ψ_{SB}, ψ_{SC} are instantaneous values of stator flux linkages in phase SA, SB and SC

Due to the large number of equations in the instantaneous form of **EQ. 5-1**, **EQ. 5-2** and **EQ. 5-3**, it is more practical to rewrite the instantaneous equations using two-axis theory (Clark transformation). Then the PM synchronous motor can be expressed as:

$$u_{S\alpha} = R_S i_{S\alpha} + \frac{d}{dt}\Psi_{S\alpha} \qquad \text{EQ. 5-4}$$

$$u_{S\beta} = R_S i_{S\beta} + \frac{d}{dt}\Psi_{S\beta} \qquad \text{EQ. 5-5}$$

$$\Psi_{S\alpha} = L_S i_{S\alpha} + \Psi_M \cos(\Theta_r) \qquad \text{EQ. 5-6}$$

$$\Psi_{S\beta} = L_S i_{S\beta} + \Psi_M \sin(\Theta_r) \qquad \text{EQ. 5-7}$$

$$\frac{d\omega}{dt} = \frac{p}{J}\left[\frac{3}{2}p(\Psi_{S\alpha} i_{S\beta} - \Psi_{S\beta} i_{S\alpha}) - T_L\right] \qquad \text{EQ. 5-8}$$

where:

α,β = Stator orthogonal coordinate system

$u_{S\alpha,\beta}$ = Stator voltages

$i_{S\alpha,\beta}$ = Stator currents

$\Psi_{S\alpha,\beta}$ = Stator magnetic fluxes

Ψ_M = Rotor magnetic flux

R_S = Stator phase resistance

L_S = Stator phase inductance

ω / ω_F = Electrical rotor speed / fields speed

p = Number of poles per phase

J = Inertia

T_L = Load torque

Θ_r = Rotor position in α,β coordinate system

EQ. 5-4 through **EQ. 5-8** represents the model of a PM synchronous motor in the stationary frame α, β fixed to the stator. The main idea of the vector control is to decompose the vectors into magnetic field generation and torque generation parts. In order to do so, it is necessary to set up a rotary coordinate system attached to the rotor magnetic field, generally called a "d-q coordinate system" (Park transformation). Thus, **EQ. 5-4** through **EQ. 5-8** can be rewritten as:

$$u_{Sd} = R_S i_{Sd} + \frac{d}{dt}\Psi_{Sd} - \omega_F \Psi_{Sq} \qquad \text{EQ. 5-9}$$

$$u_{Sq} = R_S i_{Sq} + \frac{d}{dt}\Psi_{Sq} + \omega_F \Psi_{Sd} \qquad \text{EQ. 5-10}$$

$$\Psi_{Sd} = L_S i_{Sd} + \Psi_M \qquad \text{EQ. 5-11}$$

$$\Psi_{Sq} = L_S i_{Sq} \qquad \text{EQ. 5-12}$$

$$\frac{d\omega}{dt} = \frac{p}{J}\left[\frac{3}{2}p(\Psi_{Sd} i_{Sq} - \Psi_{Sq} i_{Sd}) - T_L\right] \qquad \text{EQ. 5-13}$$

The expression of electromagnetic torque is calculated as follows:

$$t_e = \frac{3}{2}p(\Psi_{Sd} i_{Sq} - \Psi_{Sq} i_{Sd}) \qquad \text{EQ. 5-14}$$

Model of PMSM in Rotating Reference Frame

In order to produce the largest torque, an optimal operation is achieved by stator current control which ensures that the stator current space vector contains only a quadrature component,below the base speed and the d-component is maintained as $i_{sd}=0$. This is achieved in the reference frame fixed to the rotor. Employing equations **EQ. 5-9** through **EQ. 5-12** create a model of PMSM expressed in rotating reference frame as follows:

$$u_{Sd} = R_S i_{Sd} + L_S \frac{d}{dt} i_{Sd} - L_S \omega_F i_{Sq} \qquad \text{EQ. 5-15}$$

$$u_{Sq} = R_S i_{Sq} + L_S \frac{d}{dt} i_{Sq} + L_S \omega_F i_{Sq} + \omega_F \Psi_M \qquad \text{EQ. 5-16}$$

EQ. 5-14 demonstrates that the torque is dependent and can be directly controlled by the current i_{sq} only. The expression of electromagnetic torque is similar to the expression for electromagnetic torque produced by a separately excited DC motor, calculated as follows:

$$t_e = \frac{3}{2}p(\Psi_M i_{Sq}) \qquad \text{EQ. 5-17}$$

This analogy is a fundamental basis for various forms of vector control.

5.1.2 Block Diagram of Current Vector Control

In general, motor behavior is controlled directly by measuring the motor's position using an appropriate position transducer, and torque is controlled indirectly by suitable control of the motor phase currents. **Figure 5-1** shows the basic structure of current vector control of the PM synchronous motor. To perform current vector control, it is necessary to follow these steps:

- Measure the motor quantities (phase voltages and currents)
- Measure the rotor position and speed
- Perform the transformation from a 3-phase to a 2-phase system (α,β) using a Clarke transformation
- Transform stator currents to the d-q coordinate system using a Park transformation
- The stator current torque-producing (i_{sq}) and flux-producing (i_{sd}) components are separately controlled
- The output stator voltage space vector is calculated using the decoupling block
- The stator voltage space vector is transformed by an inverse Park transformation back from the d-q coordinate system to the 2-phase system fixed with the stator
- Using the space vector modulation, the output 3-phase voltage is generated

Figure 5-1 also shows the proportioning of tasks between software code modules and the dedicated motor control peripherals for the 56F8300 hybrid controller.

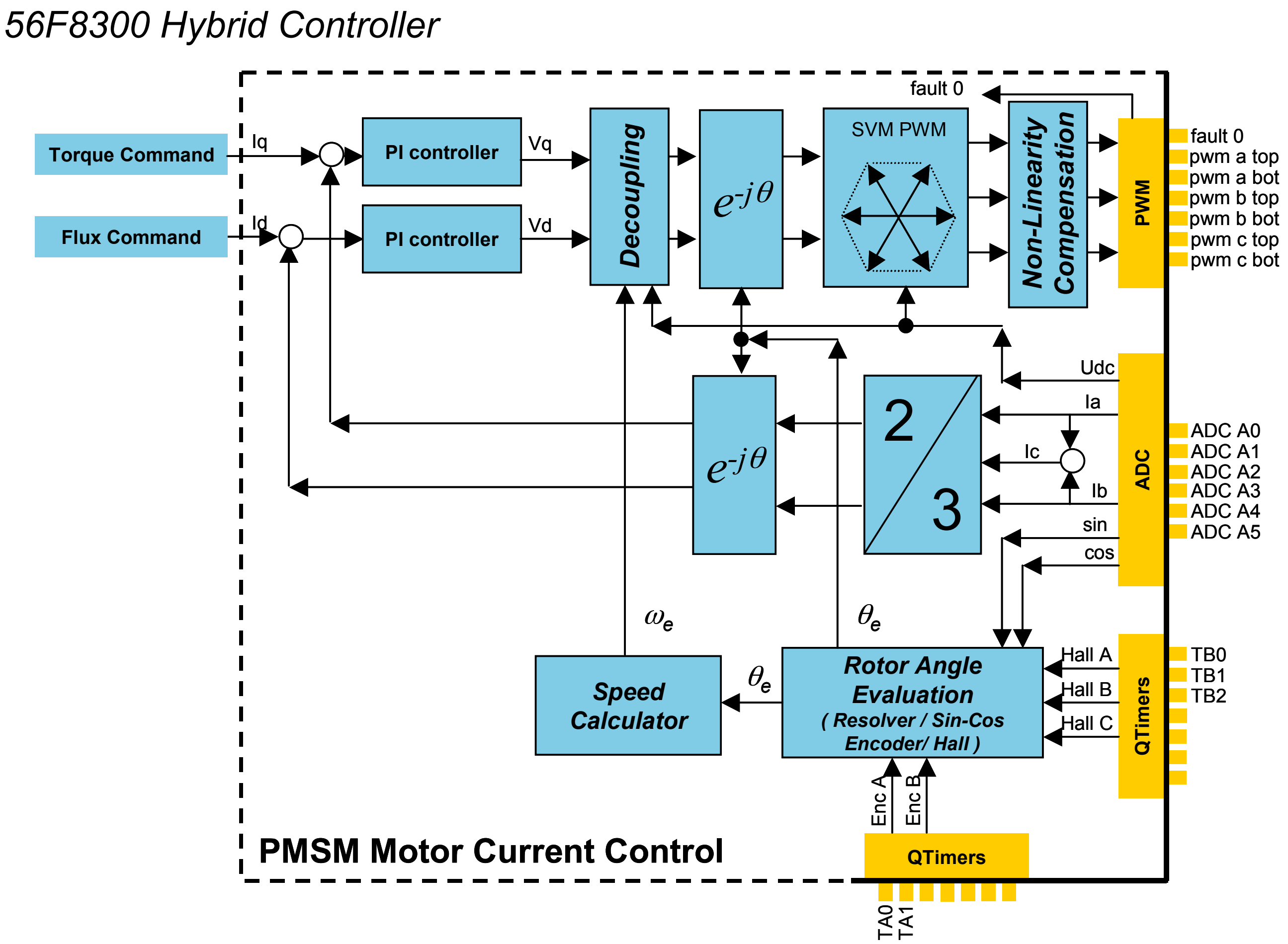

Figure 5-1. Block Diagram of Vector Current Control for the PMSM

5.1.3 Vector Transformations

The introduction of reference frames in the analysis of electrical machines is not only useful in analysis, but has also provided a powerful tool for the implementation of sophisticated control techniques. The software for a 56F8300 hybrid controller uses mathematical vector transformations, known as Park and Clarke Transformations, that ensure that the 3-phase set of currents applied to the motor is synchronized to the actual motor shaft rotation under all operating conditions. This synchronism ensures that the motor always produces the optimal torque per ampere; i.e., operates at optimal efficiency. The vector rotations require real-time calculation of the measured rotor angle sine and cosine, plus a number of multiply-and-accumulate operations.

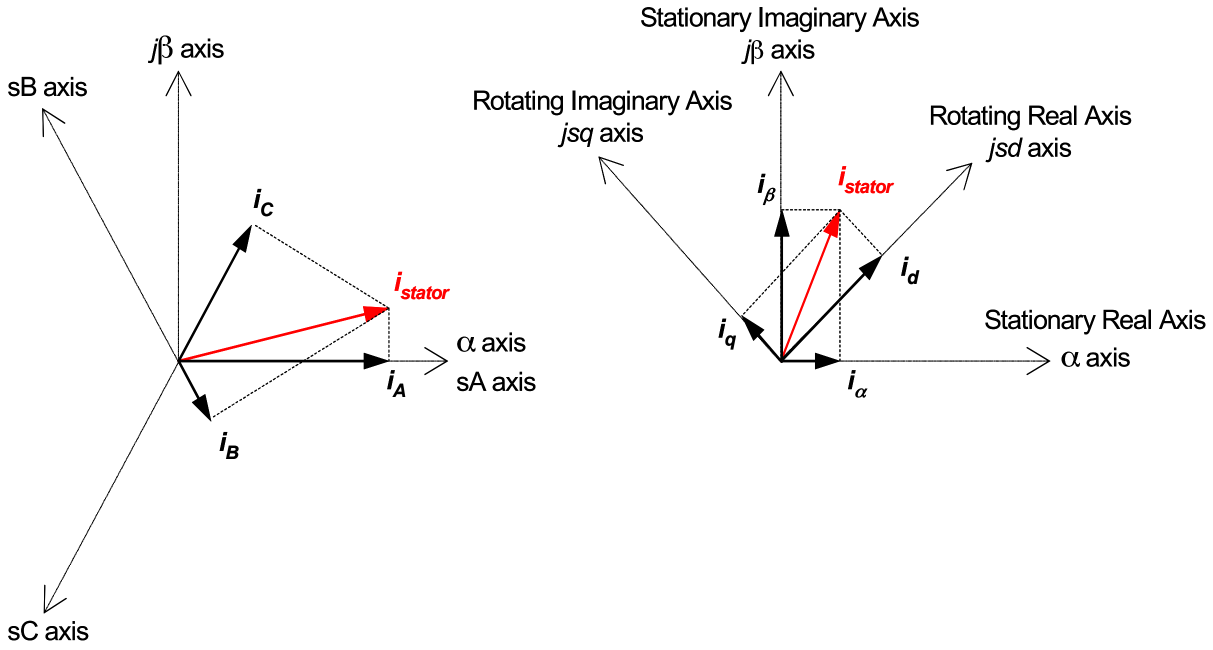

Figure 5-2. Current Space Vector Transformation into a) 2-Phase Stationary Reference Frame and b) Rotating ReferenceFrame

3-phase to 2-phase Transformation (Clarke)

A 3-phase AC machine is conventionally modeled using phase-variable notation; however, the phase quantities are not independent variables. As a result of this redundancy in the phase variable representation, it is possible to transform the system to an equivalent 2-phase representation.

The forward Clarke transformation converts a 3-phase system (a,b,c) to a 2-phase co-ordinate system (α,β). Figure 5-2 shows a graphical construction of the space vector and projection of the space vector to the quadrature-phase components α,β.

If we assume $i_{sa} + i_{sb} + i_{sc} = 0$, the quadrature-phase (2-phase system) components can be expressed utilizing only two phases of the 3-phase system:

$$\begin{aligned} i_{s\alpha} &= i_{sa} \\ i_{s\beta} &= \frac{1}{\sqrt{3}} i_{sa} + \frac{2}{\sqrt{3}} i_{sb} \end{aligned} \qquad \text{EQ. 5-18}$$

The inverse Clarke transformation reverts from a 2-phase (α,β) to a 3-phase i_{sa}, i_{sb}, i_{sc} system, and results from the following equations:

$$\begin{aligned} i_{sa} &= i_{s\alpha} \\ i_{sb} &= -\frac{1}{2} i_{s\alpha} + \frac{\sqrt{3}}{2} i_{s\beta} \\ i_{sc} &= -\frac{1}{2} i_{s\alpha} - \frac{\sqrt{3}}{2} i_{s\beta} \end{aligned} \qquad \text{EQ. 5-19}$$

Vector Rotation (Park Transformation)

The components $i_{s\alpha}$ and $i_{s\beta}$, calculated with a Clarke transformation, are attached to the stator reference frame α, β. In vector control, it is necessary to have all quantities expressed in the same reference frame. The stator reference frame is not suitable for the control process. The space vector i_s rotates at a rate equal to the angular frequency of the phase currents. The components $i_{s\alpha}$ and $i_{s\beta}$ depend on time and speed. Transformation of these components, from the stator reference frame to the d-q reference rotating frame, is at the same speed as the angular frequency of the phase currents, so the i_{sd} and i_{sq} components do not depend on time and speed. A new reference frame is defined in which the axes are made to rotate at the same rate as angular frequency of the phase quantities, stationary current, voltage and flux linkage space vector result.

Consider applying the vector rotation through an angle, θ_e, to the current space vector. The current space vector in this new reference frame is given by its components, i_{sd} and i_{sq}, and determined by the following equations:

$$\begin{aligned} i_{sd} &= i_{s\alpha}\cos\theta_e + i_{s\beta}\sin\theta_e \\ i_{sq} &= -i_{s\alpha}\sin\theta_e + i_{s\beta}\cos\theta_e \end{aligned} \qquad \text{EQ. 5-20}$$

The component i_{sd} is called the direct axis component (the flux-producing component) and i_{sq} is called the quadrature axis component (the torque-producing component).

The inverse vector rotation, to transform from a rotating to a stationary reference frame, is usually called the inverse Park transformation. The rotation d-q to α,β to stationary coordinate system is calculated by the following equations:

$$\begin{aligned} i_{s\alpha} &= i_{sd}\cos\theta_e - i_{sq}\sin\theta_e \\ i_{s\beta} &= i_{sd}\sin\theta_e + i_{sq}\cos\theta_e \end{aligned} \qquad \text{EQ. 5-21}$$

5.1.4 Decoupling Circuit

For purposes of the rotor flux-oriented vector control, the direct-axis stator current i_{sd} (the rotor flux-producing component) and the quadrature-axis stator current i_{sq} (the torque-producing component) must be controlled independently. However, the equations of the stator voltage components are coupled. The direct axis component u_{sd} also depends on i_{sq}, and the quadrature axis component u_{sq} also depends on i_{sd}. The stator voltage components u_{sd} and u_{sq} cannot be considered as decoupled control variables for the rotor flux and electromagnetic torque. The stator currents i_{sd} and i_{sq} can only be independently controlled (decoupled control) if the stator voltage equations are decoupled, then indirectly controlled by controlling the terminal voltages of the induction motor.

The equations of the stator voltage components in the d-q coordinate system **EQ. 5-15** and **EQ. 5-16** can be reformulated and separated into two components: linear components $u_{sd}^{lin}, u_{sq}^{lin}$ and decoupling components $u_{sd}^{decouple}, u_{sq}^{decouple}$. The equations are decoupled as follows:

$$u_{Sd} = u_{sd}^{lin} + u_{sd}^{decouple} = \left[R_S i_{Sd} + L_S \frac{d}{dt} i_{Sd}\right] - [L_S \omega_F i_{Sq}] \qquad \text{EQ. 5-22}$$

$$u_{Sq} = u_{sq}^{lin} + u_{sq}^{decouple} = \left[R_S i_{Sq} + L_S \frac{d}{dt} i_{Sq}\right] + [L_S \omega_F i_{Sq} + \omega_F \Psi_M] \qquad \text{EQ. 5-23}$$

The voltage components $u_{sd}^{lin}, u_{sq}^{lin}$ are the outputs of the current controllers which control the i_{sd} and i_{sq} components. They are added to the decoupling voltage components $u_{sd}^{decouple}$, $u_{sq}^{decouple}$, creating direct and quadrature components of the terminal output voltage. This means that the voltage on the outputs of the current controllers is:

$$u_{sd}^{lin} = R_S i_{Sd} + L_S \frac{d}{dt} i_{Sd} \qquad \text{EQ. 5-24}$$

$$u_{sq}^{lin} = R_S i_{Sq} + L_S \frac{d}{dt} i_{Sq} \qquad \text{EQ. 5-25}$$

And the decoupling components are:

$$u_{sd}^{decouple} = -L_S \omega_F i_{Sq} \qquad \text{EQ. 5-26}$$

$$u_{sq}^{decouple} = L_S \omega_F i_{Sq} + \omega_F \Psi_M \qquad \text{EQ. 5-27}$$

As shown, the decoupling algorithm transforms the non-linear motor model to linear equations, which can be controlled by general PI or PID controllers instead of complicated controllers.

5.1.5 Space Vector Modulation and Inverter Non-Linearity Compensation

Space Vector Modulation (SVM) can directly transform the stator voltage vectors from an α,β coordinate system to Pulse Width Modulation (PWM) signals (duty cycle values).

The standard technique of the output voltage generation uses an inverse Clarke transformation to obtain 3-phase values. Using the phase voltage values, the duty cycles needed to control the power stage switches are then calculated. Although this technique gives good results, SVM is more straightforward, although valid only for transformation from the α,β coordinate system.

The basic principle of the standard SVM technique can be explained with the help of the power stage schematic diagram depicted in **Figure 4-4** Regarding the 3-phase power stage configuration, eight possible switching states (vectors) are feasible. They are given by combinations of the corresponding power switches.The graphical representation of all combinations is the hexagon shown in **Figure 5-3**. There are six non-zero vectors, U_0, U_{60}, U_{120}, U_{180}, U_{240}, U_{300}, and two zero vectors, O_{000} and O_{111}, defined in α,β coordinates.

The combination of ON/OFF states of the power stage switches for each voltage vector is coded in **Figure 5-3** by the three-digit number in parenthesis. Each digit represents one phase. For each phase, a value of one means that the upper switch is ON and the bottom switch is OFF. A value of zero means that the upper switch is OFF and the bottom switch is ON.

Figure 5-3. Basic Space Vectors and Voltage Vector Projection

SVM is a technique used as a direct bridge between vector control (voltage space vector) and PWM. The SVM technique consists of several steps:

1. Sector identification
2. Space voltage vector decomposition into directions of sector base vectors U_x, $U_{x\pm60}$
3. PWM duty cycle calculation

The principle of SVM is the application of the voltage vectors U_{XXX} and O_{XXX} for certain instances in such a way that the “mean vector” of the PWM period T_{PWM} is equal to the desired voltage vector.

This method gives the greatest variability arranging of the zero and non-zero vectors during the PWM period. One can arrange these vectors to lower switching losses; another approach yields a different result, such as center-aligned PWM, edge-aligned PWM, minimal switching, etc.

One should notice that there is inherited non-linearity caused by the non-ideal characteristics of the power switches. The most significant non-linearity is introduced by the necessary blanking time (so called dead time) to avoid short circuiting the DC link during the commutations. This dead time, added to the device's inherent turn-on and turn-off delay times, introduces a magnitude and phase error in the output voltage. The second main non-linear effect is due to the finite voltage drop across the switch during the on-state. This introduces an additional error in the magnitude of the output voltage, although somewhat smaller, which must be compensated.

Regardless of the method used, all dead time compensation techniques are based on the polarity of the current, so current detection becomes an important issue. This is especially true around the zero-crossings, where an accurate measurement is needed to correctly compensate for the dead time. The voltage drop across the power switches can be compensated by modeling it, using as a constant voltage pulse either a linear model of power switches or look-up tables.

5.1.6 PI Controller

PI controllers are universally known because of their flexibility and relatively easy tuning. This section outlines the PI representation in continuous and discrete time domains, which is essential for digital implementation on hybrid controllers. A numerical routine is then performed, converting the continuous form onto discrete time representation for the PI controller.

The function described here calculates the Proportional-Integral (PI) algorithm according to equations represented in continuous time domain, as follows:

$$u(t) = K_c\left[e(t) + \frac{1}{T_i}\int_0^{\tau} e(\tau)d\tau\right] \qquad \text{EQ. 5-28}$$

The transition from a continuous to a discrete time domain requires that the integral operation must be approximated by a discrete numerical integration. There are several methods for replacing the time-continous integral. Here, the integral is approximated by the Backward Euler approach, also known as the Backward Rectangular method. The PI algorithm in discrete time domain follows:

$$u(k) = K_c \cdot e(k) + u_I(k-1) + K_c \cdot \frac{T}{T_i} \cdot e(k) \qquad \text{EQ. 5-29}$$

The values of K_c, and K_c/T_i (known as K_i) must be chosen carefully to ensure a proper transient response. By properly designing the PI controller, it is possible to achieve a transient response to a step input that exhibits a relatively small, or even no, overshoot. The transient response, however becomes slower, because the PI controller is a low-pass filter, which attenuates the high frequency components of the signal. The phenomenon of wind-up effect is avoided by limiting the integral part with respect to the controller output range. This feature is incorporated into the routines presented herein.

5.1.7 Angle Tracking Observer for Resolver

Resolvers are absolute angle transducers and are mounted on the motor shaft to get the motor's absolute angular position. Resolvers are often used for angle sensing in a noisy environment due to their rugged construction and their ability to reject common mode noise.The method for obtaining and digitizing the angular position of a resolver is also known as Resolver-to-Digital conversion (R/D conversion).

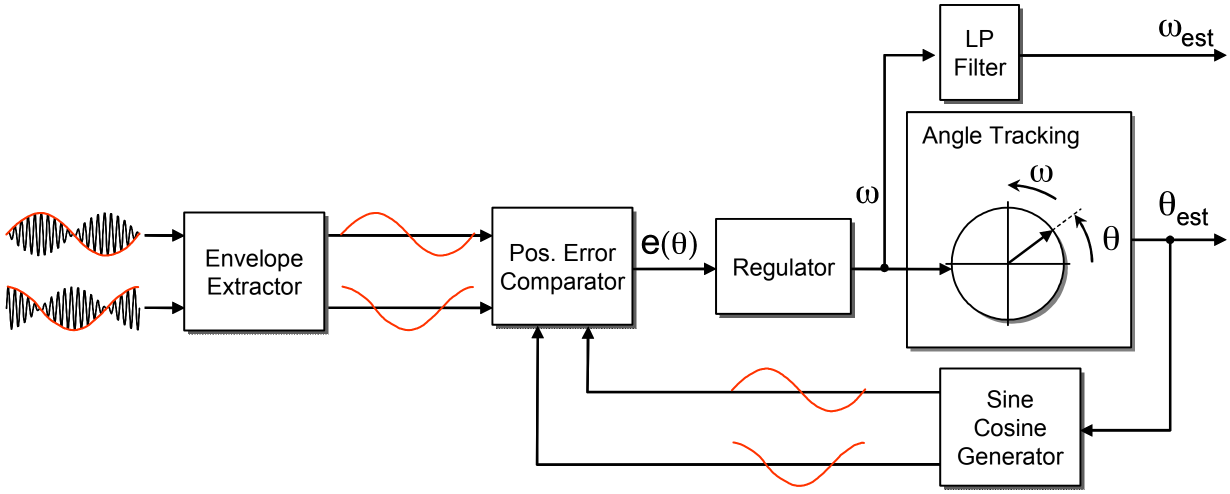

Figure 5-4. Block Diagram of Resolver Angle Extraction

The most common solution to extracting the rotor angle and speed from the resolver output signals is either a *Trigonometric* or *Angle Tracking Observer* method. It should be noted that both methods require a fast and highly accuracy measurement of the resolver output signals to be carried out.

The shaft angle can be determined by an *Inverse Tangent* function of the quotient of the sampled resolver output voltages U_{sin} and U_{cos}. This determination can be expressed in terms of resolver output voltages. An indispensable pre-condition of the accurate rotor angle estimation is to sample the resolver output signals simultaneously and close to their period peaks.

The second method (algorithm) widely used for estimation of the rotor angle and speed is generally known as an *Angle Tracking Observer;* see **Figure 5-4**. A great advantage of the *Angle Tracking Observer* method over the *Trigonometric* method is that it yields smooth and accurate estimations of both the rotor angle and rotor speed. The *Angle Tracking Observer* compares values of the resolver output signals, with their corresponding estimations, U_{sin} and U_{cos}. As in any common closed-loop system, the intent is to minimize observer error. The observer error is calculated by subtracting the estimated resolver rotor angle from the actual rotor angle. For more details on implementating speed and position sensing using a resolver, please refer to [12], **References**.

5.1.8 Evaluation Absolute Position by Digital Encoder

Digital information about rotor position can be obtained by the special arrangement of a high-resolution digital encoder providing relative position information, in combination with a low-resolution commutation sensor. This positional sensing technique is described in **Section 4.**. The quadrature decoder circuit of the hybrid controller is used to decode the quadrature encoder signals. The maximum counting resolution is 4x input signal. The quadrature decoder samples both incoming pulses. Based on the previous pulse information of the two signals and the present state, it outputs a count signal and a direction signal to internal position counts. The advanced timer modules involved can be used to decode the primary and secondary external inputs as quadrature encoded signals as well.

As seen in **Figure 5-5**, the commutation sensor provides information with 60 electrical degree granularity. Since the periphery involved for all position signals are handled only by the quadrature timer, minumum software is needed to execute, so most of the processing is done by the peripheries themselves, and only initialization and synchornization are performed by interrupt involvement.

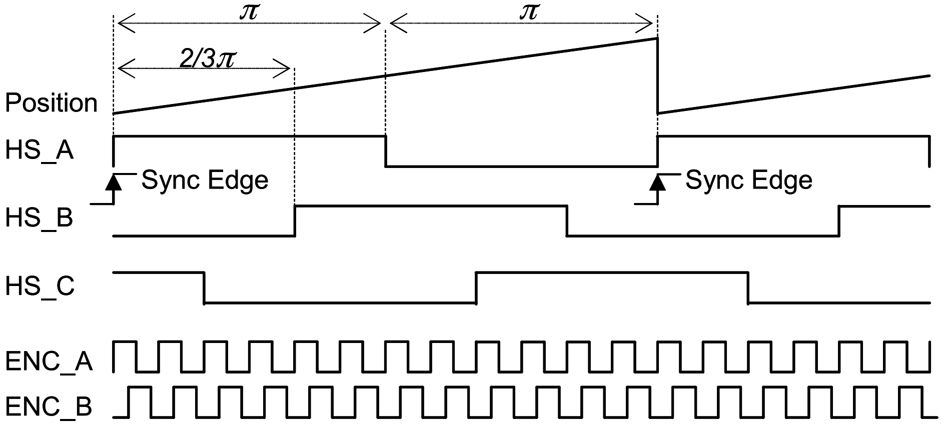

Figure 5-5. Absolute Position Encoder Signals

Position information can be evaluated by dedicated peripherals for the 56F8300 microcontroller, where it is possible to achieve high resolution of the rotor angular position and know the absolute position.

5.2 EMB actuation

The characteristics of each EMB system are determined by both hardware and software. The software plays a greater role in defining the character and brand of the new brake systems in vehicles, and the vehicle manufacturer, as system integrator, must define the behavior of the vehicle in terms of function, performance, comfort, endurance, safety, etc., to differentiate with respect to competitors.

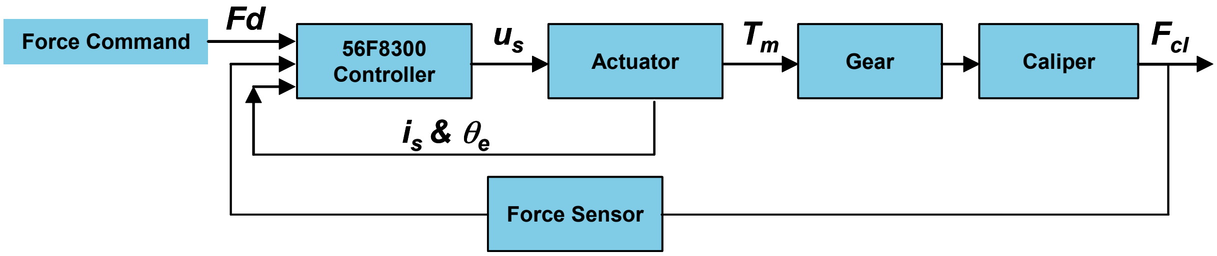

Figure 5-6. EMB Control System Structure

The base brake function allows interpretation of the driver’s demand and the brake actuation systems to decelerate the vehicle. The electronic pedal module contains a device that emulates the desired force versus displacement characteristics, as well as the force and travel sensors required to interpret the driver’s demand.

Having an architecture with brake clamp force control rather than position control is acceptable during brake actuation. During stand-by, however, position control is required to manage the air gap between the brake pads and brake rotor. Both modes of operation are achieved by switching between outer position and force control loops across the contact point between the brake pads and disc brake rotor.

Advanced braking functions such as anti-lock braking, traction control, vehicle stability, and chassis control allow optimization of vehicle braking and stability, but are not required for basic deceleration performance.

Another integrated part of the EMB software which can be adopted for running on an ECM is the dedicated operating system, as Time-Triggered OSEK, designed according to the time-triggered concept for deterministic system behavior, and allowing a complete separation of tasks (task encapsulation). If a faulty task occurs, the operating system is able to enter a "silent state" (failsilent). A fault-tolerant communication system conveys the real time image of each of the wheel brake units.

5.3 Air Gap Adjustment of Brake Pads

In contrast to hydraulic brakes, where the clearance of brake pads adjusts automatically when the hydraulic pressure is released, electro-mechanical brakes must actively adjust the clearance by actuator rotor movement.

To perform precise actuator rotor release, the “kiss position” point of contact between the brake pads and the disk must be known. The inner brake pad can then be moved to a defined position in order to adjust the desired clearance. It’s also necessary to know the contact point between the pads and the disk because this point is also the start braking position, when clamping brake force is about to begin.

The “kiss position” detection algorithm is first run for initialization of the EMB actuator when the automobile system is started. It also processes the brake signals of individual braking applications during normal driving, to preserve the zero value and to continually adapt this “kiss position” to be more immune to the brake pads’ wear. The car braking behavior is preserved for the driver’s comfort, and if an individual brake pad must be replaced, the EMB controller can issue a notification.

5.4 Communication

The existing CAN communication interface and its protocols are not deterministic (able to automatically transfer messages at pre-set points), as the timing of messages is not predictable. Because fault-tolerant, safety-critical applications like brake-by-wire systems required that messages must be transferred at predictable times (deterministic), another communications system is needed.

The FlexRay communication system will support the needs of future in-car control applications. The FlexRay communications protocol is at the core of the FlexRay system. It provides flexibility and determinism, incorporating the advantages of familiar synchronous and asynchronous protocols into a scalable static and dynamic message transmission. The protocol also supports:

- Fault-tolerant clock synchronization via a global time base
- Collision-free bus access
- Guaranteed message latency
- Message-oriented addressing via identifiers
- Scalable system fault-tolerance via the support of either single or dual channels

6. Freescale Solution for EMB System

To support Freescale's customers who design an EMB system, a complete application core targeted for EMB systems has been developed, which can be used directly for developing an EMB application and includes both hardware and software. This EMB application comprises the following features:

System

- Field-oriented control of the sinusoidal PMAC with either an attached Hall-effect with encoder or resolver
- 12/24 VDC 3-phase motor-driving capability
- CAN, RS-232, and JTAG communication available

Hardware

- Single motor controller board
- 56F8300 CPU
- 12/24 battery input voltage level
- Able to drive a 3-phase load
- Digital encoder inputs
- Commutation sensor inputs
- Wheel speed digital inputs
- Three analog inputs, 0-3.3V/12-bit

Software

- Field-oriented control - current control loop
- Position evaluation based on encoder or resolver
- Speed measurement
- PC master software interface

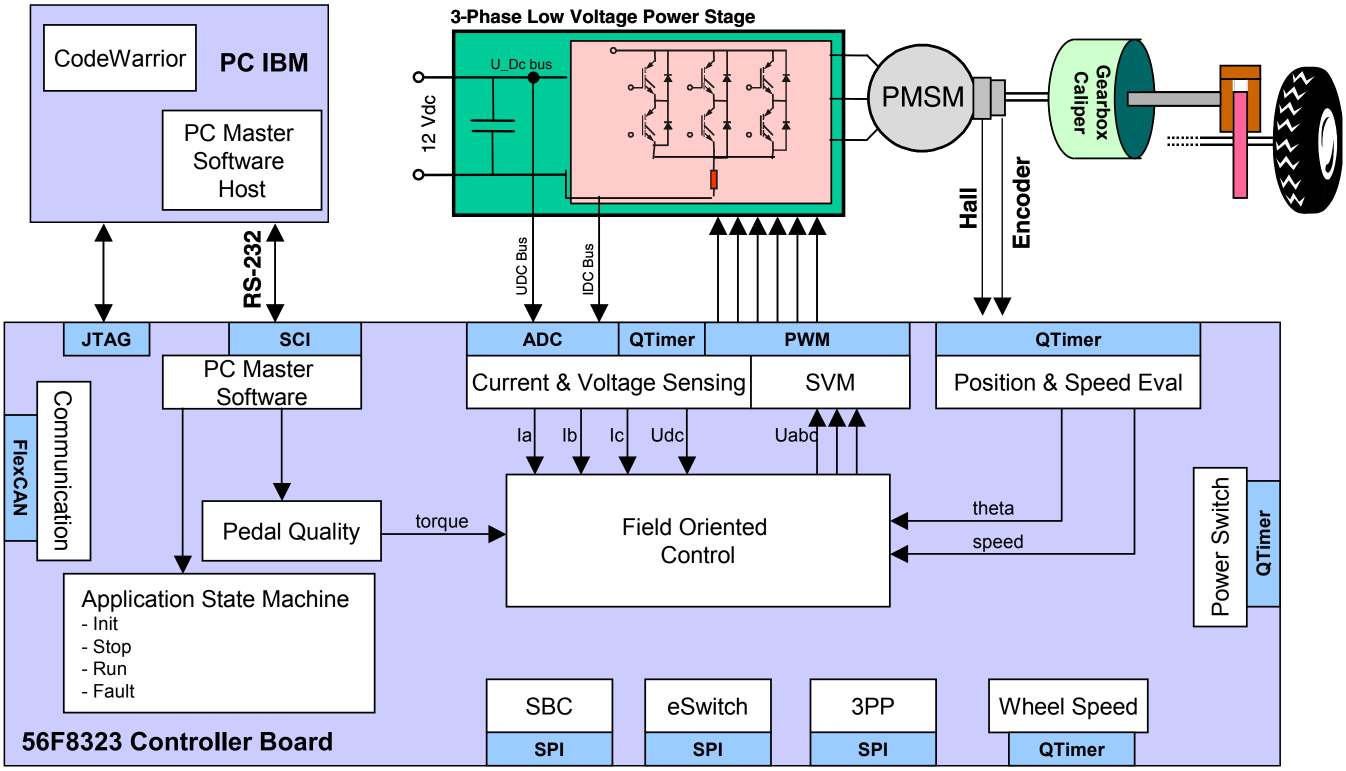

Figure 6-1. Application Block Diagram

The developed software system is based on the 3-phase Motor 56F8323 LV Board, which provides the necessary features to drive the brake actuator. The basic system is outlined in **Figure 6-1**. The rich peripheral set provided on the 56F8323 hybrid controller is fully exercisable, and, moreover, the additional automotive IC devices are accessible via programable SPI communication. User development can be based upon the software/hardware system provided.

For details about the 3-phase Motor 56F8323 LV Board, ask your Freescale representative for the board's user manual.

6.1 Single Motor Controller Board

Freescale Semiconductor's 3-phase Motor 56F8323 LV Board (see **Figure 6-3**) is a single motor controller board built around a Flash-based 56F8323 hybrid controller. This stand-alone board provides high voltage inputs/output, digital timer inputs, communication ports and various possibilities to drive different types of low-voltage motors (brushless DC, permanent magnet synchronous, etc.).

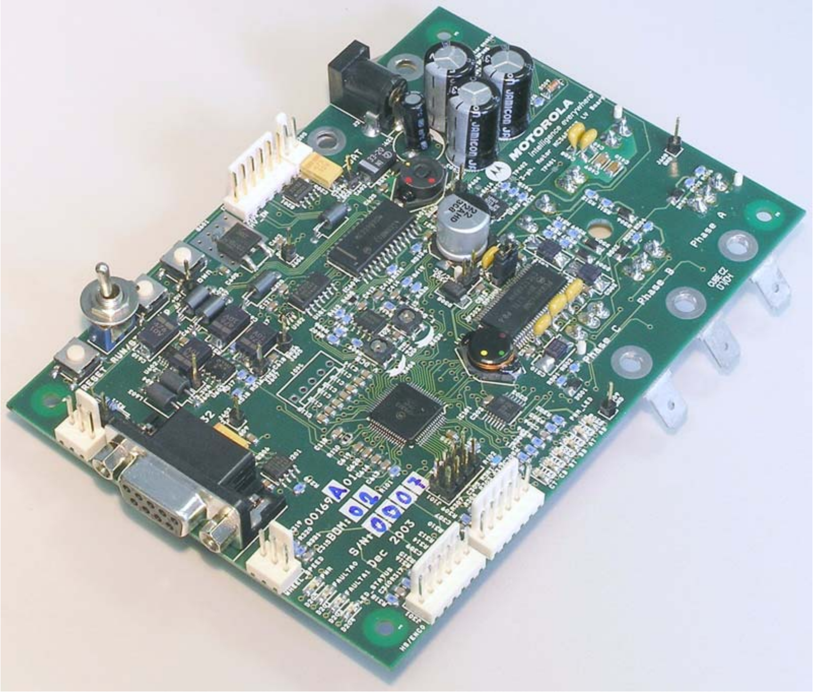

Figure 6-2. Photo of 56F8300 Motor Controller Board

The single motor controller board includes:

- 12/24V DC battery input
- Built-in 60 Amp inverter stage
- High voltage output of 5 Amp
- High voltage inputs
- Polarity protection
- 1MBit CAN interface
- RS-232 communication port
- JTAG debug port
- User LEDs
- Digital timer inputs
- User free ADC input channels
- Motor back-EMF sensing capability

6.1.1 Motor Controller Architecture

The 3-phase Motor 56F8323 LV Board facilitates the evaluation of various automotive IC parts as:

- An intelligent switch, replacing electro-mechanical relays, fuses, and discrete devices in power devices
- System-basis chip combining functions such as voltage regulators, high voltage inputs, and high speed CAN transceiver
- A 3-phase driver providing an interface between an MCU and the large FETs used to drive 3-phase loads
- A 16-bit hybrid controller on a single chip, with the processing power of a Digital Signal Processor (DSP) and the functionality of a microcontroller with a flexible set of peripherals

The 3-phase Motor 56F8323 LV Board provides the necessary features for a user to write and debug software, demonstrate the functionality of this software, and interface with the customer's application-specific arrangement. This 3-phase Motor 56F8323 LV Board can be used to develop real-time software and hardware products based on Freescale’s automotive IC parts.

An illustration of the board system architecture is shown in **Figure 6-3** The key features of the 3-phase Motor 56F8323 LV Board are maintained by Freescale’s automotive IC products. Basic segments of the 3-phase Motor 56F8323 LV Board appear in this photo:

- Input electronic switch by intelligent MC33982
- 56F8323 microcontroller
- System-basis support by MC33989
- Power stage by SEMIKRON SK115MD10
- Power MOSFET driver by MC33896

The board design is optimized for analog control signal measurement, so high fidelity measurement can be achieved.

The 3-phase Motor 56F8323 LV Board is sufficiently flexible to allow a user to fully exploit the features of Freescale’s automotive IC part 56F8323, as shown in **Figure 6-3**.

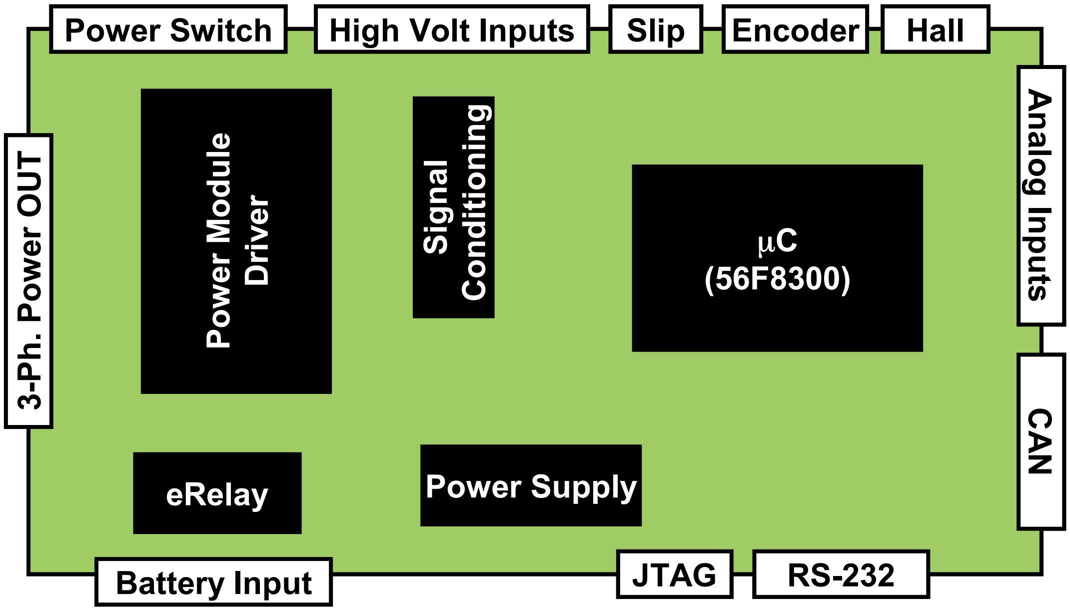

Figure 6-3. Block Diagram of 56F8300 Motor Controller Board

6.1.2 Board Connectivity

The stand-alone 3-phase Motor 56F8323 LV Board contains many features which are programmable and accessible in connection with supporting development tools, such as Metrowerks' CodeWarrior IDE and Freescale's PC master software debug utility. A simple preview of the key ICs, connectors, headers, and jumpers can be seen in **Figure 6-3**.

- Integrated circuits: 56F8323, MC33989, MC33896
- Connectors: Hall Sensors and Encoder, CAN, RS-232, high voltage input and output, power input, phase current inputs
- Terminals: Power input, 3-phase output
- Jumpers: CAN terminator, MC33896 DC/DC inverter and charge pump
- Header: JTAG
- Buttons: Reset, user up and down buttons
- Switch: User Run/Stop switch
- Pots: Overcurrent, Overvoltage
- Integrated circuit: MC33982

6.2 Voltage Sensing

The DCBus voltage sensor is represented by a simple voltage divider. DCBus voltage does not change rapidly and is nearly constant, with the ripple given by the battery supply structure. The measured DCBus voltage must be filtered to eliminate noise. One of the easiest and fastest techniques is the first order filter, which calculates the average filtered value recursively from the last two samples and coefficient C:

$$u_{DCBusFilt}(n+1) = (Cu_{DCBusFilt}(n+1) - Cu_{DCBusFilt}(n)) - u_{DCBusFilt}(n) \quad \text{EQ. 6-1}$$

To speed up initialization of voltage sensing (the filter has exponential dependency with a constant of 1/N samples), the moving average filter, which calculates the average value from the last N samples, can be used for initialization:

$$u_{DCBusFilt} = \sum\nolimits_{n=1}^{-N} u_{DCBus}(n) \quad \text{EQ. 6-2}$$

6.3 Motor Phase Current Sensing

It is possible to eliminate or minimize the need for a discrete current sensor for sinusoidal PMAC machines. The 3-phase current values for a sinusoidal PMAC motor can be reconstructed from DC Link current measurements by using a single shunt resistor with appropriate control. The success of such a scheme depends on modifications of the basic PWM algorithm to ensure that the single current sensor is granted access to each of the three-phase currents in a sufficient period of time.

The phase current feedback signal is proportional to the DCBus current and is provided by the circuitry shown in **Figure 6-4** The motor controller board introduced here provides only a DC Link shunt resistor, which is placed on the negative DCBus rail for motor current sensing. Knowing 3-phase information is required to control a sinusoidal PMAC machine, so advanced sampling and evaluation techniques must be incorporated.

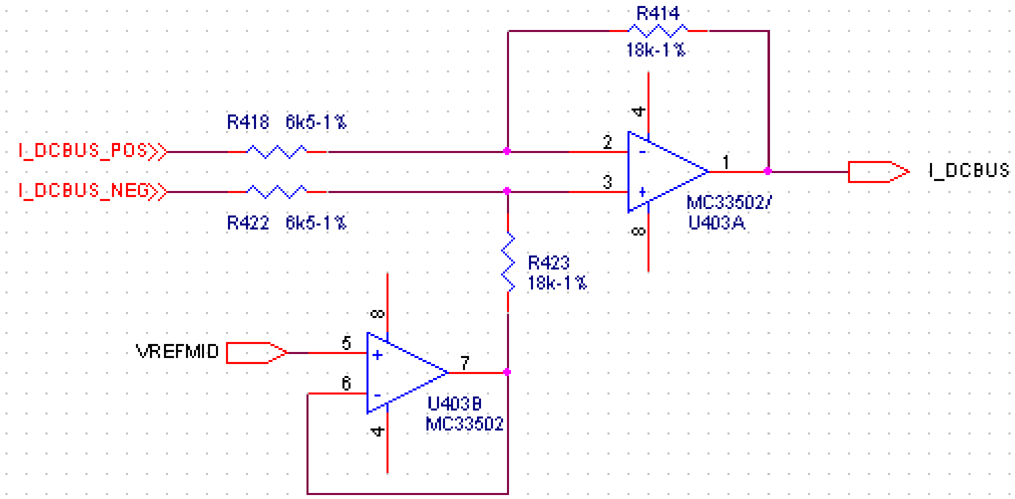

Figure 6-4. DCBus Current Sensing Circuitry

6.4 Position and Speed Sensing

All devices in Freescale's 56F8300 series include a Quadrature Decoder. This peripheral is commonly used for position and speed sensing. The Quadrature Decoder's position counter counts up/down each edge of the Phase A and Phase B signals according to their order. On each revolution, the position counter might be cleared by an index pulse, if provided by the encoder sensor.

Field-Oriented Control (FOC), or another named sinusoidal PMAC vector control, requires exact knowledge of the zero rotor position, where it is assumed that the sinusoidal PMAC rotor is aligned to the d axis in the rotating reference frame. Therefore, employing a Quadrature Decoder to decode the encoder's signals requires either the calculation of a position offset, which initializes the Quadrature Decoder position counter to correspond with the aligned rotor position (zero position), or the coupling of defined rotor position instances with the commutation sensor signals (HS_A, HS_B, HS_C).

To avoid calculation of the rotor position offset, position initialization is performed using the signal information from the commutation sensor, which reveals the exact position information every 60 electrical degree assigned to the individual edge transition. It's necessary to detect this edge transition and exploit this knowledge to correct the decoder position counter to be precisely aligned with the zero rotor position.

After reset, the position sector of the sinusoidal PMAC rotor can be evaluated using the status of individual Hall-effect sensors signals. The Hall-effect sensor status determines the initial position information for the decoder periphery. Since the position status has 60 electrical granularity, a position counter is initialized into the center of a particular sector with the *InitEncoder* variable. The motor control algorithm is provided with this rough information primarily for motor start-up only, allowing the user to spin the sinusoidal PMAC motor without having to move the rotor to an aligned position. This position initialization process is described in **Figure 6-5**. Additionally, the synchronization variable of edge detection occurrence (*RightEdge, LeftEdge*) are prepared.

Since the motor control algorithm has adequate position information, the sinusoidal PMAC motor can be driven correspondingly. If any following movement occurs, inherently generated incoming edges provide precise information to synchronize the motor control algorithm to the exact rotor position. This is accomplished by software execution, as depicted in **Figure 6-5**.

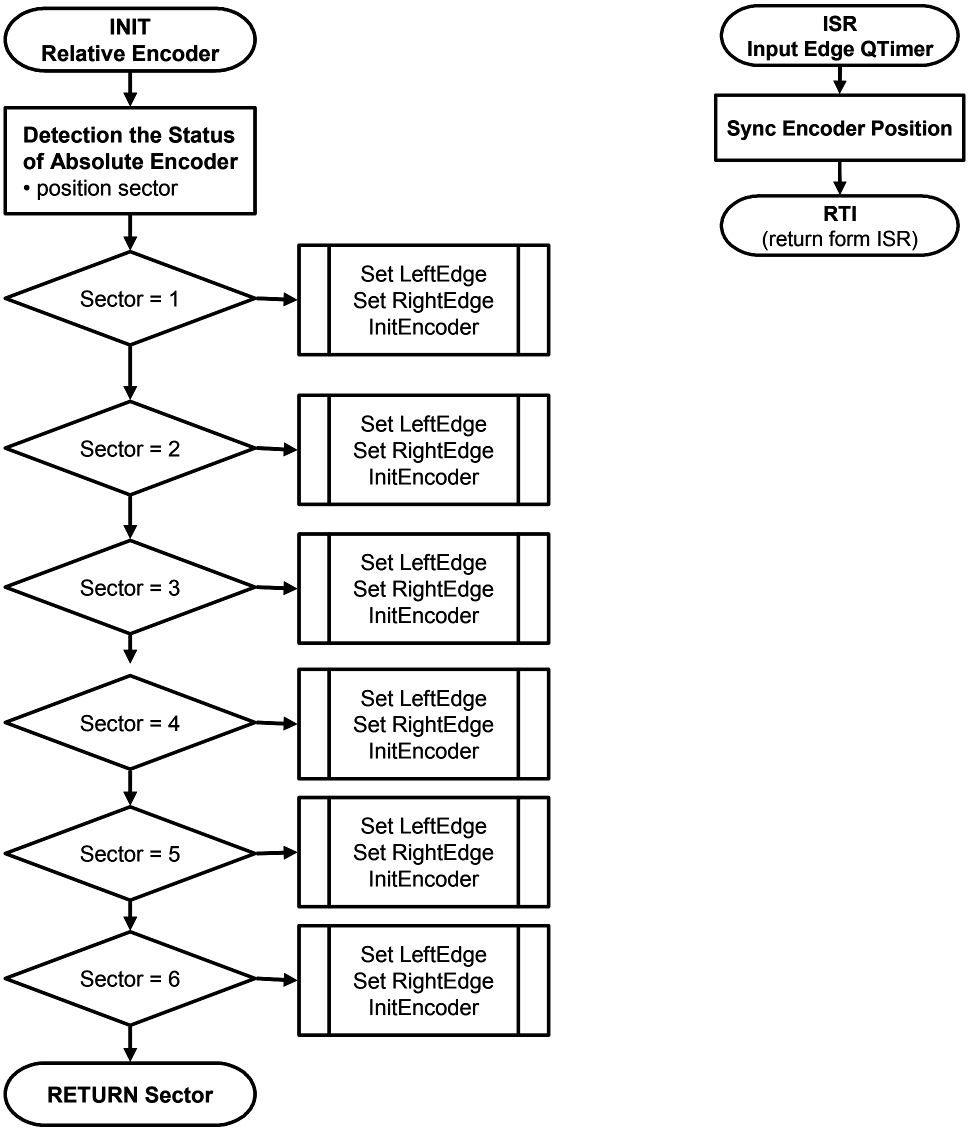

Figure 6-5. Position Initialization and Position Synchronization

6.5 Software Implementation

This section briefly describes the software structure of the motor actuator control. This application benefits from the core software structure developed in [2], **References**.

6.5.1 Numeric Value Scaling

The sinusoidal PMAC machine control algorithm and overall application uses a fractional representation for all real quantities. The N-bit signed fractional format is represented using 1.[N-1] format (1 sign bit, N-1 fractional bits). Signed fractional numbers (SF) lie in the following range:

$$-1.0 \le \mathrm{SF} \le +1.0 \text{ -}2^{-[N-1]} \qquad \text{EQ. 6-3}$$

For words and long word signed fractions, the most negative number that can be represented is -1.0, whose internal representation is \$8000 and \$80000000, respectively. The most positive word is \$7FFF or 1.0 - 2^{-15}, and the most positive long word is \$7FFFFFFF or 1‘.0 - 2^{-31}

The following equation shows the relationship between a real and a fractional representation:

$$\text{Fractional Value} = \frac{\text{Real Value}}{\text{Real Quantity Range}} \qquad \text{EQ. 6-4}$$

where:

Fractional Value is the fractional representation of the real value [Frac16]

Real Value is the real value of the quantity [V, A, RPM, etc.]

Real Quantity Range Max is the maximum of the quantity range, defined in the application [V, A, RPM, etc.]

The C language standard does not have any fractional variable type defined. Therefore, fractional operations are provided by CodeWarrior intrinsic functions (e.g. mult_r()). As a substitute for fractional type variables, the application uses types Frac16 and Frac32. These are defined as integer 16-bit and 32-bit signed variables. The difference between Frac16 and pure integer variables is that Frac16 and Frac32 declared variables should only be used with fractional operations (intrinsic functions).

A recalculation from a real to a fractional form and a Frac16 or Frac32 value is made through the following equations:

$$\text{Frac16 Value} = 32768 \cdot \frac{\text{Real Value}}{\text{Real Quantity Range Max}} \qquad \text{EQ. 6-5}$$

for Frac16 16-bit signed value and:

$$\text{Frac32 Value} = 2^{31} \cdot \frac{\text{Real Value}}{\text{Real Quantity Range Max}} \qquad \text{EQ. 6-6}$$

for Frac32 32-bit signed value.

$$\text{Fractional Value} = \frac{\text{Real Value}}{\text{Real Quantity Range Max}} \qquad \text{EQ. 6-7}$$

Fractional form, a conversion from Fraction Value to Frac16 and Frac32 value, can be provided by the C language macro.

6.6 Actuator Control Algorithms

The benefits of structured modular software design are well known. This is especially true for large complex automotive systems with many interacting software sub-blocks. As depicted in **Figure 5-1**, there are several functional blocks needed for the actuator control process, which are available in modular form.

The sinusoidal PMAC control process provides most of the motor control functionality. It is regularly executed mainly in the current processing interrupt. The current processing is called from within ADC Complete Interrupt, whown in **Figure 6-6**, once per two PWM reloads, with a period 100µs. It can also be set on each PWM reload (50µs).

Additionally, there might be a PC master software recorder *pcmasterdrvRecorder()* routine used to monitor the application.

As indicated by the flowchart of the ADC Complete Interrupt Service Routine (ISR), the sampling of analog values is processed to gain appropriate information for overall control process and monitoring. This procedure is followed by sensing the digital values of rotor position and speed measurement is performed by software computation. Knowing the appropriate control information, the PMSM current control is followed and individual software functional blocks are executed (see **Figure 6-6**). When the PMSM current control process is performed, the required voltage vector is generated and is ready to apply by updating the PWM value registers.

A 56F8300 hybrid controller provides programmable fault protection. Fault protection can disable any combination of PWM pins. Faults are generated by a logic one on any of the fault pins. Each fault pin can be mapped arbitrarily to any of the PWM pins. When fault protection hardware disables the PWM pins, the PWM generator continues to run, and only the output pins are deactivated. If a fault is latched in, it must be cleared and the PWM fault interrupt service routine is invoked (see **Figure 6-6**) prior to enabling the PWM, to prevent an unexpected interrupt.

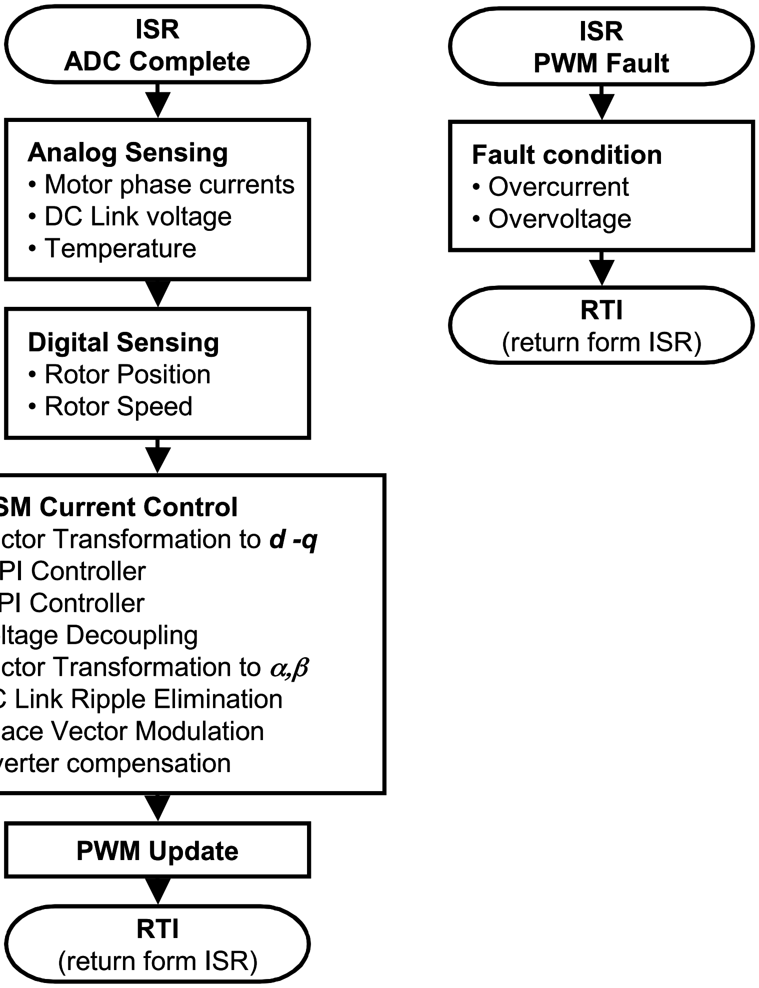

Figure 6-6. Motor Control Processing

6.7 Application State Diagram

The main software flow chart (see **Figure 6-7**) incorporates the main routine entered from the reset state. The main routine includes the application initialization, application control, and fault detection processes. The hybrid 56F8300 processor enters the reset processing state when a hardware reset signal is asserted. Upon exiting the reset state, the core enters the normal processing state and starts to execute the internal code. This application initializes the system basis chip first to enable correct board power-up. This procedure is followed by the standard microprocessor start-up sequence. It is necessary to set the occupied peripherals to their correct initial values.

Before the motor control algorithm enters the execution state, the 3-phase power driver IC is instructed to power up the boostrap capacitors, enabling the power MOSFET module to fully function. Since the application software must be able to provide its own application status and set up additional on-board IC devices, the communication channels, such as CAN and RS-232, must be correctly initialized.

Application control is a software mechanism incorporating a state machine which governs the overall application operation, and upon available data, runs appropriate software states. In run-time conditions, the hardware status, such as temperature, IC device status, and so on, is monitored.

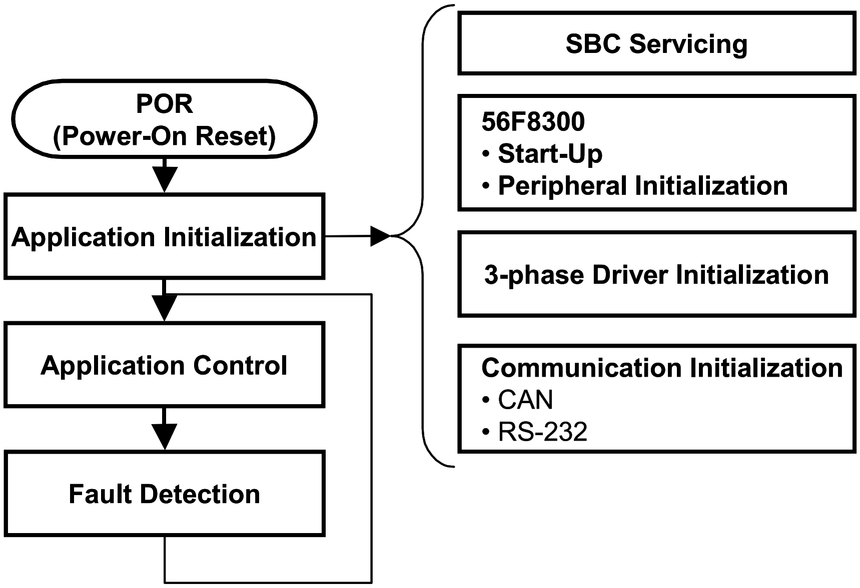

Figure 6-7. Main Software Flowchart

The application state control is depicted in **Figure 6-8** in more detail, and reveals the application software in the state machine form. When the normal execution state of the hybrid controller is entered, the software starts executing the INIT state, and the initialization procedures take place. As soon as the initialization is performed, the state corresponding to the condition of the control variables is entered.

The STOP state provides an actuator standstill. In the STOP state, the brake drive is disabled, PWM output pads are disabled, and the PMSM current control process is called by the ADC Complete ISR. The application waits for the start command. If any fault is detected in the STOP state, the application enters the FAULT state. If no fault is present and the start command is accepted, the application transits to the RUN state, and electric motor actuation is started upon the brake command.

The RUN state can be entered from the STOP state. The RUN state performs a motor brake actuation. In the RUN state, the brake drive is enabled and the actuator develops the appropriate shaft torque in the brake mechanism. The PWM output pads are enabled and the full PMSM current control process is called by the ADC Complete ISR. If any fault in the RUN state is detected, the application enters the FAULT state.

The FAULT state can be entered from any state. In the FAULT state, the brake actuator is disabled and the application waits for the fault to be cleared. When it detects that the fault has disappeared and the fault clear command is accepted, the switch command is moved to the stop position and the application transits to the INIT state.

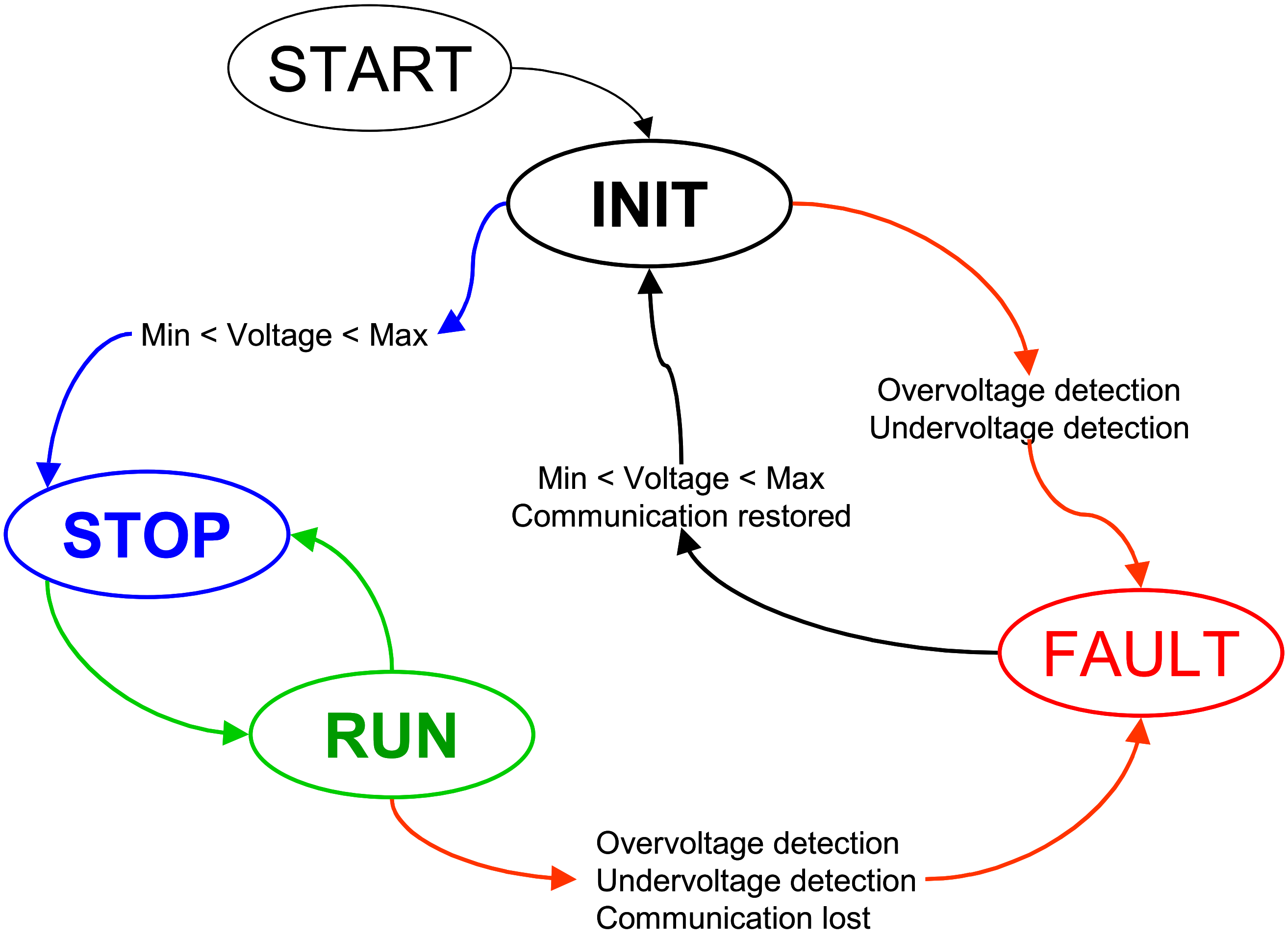

Figure 6-8. Generic Application State Machine

7. References

1. Bose, K. B. (2002). *Modern Power Electronics and AC Drives,* Prentice Hall, Inc. ISBN 0-13-016743-6, New York.
2. *3-Phase PM Synchronous Motor Torque Vector Control Using 56F805,* DRM018/D, Freescale Semiconductor, Inc.
3. Ch. Line, Ch. Manzie and M. Good (2004). *Control of an Electromechanical Brake for Automotive Brake-By-Wire Systems with an Adapted Motion Control Architecture*, SAE Automotive Dynamics, Stability & Controls Conference and Exhibition, P-386, Detroit, Michigan
4. R. Schwarz, R. Isermann, J. Böhm, J. Nell and P. Rieth (1999). *Clamping Force Estimation for a Brake-by-Wire Actuator*, International Congress and Exposition Detroit, SP-1413, Detroit, Michigan.
5. I. Petersen (2003). *Wheel Slip Control in ABS Brakes using Gain Scheduled Optimal Control with Constraints,* PhD. Thesis, NUST Trondheim, Norway
6. N. A. Kelling, and W. Heck (2002), *The BRAKE Project - Centralized Versus Distributed Redundancy for Brake-by-Wire Systems*, SAE 2002 World Congress, SP-1658, Detroit, Michigan
7. D. F. Reuter, E. W. Lloyd, J. W. Zehnder II and J. A. Elliott (2003). *Hydraulic Design Considerations for EHB Systems*, 2003 SAE World Congress, SP-1780, Detroit, Michigan
8. J. Langenwalter and B. Kelly (2003), *Virtual Design of a 42V Brake-by-Wire System*, SAE 2003 World Congress & Exhibition, SP-1769, Detroit, Michigan
9. *Electro-mechanical Braking (Brake-By-Wire)*, SG2008, Freescale Semiconductor, Inc.
10. *DSP56800E Reference Manual*, DSP56F800ERM, Freescale Semiconductor, Inc.
11. *56F8300 Peripheral User Manual*, MC56F8300UM, Freescale Semiconductor, Inc.
12. *DSP56F80x Resolver Driver and Hardware Interface,* AN1942, Freescale Semiconductor, Inc.

How to Reach Us:

Home Page:
www.freescale.com

E-mail:
support@freescale.com

USA/Europe or Locations Not Listed:
Freescale Semiconductor
Technical Information Center, CH370
1300 N. Alma School Road
Chandler, Arizona 85224
+1-800-521-6274 or +1-480-768-2130
support@freescale.com

Europe, Middle East, and Africa:
Freescale Halbleiter Deutschland GmbH
Technical Information Center
Schatzbogen 7
81829 Muenchen, Germany
+44 1296 380 456 (English)
+46 8 52200080 (English)
+49 89 92103 559 (German)
+33 1 69 35 48 48 (French)
support@freescale.com

Japan:
Freescale Semiconductor Japan Ltd.
Headquarters
ARCO Tower 15F
1-8-1, Shimo-Meguro, Meguro-ku,
Tokyo 153-0064, Japan
0120 191014 or +81 3 5437 9125
support.japan@freescale.com

Asia/Pacific:
Freescale Semiconductor Hong Kong Ltd.
Technical Information Center
2 Dai King Street
Tai Po Industrial Estate
Tai Po, N.T., Hong Kong
+800 2666 8080
support.asia@freescale.com

For Literature Requests Only:
Freescale Semiconductor Literature Distribution Center
P.O. Box 5405
Denver, Colorado 80217
1-800-441-2447 or 303-675-2140
Fax: 303-675-2150
LDCForFreescaleSemiconductor@hibbertgroup.com

Information in this document is provided solely to enable system and software implementers to use Freescale Semiconductor products. There are no express or implied copyright licenses granted hereunder to design or fabricate any integrated circuits or integrated circuits based on the information in this document.

Freescale Semiconductor reserves the right to make changes without further notice to any products herein. Freescale Semiconductor makes no warranty, representation or guarantee regarding the suitability of its products for any particular purpose, nor does Freescale Semiconductor assume any liability arising out of the application or use of any product or circuit, and specifically disclaims any and all liability, including without limitation consequential or incidental damages. “Typical” parameters that may be provided in Freescale Semiconductor data sheets and/or specifications can and do vary in different applications and actual performance may vary over time. All operating parameters, including “Typicals”, must be validated for each customer application by customer’s technical experts. Freescale Semiconductor does not convey any license under its patent rights nor the rights of others. Freescale Semiconductor products are not designed, intended, or authorized for use as components in systems intended for surgical implant into the body, or other applications intended to support or sustain life, or for any other application in which the failure of the Freescale Semiconductor product could create a situation where personal injury or death may occur. Should Buyer purchase or use Freescale Semiconductor products for any such unintended or unauthorized application, Buyer shall indemnify and hold Freescale Semiconductor and its officers, employees, subsidiaries, affiliates, and distributors harmless against all claims, costs, damages, and expenses, and reasonable attorney fees arising out of, directly or indirectly, any claim of personal injury or death associated with such unintended or unauthorized use, even if such claim alleges that Freescale Semiconductor was negligent regarding the design or manufacture of the part.

freescale™
semiconductor

Freescale™ and the Freescale logo are trademarks of Freescale Semiconductor, Inc.
All other product or service names are the property of their respective owners.
This product incorporates SuperFlash® technology licensed from SST.
© Freescale Semiconductor, Inc. 2004. All rights reserved.

AN1999
Rev. 0
11/2004