3-Phase SR Motor Control with Hall Sensors Using a 56F80x, 56F8100 or 56F8300 Device

Design of Motor Control Application

Radim Visinka

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

Contents

1. Introduction

This Application Note describes a design example with a 3-phase Switched Reluctance (SR) motor drive. It is based on Freescale's 56F80x or 56F8300 dedicated motor control devices. The software design takes advantage of Processor Expert (PE) software.

SR motors are gaining wider popularity for variable speed drives. This is due to their simple low-cost construction characterized by an absence of magnets and rotor winding, high performance over wide speed range, and fault-tolerant design of the power stage. Availability and cost of the necessary electronic control make the SR drive a viable alternative to other commonly used motors, like AC, BLDC, PM Synchronous or universal motors, for a number of applications.

This application is a speed closed-loop SR drive using a Hall position sensor. The application serves as an example of an SR motor control system design using a Freescale controller with PE support. It also illustrates the use of dedicated motor control libraries that are included in the PE.

This application note includes a description of the Freescale controller's features, basic SR motor theory, system design concept, hardware implementation and software design, including utilization of the PC master software visualization tool.

2. Advantages and Features of Freescale's Controller

The Freescale 56F80x (56800 core) and 56F8300 (56800E core) families are well suited for digital motor control, combining a DSP's calculation capability with an MCU's controller features on a single chip. These controllers offer many dedicated peripherals, such as a Pulse Width Modulation (PWM) unit, an Analog-to-Digital Converter (ADC), timers, communication peripherals (SCI, SPI, CAN), on-board Flash and RAM. Generally, all family members are appropriate for use in Switched Reluctance motor control.

The following sections use a specific family member to describe the family's features.

2.1 56F805, 56800 Core Family

The 56F805 provides the following peripheral blocks:

- Two Pulse Width Modulator units (PWMA and PWMB), each with six PWM outputs, three Current Sense inputs, and four Fault inputs, fault-tolerant design with dead time insertion, supports both center-aligned and edge-aligned modes
- 12-bit Analog-to-Digital Convertors (ADCs), supporting two simultaneous conversions with dual 4-pin multiplexed inputs, ADC can be synchronized by PWM
- Two Quadrature Decoders (Quad Dec0 and Quad Dec1), each with four inputs, or two additional Quad Timers, A & B
- Two dedicated general purpose Quad Timers totalling six pins: Timer C with two pins and Timer D with four pins
- CAN 2.0 B-compliantmModule with 2-pin ports used to transmit and receive
- Two Serial Communication Interfaces (SCI0 and SCI1), each with two pins, or four additional GPIO lines
- Serial Peripheral Interface (SPI), with a configurable 4-pin port, or four additional GPIO lines
- Computer Operating Properly (COP) / Watchdog timer
- Two dedicated external interrupt pins
- Fourteen dedicated General Purpose I/O (GPIO) pins; 18 multiplexed GPIO pins
- External reset pin for hardware reset
- JTAG/On-Chip Emulation (OnCE)
- Software-programmable, Phase Lock Loop-based frequency synthesizer for the controller core clock

Table 2-1. Memory Configuration for 56F80x Devices

	56F801	56F803	56F805	56F807
Program Flash	8188 x 16-bit	32252 x 16-bit	32252 x 16-bit	61436 x 16-bit
Data Flash	2K x 16-bit	4K x 16-bit	4K x 16-bit	8K x 16-bit
Program RAM	1K x 16-bit	512 x 16-bit	512 x 16-bit	2K x 16-bit
Data RAM	1K x 16-bit	2K x 16-bit	2K x 16-bit	4K x 16-bit
Boot Flash	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit

2.2 56F8346, 56800E Core Family

The 56F8346 provides the following peripheral blocks:

- Two Pulse Width Modulator units (PWMA and PWMB), each with six PWM outputs, three Current Sense inputs, and three Fault inputs for PWMA/PWMB; fault-tolerant design with dead time insertion, supporting both center-aligned and edge-aligned modes
- Two 12-bit Analog-to-Digital Converters (ADCs), supporting two simultaneous conversions with dual 4-pin multiplexed inputs; the ADC can be synchronized by PWM modules
- Two Quadrature Decoders (Quad Dec0 and Quad Dec1), each with four inputs, or two additional Quad Timers, A & B
- Two dedicated general purpose Quad Timers totaling three pins: Timer C with one pin and Timer D with two pins
- CAN 2.0 B-compliant module with 2-pin ports used to transmit and receive
- Two Serial Communication Interfaces (SCI0 and SCI1), each withtwo pins, or four additional GPIO lines
- Serial Peripheral Interface (SPI), with a configurable 4-pin port, or four additional GPIO lines
- Computer Operating Properly (COP) / Watchdog timer
- Two dedicated external interrupt pins
- 61 multiplexed General Purpose I/O (GPIO) pins
- External reset pin for hardware reset
- JTAG / On-Chip Emulation (OnCE)
- Software-programmable, Phase Lock Loop-based frequency synthesizer for the controller core clock
- Temperature Sensor system

Table 2-2. Memory Configuration for 56F8300 Devices

	56F8322	56F8323	56F8345	56F8346	56F8347
Program Flash	16K x 16-bit	16K x 16-bit	64K x 16-bit	64K x 16-bit	64 x 16-bit
Data Flash	4K x 16-bit	4K x 16-bit	4K x 16-bit	4K x 16-bit	4K x 16-bit
Program RAM	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit
Data RAM	4K x 16-bit	4K x 16-bit	4K x 16-bit	4K x 16-bit	2K x 16-bit
Boot Flash	4K x 16-bit	4K x 16-bit	4K x 16-bit	4K x 16-bit	4K x16-bit

Table 2-2 Memory Configuration for 56F8300 Devices, continued

	56F8355	56F8356	56F8357	56F8365	56F8366	56F8367
Program Flash	128K x 16-bit	128K x 16-bit	128K x 16-bit	256K x 16-bit	128K x 16-bit	128K x 16-bit
Data Flash	4K x 16-bit	4K x 16-bit	4K x 16-bit	16K x 16-bit	4K x 16-bit	4K x 16-bit
Program RAM	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit	2K x 16-bit
Data RAM	8K x 16-bit	8K x 16-bit	8K x 16-bit	16K x 16-bit	4K x 16-bit	8K x 16-bit
Boot Flash	4K x 16-bit	8K x 16-bit	8K x 16-bit	16K x 16-bit	8K x 16-bit	8K x 16-bit

2.3 Peripheral Description

In addition to the fast Analog-to-Digital converter, the most interesting peripherals, from the SRM application point of view, are the Pulse-Width-Modulation (PWM) unit and the 16-bit Quad Timer.

The PWM module offers a lot of freedom in its configuration, enabling efficient control of the SR motor. It has the following features:

- Three complementary PWM signal pairs, or six independent PWM signals
- Supports complementary channel operation
- Dead time insertion
- Separate top and bottom pulse width correction via current status inputs or software
- Separate top and bottom polarity control
- Edge-aligned or center-aligned PWM signals
- Resolution of 15 bits
- Integral reload rates from one to 16; half-cycle reload capability
- Individual software-controlled PWM output
- Programmable fault protection
- Polarity control
- 20-mA current sink capability on PWM pins
- Write-protectable registers

The SR Motor control application utilizes the PWM module set in the independent PWM mode, permitting fully independent generation of control signals for all switches of the power stage. In addition to the PWM generators, the PWM outputs can be controlled separately by software, allowing the setting of the control signal to logical 0 or 1. Thus, the state of the control signals can be changed immediately at a given rotor position (phase commutation) without changing the content of the PWM value registers.

The Quad Timer is an extremely flexible module, providing all required services related to time events. It has the following features:

- Each timer module consists of four 16-bit counters / timers
- Counts up / down
- Counters are cascadable

- Programmable count modulo
- Maximum count rate equals peripheral clock / 2 when counting external events
- Maximum count rate equals peripheral clock when using internal clocks
- Count once or repeatedly
- Counters are preloadable
- Counters can share available input pins
- Each counter has a separate prescaler
- Each counter has capture and compare capability

The SR motor application utilizes three channels of the Quadrature Timer module in Input Capture mode. It enables sensing of the rotor position using position Hall sensors.

3. Target Motor Theory

3.1 Switched Reluctance Motors

A Switched Reluctance (SR) motor is a rotating electric machine where both stator and rotor have salient poles. The stator winding comprises a set of coils, each of which is wound on one pole. The rotor is created from lamination in order to minimize the eddy-current losses.

SR motors differ in the number of phases wound on the stator. Each has a certain number of suitable combinations of stator and rotor poles. **Figure 3-1** illustrates a typical 3-phase SR motor with a six stator poles/ four rotor poles (6/4) configuration.

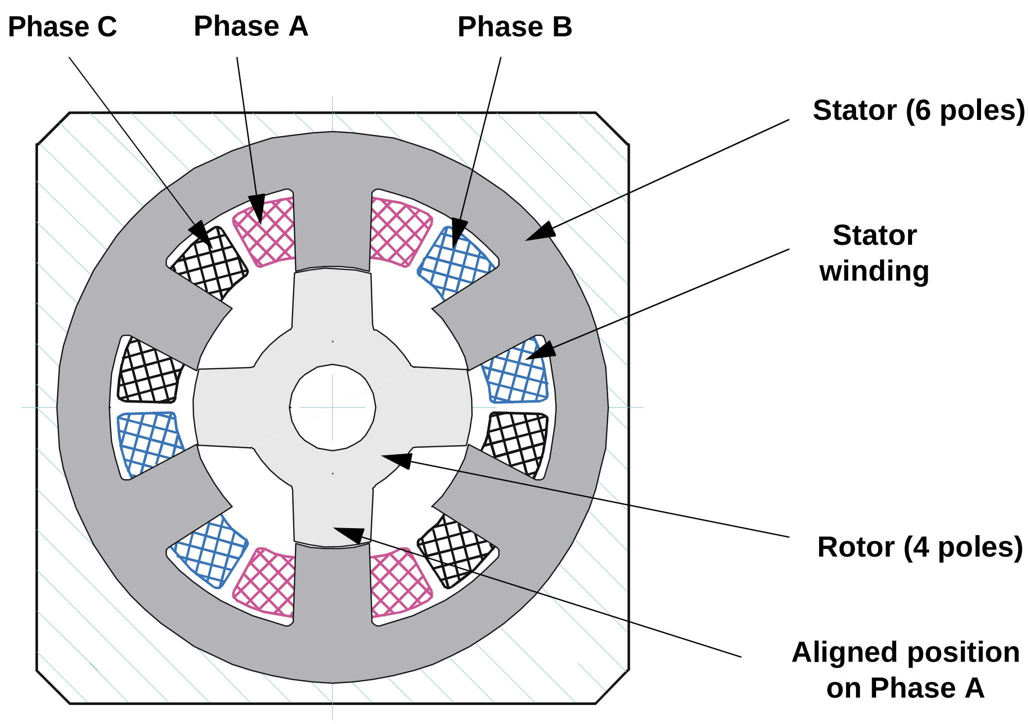

Figure 3-1. 3-Phase 6 / 4 SR Motor

The motor is driven by a sequence of current pulses applied in each phase. The individual phases are consequently energized, forcing the motor to rotate. The current pulses must be applied to the respective phase in an exact rotor position to the energized phase. When any pair of rotor poles is exactly in line with the stator poles of the selected phase, the phase is said to be in an aligned position—the rotor is in the position of maximal stator inductance (see **Figure 3-1**). If the interpolar axis of the rotor is in-line with the stator poles of the selected phase, the phase is said to be in an unaligned position—the rotor is in a position of minimal stator inductance. The inductance profile of SR motors is a triangular profile with its maximum in the aligned position and its minimum in the unaligned position. **Figure 3-2** illustrates the idealized triangular inductance profile of all three phases of an SR motor with Phase A highlighted. The individual Phases A, B, and C are shifted by 120 electrical degrees relative to each other. The interval, when the respective phase is powered, is called the dwell angle, θ_{dwell}. It is defined by the turn-on angle, θ_{on}, and the turn-off angle, θ_{off}.

When the voltage is applied to the stator phase, the motor creates torque in the direction of increasing inductance. When the phase is energized in its minimum inductance position, the rotor moves to the forthcoming position of maximal inductance. The movement is defined by the magnetization characteristics of the motor. The typical current profile for the constant phase voltage is shown in **Figure 3-2**. For a constant phase voltage the phase current has its maximum in the position when the inductance starts to increase. This corresponds to the position when the rotor and the stator poles start to overlap. After the phase is turned off, the phase current falls to zero. The phase current, present at the region of decreasing inductance, generates negative torque. The torque, generated by the motor, is controlled by the applied phase voltage and by the appropriate definition of switching turn-on and turn-off angles. For more details, see [1], **References**.

The position of the rotor must be measured during motor operation, since the phases must be properly energized. This can be achieved by the position sensor, or using sensorless techniques, evaluating the motor current and voltage.

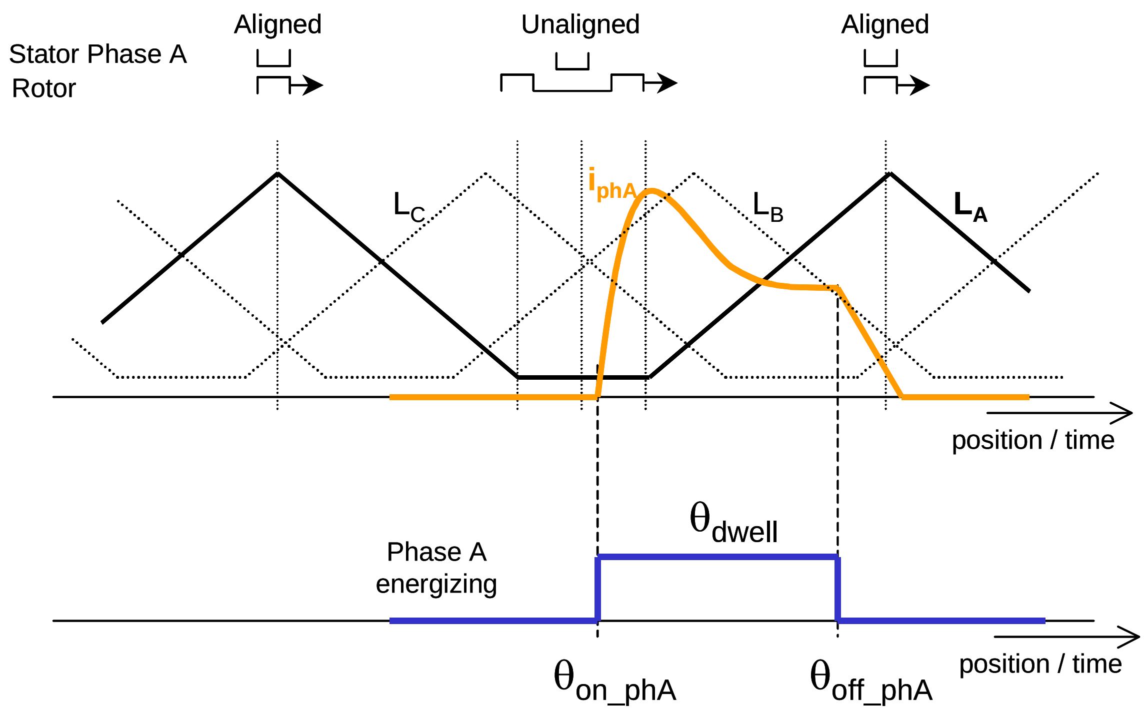

Figure 3-2. Phase Energizing

The motor itself is a low-cost, simply constructed machine. High speed operation is possible, making the motor suitable for high-speed applications, like vacuum cleaners, fans, white goods, etc. The disadvantage of the SR motor is the need for shaft position information for proper switching of individual phases. Also, the motor structure causes noise and torque ripple. The higher the number of poles, the smoother the torque ripple, but motor construction and control electronics become more expensive. Torque ripple can also be reduced by advanced control techniques, such as phase current profiling.

3.2 Magnetization Characteristics of an SR Motor

The SR motor is a highly non-linear system. The non-linear theory describing the behavior of the motor is readily available, and a mathematical model can be created based on the theory. On one hand, it enables simulation of the SR motor system and, on the other hand, development and implementation of sophisticated algorithms for controlling the SR motor is feasible.

The SR motor's electromagnetic circuit is characterized by non-linear magnetization. **Figure 3-3** illustrates a magnetization characteristic for a specific SR motor. It is a function between the magnetic flux, ψ, the phase current, i, and the motor position, θ. The influence of the phase current is most apparent in the aligned position, where saturation effects can be observed.

The magnetization characteristic curve defines the motor's non-linearity. The torque generated by the motor phase is a function of the magnetic flux; therefore, the phase torque is not constant for constant phase current for different motor positions. This causes SR motor torque ripple and noise.

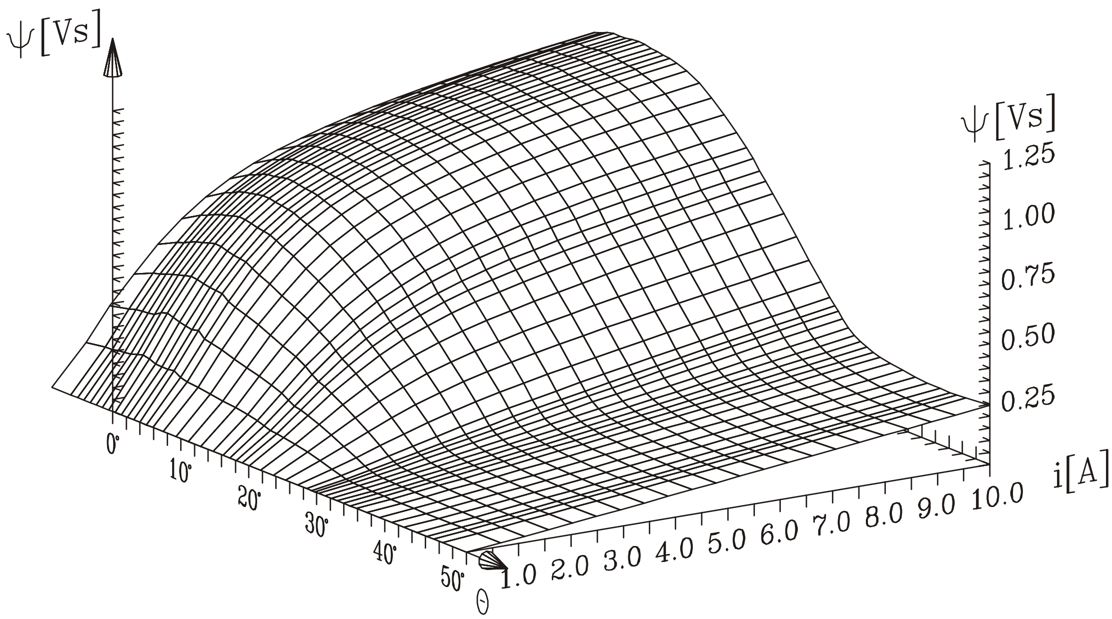

Figure 3-3. Magnetization Characteristics of the SR Motor

3.3 Digital Control of an SR Motor

The SR motor is driven by voltage strokes coupled with the given rotor position. The profile of the phase current together with the magnetization characteristic defines the generated torque and thus the motor's speed, requiring electronic control for motor operation. Several power stage topologies are being implemented, according to the number of motor phases and the desired control algorithm. The particular structure of the SR power stage structure defines the freedom of control for an individual phase.

The most used topology is the power stage with two independent power switches per motor phase. Such a power stage for 3-phase SR motors is illustrated in **Figure 3-4**. It enables control of the individual phases fully independent of the other and thus permits the widest freedom of control. Other power stage topologies share some of the power devices for several phases, thus saving on power stage cost, but the phases cannot be fully controlled independently. In contrast to AC power stages, note that the particular topology of the SR power stage is fault tolerant because it eliminates the possibility of a short circuit.

During normal operation, the electromagnetic flux in an SR motor is not constant and must be built for every stroke. In the motoring period, these strokes correspond to the rotor position when the rotor poles are approaching the corresponding stator pole of the excited phase. As shown in **Figure 3-4**, in Phase A, the stroke can be established by switching the switches Q1 and Q2. At low-speed operation, the Pulse Width Modulator (PWM), applied on the corresponding switches, modulates the voltage level.

Two basic switching techniques can be applied:

- **Soft Switching**, where one transistor is left truned on during the entire commutation period and PWM is applied to the other transistor

- **Hard Switching**, where PWM is applied simultaneously to both transistors

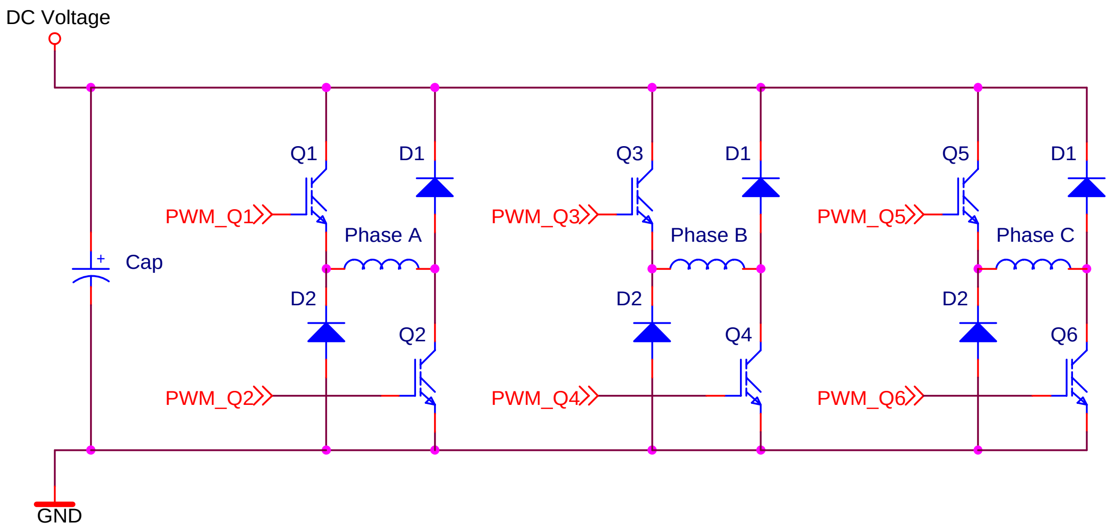

Figure 3-4. 3-Phase SR Power Stage

Figure 3-5 illustrates both soft and hard switching PWM techniques. The control signals for the upper and the lower switch of the previously described power stage define the phase voltage and thus the phase current. The soft switching technique generates lower current ripple compared to the hard switching technique. Also, it produces lower acoustic noise and less EMI. Therefore, soft switching techniques are often preferred for motor operation.

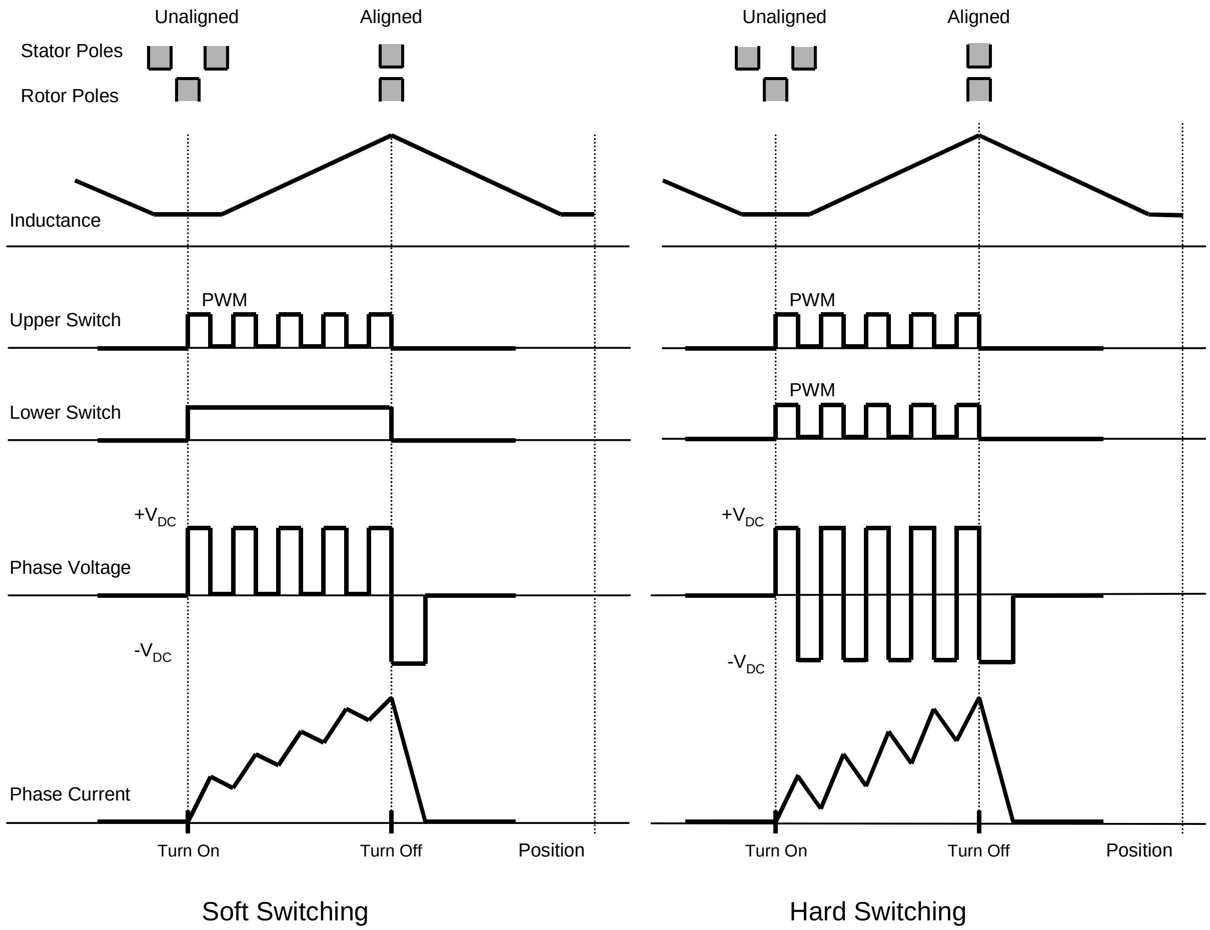

Figure 3-5. Soft Switching and Hard Switching

4. Control Techniques for Switched Reluctance Motors

A number of control techniques for SR motors exists. They differ in the structure of the control algorithm and in position evaluation. The control technique described in this application note incorporates a voltage control algorithm with the evaluation of the position using Hall sensors.

4.1 Voltage Control of an SR Motor

Voltage control of an SR motor represents one of the basic control algorithms. In the algorithm, the voltage applied to the motor phases is constant during the complete sampling period of the speed control loop and the commutation of the phases is linked with the position of the rotor.

The voltage applied to the phase is controlled directly by a speed controller. The speed controller processes the speed error - the difference between the desired speed and the actual speed - and generates the desired phase voltage. The phase voltage is defined by a PWM duty cycle implemented at the DCBus voltage of the SR inverter. The phase voltage is constant during a complete dwell angle. The technique is illustrated in

Figure 4-1. The current and the voltage profiles can be seen in **Figure 4-2**. The phase current is at its peak at the position when the inductance starts to increase (the stator and rotor poles start to overlap) due to the change in the inductance profile.

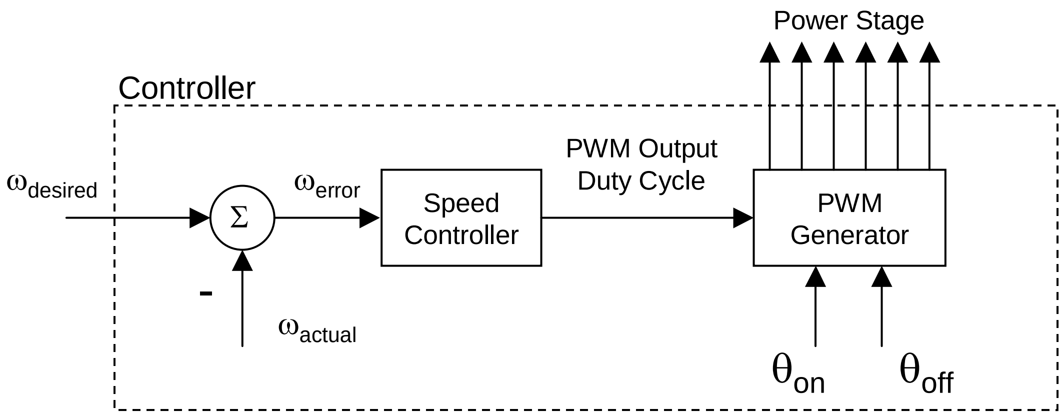

Figure 4-1. Voltage Control Technique

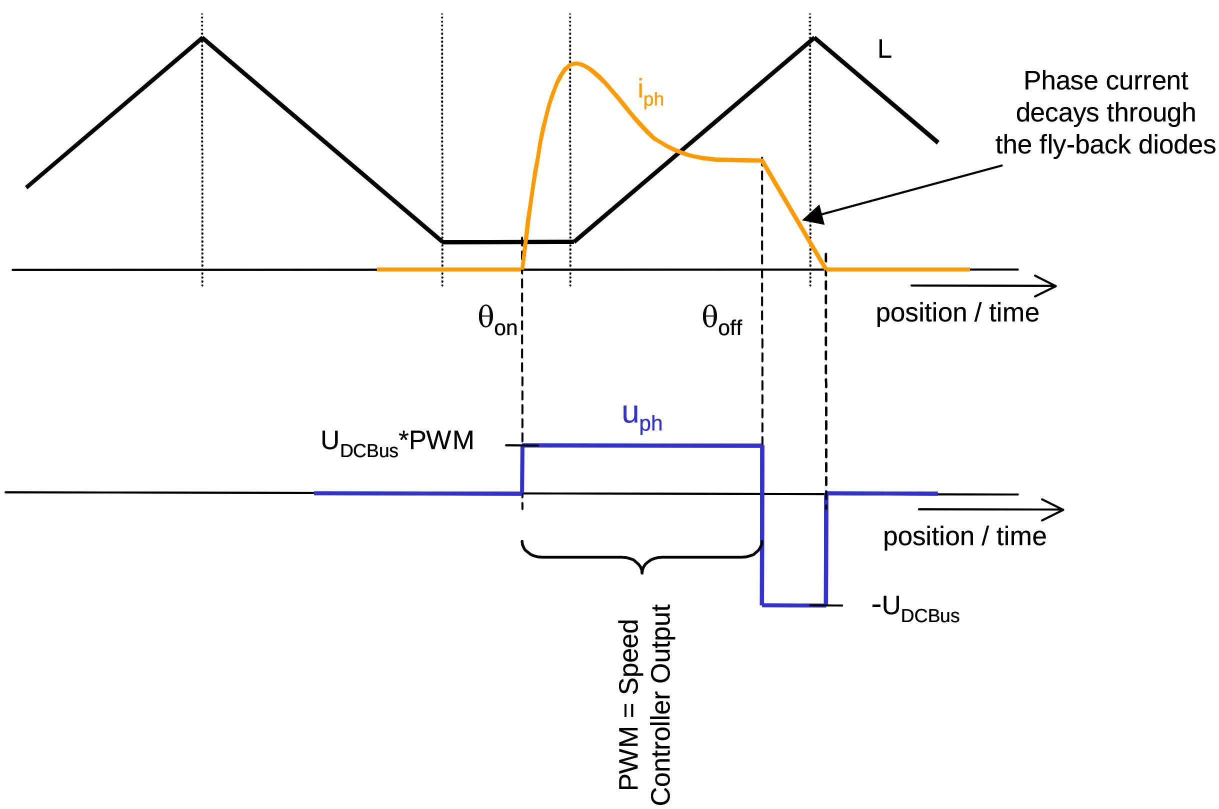

Figure 4-2. Voltage Control Technique - Voltage and Current Profiles

4.2 Position Sensing Using Hall Sensors

The SR motor requires position sensing for its operation. Hall sensors represent one widely used type of position sensor.

The position Hall sensors consist of an eight-segmented disk and three Hall sensors mounted 120^{o} from each other. The segmented disc is mounted on the motor shaft. The number of rotor poles defines the ratio between the mechanical revolution and the electrical period. If there are four rotor poles, the ratio is 4:1. In such a configuration, the generated logic signals together provide 24 edges per one mechanical revolution, or six edges per one electrical period. The electrical resolution is then 60 electrical degrees.

The signals from the sensors are positioned in such a way that the rising edge of the position sensor signal is in the aligned position of the individual phase:

- Rising edge on Sensor A for aligned position of Phase A
- Rising edge on Sensor B for aligned position of Phase B
- Rising edge on Sensor C for aligned position of Phase C

Both the idealized profile of inductances and the alignment of the Hall sensors are illustrated in **Figure 4-3**. The figure also illustrates the selection of energized phases for motor start-up and running, described in the following sections.

Note that the shape of the signals does not depend on the direction of rotation; the rising edge is always in the aligned position.

4.3 Control Technique of SR Motors Using Hall Sensors

The control technique must provide both reliable motor start up from any position and the proper commutation of the phases during motor operation. Both start up and commutation are based on the position of the Hall sensors.

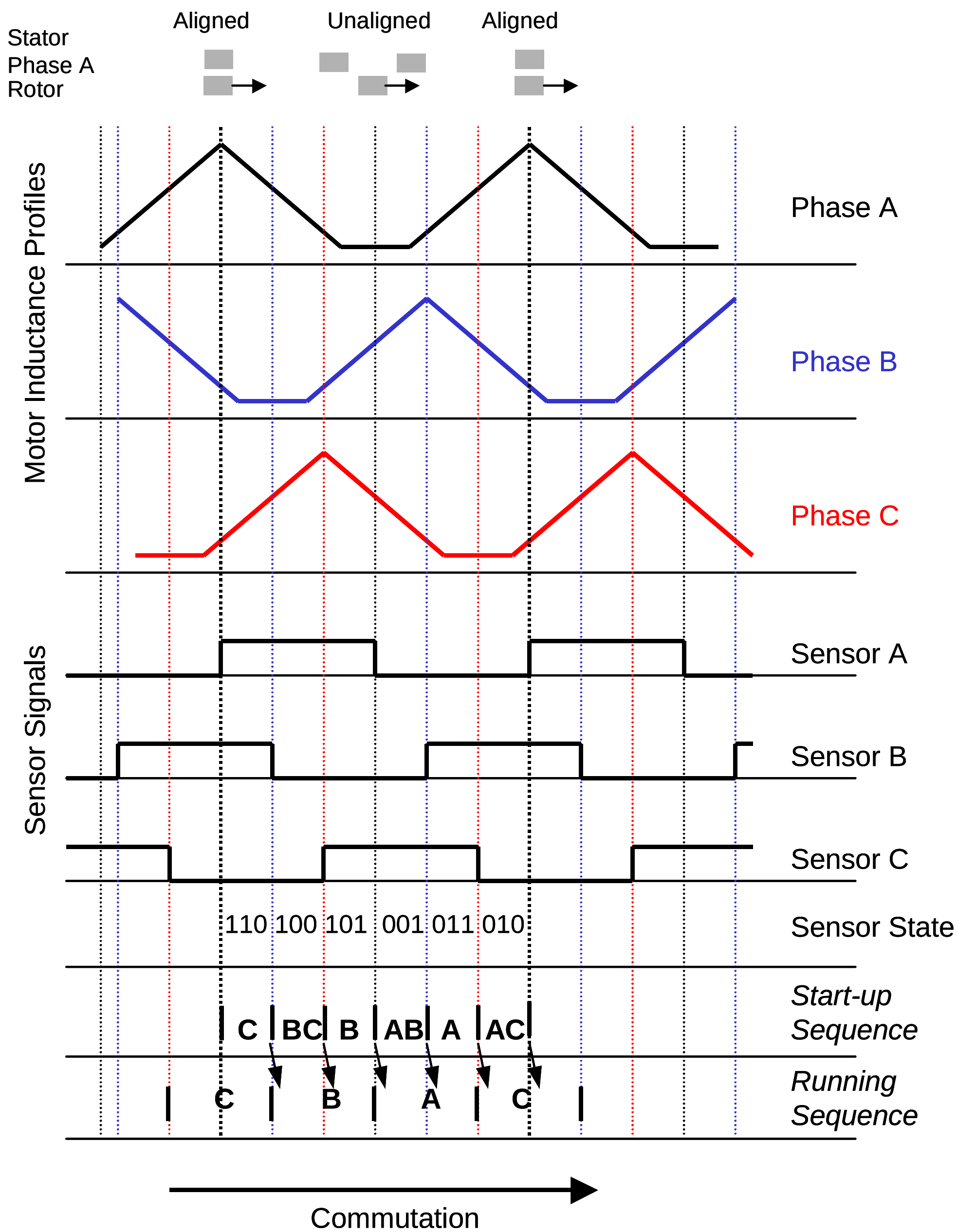

Figure 4-3. Control Technique

4.3.1 Start-up

The start-up algorithm provides the motor's start process. During start-up, the state of the individual position sensors is sensed, and the phases are powered in the defined sequence in order to start the motor in the defined direction of rotation.

Figure 4-4 illustrates the start-up procedure flow chart.

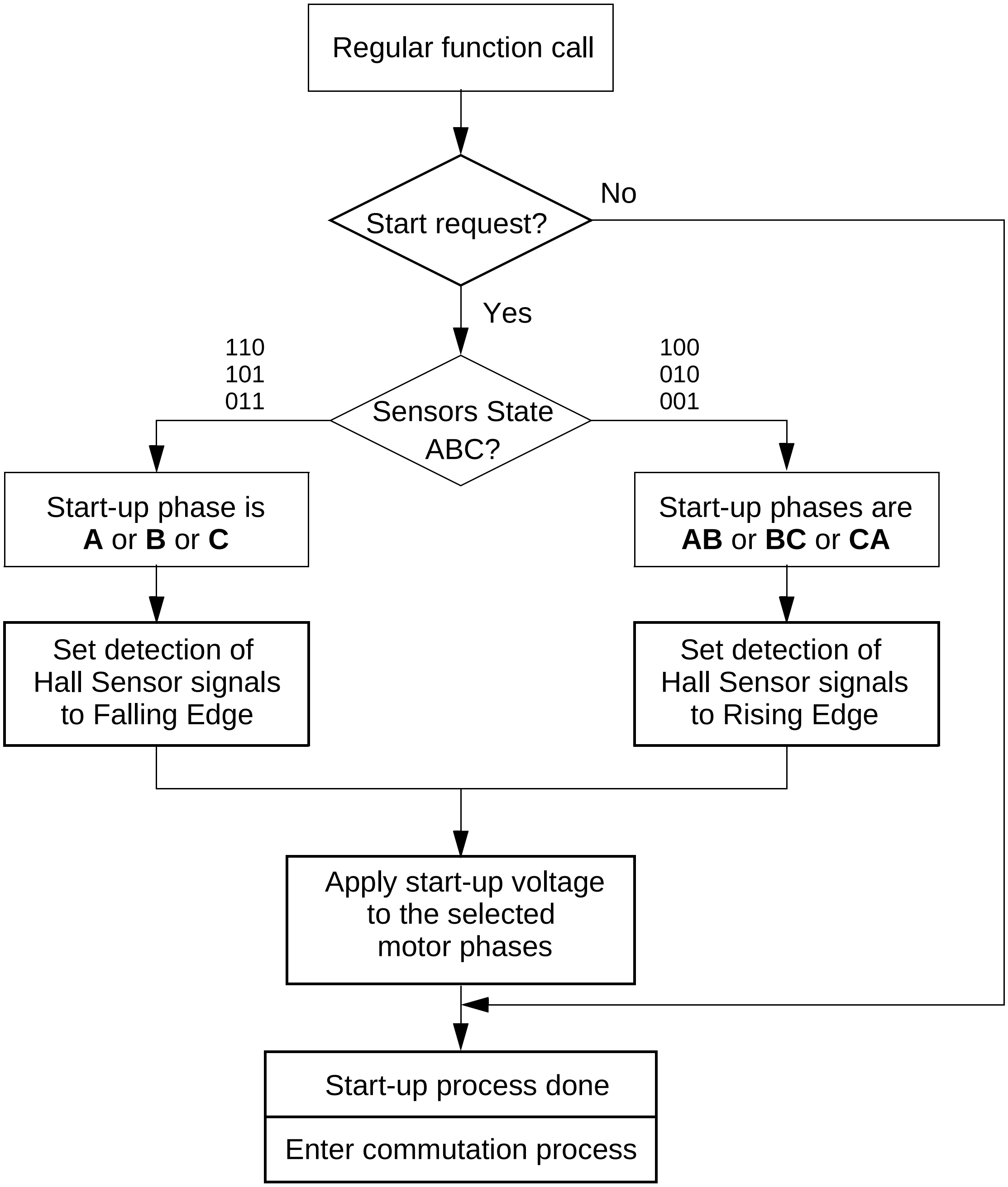

Figure 4-4. Flow Chart—Start-Up Process

During the start-up process, the start-up command is checked regularly. When the start-up command is accepted, the actual state of the sensors is checked and the desired start-up phases are selected. The selection of the phases depends on the actual start-up position of the rotor. It is influenced by the following aspects:

- The position sensor has limited precision due its mechanical construction, with the result that the actual rotor position might be shifted somewhat with respect to the sensed position
- The resolution of the sensor during stand still is 60 electrical degrees (six pulses per electrical period). It is too wide for a reliable determination of the single phase or two phases that should be powered first to start the motor.

Due to these limitations of the Hall sensors, there are some start-up positions where just one phase can be powered, and other start-up positions where two motor phases must be powered simultaneously in order to start the motor reliably. The selection of the start-up phases is defined by the torque that the individual phases can generate in the start-up position, and its relation to the Hall sensors.

As stated in **Section 3.**, when the voltage is applied to the stator phase, the motor creates torque in the direction of the increasing inductance. Note that the value of the applied start-up voltage is limited by a maximal phase current, so it must depend on the parameters of the motor.

In several states of the Hall sensors, the inductance profile is steadily rising over an entire 60 electrical degrees interval (see **Figure 4-3**), enabling the motor to generate the desired start-up torque sufficient to power just one phase :

— Sensor state “110” : power Phase C

— Sensor state “101” : power Phase B

— Sensor state “011” : power Phase A

When the appropriate voltage is applied to the selected phase, the motor starts to rotate.

In the other positions, the inductance is not steadily rising over an entire interval of 60 electrical degrees; therefore, the desired torque might not be generated. For example, if the interval of 60 electrical degrees of the Hall sensors state is “100”:

- Phase A generates a negative torque
- Phase B cannot be powered alone due to the flat inductance at the beginning of the interval, causing poor torque generation
- Phase C cannot be powered alone due to the flat inductance at the end of the interval, causing poor torque generation. Also, the possible inaccuracy of the sensor can even cause decreasing inductance at the end of the interval, generating a negative torque.

Therefore, for the Hall sensors’ state of “100”, both Phases B and C must be powered simultaneously, ensuring the generation of the correct torque for the full interval of 60 electrical degrees.

Similarly, simultaneously powered phases are defined for the other “non-definite” positions:

— Sensor state “100” : power Phases B and C
— Sensor state “001” : power Phases A and B
— Sensor state “010” : power Phases A and C

When both phases are powered, the motor starts to move in the direction of increasing inductance. When the Hall sensors generate a rising edge, the corresponding phase must be turned off, because it approaches the interval of falling inductance (negative torque), and only one phase stays powered.

When the motor starts to rotate in the desired direction, the start-up procedure is left and the commutation process is entered; see **Figure 4-4**.

4.3.2 Commutation

During a standard operation in the commutation phase, just one phase is powered at a time. The control technique uses fixed turn-on and turn-off angles for switching the phases. The speed of the motor is controlled by the voltage, applied to the motor phase using the PWM technique.

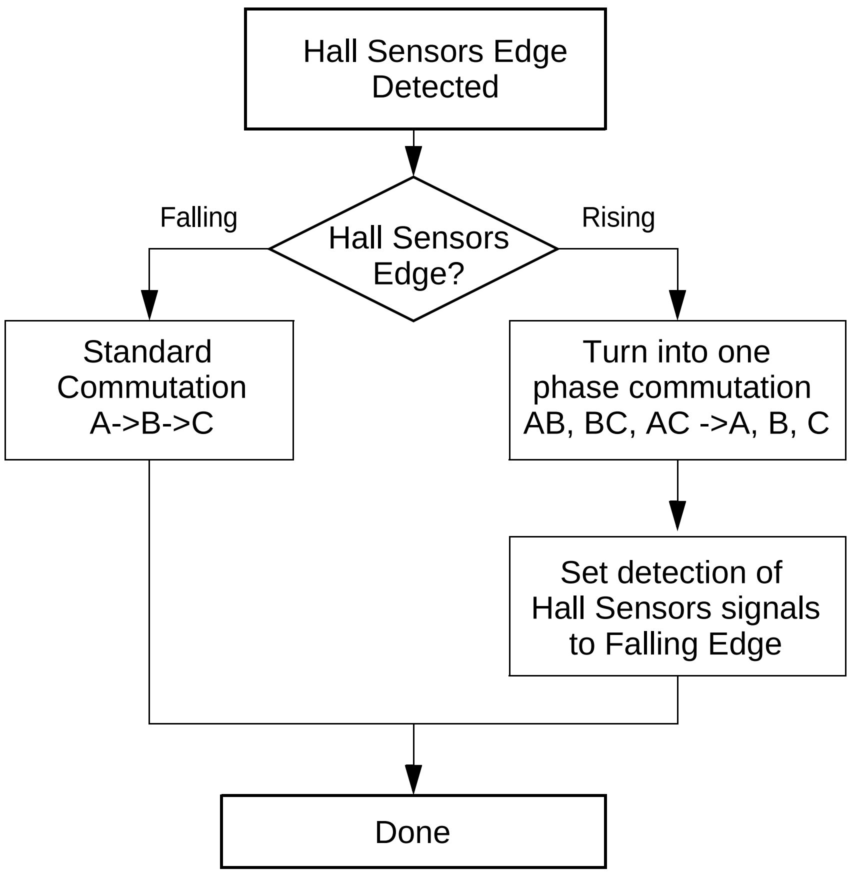

Figure 4-5. Flow Chart—Commutation Phase

Figure 4-5 illustrates the flow chart for the commutation phase. The process begins with the sensing of the edge from the Hall sensor signals. First, the polarity of the signal edge is evaluated. This is important for a smooth transition between the start-up and the commutation phase. For a rising signal edge, two phases were powered during the motor start-up. One of these two phases is turned off and only one phase stays powered:

— Rising edge of Sensor A: turn off Phase A, keep Phase C turned on
— Rising edge of Sensor B: turn off Phase B, keep Phase A turned on
— Rising edge of Sensor C: turn off Phase C, keep Phase B turned on

The edge polarity detection of the Hall sensor signals is then changed to the falling edge, thus switching to the standard commutation process.

During the standard commutation procedure, only falling edges from the Hall sensors are detected. The turn-on and turn-off angles are directly determined by the position signals of the Hall sensors. When the falling edge on the sensor signal occurs, the corresponding phase is turned off and the following phase in the direction of rotation is turned on. The phases commutate in the sequence C-B-A-C.

— Falling edge of Sensor A: turn off Phase A, turn on Phase C
— Falling edge of Sensor B: turn off Phase B, turn on Phase A
— Falling edge of Sensor C: turn off Phase C, turn on Phase B

The commutation algorithm is accessed each time the falling edge is sensed. The phases of the motor are sequentially powered and the motor rotates in the desired direction of rotation.

The algorithm introduced here is simple, but gives acceptable results for the considered speed range and is a good starting point for further development of the SR algorithms. The improvement of the method can be represented by the adjustment of the turn-on and turn-off angles, according to the motor speed. It is especially important for higher speeds, where the increase of the phase voltage is not possible due to the PWM saturation of 100%. The current controlled SR drive can be developed when an inner current control loop is added. Also, the sensorless SR drive can be developed on the basis of the algorithm introduced.

5. System Design Concept

5.1 System Outline

The system is designed to drive a 3-phase SR motor. The application meets the following performance specifications:

- Voltage control of SR motor using Hall sensors
- Targeted for 56F8xxEVM or 56F83xxEVM
- Runs on 3-phase SR High-Voltage (HV) motor control development platform at a variable line voltage of 115-230V AC (range -15% to +10%)
- Runs on 3-phase SR Low-Voltage (LV) motor control development platform at a voltage of 12V DC
- Control technique incorporates:

- Voltage SRM control with speed-closed loop
- Rotation in one direction
- Motoring mode
- Start from any motor position without rotor alignment
- Minimum speed of 700rmp
- Maximum speed of 2500rpm for an HV SR motor at an input power line of 230V AC
- Maximum speed of 1500rpm for an HV SR motor at an input power line of 115V AC
- Maximum speed of 1500rpm for an LV SR motor at an input power line of 12V DC

- Manual interface
  - RUN / STOP switch
  - UP / DOWN push button control
  - LED indication
- PC master software control interface
  - Motor start / stop
  - Speed set-up
- PC master software monitor
  - PC master software graphical control page
    - Required speed
    - Actual motor speed
    - PC master remote control mode
    - Start / stop status
    - Drive fault status
    - DCBus voltage level
    - System status
  - PC master software speed scope
    - Observes actual and desired speeds
- Fault protection for:
  - DCBus overvoltage

— DCBus undervoltage

— DCBus overcurrent

— Overtemperature

The SR drive is designed to power both low-voltage and high-voltage SR motors equipped with Hall sensors. The motors have the following specifications:

Table 5-1. Specifications of the Motor and Hall Sensors

Motor	Motor Type	3-Phase SR Motor 6 / 4 (Stator / Rotor) Poles	
	Speed Range	< 5000rpm	
	Nominal Voltage	High-Voltage Motor	300V
		Low-Voltage Motor	10V
	Nominal Current	High-Voltage Motor	3 x 1.2A
		Low-Voltage Motor	3 x 28.5A
Position Sensor	Sensor Type	3-Phase Hall sensors	
	Number of Disc Segments	8	
	Sensor layout	Sensors distributed at 60 mechanical degrees angles to one another	

5.2 Application Description

A standard system concept is chosen for the drive; see **Figure 5-1**. The system incorporates the following hardware components:

- Choice of development platforms:
  - — **3-phase SR high-voltage development platform**, a high-voltage power stage with optoisolation board and a high-voltage SR motor with attached brake

  or

  - — **3-phase SR low-voltage development platform**, a low-voltage power stage and a low-voltage SR motor with attached brake
- Feedback sensors:
  - — Position (Hall sensors)
  - — DCBus voltage
  - — DCBus current
  - — Temperature
- Digital Signal Controller

The controller runs the main control algorithm. According to the user interface input and feedback signals, it generates 3-phase PWM output signals for the SR motor inverter.

The drive can basically be controlled in two operating modes:

- In the **Manual** operating mode, the required speed is set by a RUN / STOP switch and UP and DOWN push buttons
- In the **PC master software** operating mode, the required speed is set by the PC master software

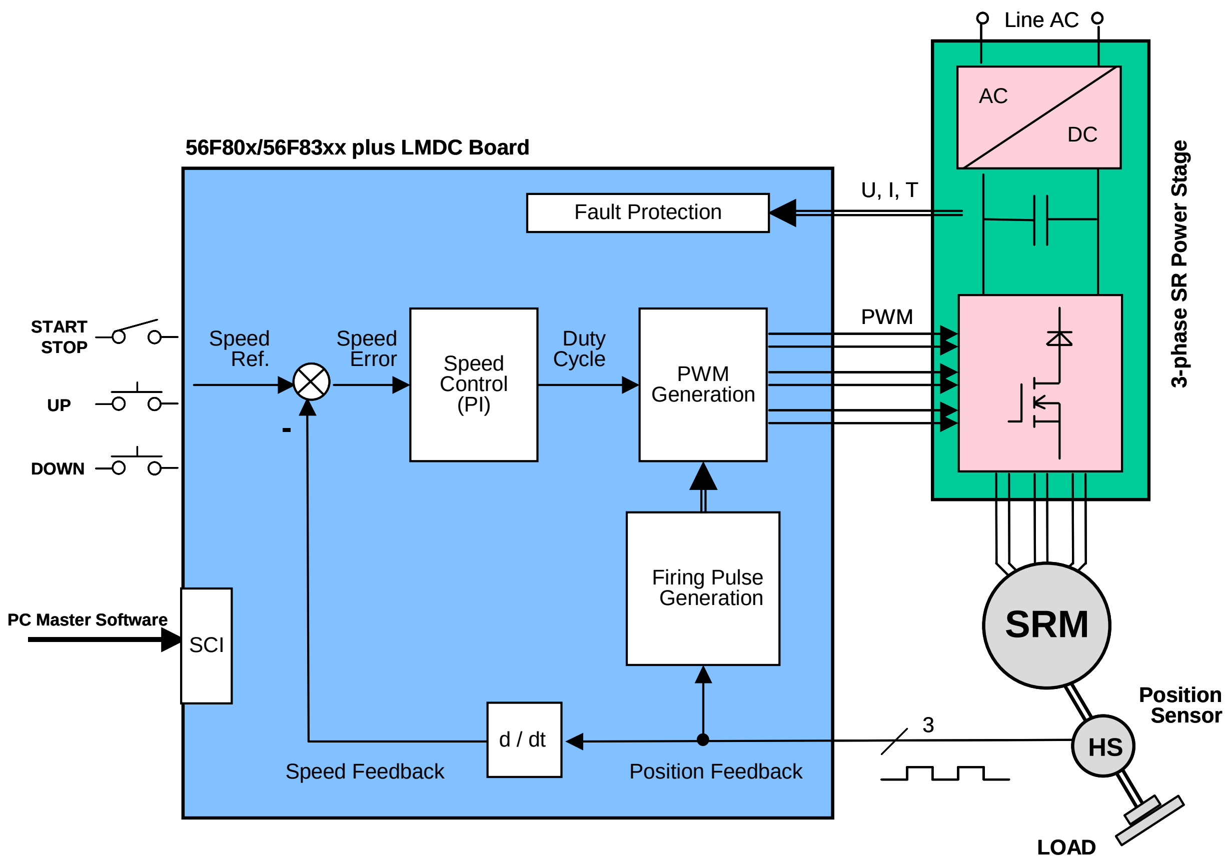

Figure 5-1. System Concept

After reset, the drive is initialized and automatically enters the Manual operating mode.

Note: PC master software can only take over control when the motor is stopped.

When the Start command is detected (using the RUN / STOP switch or the PC master software button, “Start”) and no fault is pending, the application can be started. First, the start-up sequence is performed. The state of the Hall sensors position signals is sensed and the individual motor phases are powered in order to start the motor in the requested direction of rotation. When the motor begins to rotate, the commutation process is enabled.

The edges of the Hall sensors' position signals are captured by the Input Capture function of the controller's on-chip Quad Timer module. The switching pattern for the PWM control signals is determined based on these captured signals.

The actual speed of the motor is determined by the Hall sensor signals. The reference speed is calculated according to the control signals (RUN / STOP switch, UP / DOWN push buttons) and PC master software commands (when controlled by PC master software). The acceleration / deceleration ramp is implemented. The comparison between the reference speed and the measured speed gives a speed error. Based on the speed error, the speed controller generates the desired PWM duty cycle. Finally, according to the determined switching pattern and the calculated duty cycle, the controller's on-chip PWM module generates PWM signals for the SR motor power stage.

The DCBus voltage, the DCBus current and the power stage temperature are measured during the control process. The measurements are used to protect the drive from DCBus overvoltage, DCBus undervoltage, DCBus overcurrent and overtemperature. The DCBus undervoltage and overtemperature protection are performed by software, while the DCBus overcurrent and the DCBus overvoltage fault signals utilize the Fault inputs of the controller's on-chip PWM module. If any of the previously mentioned faults occur, the PWM outputs are disabled in order to protect the drive. The fault state can only be exited when the fault conditions have disappeared and the RUN / STOP switch is moved to the STOP position.

5.3 Hardware Implementation

This section details the hardware implementation for targeting the 56F83xxEVM.

As stated earlier, the application runs on Freescale's motor control controllers using EVM Boards and a dedicated 3-Phase SR platform.

The application can be controlled by Freescale's motor control controller 56F83xx.

The application can run on both of the following motor platforms:

- 3-Phase SR Low-Voltage platform
- 3-Phase SR High-Voltage platform

The application hardware set up is shown in **Figure 5-2** and **Figure 5-3**. The application software is identical for all controllers and both SR platforms.

A detailed application hardware set up can be found in the document **Targeting Freescale's 56F83xx Platform**.

Dedicated descriptions of individual boards can be found in the comprehensive user manual belonging to each board or on the Freescale website:
www.freescale.com
Each manual includes the schematic of the board, description of individual function blocks and a bill of materials. An individual board can be ordered as a standard product. The following chapters illustrate the configuration of the both the SR high-voltage platform and the SR low-voltage platform, together with references to the documentation.

5.3.1 3-Phase SR Low-Voltage Platform

The system configuration is shown in **Figure 5-2**.

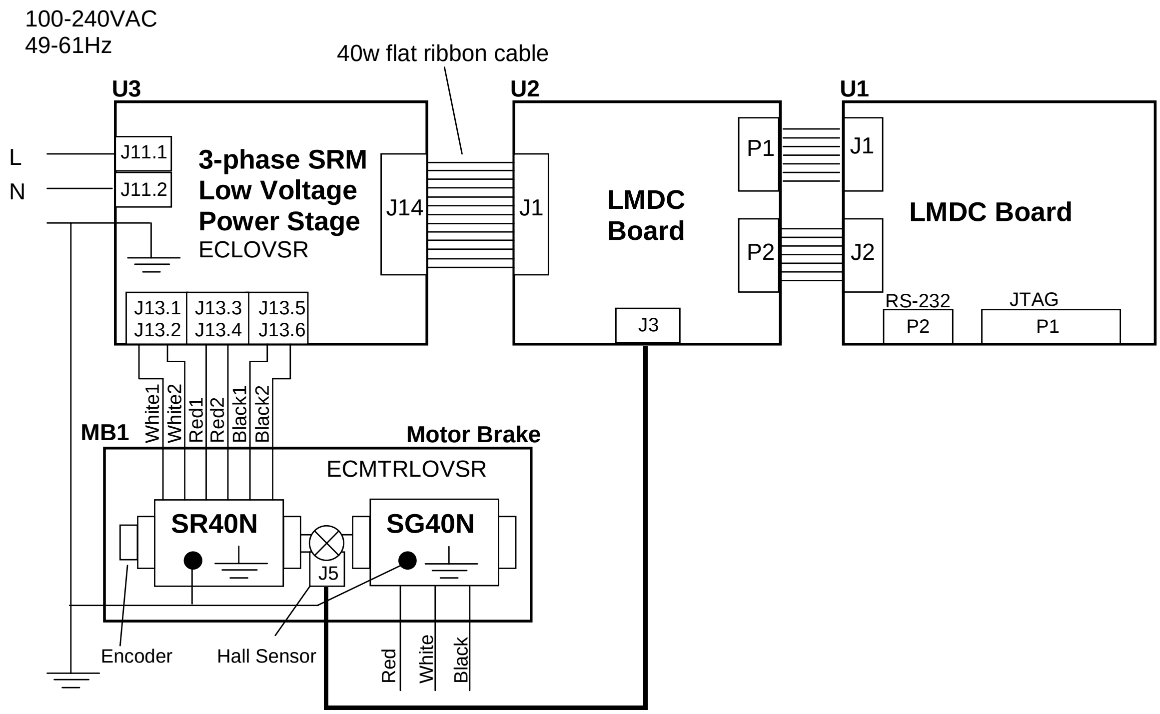

Figure 5-2. 3-Phase SR Low-Voltage Platform Configuration

All system parts are supplied and documented according to the following references:

- U1 - Controller Board for 56F8300
  - — Supplied as MC56F83xxEVM
  - — Described in **56F83xxEVMUM Evaluation Module User's Manual** for the specific device being implemented
- U2 - Legacy Motor Daughter Card (LMDC)
  - — Supplies limited; please contact your Freescale representative
- U3 - 3-Phase SR Low-Voltage Power Stage
  - — Supplied as Freescale Part # ECLOVSR
  - — Described in **Freescale Embedded Motion Control 3-Phase Switched Reluctance Low-Voltage Power Stage User's Manual**
- MB1 - Motor-Brake SR40N + SG40N
  - — Supplied as Freescale Part # ECMTRLOVSR

5.3.2 3-Phase SR High-Voltage Platform

The system configuration is shown in **Figure 5-3**.

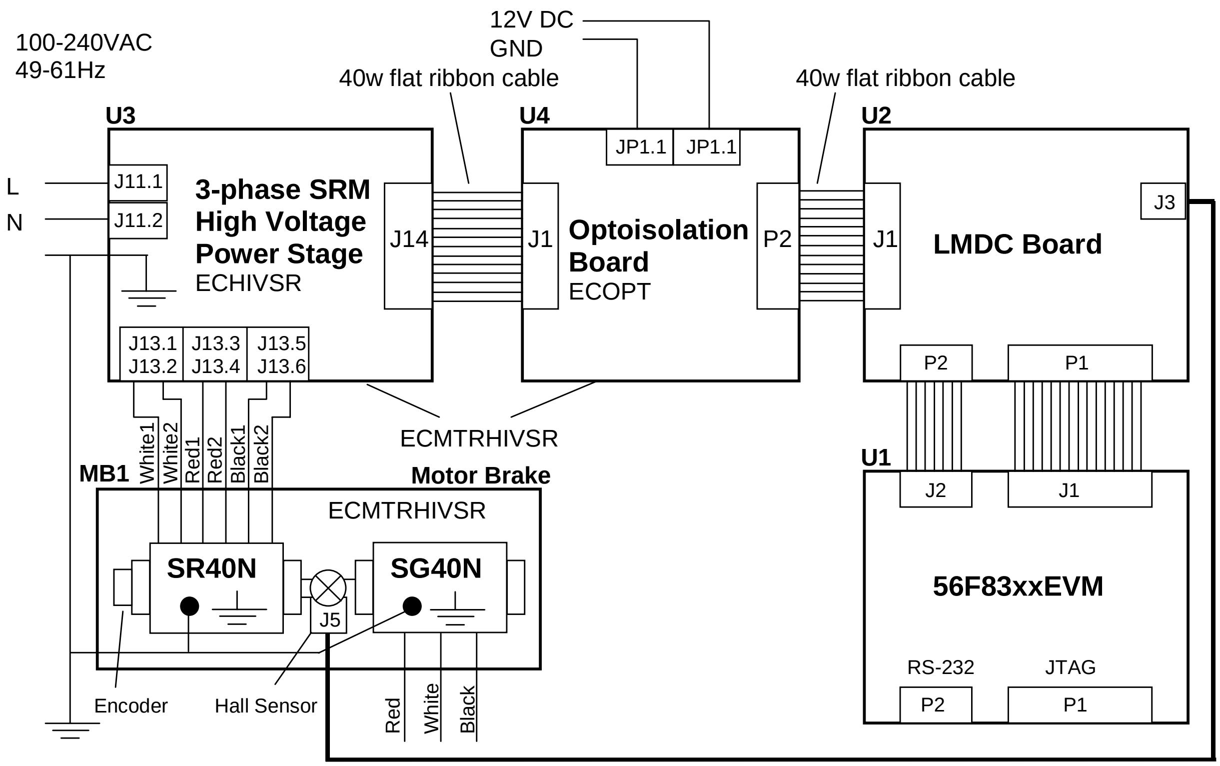

Figure 5-3. 3-Phase SR High-Voltage Platform Configuration

All system parts are supplied and documented according to the following references:

- U1 - Controller Board for 56F8300
  - Supplied as MC56F83xxEVM
  - Described in **56F83xxEVMUM Evaluation Module Hardware User's Manual** for the specific device being implemented
- U2 - Legacy Motor Daughter Card (LMDC)
  - Supplies limited; please contact your Freescale representative
- U3 - 3-Phase SR High-Voltage Power Stage
  - Supplied in a kit with the Optoisolation Board as Freescale Part #ECOPTHIVSR
  - Described in **Freescale Embedded Motion Control 3-Phase Switched Reluctance High-Voltage Power Stage User's Manual**
- U4 - Optoisolation Board
  - Supplied with a 3-phase SR High Voltage Power Stage as Freescale Part #ECOPTHIVSR

    Or

- — Optoisolation board is supplied separately as Freescale Part #ECOPT
- — Described in **Optoisolation Board User's Manual**
- MB1 Motor-Brake SR40V + SG40N
  - — Supplied as Freescale Part #ECMTRHIVSR

Warning: To avoid electric shock and potential damage to the development equipment, the use of optoisolation (optocouplers and optoisolation amplifiers) is strongly recommended during development.

6. Software Design

This section explains software design for targeting the 56F83xxEVM and describes the design of the software blocks of the drive. The software will be described in terms of:

- Control algorithm data flow
- Software implementation

6.1 Data Flow

The control algorithm of the closed-loop SR drive is described in **Figure 6-1**.

The desired speed is set either by using the manual interface, or by PC master software. The speed command is generated according the defined acceleration ramp of the motor. The actual speed is calculated from the time captured between the detected edges of the Hall sensors. The speed controller utilizes both the speed command and the actual speed and generates the desired PWM duty cycle.

When the edge from the Hall sensor signal is detected, a new commutation pattern for the motor phases is generated. The output voltage is then generated according to the desired duty cycle, the actual DCBus voltage, and the new commutation pattern, using the controller's on-chip PWM module.

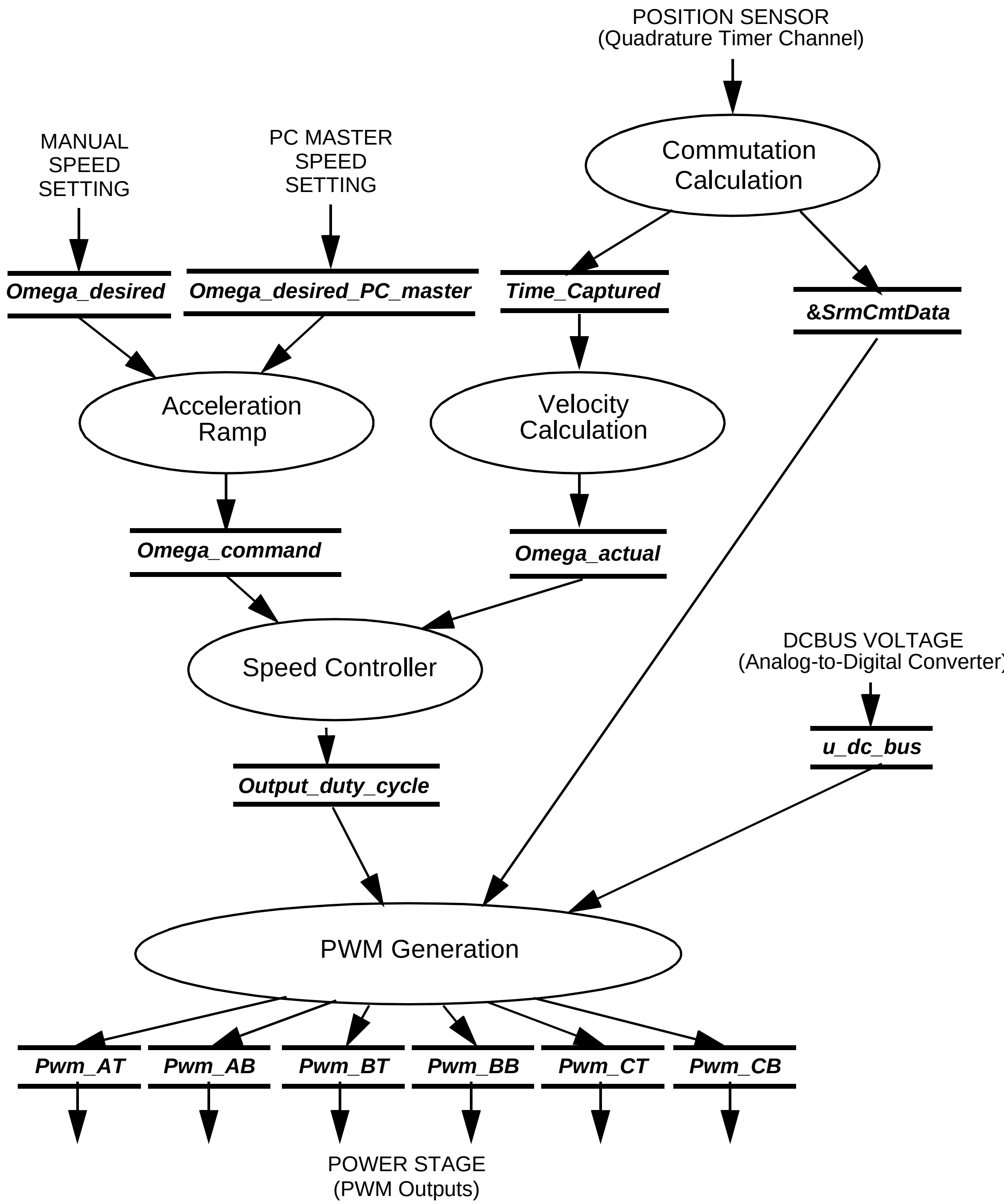

Figure 6-1. Data Flow

The individual processes are described in detail in the following sections.

6.1.1 Acceleration Ramp

The process calculates the actual speed command based on the desired speed according to the acceleration / deceleration ramp. The desired speed is controlled either manually using the push buttons (Manual operating mode), or by PC master software (PC master software operating mode).

6.1.2 Commutation Calculation

The process services the position Hall sensor signals. It generates PWM commutation patterns and captures the time between the last two edges of the Hall sensor signals.

The Hall sensors generate a stream of pulses that are directed to the on-chip Quad Timer module. Since the position sensor utilizes three Hall sensors, three channels of the Quad Timer are used. The input capture function of the Quad Timer invokes the calculation of the process when the correct edge of the Hall sensor appears.

The controller's on-chip PWM module is used in the mode of generation of independent output signals that can be controlled either by software or by the PWM module.

The commutation technique distinguishes three following cases:

- When the PWM output must be modulated, the PWM generator controls the channel directly
- When the PWM output must be switched to the inactive state (0), the software output control of the corresponding PWM channel is handed over and the channel is turned off manually
- When the PWM output must be switched to the active state (1), the software output control of the corresponding PWM channel is handed over and the channel is turned on manually

The on-chip PWM module enables control of the outputs of the PWM module either by the PWM generator or by using the software. Setting the output control enable bit, OUTCTLx, enables software to drive the PWM outputs instead of the PWM generator. In an independent mode, with OUTCTLx = 1, the output bit OUTx controls the PWMx channel. Setting or clearing the OUTx bit activates or deactivates the PWMx output. The OUTCTLx and OUTx bits are in the PWM output control register.

The control technique requires the preparation of the output control register. For the calculation of the OUTCTLx and OUTx bits in the PWM output control register, a dedicated commutation algorithm, 3-Phase SR Motor Commutation Handler for Hardware Configuration 2-Switches-per-Phase, *srmcmt3ph2spp*, has been developed. The algorithm generates the output control word according to the desired action and the desired direction of rotation. For example, when Phase A must be turned off, the algorithm sets the corresponding OUTCTLx bits to enable the output control of the required PWMs and clears OUTx bits to turn the PWMs off. The other bits of the output control register are not affected. A detailed description of the algorithm can be found in the Processor Expert documentation.

6.1.3 Velocity Calculation

This process calculates the motor's actual speed. It reads the time between the following falling edges of the Hall sensors output and calculates the actual motor speed, *Omega_actual*. A software filter of the speed measurement is incorporated in the process for better noise immunity. The actual motor speed is calculated as an average value of the last four measurements.

6.1.4 Speed Controller

The process calculates the output duty cycle of the PWM according to the speed error. The speed error is the difference between the actual speed, *Omega_actual*, and the speed command, *Omega_command*. The PI controller is implemented. The constants of the speed controller are tuned experimentally according to the actual load and the rating of the power stage.

6.1.5 PWM Generation

This process sets the on-chip PWM module for generation of the control pulses for the 3-phase SR motor power stage. The generation of the pulses is based on the software control register, generated by the commutation calculation process, on the required duty cycle generated by the process speed controller. The calculated software control word is loaded into the proper PWM register and the PWM duty cycle is updated according to the required duty cycle. The PWM generation process is accessed regularly in a rate given by the PWM frequency. It is frequent enough to ensure the precise generation of commutation pulses.

6.2 Software Implementation

The general software diagram incorporates the Main routine entered from reset and the interrupt states (see **Figure 6-2**).

The Main routine provides board identification, initialization of the controller, and initialization of the application, then enters an infinite background loop, which contains a Scheduler routine.

The Scheduler routine provides the timing sequence for two tasks, Timeout 1 and Timeout 2. The Timeout 1 and Timeout 2 flags are set by software timer interrupts. The Scheduler utilizes these flags and calls the required routines:

- The routine in Timeout 1:
  - Handles the user interface
  - Calculates the required speed, the start-up routines and the speed ramp (acceleration / deceleration)
- The routine in Timeout 2:
  - Executes the speed controller

To avoid software bottlenecks, the Timeout 1 and Timeout 2 tasks are performed in the run state, instead of in the interrupt routines. Since the usual time periods are in the range of milliseconds, this solution is fully sufficient. Note that these periods define the critical time period for the task scheduler.

The following interrupt service routines are utilized:

- Input Capture ISR, which services signals generated by Hall sensors
- Fault ISR, which services faults invoked by external hardware faults
- PWM Reload ISR, which services an update of the PWM registers
- Timer ISR, which services the generation of a time base for software timers
- ADC ISR, which services the results of an Analog-to-Digital conversion
- SCI ISR, which services the communication with the PC master software

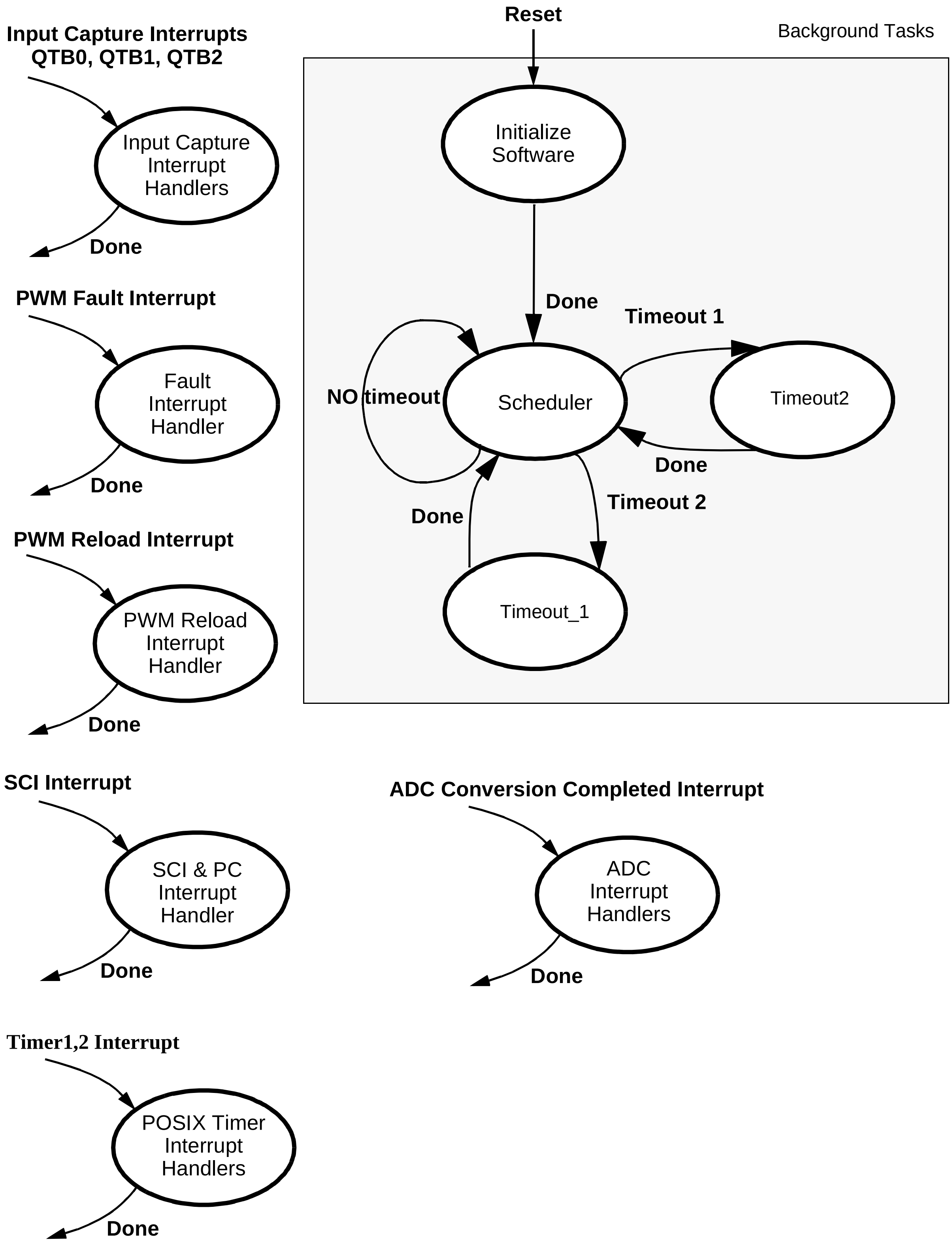
Reset
Background Tasks
Input Capture Interrupts
QTB0, QTB1, QTB2
Input Capture Interrupt Handlers
Done
Initialize Software
Done
Timeout 1
PWM Fault Interrupt
Fault Interrupt Handler
NO timeout
Scheduler
Timeout2
Done
Done
Timeout 2
Done
PWM Reload Interrupt
PWM Reload Interrupt Handler
Timeout_1
Done
SCI Interrupt
ADC Conversion Completed Interrupt
SCI & PC Interrupt Handler
ADC Interrupt Handlers
Done
Done
Timer1,2 Interrupt
POSIX Timer Interrupt Handlers
Done

Figure 6-2. State Diagram - General Overview

6.2.1 Initialization

The Main routine provides initialization of the controller:

- Disables interrupts
- Initializes the controller's PLL
- Disables COP and LVI
- Initializes POSIX Timers for scheduler time reference:
  - Timer 1 for time reference Timeout 1
  - Timer 2 for time reference Timeout 2
- Initializes LED
- Initializes PWM module:
  - Edge-aligned independent PWM mode, positive polarity
  - PWM modulus defines the PWM frequency
  - PWM interrupt reload occurs each PWM pulse
  - FAULT2 (DCBus overcurrent fault) in Manual mode, interrupt enabled
  - FAULT1 (DCBus overvoltage fault) in Manual mode, interrupt enabled
  - Associate interrupt with PWM reload event
  - Associate interrupt with PWM fault event
- Initializes Quadrature Decoder
  - Sets the on-chip digital filter of the Quadrature Decoder inputs
  - Connects Quadrature Decoder signals to Quad TimerB
- Initializes Quad TimerB, channels B0, B1, B2
  - Input Capture on the falling edge
  - Set positive polarity
  - Associate interrupt to the IC event
- Sets up I/O ports (brake, switch, push buttons)
  - Brake
  - Switch on GPIO
  - Push buttons on GPIO
- Initializes the Analog-to-Digital Converter
  - ADC set for sequential sampling, single conversion
  - Associate interrupt with ADC conversion completed event
  - Channel 0 = DCBus voltage
  - Channel 5 = temperature
- Initializes control algorithm

    - — Speed controller
    - — Control algorithm parameters
- Enables interrupts
- Starts ADC conversion
- Identifies the voltage level

6.2.2 Interrupts

The interrupt handlers have the following functions:

- *Input Capture Interrupt Handlers* read the time between the two subsequent falling edges of the Hall sensor, generate a commutation pattern and calculate the actual speed of the motor. Each of three position Hall sensors utilizes a separate Input Capture Interrupt. The description of the commutation pattern calculation is in **Section 4.3.2** and **Section 6.1.2**. Speed measurement is described in **Section 6.1.3**.
- *Fault Interrupt Handlers* take care of the fault interrupts. The PWM Fault ISR is the highest-priority interrupt implemented in the software. If DCBus overcurrent or DCBus overvoltage faults are detected, the external hardware circuit generates the corresponding fault signal that is detected on the controller's Fault input pin. The signals automatically disable the motor control PWM outputs in order to protect the power stage and generate a Fault interrupt, where the fault condition is handled. The routine records the corresponding fault source to the fault status register.
- *PWM Reload Interrupt Handler* provides phase commutation and generates the required voltage strokes for the SR motor. It loads the calculated commutation pattern to the PWM software control registers and the calculated duty cycle to all six PWM value registers.
- *POSIX Timer Interrupt Handlers* generate the two time-out references for the scheduler.
- *ADC Interrupt Handler* takes care of the ADC conversion process; it starts the conversion, reads converted value of voltage and temperature. It also provides software protection against overtemperature and DCBus undervoltage using filtered values of the DCBus voltage and the temperature of the power module. If power module overtemperature and DCBus undervoltage occur, the handlers disable the motor and set the records of the corresponding fault source to the fault status register.
- *PC and SCI Interrupt Handler*s provide the PC master software's SCI communication and service routines. These routines are fully independent of the motor control tasks.

6.2.3 Scheduler

The Scheduler routine provides the timing sequence for two timed outputs - Timeout 1 and Timeout 2.

6.2.3.1 State - Timeout 1

This state is accessed from the main Scheduler in the Timeout 1 period (10ms). The following sequence is performed:

- The status of the RUN / STOP switch is scanned. The state of the switch is filtered through two sequential samples in order to increase noise protection. An algorithm also protects the drive against "start after reset" when the RUN / STOP switch is left in the start position.
- According to the operating mode, the desired speed is calculated:
  - According to the push buttons in Manual mode
  - According to the command from the PC in PC master operating mode
- The drive is enabled or disabled according to the control commands and fault status; if the drive is stopped, all required drive variables are initialized
- If required, a start-up routine is performed and the start-up switching pattern is generated. For a detailed description, see **Section 4.3.1**.
- The command speed is calculated using the acceleration / deceleration ramp, according to the desired speed
- Subsequent ADC conversion is started, ensuring that the ADC is started periodically
- The LED is controlled according to the state of the drive; it can indicate the Stop state, the Run state or the Fault state.

6.2.3.2 State - Timeout 2

This state is accessed from the main Scheduler in the Timeout 2 period (15ms). The speed controller is performed and the corrected PWM duty cycle is calculated in this state. The speed controller constants are determined experimentally and set during the initialization of the chip.

7. Implementation Notes

This section explains implementation notes for targeting the 56F83xxEVM.

7.1 Scaling of Quantities

The SR motor control application uses a fractional representation for all real quantities, except time. The N-bit signed fractional format is represented using the 1.[N-1] format (1 sign bit, N-1 fractional bits). Signed fractional numbers (SF) lie in the following range:

$$-1.0 \leq SF \leq +1.0 - 2^{-[N-1]} \qquad \text{EQ. 7-1}$$

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 the real and the fractional representations:

$$\text{Fractional Value} = \frac{\text{Real Value}}{\text{Real Quantity Range}} \qquad \text{EQ. 7-2}$$

Where:

Fractional Value = A fractional representation of the real value [Frac16]

Real Value	=	The true value of the quantity [V, A, rpm, etc.]
Real quantity range	=	The maximum range of the quantity, defined in the application [V, A, rmp, etc.]

7.1.1 Voltage Scaling

All application voltages, (DCBus voltage, DCBus undervoltage limit, start-up voltage) are scaled relative to the maximum measurable voltage. In the case of the DCBus voltage, the scaling equation is:

$$u\_dc\_bus = \frac{V_{DC\_BUS}}{V_{MAX}} \quad \text{EQ. 7-3}$$

Where:

u_dc_bus	=	The DC Bus voltage variable [Frac16]
V_{DC_BUS}	=	The measured DCBus voltage [V]
V_{MAX}	=	The maximum measurable DCBus voltage, given by the design of the power stage [V]

In the application, V_{MAX} = 407V for the high-voltage platform and V_{MAX} = 15.9V for the low-voltage platform.

7.1.2 Speed Scaling

All application speed variables (desired speed, actual motor speed, desired start-up speed, speed command, speed limits, push button speed increments) are scaled relative to the maximum measurable speed of the drive. For the desired start-up speed, the scaling equation is:

$$omega\_desired\_startup = \frac{\omega_{start\_up}}{\omega_{MAX}} \quad \text{EQ. 7-4}$$

Where:

$omega\_desired\_startup$	=	The desired start-up speed variable [Frac16]
$\omega_{start-up}$	=	The desired start-up speed [rpm]
ω_{MAX}	=	The maximum measurable speed of the drive [rpm]

In the application, ω_{MAX} = 3000rpm.

7.1.3 Duty-Cycle Scaling

All application duty-cycle variables (output duty-cycle, high- and low-duty-cycle limits for the speed controller) are scaled relative to the maximum applicable duty cycle of the drive. For output duty cycle, the equation is:

$$output\_duty\_cycle = \frac{duty\_cycle_{output}}{duty\_cycle_{MAX}} \quad \text{EQ. 7-5}$$

Where:

output_duty_cycle	=	The output duty-cycle variable [Frac16]
$duty\_cucle_{output}$	=	The desired output duty-cycle [%]
$duty\_cycle_{MAX}$	=	The maximum applicable duty-cycle [%]

In the application, $duty\_cycle_{MAX}$ = 100 %.

7.2 Velocity calculation

The actual speed of the motor is calculated from the time, *TimeCaptured,* captured by the on-chip Quad Timer between the two following edges of the position Hall sensors. The actual speed *OmegaActual* is calculated according to the following equation:

$$OmegaActual = \frac{SpeedCalcConst}{TimeCaptured} \quad \text{EQ. 7-6}$$

Where:

OmegaActual	=	The actual speed [rpm]
TimeCaptured	=	The time, in terms of number of timer pulses, captured between two edges of the position sensor [-]
SpeedCalcConst	=	A constant defining the relation between the actual speed and number of captured pulses between the two edges of the position sensor

The constant *SpeedCalcConst* is calculated as:

$$SpeedCalcConst = 2^{15} \times \frac{SpeedMin}{SpeedMax} \quad \text{EQ. 7-7}$$

Where:

SpeedMin	=	The minimum measurable speed [rpm]
TimeCaptured	=	The maximum measured speed [rpm]

Minimum measured speed, *SpeedMin*, is calculated by the configuration of the sensors and parameters of the controller's on-chip timer, used for speed measurement. It is calculated as:

$$SpeedMin = \frac{\frac{1}{NoPulsesPerRev} \times 60}{\frac{2^{15}}{BusClockFreq} \times Presc} \qquad \text{EQ. 7-8}$$

Where:

NoPulsesPerRev	=	The number of sensed pulses of the position sensor per a single revolution [-]
Presc	=	The prescaler of the Quad Timer used for the speed measurements
SpeedCalcConst	=	The controller's Bus Clock Frequency [Hz]

Maximum measured speed, *SpeedMax,* is selected:

$$SpeedMax = k \times SpeedMin \qquad \text{EQ. 7-9}$$

In this equation, *k* is an integer constant greater than 1.

The speed calculation constant is determined as:

$$SpeedCalcConst = BusClockFreq \times \frac{60}{NoPulsesPerRev \times Presc \times SpeedMax} \qquad \text{EQ. 7-10}$$

Where:

NoPulsesPerRev	=	12 Hall sensors pulses per 1 revolution of the motor
Presc	=	128
BusClockFreq	=	$30 * 10^6$Hz
SpeedMax	=	3000rpm

Then, *SpeedCalcConst* = 390 [rev^{-1}]

8. Processor Expert (PE) Implementation

This section explains PE Implementation for targeting the 56F83xxEVM.

PE is a collection of beans, APIs, libraries, services, rules and guidelines. This software infrastructure is designed to let 56F80x and 56F8300 software developers create high-level, efficient, and portable code; the application code is available in PE. This chapter describes how the SR motor control application is written under PE.

8.1 Beans and Library Functions

The SR motor control application uses the following beans:

- ADC bean
- Timer bean
- Quad Timer bean
- Quadrature Decoder bean
- PWM bean
- LED bean
- Switch bean
- Button bean
- Brake bean

The SR motor control application uses the following library functions:

- *srmcmt3ph2sppPhOff (*srm commutation algorithm, *MC_SrmCommutation* bean*)*
- *srmcmt3ph2sppInit (*srm init algorithm, *MC_SrmCommutation* bean*)*
- *srmcmt3ph2sppSoftSw (*srm commutation algorithm, *MC_SrmCommutation* bean*)*
- *controllerPItype1* (PI controller, *MC_Controller* bean)
- *rampGetValue* (ramp generation, *MC_Ramp* library)

8.2 Beans Initialization

Each peripheral on the controller or on the EVM board is accessible through a bean. The use of bean initialization for all peripherals is described in this section. For a more detailed description of drivers, see the **Targeting Freescale 56F83xx Platform** manual for the specific device being implemented.

To use a bean, follow these steps:

- Add the required bean:
  - — Right click *Beans* under the *Processor Expert* tab in the project window
  - — Select *Add Beans*, which opens PE's *Bean Selector* window
  - — Select the desired bean
- Configure the added bean
- Call the bean's *init* function or us PE initialization, by selecting *Call init* in the CPU *init* code

Access to individual bean functions is provided from *PESL* support by the *ioctl* or *PESL* function call. To enable access to these functions, *PESL support* should be enabled in the *CPU bean* used.

8.3 Interrupts

When configuring a bean in PE, the user defines the callback functions called during interrupts.

8.4 PC Master Software

PC master software was designed to provide a debugging, diagnostic and demonstration tool for development of algorithms and applications. It consists of components running on PCs and components running on the target controller, connected by an RS-232 serial port. A small program is resident in the controller that communicates with the PC master software to parse commands, return status information to the PC, and process control information from the PC. The PC master software executing on a PC uses Microsoft Internet Explorer as a user interface to the PC.

To enable the PC master software operation on the controller target board application, add the *PC_Master* bean to the application. The *PC_Master* bean is located under *CPU External Devices -> Display* in PE's *Bean Selector*.

The PC master bean automatically includes the SCI driver and installs all necessary services. This means there is no need to install the SCI driver, because the *PC_Master* bean encapsulates its own SCI driver.

The default baud rate of SCI communication is 9600 and is set automatically by the PC master software driver.

A detailed PC master software description is provided in the PE documentation.

The 3-Phase SR Motor Control with Hall Sensors utilizes PC master software for remote control from a PC. It enables the user to:

- Take over the control of the PC master software
- Start / stop control
- Sets the motor speed

Variables read by the PC master software as a default and displayed to the user are:

- Required motor speed
- Actual motor speed
- Application's operating mode
- Start / Stop status
- DCBus voltage level
- Identified line voltage

The control page of the PC master software is illustrated in **Figure 8-1**. The profiles of the required and actual speeds can be seen in the Speed Scope window.

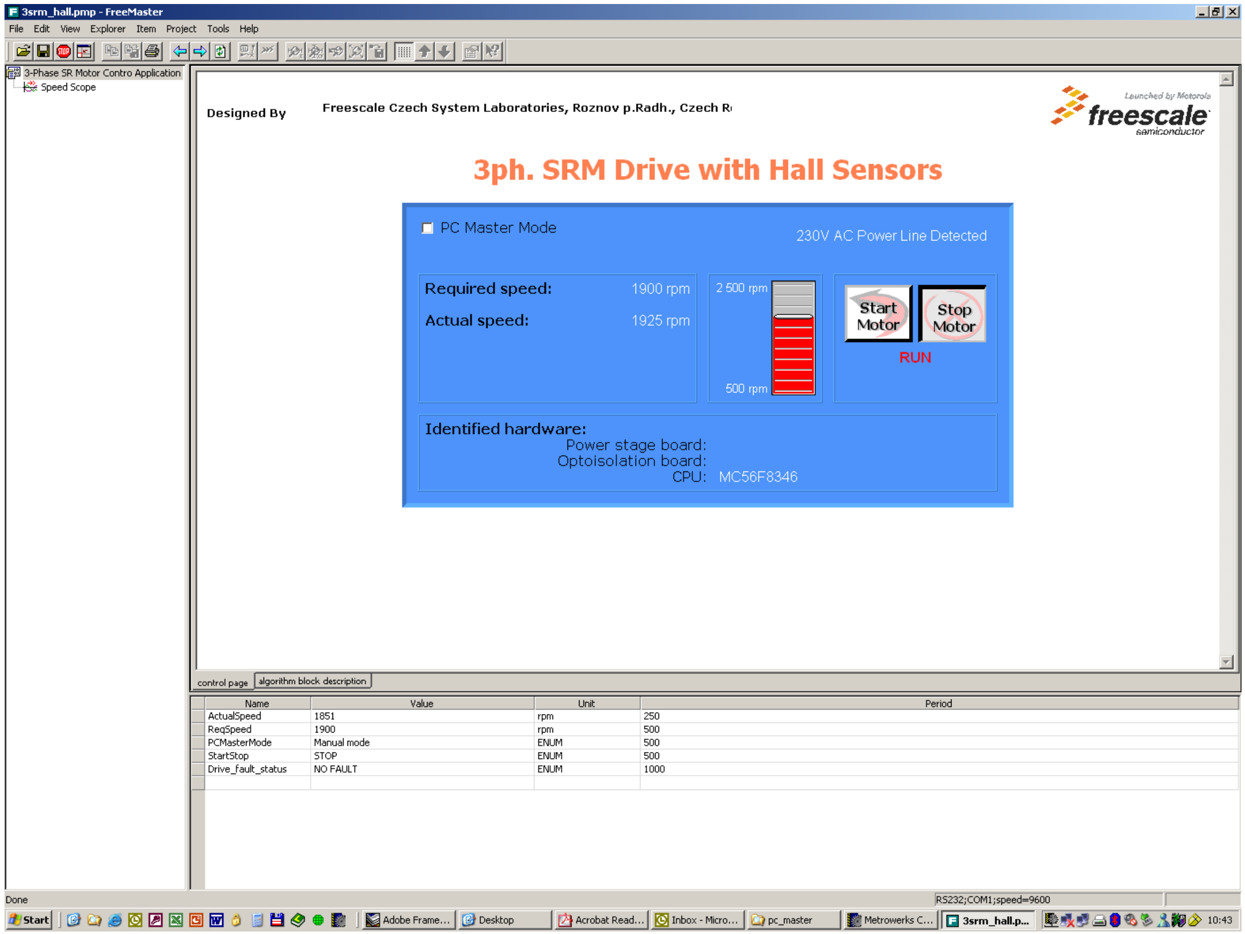

Figure 8-1. PC Control Window

9. Digital Signal Controller Use

Table 9-1 shows how much memory is needed to run the 3-phase SR drive in a speed closed-loop using Hall sensors. The PC master software recorder buffer is set to 512 words and the bulk of the controller's memory is still available for other tasks.

Table 9-1. RAM and FLASH Memory Use for PE 2.94 and CodeWarrior 6.1.2

Memory (in 16 bit Words)	**Available for 56F8300 Controller**	**Used Application + Stack**	**Used Application without PC Master Software, SCI**
Program Flash	64K	6438	3282
Data Flash	4K	21	8
Program RAM	2K	0	0
Data RAM	4K	2508 + 512 stack	324 + 512 stack

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

AN1912

Rev. 2
9/2005