Using the dsPIC30F for Sensorless BLDC Control

Author: Charlie Elliott
Smart Power Solutions, LLP
Co-author: Steve Bowling
Microchip Technology Inc.

INTRODUCTION

This application note describes a fully working and highly flexible software application for using the dsPIC30F to control brushless DC (BLDC) motors without position sensors. The software makes extensive use of dsPIC30F peripherals for motor control. The algorithm implemented for sensorless control is particularly suitable for use on fans and pumps. The program is written in C and has been specifically optimized and well annotated for ease of understanding and program modification.

Software Features

- Back EMF zero-crossing routine precludes the need for position sensing components.
- Application includes adjustable parameters and two selectable starting methods to match the particular load.
- Detects if the sensorless algorithm gets lost.
- Restarts the sensorless control without stopping the motor.
- Controls braking current to regulate DC bus voltage.
- Commutation scheme allows up to 30° phase advance to be linearly introduced as the speed increases for improved motor efficiency and extended speed range.
- Four different ways of controlling the motor speed.
- Simple user interface with LCD display and push buttons lets you adjust over 40 parameters.
- Software consumes approximately 5 MIPS (worst case) and requires approximately 16 Kbytes of program memory.
- Without the user interface and debug code, the application code fits into less than 12 Kbytes of program memory, making it compatible with the smallest memory dsPIC30F device planned (dsPIC30F2010).

Known Limitations

- As delivered, the maximum output frequency at which the sensorless system works reliably is approximately 150 Hz. However, this limitation allows very common 4-pole motors to run at up to 4500 RPM.
- The output frequency can be extended up to approximately 250 Hz (7500 RPM for a 4-pole motor) if phase advance is used. Higher speeds are possible with software modifications.
- Hard modulation of diagonally opposite inverter switches is supported.
- The system supports motoring in closed-loop commutation as would be required for a typical fan or pump.

BACKGROUND

The brushless DC (BLDC) motor is used for both consumer and industrial applications owing to its compact size, controllability and high efficiency. Increasingly, it is also used in automotive applications as part of a strategy to eliminate belts and hydraulic systems, to provide additional functionality and to improve fuel economy. The continuing reduction in cost of magnets and the electronics required for the control of BLDC motors has contributed to its use in an increasing number of applications and at higher power levels.

The BLDC motor is usually operated with one or more rotor position sensors since the electrical excitation must be synchronous to the rotor position. For reasons of cost, reliability, mechanical packaging and especially if the rotor runs immersed in fluid, it is desirable to run the motor without position sensors – so called sensorless operation.

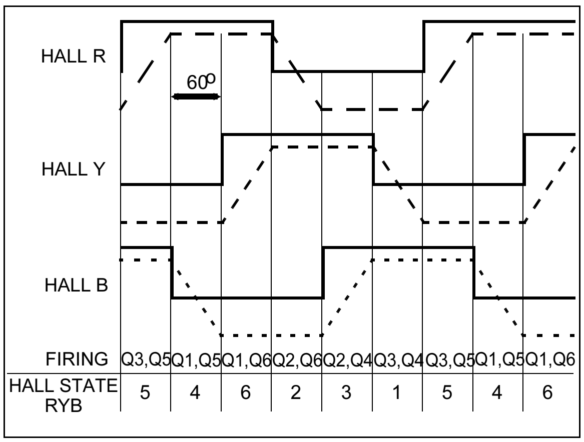
Instead of elaborating on operation of the BLDC with position sensors, it is assumed that the reader is already familiar with this technique. Microchip Application Note AN857 contains a very useful introduction to BLDC motor control. Alternative explanations may be found in the text books listed in the bibliography. It should be noted that the sensorless scheme described here is a more advanced form of the one described in AN857. Finally it should be pointed out that all the discussions here, and the application software, assume a 3-phase motor is to be used.

Sensorless Techniques for BLDC Motor Commutation

The methods discussed here are applicable only to 3-phase motors of standard construction (no search coils or deliberate asymmetries). It is also assumed that conventional 120° blocks of energization are used such that there are periods of time when one phase has zero current flowing and is not being actively driven. The driven phases must be switched, or commutated, at periodic intervals to run the motor.

To allow correct commutation of the motor, the absolute position within an electrical cycle must be measured. For conventional energization, six equally spaced commutations are required per electrical cycle. This is usually implemented using three hall-effect or optical switches with a suitable disk on the rotor. Continuous position information is not required, just detection of the required commutation instances. Figure 1 shows the three sensor outputs along with the corresponding Back EMF (BEMF) voltage waveform for each phase.

FIGURE 1: BLDC COMMUTATION DIAGRAM



To detect rotor position by monitoring a property of the motor, clearly this property must vary with position. Furthermore, it is desirable if the property establishes a unique position within an electrical cycle, which adds robustness to the sensorless technique. The variation in phase flux-linkage with position produces torque. This effect can be dissected into reluctance and BEMF components, both of which may vary with current, as well as position. BEMF also varies linearly with speed. The variation of the Reluctance or BEMF can either be monitored directly or their effect on a secondary quantity can be used instead.

RELUCTANCE VARIATION METHODS

Reluctance is the magnetic equivalent to electrical resistance in the magnetic Ohm's law as given by Equation 1:

EQUATION 1: MAGNETIC OHM'S LAW

$$\Re = \frac{MMF}{\Phi}$$

In this equation:

$\Re$ = Reluctance

MMF = Magno-Motive Force

Φ = Flux

Reluctance represents how easy it is for flux to flow around the magnetic circuit formed by the steel, air-gap and magnets. Magnets form very good flux sources and are equivalent to a current source. Phase windings form a good MMF source and are the equivalent of voltage sources. Under low levels of magnetic loading, steel has a low reluctance and is unsaturated. Under high levels of magnetic loading (> 1.5 T typically), the reluctance of the steel rapidly starts to increase as saturation begins. Air has a very high reluctance, which is independent of magnetic loading. Magnet material behaves in a similar manner.

Since the reluctance varies with position, it can be used as the basis for sensorless operation. In all BLDC motors, there will be some variation in reluctance with angle. From the terminals of the machine, the reluctance variation will be apparent as a variation in the inductance. This variation has the distinct advantage that the variation is detectable at zero speed. However, prior knowledge of the $L(i,\theta)$ characteristics of the motor to be controlled is required.

Unfortunately, the reluctance variation with position is too small to be measured reliably for many BLDC motors. This characteristic is especially true of motors with surface mounted magnets because the effective air-gap is large. As a result, the dominant part of the magnetic reluctance is constant, thus making any residual variation with position difficult to measure. The reluctance variation also tends to be low in motors that have been specifically designed for low torque ripple production because varying reluctance gives rise to an additional torque component. Buried and interior magnet motors often have significant reluctance variation with angle, but they tend to be energized with sinusoidal voltages and, therefore, will not be considered further.

BEMF METHODS

The BEMF waveform of the motor varies as both a function of position and speed. Detection of position using the BEMF at zero and low speeds is, therefore, not possible. Nevertheless, there are many applications (e.g., fans and pumps) that do not require positioning control or closed-loop operation at low speeds. For these applications, a BEMF method is very appropriate. There are many different methods of using the BEMF. The majority of these methods can be summarized as follows:

- Machine terminal voltage sensing
  - Either by direct measurement or inference (knowledge of switch states and DC bus voltage).
- Mid-point voltage sensing
  - Only works for Y connected motors with particular BEMF properties.
  - 4th wire not actually required. Can re-create star point using resistor networks and difference operation.
- Bus current gradient sensing
  - Relies on characteristic bus current shape due to commutation changing as rotor leads/lags.
  - Can not use fast bus current control.

FLUX-LINKAGE VARIATION METHODS

Detection of the variation of flux-linkage with position effectively combines the reluctance and BEMF methods into one. The phase voltage is given by Equation 2:

EQUATION 2: BEMF PHASE VOLTAGE

$$V_{PH} = iR + (d\Psi)/(dt)$$

This method offers the potential of seamless operation from zero speed for either square or sinusoidal energization. Closed-loop observers are required to correctly determine position from the open-loop integration of applied voltage and the measured phase currents, which requires detailed prior-knowledge of the $\psi(i,\theta)$ characteristics of the motor, as well as significant processing power.

Implementation of the Chosen Sensorless Technique

The particular method implemented is based on detecting the instances when the BEMF of an inactive phase is zero. Apart from the amplification of the bus-shunt signal, which is optional, and the power switch gate drivers, the implementation is single-chip with the dsPIC30F providing all of the control functionality.

The so-called BEMF "zero crossing" technique was chosen because:

- It is suitable for use on a wide range of motors.
- It can be used on both Y and Δ connected 3-phase motors in theory. Certain classes of Δ connected motors may not work.
- It requires no detailed knowledge of motor properties.
- It is relatively insensitive to motor manufacturing tolerance variations.
- It will work for either voltage or current control.

The zero-crossing technique is suitable for a wide range of applications where closed-loop operation near zero speed is not required. Its application on fans and pumps is particularly appropriate.

Provided the speed is greater than zero, there are only two positions per electrical cycle when the BEMF of a phase is zero, and these positions can be distinguished by the slope of the BEMF through the zero crossing as shown in Figure 2.

Each sector corresponds to one of six equal 60° portions of the electrical cycle. (The sector numbering is completely arbitrary but matches that used throughout the software.) Commutations occur at the boundary of each of the sectors. Therefore, it is the sector boundaries that need to be detected. There is a 30° offset between the BEMF zero-crossings and required commutation positions, which must be compensated for to ensure efficient and smooth operation of the motor.

FIGURE 2: ZERO CROSSING DETECTION

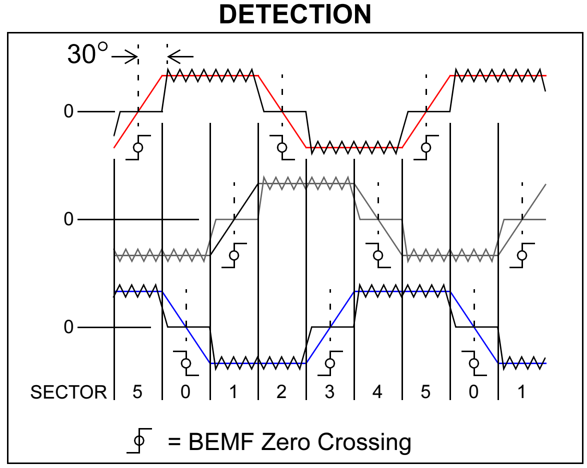
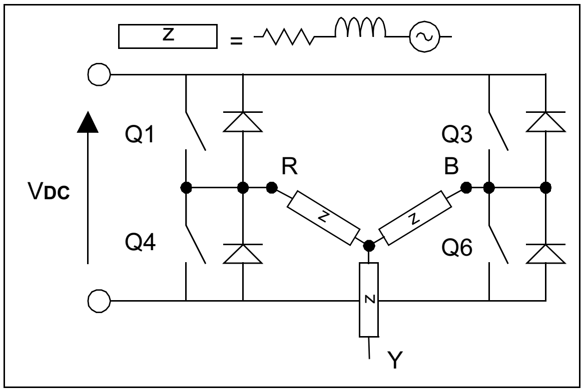


Figure 2 also shows the individual idealized phase BEMF waveforms. Assuming only the three motor leads are available for sensing the BEMF, then the voltage of the star point of the motor must be determined because the BEMF waveform will be offset by the star point voltage.

Because the method of operation is different for delta-connected motors, they are discussed in Appendix B.

Recalling that only two phases are actively driven, and with currents of opposite directions flowing at any one time, Figure 3 clarifies the situation where phase Y is the one to be used for BEMF sensing purposes.

FIGURE 3: BEMF SENSING HARDWARE EXAMPLE



For positive current in phase R (defined as current flowing toward the star point) and negative current in phase B, Q1 and Q6 would be controlled, corresponding to sector 1 in the previous diagrams. Assuming the two ends of the active phases are always connected to opposite rails of the DC supply by symmetry, the star point is always at ½ V_{DC}, irrespective of the polarity of the voltage across the two active phase windings. However, the ½ V_{DC} value will only be true if the phases are identical in terms of R, L and BEMF, and the switch and diode drops are equal. Assuming this to be the case for the moment, it therefore appears that the BEMF zero crossing will be biased by ½ V_{DC}, which is simple to take into account.

In its simplest form, the BEMF zero crossing method can be implemented as follows:

- Monitor all three phase terminal voltages and V_{DC} via potential dividers fed into the ADC.
- Detect during appropriate sectors when the phase BEMFs cross ½ Vdc. Only one phase voltage need be monitored for a given sector.
- Measure the time for 60°, the time between zero crossings, by using one of the timers available. Dividing this value by two and loading it into another timer, the implicit 30° offset required for correct commutation can be cancelled.

In practice, the implementation is not much more sophisticated than this despite the fact that the BEMF waveforms measured are influenced by several second order effects as follows:

- Phase winding demagnetization at the end of a block of energization causes the phase terminal being sensed to be clamped to one of the DC bus rails as the energy stored in the winding flows back to the supply via the inverter diodes. Care must be taken that the action of the diode on the phase terminal voltage does not cause a false zero crossing.
- Mutual coupling from active phases due to PWM action causing "noise" to be superimposed on top of the BEMF. The noise tends to be at a minimum at the zero crossing position itself.
- Deviation in the star point voltage from ½ V_{DC}.
  - If the phase current is zero for part of the PWM cycle, the output terminals of the active phases are left to float. This effect of the phase current being zero for part of a cycle is often referred to as discontinuous current.
  - Unequal switch/diode voltage drops between the high side and low side devices will not cause any perceivable issue for the majority of systems. On exceptional systems, a small imbalance will result between the width of the energization of the positive and negative current regions.
  - Non-trapezoidal BEMF means that the star point voltage moves because the two active phase BEMFs are not equal and opposite in magnitude. Most BLDC motors will have a BEMF waveshape somewhere between a trapezoid and a sinusoid. In practice, this characteristic does not cause a problem since all it does is modify the slope of the apparent BEMF being monitored either side of the zero crossing position.

The phase winding demagnetizing issue is easily taken care of in software by discarding the first few samples of the BEMF after commutation. Both the mutual coupling PWM "noise" and the discontinuous current issue are eliminated by not filtering the BEMF waveforms appreciably with hardware and carefully choosing the sample point of the signal with respect to the PWM waveform. The special event trigger from the motor control PWM module is used to initiate conversion of the ADC signals just before the switches turn off.

THE APPLICATION SOFTWARE

MPLAB® 6.40 was used for the development environment and the Microchip C30 optimizing compiler (v1.10.02) was used for compilation. The MPLAB ICD 2 was used for debug and programming. The motors used for development were from the Hurst Manufacturing NT Dynamo™ standard range of products.

The majority of the code is written in C with in-line assembler used in a few places where necessary for efficiency or functionality. Table 1 describes the content and function of the 16 individual source code files.

Hardware Resources

As provided, the code consumes 15,594 bytes of program memory space with the compiler Level 1 optimizations enabled. This amount of memory includes the user interface code and constant values stored in the program space. You will probably want to remove the user interface code for your final application. Removal of the user interface code will easily allow the application to fit into the smallest dsPIC® device variants.

The application requires 276 bytes of data memory storage. The rest of the device memory is available as dynamic storage for the software stack.

As written, the application allocates two rows (64 program memory locations) of device program memory to use as non-volatile storage of software parameters. There are 45 parameters total in the application.

The software was written to run at a CPU speed of 7.38 MIPS. This operating speed can be achieved by using the 4X PLL on the dsPIC device and using a 7.38 MHz crystal or external clock source. The software requires a maximum of 5 MIPS execution speed so plenty of CPU bandwidth is available for other application tasks. The software can be modified for operation at higher CPU speeds by modifying constants in the `defs.h` file.

Although the source code is thoroughly commented, the major routines specific to motor control are explained in the flow charts contained in Appendix A. Table 2 explains which of the dsPIC30F peripherals are used and for what purpose.

TABLE 1: SOURCE CODE FILES

Filename	Purpose of File	Functions Inside
defs.h	#define macro values used throughout the software	
extern_globals.h	External declaration of global variables	
flash_routines.c	Low-level routines for erasing and writing to Flash program memory	erase_flash_row program_flash
globals.h	Declaration of global variables	
hardware.h	#define macros specific to the dsPIC30F motor control development PCB	
inline_fns.h	Header file containing functions that are compiled in line for efficiency and then called by ADC ISR	check_zero_crossing current_control acquire_position
ISRs.c	All Interrupt Service Routines as well as any Trap Servicing Routines	AddressError StackError MathError PWMInterrupt FLTAInterrupt ADCInterrupt T1Interrupt T2Interrupt T3Interrupt
lcd_drivers.c	Low-level routines for accessing the 2x16 LCD display	Too many to list individual routines
lcd_messages.h	String constants used for messages on the LCD display	
main.c	Initialization and background code	main
medium_event.c	The medium event rate handler itself and all code called by it apart from one which is part of user_interface. The medium event handler executes every 10 msec.	medium_event_handler speed_loop voltage_control starting_code

TABLE 1: SOURCE CODE FILES (CONTINUED)

Filename	Purpose of File	Functions Inside
parameters.h	All the user parameter default values and details on minimum-maximum values, increment rates and editing strings	
setup.c	All setup code for the peripherals called during initialization	setup_ports setup_motor_pwms setup_adc setup_qei setup_timers WriteConfig
slow_event.c	The slow event handler only. Although the user interface functions are called from the handler, the code is separate. The slow event handler executes every 100 msec.	slow_event_handler
user_interface.c	Various routines which implement the user interface via the LCD display and push-button switches.	screen_handler process_switches save_parameter process_parameters debounce_switches edit_screen uint_to_string nibble_to_hex run_screen
xlcd.h	#define macros for use by the lcd_drivers	

TABLE 2: dsPIC30F PERIPHERAL USAGE

dsPIC30F Peripheral	Function and Configuration
Motor Control PWM Module	Used to drive the 3-phase inverter with 16 kHz PWM modulation of diagonally opposite switches. Outputs are configured in independent mode, and the Special Event Trigger is used to initiate ADC conversions just before switches turn off.
High Speed 10-bit ADC	Used to take four simultaneous samples per PWM cycle of bus current, bus voltage, demand pot and phase voltage (1 of the 3). The samples are synchronized to the PWM module.
Quadrature Encoder Interface (QEI)	Inputs disabled but timer used in 16-bit free-running mode to provide timestamps of zero crossing detections.
TIMER2	Used in 16-bit mode to provide delay between the zero crossing events and the desired commutation times.
TIMER3	Used to provide PWM of brake chopper switch.

HARDWARE

This application was developed to run on the dsPICDEM™ MC1 Motor Control Development Board and either the dsPICDEM MC1L 3-Phase Low Voltage Power module or the dsPICDEM MC1H 3-Phase High Voltage Power module. A photograph of the control board/power module system is shown in Figure 4.

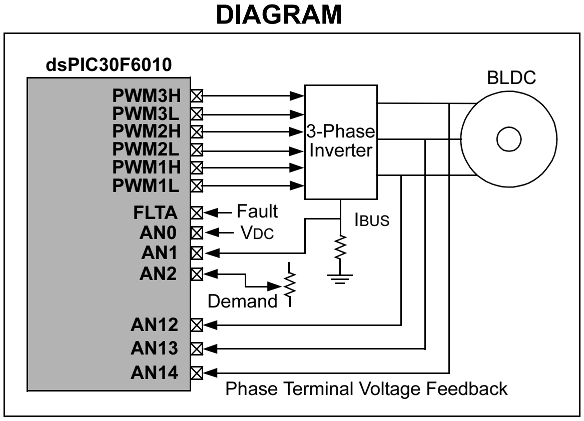
FIGURE 4: CONTROL BOARD/POWER MODULE SYSTEM

These development tools are available from Microchip (refer to the Microchip website for further details). Alternatively, you can design your own hardware, though some software modifications may be required.

Use of the dsPICDEM development tools requires some modification of jumpers on the PCB. These modifications are described in **"Modifications to the Power Module"** and **"Modifications to the Motor Control Development Board"**.

The following block diagram (Figure 5) shows the simplified hardware architecture with respect to the motor control. The LCD interface and push buttons have been omitted for clarity.

FIGURE 5: HARDWARE BLOCK DIAGRAM



The three-phase inverter, bus current sensing circuitry and voltage feedback potential dividers are all located with the power module.

GETTING STARTED

Modifications to the Power Module

To obtain the required feedback from the power module, it must be modified. This modification bridges the isolation barrier so that the phase voltage feedback (x3), V_{DC} feedback, and bus current shunt feedback signals are made available to the dsPIC30F on the control board. Follow this process:

1. Remove the lid as described in the user manual for the power module.
2. Solder low value (47R or below) resistors into LK22, 24-26 and LK30
3. If you're using a high voltage module, carefully follow the additional procedures for modifying and using the system in the non-isolated mode. This involves soldering a ground wire of a suitable current rating between J5 and J13.

Note:	You must cross the electrical isolation barrier on the High Voltage Power module to modify it for this application. Be sure to maintain the input earth (ground) and use a safety isolation transformer between the main supply and the input to the power module.

4. Set the shunt scaling links LK11-12 for desired motor use. The LK11-12 links scale the bus current feedback. If in doubt, remove LK11-12, as this will give best protection and highest gain feedback.
5. If you're using the power module at less than 50% of the maximum bus voltage rating, it is recommended that the voltage feedback scaling be reduced to obtain higher feedback voltages. Change the values of R10, R13 and R14 for the V_{DC} and R16-21 for the V_{PH} from the top of the printed circuit board without disassembling the module. See the user manual of the power module and the schematics for details.

Note:	The phase voltage and DC bus voltage feedback scaling must match to achieve correct sensorless operation.

Connecting the Motor

The motor should be connected in the normal way using three wires and earth (ground) of appropriate current rating. One advantage of the sensorless system is that the phase sequence of the motor leads is not important, because it only defines which direction is forwards. If you have a suitable position feedback device, you can also use it for diagnostic purposes.

Modifications to the Motor Control Development Board

The ADC channels on the control board must be reassigned for the application software to function correctly, because four simultaneous samples are taken of the bus current (I_{BUS}), bus voltage (V_{DC}), demand pot (POT) and one phase voltage (V_{PH}). The dsPIC30F 10-bit A/D Converter uses specific input pins for simultaneous sampling. AN0, 1, 2 are used for the V_{DC}, I_{BUS} and POT, respectively, with the CH0 MUX used to move between the V_{PH} signals on the original assignments of AN12, 13, 14. You need to make the following connections on the PCB to reassign the analog channels:

- Connect AN11 on J6 to Pin 2 of LK1 (the other pins of LK1 should be unconnected).
- Connect AN8 on J6 to Pin 2 of LK2 (the other pins of LK2 should be unconnected).
- Connect AN2 on J6 to AN7 on J6.

Note: LK1 and LK2 are used to reassign AN0 and AN1 to ensure there is no conflict between these signals and the MPLAB ICD 2, which uses these lines for the default clock and data.

Using S2 and MPLAB ICD 2

AN0 and AN1 are required to provide feedback and are also used by the MPLAB ICD for programming and debugging. Therefore, you must use S2 to switch the MPLAB ICD clock and data lines at appropriate times. This is required whether debugging is used or not. If you are using the dsPICDEM MC1 Motor Control Development Board and plan to use the MPLAB ICD 2 for debugging, follow steps 1-3 of the following procedure. If you plan to use the MPLAB ICD for device programming, you only need to perform steps 2-3.

1. Within MPLAB IDE, select the “Use EMUC1 and EMUD1” option under the Configure> Configuration Bits>Comm Channel Select window.
2. Program the device with S2 in the MPLAB ICD position.
3. After programming is complete, move S2 into the Analog position and run the software.

Setting and Tuning User Parameters

The user interface is simple and intuitive. The LCD display and push button switches let you adjust many parameters. Help strings are displayed where feasible. The function of the four push buttons is as follows:

S4	• Activates the edit menu from standby or fault conditions. • Scrolls backwards through the parameter list in the edit menu. • Reduces the value of a parameter when altering its value. • Toggles between two different screens when running.
S5	• Scrolls forward through the parameter list in the edit menu. • Increases the value of a parameter when altering its value.
S6	• Selects a parameter for alteration. • Stores the new value of a parameter.
S7	• Starts/stops/resets the system when the edit menu is not active. • Exits from both the edit menu and when altering a parameter.

Most of the parameters are self-explanatory in their function. The source file parameters.h contains additional explanations of the parameters as well as the default values and individual parameter properties. Where a statement is within quotation marks (“”), the statement corresponds to the text string displayed on the LCD. Appendix B lists the individual parameters and contains hints on the appropriate values for some parameters. The system always powers up at parameter 0, and the access is circular (i.e., moving backwards from parameter 0 moves to the last parameter). The starting parameters are explained in more detail below.

Suggested Setup Method

The default parameters are suggested as a good starting point for setup. These default values are contained in the parameters.h file and described in **Appendix A: “User Parameters”**. The system is configured for open loop operation in the sense that simple voltage control is used for both starting and running, which initially removes the need for speed and current control loop tuning.

It is suggested that you initially ignore all parameters having to do with the control loops and concentrate on adjusting the starting parameters (see **“Starting Parameters”**) to obtain reliable and non-oscillatory starting.

Once the system is running sensorless in the Open-Loop Control mode, you may wish to experiment with the control loops and other system parameters.

Hardware Parameters

Before the system can be started, you must ensure that certain hardware-related setup parameters are correct. The setting of these parameters will depend mostly on the selected motor. The setup parameters include:

- Number Motor Poles
- Blanking Count
- Voltage Scale
- Current Scale

An explanation of these parameters is provided in Appendix A.

Starting Parameters

The motor must be started open-loop due to the lack of BEMF information at low speed. Provided starting parameters are adjusted to suit the motor and the demand is neither too high nor too low, the system should then run sensorless. If the demand is too high, an over-current trip may result. If the demand is too low, the system will stall. A half turn of the demand pot is a good starting point. There are two different starting methods implemented and several parameters to be adjusted to tune the starting for the particular application. The parameters that control motor start are as follows:

- Direction Demand
- Lock Position 1 Time, Lock Position 2 Time
- Lock Position 1 Demand, Lock Position 2 Demand
- Ramp Start Speed, Ramp End Speed
- Ramp Start Demand, Ramp End Demand
- Ramp Duration
- Starting Control
- Acquire Method
- ZeroX Enable Speed
- Windmilling Demand
- Braking Ramp Time

The first thing you will need to do is to decide in which direction the motor will start and run. The direction is changed using the Direction Demand parameter. During the initial development of your project, the direction of the motor may not be important. However, some types of motors and some loads require a certain direction of rotation. You can also reverse the direction of the motor by swapping two of the power wires to the motor.

The starting routine runs the motor at a relatively low open-loop speed, then ramps the speed to a final value that produces sufficient BEMF voltage so the sensorless algorithm can begin operation. The operation of the BLDC motor in open loop mode is much like a stepper motor, although it is a very inefficient mode of operation and the motor cannot produce the rated torque when operating in this manner.

SETTING THE LOCK PARAMETERS

Before the motor is run, the algorithm energizes two pairs of windings for a brief period of time to position the rotor into two reference, or lock, positions. These two lock positions ensure that the rotor is at a known reference point before the open loop starting algorithm begins. It is very important that the position of the rotor is stable before the open loop starting begins and the four Lock Position parameters must be adjusted accordingly. If the Lock Position Demand parameters are set too high, the rotor will oscillate when it reaches the lock positions. If they are too low, the rotor will not move to the reference position. Try increasing or decreasing the Lock Position Demand parameters until the rotor moves quickly to the two lock positions with a minimum of oscillation. After the demands are set, you can then increase or decrease the Lock Position Time parameters to adjust the holding time in each lock position. Loads that have a lot of inertia, such as a large diameter fan blade, may need a longer holding time to allow the rotor oscillations to decay. The lock times for low inertia loads can generally be set to a very low value to allow the motor to start quickly. As you configure the software, try starting the motor using the S7 button and observe the rotor operation during the lock times. If you have not yet configured the remainder of the parameters, you can press S7 just after the lock times occur to cancel the motor starting routine.

SETTING THE RAMPING PARAMETERS

At the end of the second lock, the system automatically starts energizing the system in an open loop stepping manner. You must select the starting speed (Ramp Start Speed parameter) and the energization demand so that the rotor “locks on” to the energization sequence.

The system then increases the speed to the “Ramp End Speed” over the “Ramp Duration” time given while also changing the demand linearly with speed according to the two ramp demand values. The open loop stepping speed is profiled between the start and end speeds by a square law function with time as given by Equation 3:

EQUATION 3: SQUARE LAW FUNCTION WITH TIME

$$\omega = \omega_s + kt^2$$

where ω_s is the Ramp Start Speed parameter, *k* is the Ramp End Speed parameter minus the Ramp Start Speed parameter, and *t* is time as determined by the Ramp Duration parameter.

This acceleration profile has been chosen to optimize starting performance. The speed at the end of the ramp must be high enough so that there is sufficient BEMF voltage present for the system to reliably detect the zero crossings.

RAMPING PARAMETER GUIDELINES

You will need to select a start and end speed for the ramping. These speeds will depend on the rated speed of your particular motor and the BEMF voltage constant. You will have to make sure the motor is reliably accelerated to a speed at which the sensorless routine can detect the BEMF voltage. A rule of thumb that can be used is to set the Ramp Start Speed parameter to a value that is 1/60th the value of the rated motor speed. The Ramp End Speed parameter can be set to a value that is 1/6th the value of the rated motor speed. For example the Ramp Start Speed would be set to 50 RPM and the Ramp End Speed would be set to 500 RPM for a 3000 RPM motor.

Next, you will need to set the Ramp Start Demand and Ramp End Demand parameters. Assuming you are using voltage control mode (software default), starting values near 50% will generally be appropriate. The key to setting these demands is to accelerate the motor to the end speed without ‘slipping’ or excessive mechanical vibration. The best way to set these demands is to observe the rotor while starting and listen to the sound that the motor makes when it is energized. As the starting routine executes, most motors will make a ticking noise with frequency proportional to the ramp speeds. If you hear the ramp speed increasing, but the rotor appears to be spinning slowly or just oscillating in a stationary position, then the ramp demands probably need to be increased. If the rotor appears to be accelerating properly, but there seems to be excessive motor vibration, over-current trips, or excessive noise during ramping, the ramp demands are probably set too high. In most cases, you will want to set the Ramp End Demand parameter 5% to 15% higher than the Ramp Start Demand parameter. If these two parameters made equal to each other, you may observe that the motor starts ramping normally, but begins to slip as the ramp speed increases.

The Ramp Duration parameter can be adjusted to optimize starting time. In general, you should start with a relatively long ramping time to ensure the motor is starting properly. A ramping time between 2 and 4 seconds should be appropriate for most motor and load combinations. You will find that loads with greater inertia require a longer ramp time for proper acceleration. As the ramp time is decreased, you may also have to increase the Ramp Start Demand and Ramp End Demand parameters to avoid rotor slipping during startup.

SETTING THE STARTING CONTROL

You choose either current or voltage control with Starting Control parameter (#40).

Current control has the advantage of eliminating variations in the starting currents due to DC bus voltage variations or motor resistance. However the hold times often must be increased as the rotor will tend to oscillate more than if voltage control is used. Also the current control PID loop requires tuning.

If you use current control you should enter an appropriate over-current trip level, as this scales the demand. Be sure to enter the current feedback scaling correctly (see parameters.h for guidance on appropriate values).

Voltage control (default) offers the possibility of eliminating bus current sensing and the associated software for certain applications. You should only use this control method when the DC bus voltage variations are well known and the load torque is repeatable. Otherwise the starting may fail.

THE TWO DIFFERENT ACQUISITION METHODS

Two different acquisition methods, referred to as “Method 1” and “Method 2” in this document and throughout the source code, are implemented to acquire the initial position prior to running sensorless. You select which method is used by the Acquire Method parameter (#43). The application dictates which of the two methods is the most appropriate.

Method 1

With this method the system begins to looks for zero crossings once the motor speed exceeds the ZeroX Enable Speed parameter (#44). If zero crossings are detected in two consecutive sectors of an electrical cycle, then the sensorless commutation is launched. The ZeroX Enable Speed parameter should be set to a speed above which smooth motion and sufficient back EMF can be observed. This parameter is best determined and monitoring one or more of the phase voltages with an oscilloscope while adjusting the starting parameters. The phase voltages are best observed on connector J6, signals AN12, AN13 and AN14. The ZeroX Enable Speed parameter must be less than the Ramp End Speed parameter for this method to work.

When running at a constant speed with open-loop stepping energization, the rotor position will be approximately 90° (electrical) in advance of that when running correctly under sensorless control, assuming the load torque is negligible. As a result, the BEMF zero crossings occur when the phase is being energized rather than during the inactive regions and, thus, cannot be sensed. To make the zero crossings observable, it is necessary to accelerate the motor at a certain rate. During acceleration the inertia of the motor and load is used to cause a lag in position, which cancels some or all of the natural phase lead. The higher the rate of acceleration, the more lag will occur. Thus with the correct starting parameters selected and a relatively predictable mechanical load, the BEMF crossing point occurs during the de-energized period and can be detected, allowing system starting.

For many applications this will be the acquisition method of choice as it can provide fast and seamless starting. However, for this method to work correctly, the starting parameters that control the acceleration ramp must be chosen especially carefully. If the mechanical load varies or is not repeatable, then it may cause the acquisition to fail.

Method 2

Method 2 does not look for zero crossings while the speed is being increased. Instead, at the end of the speed ramp, the motor is briefly de-energized. At this time all three phase voltages are observed. The instance and sequence of the phase voltages as they rise above zero volts is used to determine both the direction of rotation and the position. Only one electrical cycle of rotation, at most, should be required for the system to acquire as two different phase voltage rising edges are required. Once it has acquired, the system is re-energized with the sensorless commutation running. This method, therefore, has the advantage of requiring little knowledge about the motor and load. All that is required is that there be sufficient back EMF and inertia so that the motor does not stall during acquisition. Also there should be no excessive oscillation of speed just before the end of the starting ramp. This method of acquisition is used to provide flying start detection (significant rotation occurring when the system is started) and windmilling detection (see Table 3)

TABLE 3: COMPARISON OF ACQUISITION METHODS

Method 1	Method 2
Advantages	
• Fast starting • No risk of motor stalling	• Easy parameter tuning • Works for unpredictable mechanical loads
Disadvantages	
• Careful adjustment of parameters needed • Predictable load required	• Needs appreciable inertia or low load at modest speeds to prevent stall

WINDMILLING

In fan applications, it is not uncommon for the fan impeller to rotate when the motor is de-energized due to ambient airflow. This phenomenon is known as windmilling. To provide robust starting, an initial check of speed and direction is made. If the motor is found to be already rotating in the same direction as demanded then method 2 is used to achieve a flying start.

If the motor is found to be rotating in the opposite sense to the demand then the motor must be decelerated to a standstill. The normal starting method can then be used to start the motor in the desired direction. This is achieved by energizing the motor open loop starting from the speed detected during the rotation check and ramping down. The time taken to reach 0 Hz is controlled by the Braking Ramp T parameter, which is set in 10 ms increments. The Windmilling Dem. parameter sets the percent demand used during the windmilling deceleration time. These two parameters should be adjusted to ensure the system stays in-lock during braking.

Starting Parameter Troubleshooting

If correct starting is not achieved with the chosen parameter settings, use the following suggested order of adjustment:

LOCKING PARAMETERS

First, ensure that the initial positioning is occurring correctly. Try making both step times longer to observe the locking movement. Then reduce the step times to acceptable values after it is clear that the motor is responding correctly. You can expect that occasionally the motor will stop in the same position as that required by the first step, which means no movement will be observed on the first step. However, a check of the phase voltages will re-assure you that correct energization is nevertheless occurring.

In general, systems with low starting loads, e.g., most fans, require a low demand to align the rotor. Because such loads may also be poorly damped, too high a demand may cause unwanted oscillation. In contrast systems with high starting loads, e.g., some pumps, require a higher demand to ensure they can move into alignment. Fortunately such loads are usually well damped. Complex loads with high cogging torque or backlash may require high demand and long step times, if indeed this type of sensorless control is suitable for them at all.

RAMP START DEMAND AND SPEED

After the second locking step the system immediately begins to step the motor at the predetermined initial rate. It is vital that the rotor is able to lock-on to this energization. If the demand is too low, or the speed too fast, the rotor may fail to lock on to the energization. Conversely, if the speed is too low and the demand too high, the rotor may have time to oscillate around a position of alignment and, thus, allow the synchronization to fail.

Observing the voltage feed back signal during the periods when a phase is not energized will establish if BEMF is showing symptoms of oscillation. The solution to these unwanted position oscillations might seem to be to reduce the demand. However, as we see in the following section the extra torque can be usefully employed to accelerate. Thus the initial value for demand must be adjusted with the speed ramping requirements in mind.

RAMPING UP TO THE ACQUIRE SPEED

Both methods of acquisition need the motor speed to be sufficiently high for BEMF information to be valid. Furthermore with Method 1, the acceleration itself is vital to produce the phase shift necessary to allow the sensing method to work. See the following scope photographs for examples of good and bad ramps.

When you consider starting parameters, remember that the system is depending on BEMF voltage information to acquire successfully. As a rule of thumb, knowing the BEMF constant of the motor, you should adjust the phase voltage feedback resistor networks to provide approximately 100 mV of feedback at the speed you are aiming to acquire. The acquire speed for Method 1 will be somewhere between the ZeroX Enable Speed parameter and the Ramp End Speed parameter. For Method 2, the speed at acquisition is given by the Ramp End Speed parameter.

EXAMPLE SCOPE PHOTOGRAPHS OF PHASE VOLTAGE FEEDBACK DURING STARTING

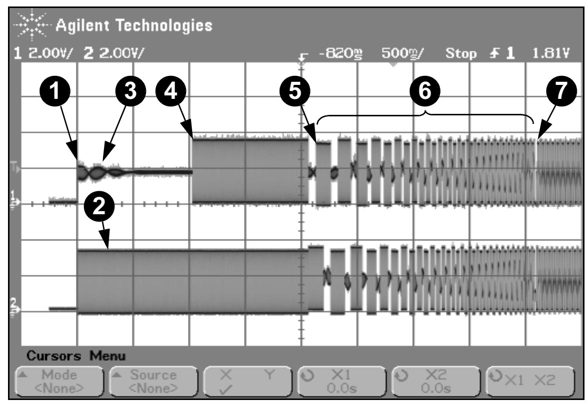
Following are scope photographs showing the following conditions:

- Locking and Open Loop Stepping (see Figure 6)
- Normal Sensorless Running
- Acquisition Method 2
- Acquisition Method 1

In all photographs, the top trace is phase A voltage feedback, and the bottom trace is phase B voltage feedback. Phase C voltage is not shown.

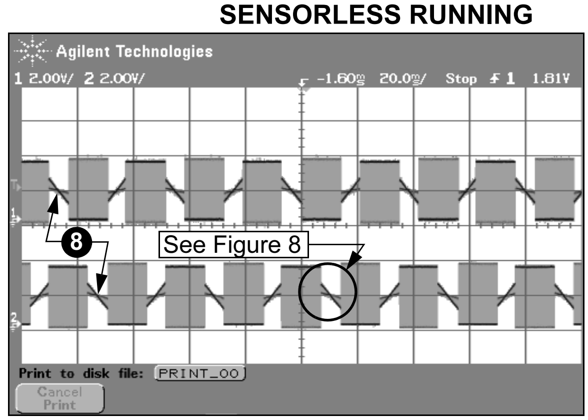
Scope Plot 1 (Figure 6) shows phase A & B voltage feedback signals as the motor is starting. The first locking step (1) begins after about 1/2 second (one time division from the left edge). The B & C phases are energized (2), and some oscillation is seen on phase A before it settles down (3). After approximately one second the second lock step occurs (4), where A and B are locked. After a further second the system begins to step (5), and ramps up in speed. It can be seen that once the system enters the stepping phase of the starting routine, energization is on for 120 electrical degrees and off for 60 electrical degrees, and the frequency of energization can be seen increasing (6). In the last time division (7) a small gap of two missing energization cycles be seen, which correspond to the point of acquisition using Method 2.

FIGURE 6: SCOPE PLOT 1: LOCKING AND OPEN LOOP STEPPING



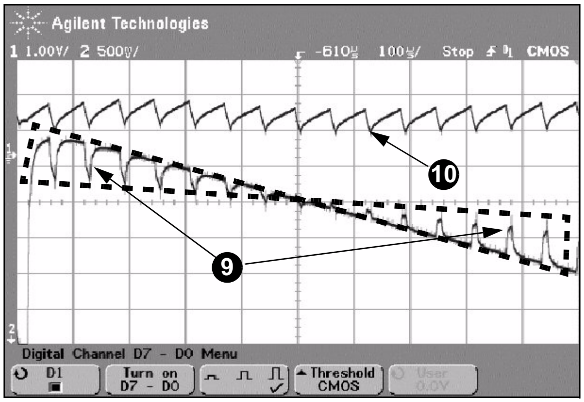
Scope Plot 2 (Figure 7) shows the system running normally, in sensorless mode, after acquisition has been successful. The BEMF zero crossing points (8) are always visible because the blocks of energization are correctly placed, as expected from a closed-loop control. In examining the progression from plot 1 to plot 2 in more detail, the two acquisition methods need to be considered separately.

FIGURE 7: SCOPE PLOT 2: NORMAL SENSORLESS RUNNING



The ‘fuzz’ on the BEMF waveform during the inactive regions can be ignored. It is a superimposed high frequency AC voltage at the PWM frequency, as shown in Figure 8.

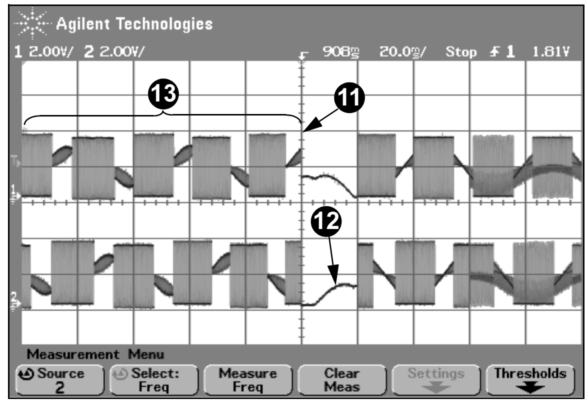
FIGURE 8: SCOPE PLOT 2A: ZERO CROSSING EVENT MAGNIFIED



This fluctuation (9) shown inside the dotted area is due to the mutual coupling effects of the motor phases and is caused by the PWM signal that is energizing the motor. The amount of mutual coupling varies as a function of the particular motor type used. The system ignores this AC component since the ADC samples synchronously to the PWM excitation. The top trace (10) shows the motor current that is a result of the applied PWM energization, but it is not really of interest in this discussion.

Plot 3 (Figure 9) shows Acquisition Method 2. The center of the plot (11) shows the point at which open-loop stepping of the motor is stopped and Method 2 is used to sense position. At this point the BEMF no longer has a ½ VDC offset normally seen during energization. The system senses the point where the phase voltage rises above the value given by the Acquire Threshold parameter (parameter #34). This condition can be seen to occur on phase B just after the center of the trace (12). The next rising edge occurred on phase C (not shown) and was used to determine direction, speed and the position. The system then transitioned to normal closed loop sensorless operation.

FIGURE 9: PLOT 3: ACQUISITION METHOD 2



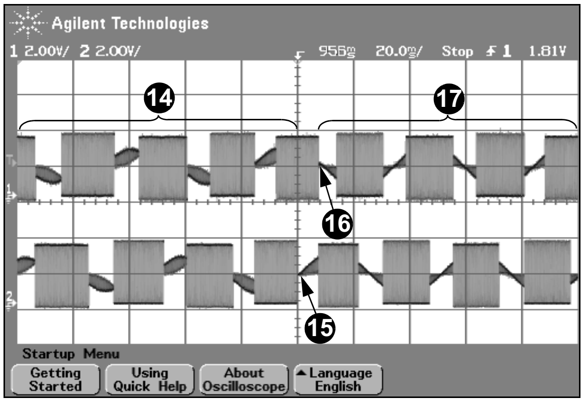
Notice that the traces up to the point of de-energization show that the portion of BEMF waveform visible when the phases are not energized do not show any BEMF zero crossings (13). The crossings are hidden by the energization. This absence of detectable crossings would cause Acquisition Method 1 to fail but is not a problem for Acquisition Method 2.

To successfully use method 2, the ramp parameters must be set so that the speed ramp during stepping meets three criteria.

- First, the rotor must stay locked to the top of the ramp. If acceleration is too fast and the rotor loses synchronization, lengthen the Ramp Duration parameter (parameter #10), reduce the Ramp End Speed parameter (parameter #7) or increase the Ramp End Demand parameter (parameter #9).
- Second, the magnitude of BEMF voltage for the Ramp End Speed parameter (parameter #7) must be sufficient.
- Third, there should not be significant speed oscillation at the top of the ramp. Too much oscillation could result in false rising edge detection and incorrect detected position. The error between actual and detected position causes incorrectly placed firing pulses, and a gross error at this point may cause the motor to stall.

In scope plot 4 (Figure 10), the system is open loop stepping to the left of center (14). BEMF voltage is visible between energization phases, but no zero crossings are shown because there is insufficient phase lag to uncover them. Phase lag increases with time as acceleration increases. The first zero crossing visible is on phase B in the center of plot (15). The next is on phase A approximately 10 milliseconds later (16). This second visible zero crossing is where the system acquires using Acquisition Method 1. After this point, correct closed loop sensorless operation begins (17).

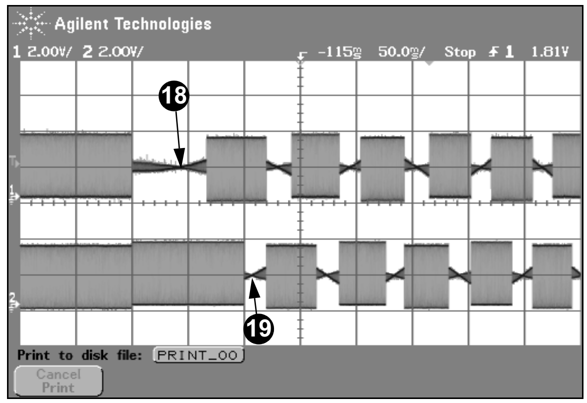
FIGURE 10: SCOPE PLOT 4: ACQUISITION METHOD 1



For some motors and loads, it is possible that the initial kick of acceleration when transitioning from rotor locking into open loop stepping may uncover zero crossings and allow very fast starting. To take advantage of this condition, the BEMF voltage magnitude must be sufficient and the ZeroX Enable Speed parameter (parameter #44) must be set suitably low at the point of acquisition.

Scope plot 5 (Figure 11) shows an example of this that was achieved on a system with a large inertia. During the first two steps of commutation the rotor has not moved very far thus revealing the zero crossing points immediately (18, 19).

FIGURE 11: SCOPE PLOT 5: ACQUISITION METHOD 1 – FAST STARTING



Running Parameters

SETTING THE CONTROL MODE

The application software has four control modes that can be selected for use during sensorless operation. These modes are as follows:

- Mode 0 – Closed Volts
- Mode 1 – Closed Current
- Mode 2 – Open Volts
- Mode 3 – Open Current

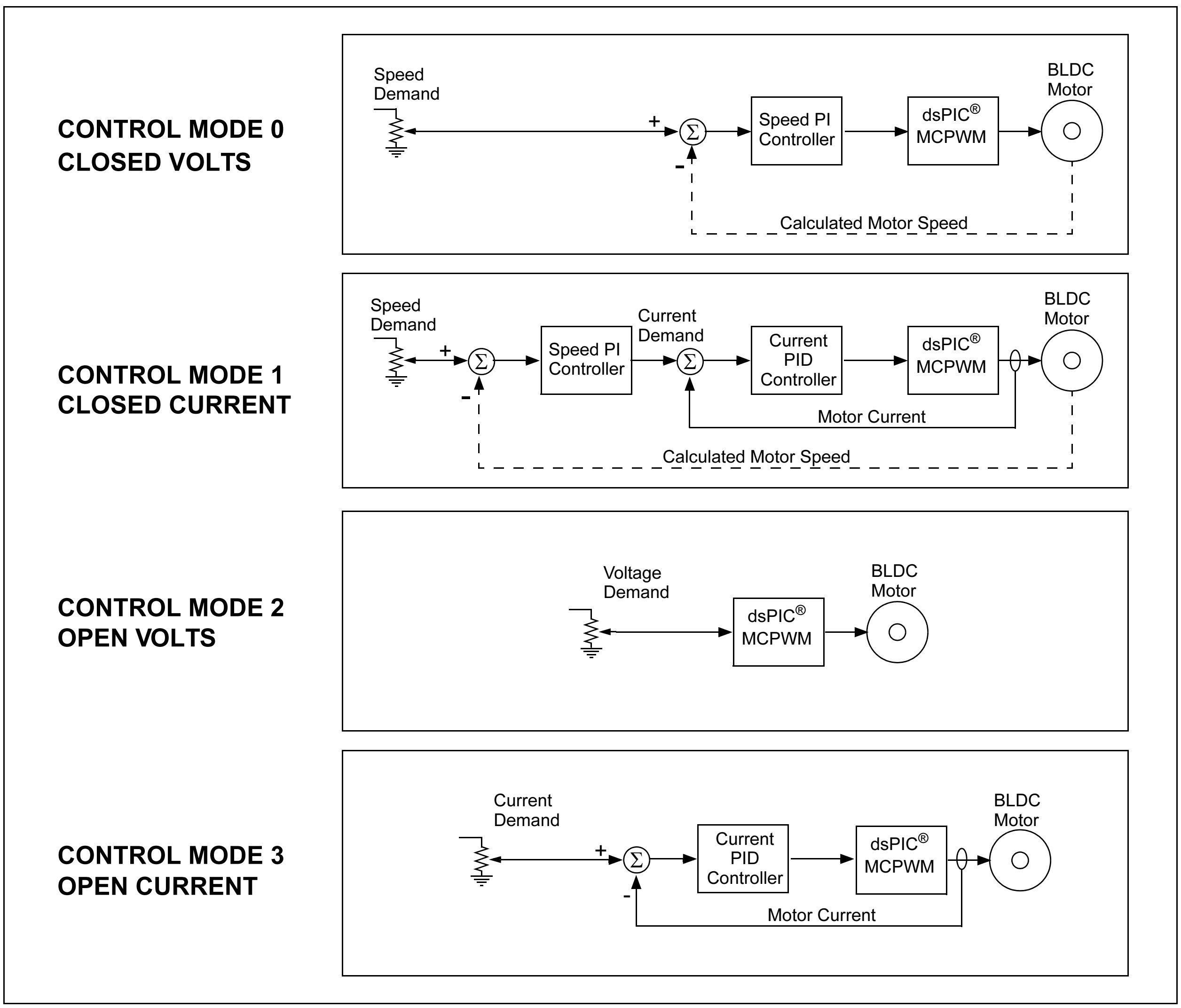
The four software modes determine whether the speed and current control loops are used, as illustrated in Figure 12.

In Mode 0, a speed control loop is used to control the PWM duty cycle delivered to the motor. The speed setting is determined by the potentiometer value, which may be scaled using the Pot X for Speed parameter (parameter #39) to achieve the desired speed range.

In Mode 1, an inner current control loop is used to set the PWM duty cycle. The speed control loop is used as an outer control loop that provides the current demand to the inner current control loop.

Mode 2 does not use any control loops to affect the motor operation. This is the default software mode. The value of the potentiometer directly affects the PWM duty cycle. Selecting Mode 2 for the initial tuning of the software for a particular motor eliminates potential issues with control loop tuning that may affect motor starting. You can use the Pot / for Duty parameter (parameter #37) to adjust scaling of the potentiometer value in this mode.

FIGURE 12: CONTROL MODES



In Mode 3, only the current control loop is enabled. The potentiometer value sets the current demand. The Pot / for Current parameter (parameter # 38) can be used to scale the current demand from the potentiometer value. When operating in Mode 3 with the motor lightly loaded, the PWM duty cycle often saturates causing the motor to run at maximum speed. This effect occurs because the motor cannot consume the amount of current requested by the current controller. When Mode 3 is used, it is helpful to connect an ammeter in series with the DC bus power supply to observe the effects of the current controller.

THE CONTROL LOOP PARAMETERS

When the software is operating in Mode 0, the Speed P Gain parameter (parameter #20) and Speed I Gain parameter (parameter #21) need to be adjusted. When the software is operating in Mode 3, the Current P Gain parameter (parameter #17), Current I Gain parameter (parameter #18) and Current D Gain parameter (parameter #19) need to be adjusted. If the software is operating in Mode 1, both the voltage and current control parameters will require adjustment.

You will have to experimentally adjust the PID parameters. The default PID parameters are conservative values that will probably work for most motors.

SETTING PARAMETER LIMITS

The Stall Time Limit parameter is a delay time that determines how long the software waits before a stall condition is indicated. If the software detects valid zero crossing events before this time expires, a stalled motor could potentially be restarted.

You may need to adjust the Over Speed Limit parameter if your motor runs at a high speed. The default value is 3300 RPM.

The Over Volts Limit parameter sets the maximum DC bus voltage in tenths of a volt increments. This parameter determines when the brake chopper circuit will begin to regulate the DC bus voltage.

The Over Current Limit parameter sets the peak DC bus current value in tenths of an amp increments. You will want to set this parameter to be several times higher than typical steady-state operating current values.

The Tolerance Check parameter determines the sensitivity of the system to variations of the commutation period, measured as a percentage. If the time period between two zero crossing events exceeds the prior time period by the Tolerance Check percentage, then the sensorless algorithm will enter a lost condition. Rapid acceleration or deceleration of the motor can cause the Tolerance Check parameter to be exceeded. For this reason, you should avoid setting this parameter to low values. Setting the Tolerance Check parameter to 99% will effectively disable the tolerance checking.

The Rotation Timeout parameter determines how long the application will wait before the starting routine locks the rotor and begins the open loop ramping process. The rotation timeout is applied when starting the motor. The rotation timeout is useful because the motor may already be spinning at the time the starting routine is begun. If the motor is already spinning at a rate to produce a sufficient level of starting back-EMF, then the acquire algorithm can determine the motor position and transition to sensorless mode before the rotation timeout delay expires.

TROUBLESHOOTING GUIDE

This table lists some possible symptoms of anomalies and provides an explanation of their likely causes.

Symptoms shown with capital letter (e.g., OVER CURRENT) refer to error messages that are displayed on the LCD.

For information on referenced parameters, see **Appendix A: "User Parameters"**.

TABLE 4: SYMPTOMS AND CAUSES

Symptom	Likely Cause
Motor will not energize	• Motor output connections are not in the correct terminals. • Power module supply not turned on or wires swapped (DC power module only).
Initial locking is very jerky	• Too much starting demand.
After locking, motor fails to begin stepping	• Initial locking sequence unsuccessful. • Ramp start speed (parameter #6) too high. • Ramp start demand (parameter #8) too low.
Rotor looses lock or oscillates during starting	• Not sufficient demand. • Ramp start speed (parameter #6) too high. • Ramp acceleration rate too high. The ramp end speed (parameter #7) is too high and/or ramp duration (parameter #10) is too short.
System does not acquire during ramp when using Acquisition Method 1 – FAILED TO START or STALLED fault	• Insufficient BEMF to acquire - increase ramp end speed (parameter #7) or change voltage feedback scaling (parameter 28 and 29). • Starting ramp not adjusted correctly to make zero crossings visible. • Ramp Start Speed (parameter #6) causing oscillatory rotation. • Power module or control board modifications not done or incorrectly carried out. • Check that S2 is in analog position.
System does not run sensorless after ramping when using Acquisition Method 2 – FAILED TO START or STALLED fault	• Insufficient BEMF to start - increase ramp end speed (parameter #7) or change voltage feedback scaling resistors. • Acquire threshold (parameter #34) set too high. • Starting ramp causing oscillatory rotation. • Power module or control board modifications not done or incorrectly carried out. • Check S2 is in analog position.
SENSORLESS LOST	• Increase Tolerance Check (parameter #10) • Ensure that input leads are not routed over the control board. • If running using speed control, check that speed loop is stable or that excessive speed overshoot does not occur. • If running using current control. check that current control loop is stable. • Check that auto-reacquire is enabled. This is especially important if accelerating/decelerating rapidly.
OVER CURRENT	• If using voltage control, reduce demand. • If using current control, check that current loop is stable. • Check motor wiring. • Check that S2 is in analog position and feedback signal wiring is correct. If system was run for the first time with S2 in the incorrect position, then the dsPIC device must be reset because an offset reading is taken of the bus current signal during initialization.
OVER VOLTAGE	• Check voltage demand parameter correct and that external brake resistor connected (if required). • Check voltage feedback scaling parameters • Check incoming supply voltage.
HARDWARE TRIP	• If SHUNT OVERCURRENT or HALL OVERCURRENT LEDs are illuminated on the power module, reduce demand, check that control loops are stable or check scaling feedback parameters. • If over voltage, check that the voltage demand (parameter #22) is correct and that the external brake resistor is connected (if required). • Check incoming supply voltage. • If over temperature, reduce motor currents or use a fan on power module heat sink.

REFERENCES

- *Modern Power Electronics and AC Drives*, B. Bose, Prentice Hall PTR, ISBN 0130167436
- *Electric Motors and Drives*, A.Hughes, Heinemann Newnes, ISBN 0750617411
- *Brushless Permanent Magnet and Reluctance Motor Drives*, T. Miller, Oxford Clarendon, ISBN 0198593694
- K. Iizuka et. al, “Microcomputer control for sensorless brushless motor”, IEEE Transactions on Industrial Applications, Vol. 21, No.4 1985, pp 595-601

APPENDIX A: USER PARAMETERS

This table explains the parameters used in the dcPIC30F sensorless BLDC motor control application described in this Application Note. Many of the descriptions include hints to help you apply these parameters to your particular application.

Parameter names are shown as they appear in the `parameters.h` file as well as the text string displayed on the LCD (for example “Lock Pos.2 Dem.” and “CONTROL MODE”.

TABLE A-1: USER PARAMETERS

Param No.	Parameter Name	Description	Default Value
0	DIRECTION DEMAND	Determines direction of rotation	FORWARD
1	CONTROL MODE	Determines method of speed control as follows: 0 – Closed loop speed control with output from speed loop directly controlling PWM duty cycle. 1 – Closed loop speed control with output from speed loop fed into current loop demand. 2 – Open loop speed control with demand pot directly controlling PWM duty cycle. 3 – Open loop speed control with demand pot directly controlling current demand. Potentiometer VR2 provides the demand.	3
2	Lock Pos.1 Time	Time for first starting lock in 10 ms increments	100
3	Lock Pos.2 Time	Time for second starting lock in 10 ms increments	100
4	Lock Pos.1 Dem.	% Demand for first starting lock	44
5	Lock Pos.2 Dem.	% Demand for second starting lock	44
6	Ramp Start Speed	RPM for beginning of starting ramp	20
7	Ramp End Speed	RPM for end of starting ramp	250
8	Ramp Start Dem.	% Demand for start of ramp	49
9	Ramp End Dem.	% Demand for end of ramp	50
10	Ramp Duration	Duration of ramp in 10ms increments	200
11	Phase Adv. Start	Start Speed for phase advance in RPM Hint: Appropriate phase advance improves system efficiency and extends the speed range over which the sensorless control works. Hint: To disable phase advance, set this to a value above the over-speed trip level.	1500
12	Phase Adv. Slope	Slope of phase advance in degrees per 1000 RPM. E.g., if = 10, this would give 10°/1000 RPM	25
13	Stall Time Limit	Length of time in 10 ms increments that no rotation detected before trip occurs.	100
14	Over Speed Limit	Over Speed trip in RPM	3300
15	Over Volts Limit	Over Voltage trip in 0.1V increments	500
16	Over Current Lim	Over Current trip in 0.1 A increments	100
17	Current P Gain	Current Loop PID gains. These gains are scaled up by 512 so that fractional gains may be used (e.g., 256 = ½). These gains will require adjustment in control modes 0 or 3.	900
18	Current I Gain		100
19	Current D Gain		0
20	Speed P Gain	Speed Loop PI gains. These gains are scaled up by 16384 so that fractional gains may be used (e.g., 8192 = ½). These gains will require adjustment in control modes 0 and 1.	1000
21	Speed I Gain		10
22	Voltage Demand	Voltage Demand for brake chopper. Hint: To disable brake chopper, set this parameter to a value above the over voltage trip level.	490

TABLE A-1: USER PARAMETERS (CONTINUED)

Param No.	Parameter Name	Description	Default Value
23	Volts P Gain	Brake chopper PI gains. Note that these gains are scaled up by 512 so that fractional gains may be used (e.g., 256 = ½).	10000
24	Volts I Gain		10
25	No. Motor Poles	Number of motor rotor poles, not pole pairs. The number should therefore be even. This parameter affects speed scaling of displayed RPM values.	10
26	Current Scale X	These parameters specify the mathematical scaling of the voltage and current feedback signals and depend upon the resistive divider ratios chosen for the hardware. See the parameters.h file for guidance on setting these values.	100†
27	Current Scale /		539†
28	Volts Scale X		100*
29	Volts Scale /		1305*
30	Tolerance Check	% Variation between zero crossing time stamps before system is lost. Do not set this value below 10%. To disable tolerance checks set it to 99%. Value may need to be increased if fast acceleration/deceleration is required.	50
31	Auto Re-acquire	If enabled, this parameter will let the system automatically reacquire when in lost state.	1
32	Blanking Count	Number of PWM cycles for which zero crossing detection is disabled after a commutation occurs. This delay provides time for the current in the non-driven phase to discharge before the algorithm begins to look for zero crossing.	1
33	Zero X Level Thd	Number of V_{PH} samples required above ½ V_{DC}, or below if rising edge detect, before software begins to look for samples on the other side of the zero crossing threshold. This parameter makes the zero crossing detect algorithm more robust by forcing the software to detect a minimum number of samples above and below the ½ V_{DC} reference point before a valid zero crossing is recognized.	2
34	Acquire Threshld	ADC value used for the rising edge detect when acquiring the position. A value of 10 would give a 50 mV threshold level. Lower values may allow acquisition at lower BEMF voltages (speed) but may be less reliable due to noise and offset voltages.	10
35	Acquire Level Td	The number of samples of the phase voltage that have to be < the acquire threshold before the phase voltage is checked to be above the acquire threshold. Because the ADC is continuously cycling through the three phase voltage channels, the effective resolution of this parameter is 3.	6
36	Rotation Timeout	Length of time in 10 ms increments in which rotation must be seen for system to automatically acquire and launch sensorless rather than lock and ramp. Setting a longer time may allow the motor to start immediately at a lower speed if there is sufficient BEMF. Setting too long a time may result in failed starting.	5
37	Pot / for Duty	When CONTROL MODE = 2, this parameter scales the pot ADC reading (0-1023) to duty cycle value loaded into PWM generators. For most applications this should left at 1 to give the pull range of PWM.	1
38	Pot / for Currnt	When CONTROL MODE = 3, this parameter scales the pot ADC reading (0-1023) to current demand value in ADC counts.	8
39	Pot X for Speed	When CONTROL MODE = 0 or 1, this parameter scales the pot ADC reading (0-1023) to speed demand value in RPM.	3

TABLE A-1: USER PARAMETERS (CONTINUED)

Param No.	Parameter Name	Description	Default Value
40	Starting Control	When this parameter is set to 1, voltage control is used for starting. When set at 0, current control is used.	1
41	Windmilling Dem.	This parameter is used if windmilling is detected during starting (rotor going opposite direction to the demanded direction). It sets the current demand value used for decelerating the motor to rest.	20
42	Braking Ramp T	This parameter sets the length of time, in 10 ms increments, that the motor speed is reduced to zero when windmilling.	200
43	Acquire Method	When set to 0, Acquisition Method 1 starting is used. When set to 1, Acquisition Method 2 starting is used.	1
44	ZeroX Enable Spd	When using Acquisition Method 1 starting, this parameter sets the open loop stepping speed at which zero crossing detection is enabled.	100

† These parameters are suitable for a low voltage power module.

* These values assume a low voltage power module with LK11 and LK12 open circuit.

APPENDIX B: SOURCE CODE LISTING

The latest software version can be downloaded from the Microchip web site (www.microchip.com). You will find the source code appended to the electronic version of this application note. At the time of this writing, the most current software version is V3.01.

APPENDIX C: PROGRAM FLOW CHARTS

FIGURE C-1: MAIN

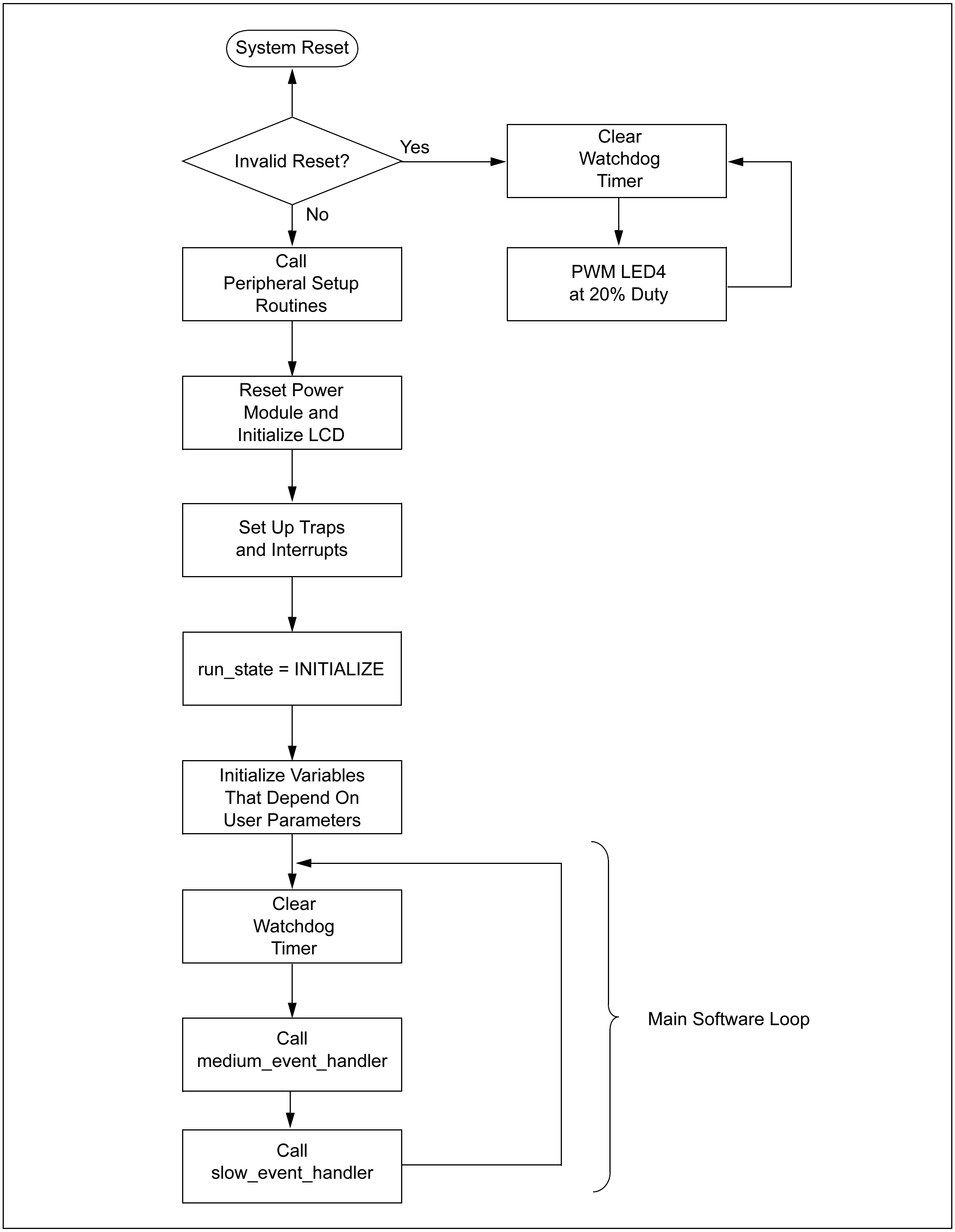


FIGURE C-2: MEDIUM EVENT HANDLER

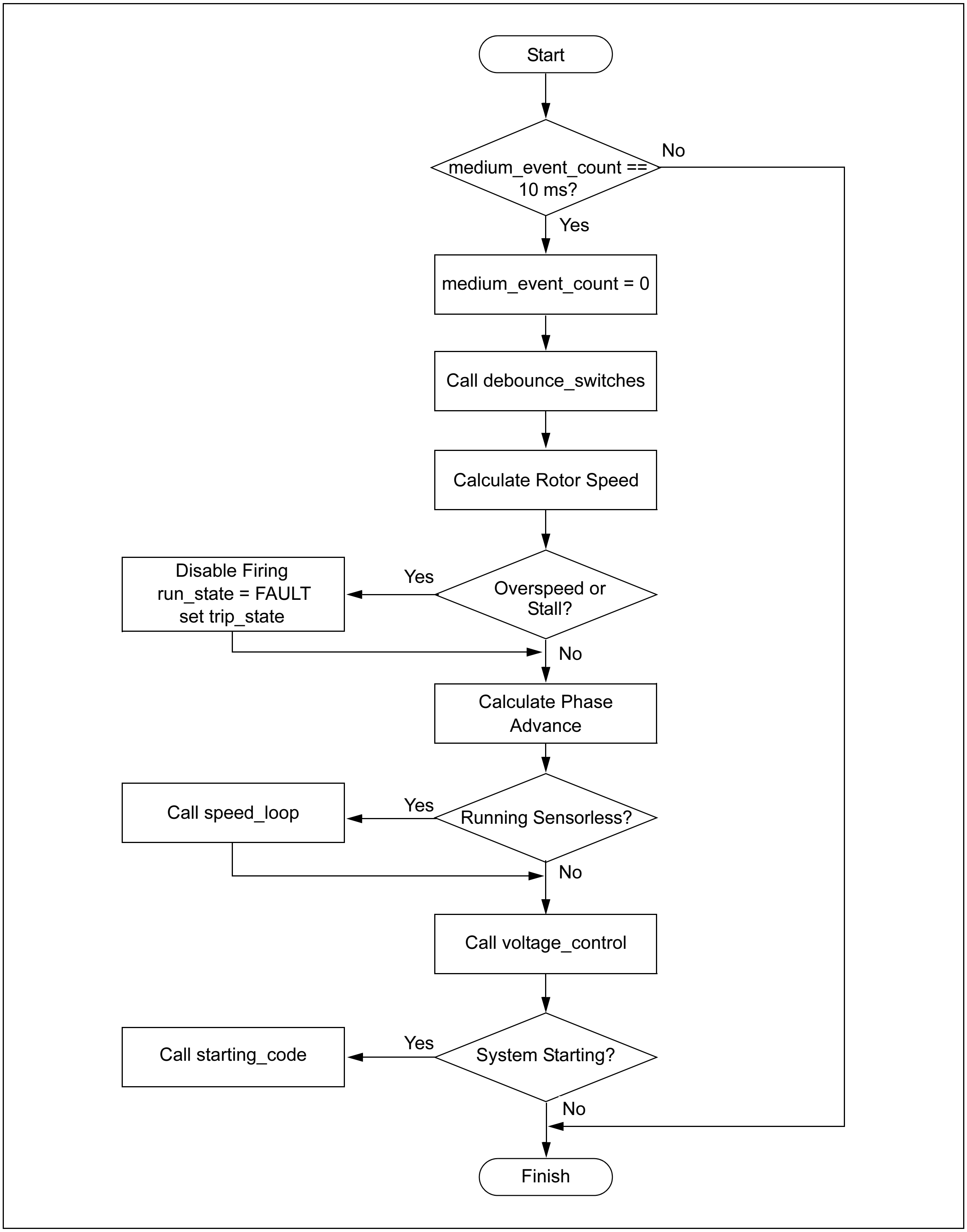


FIGURE C-3: STARTING CODE

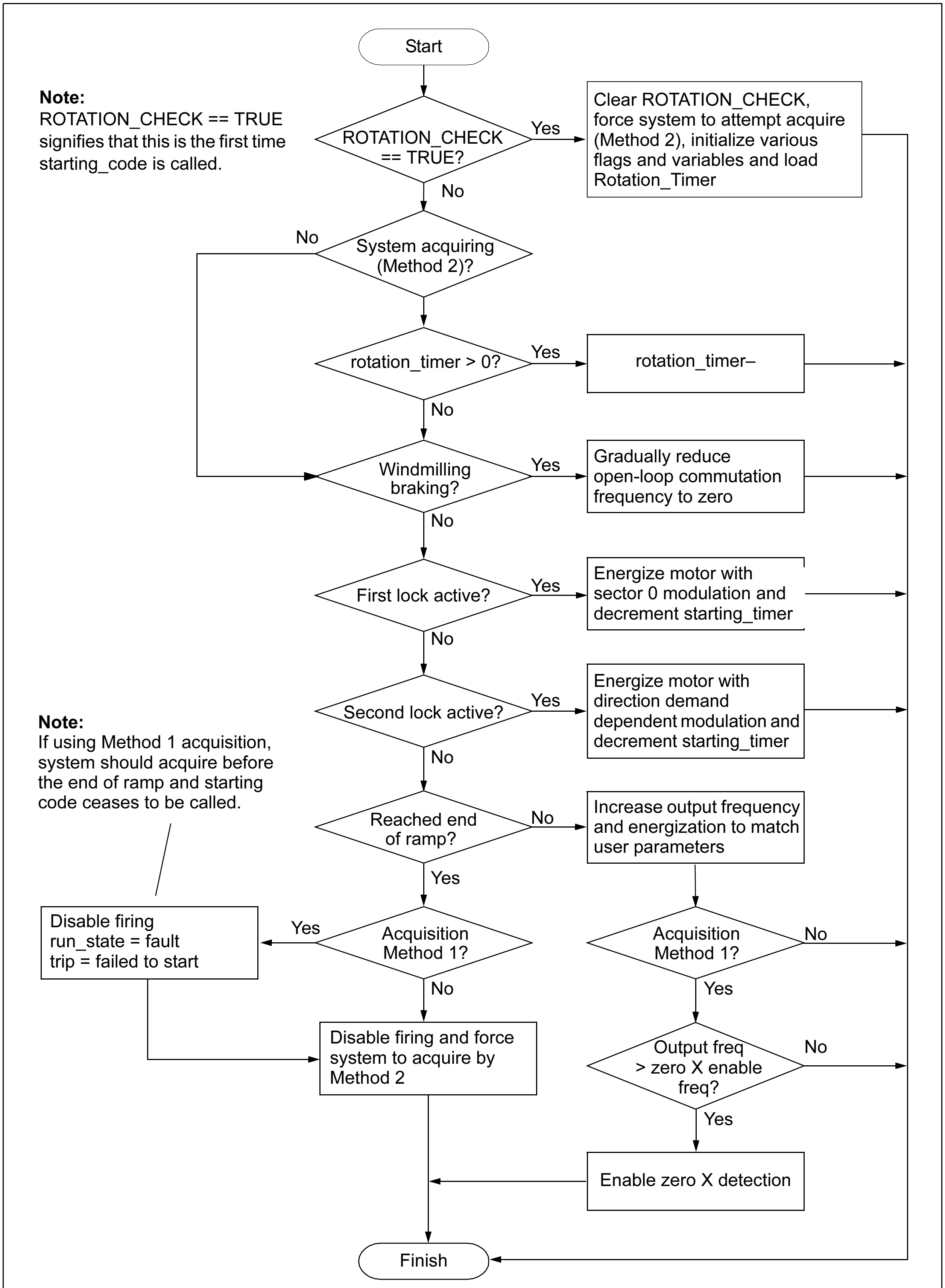


FIGURE C-4: SLOW EVENT HANDLER

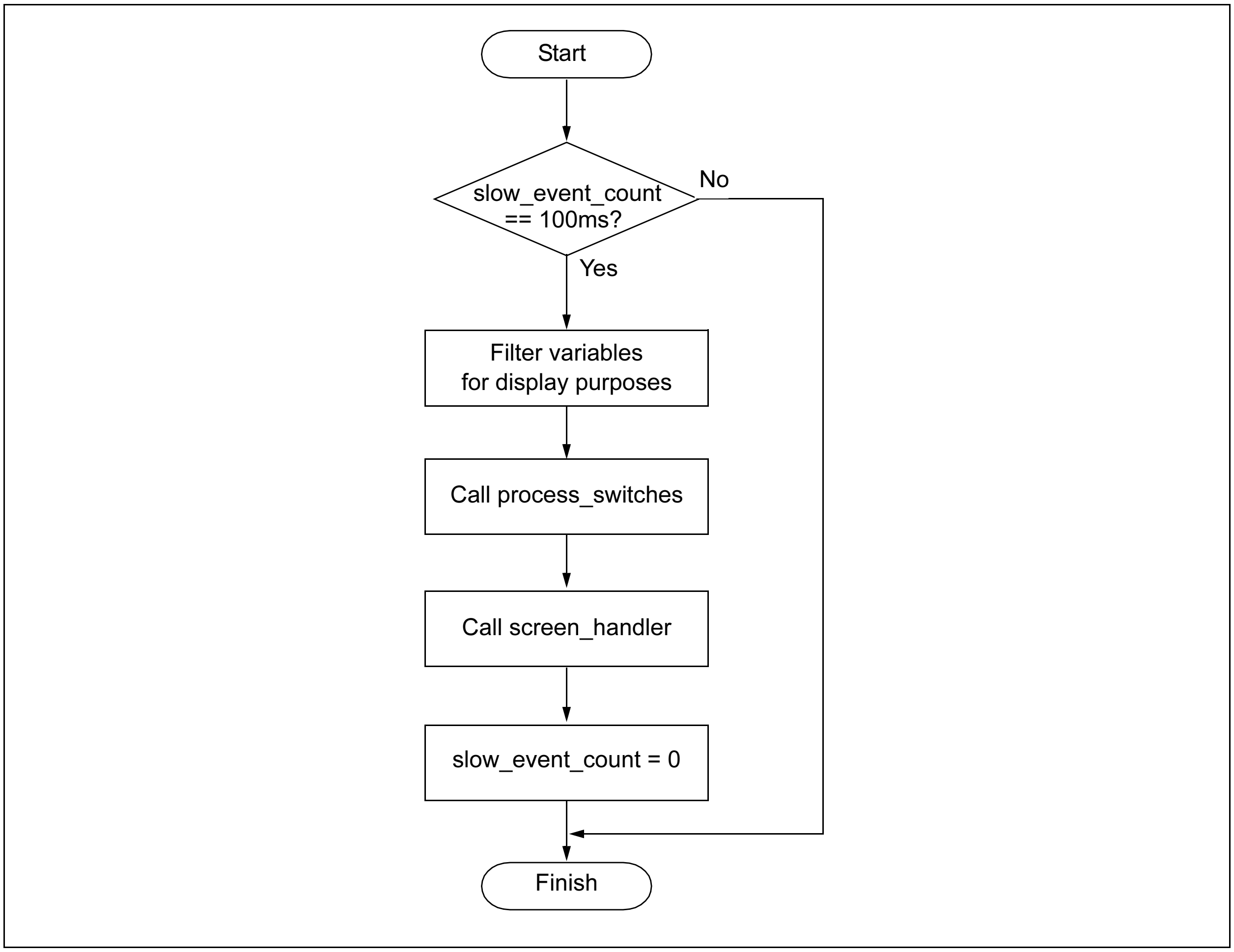


FIGURE C-5: PWM INTERRUPT SERVICE ROUTINE

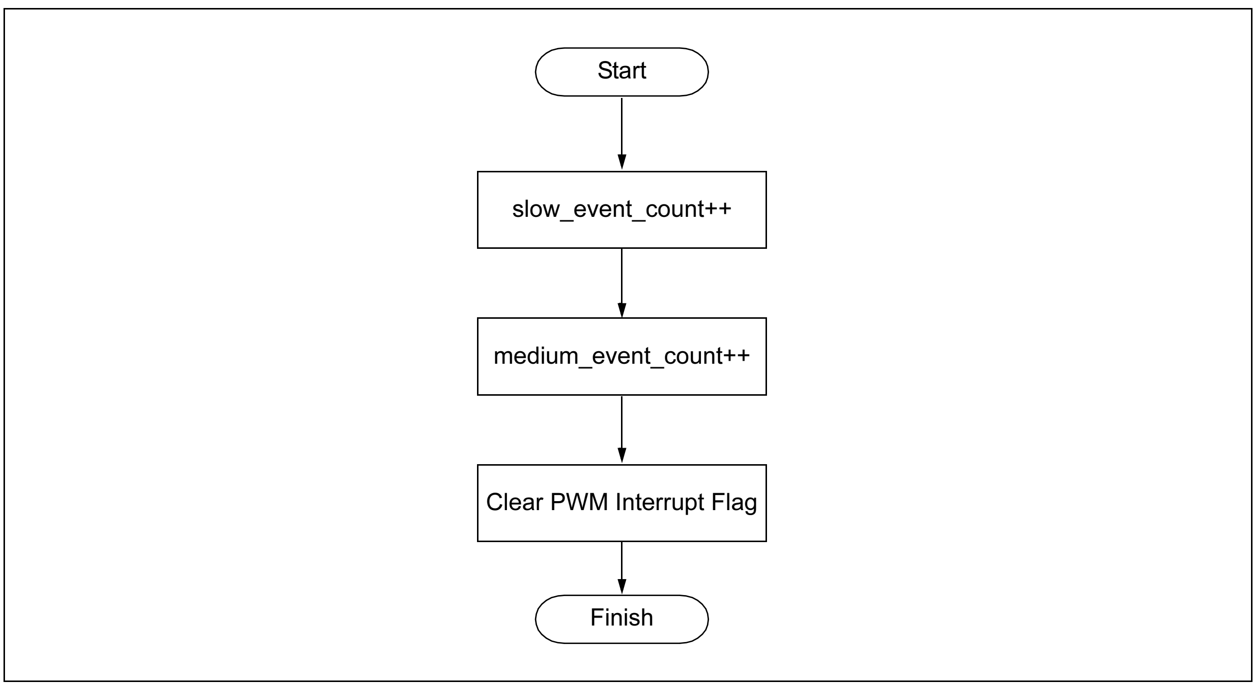


FIGURE C-6: ADC INTERRUPT SERVICE ROUTINE

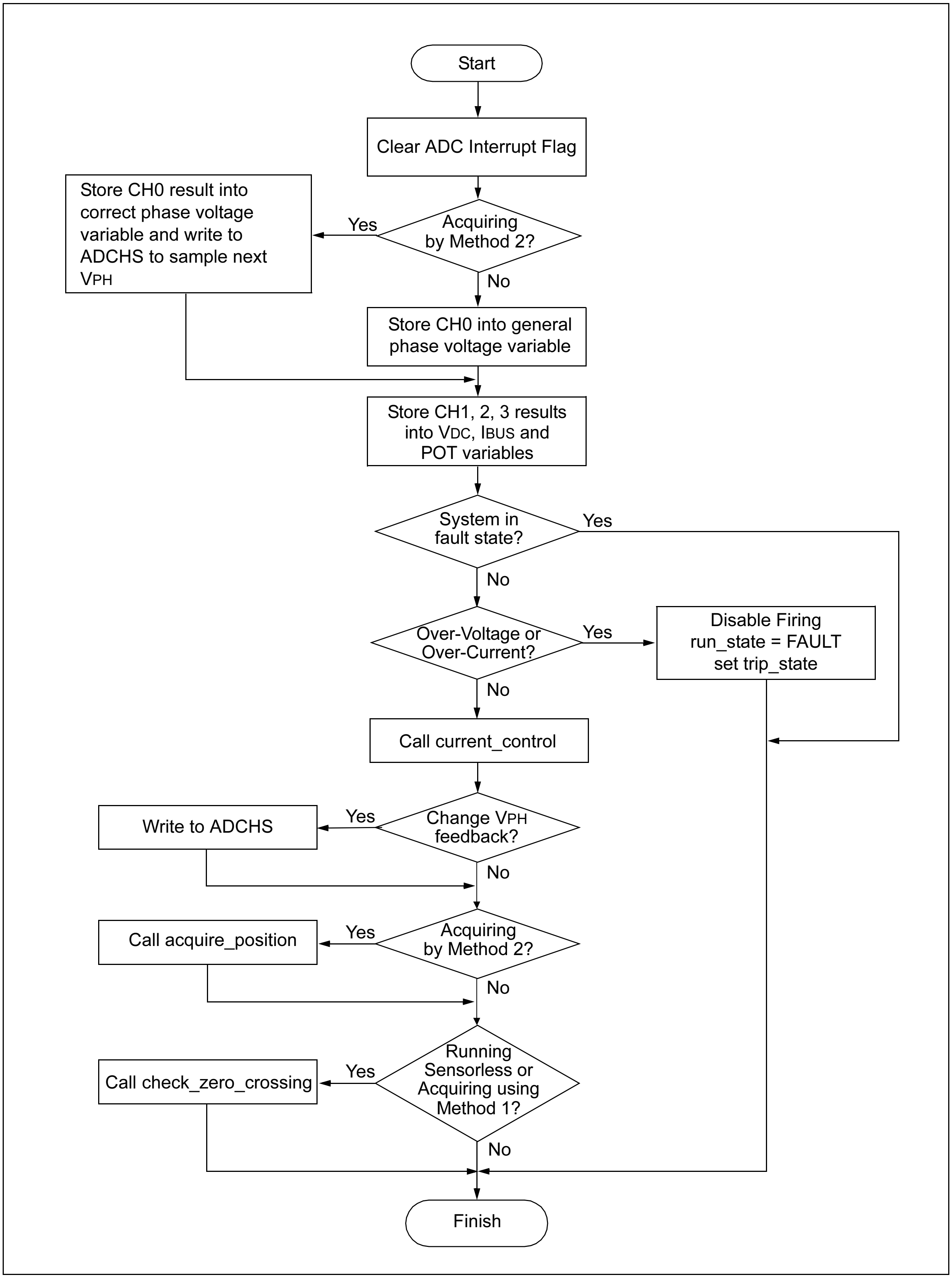


FIGURE C-7: ACQUIRE POSITION (METHOD 2)

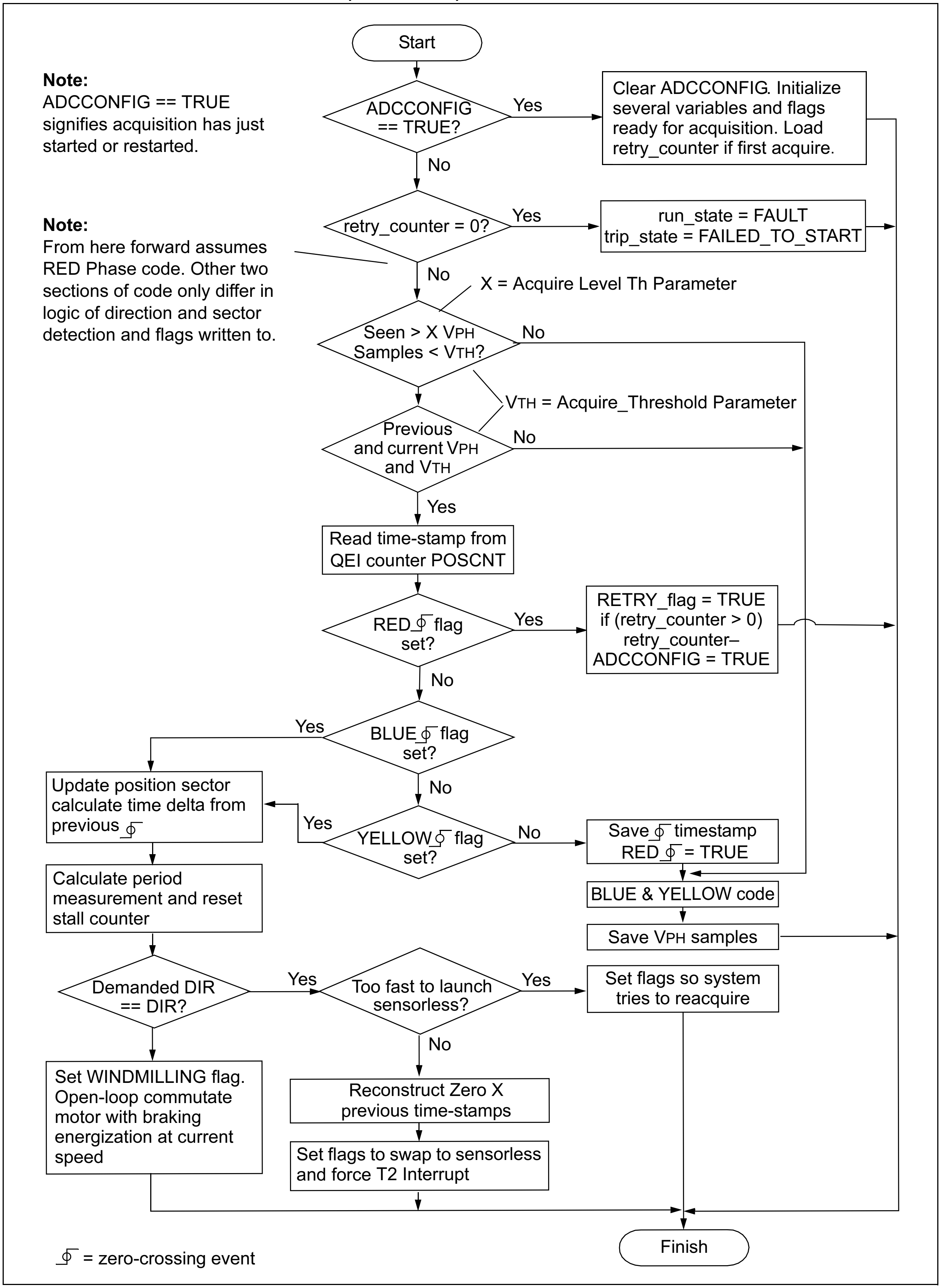


FIGURE C-8: CHECK ZERO CROSSING

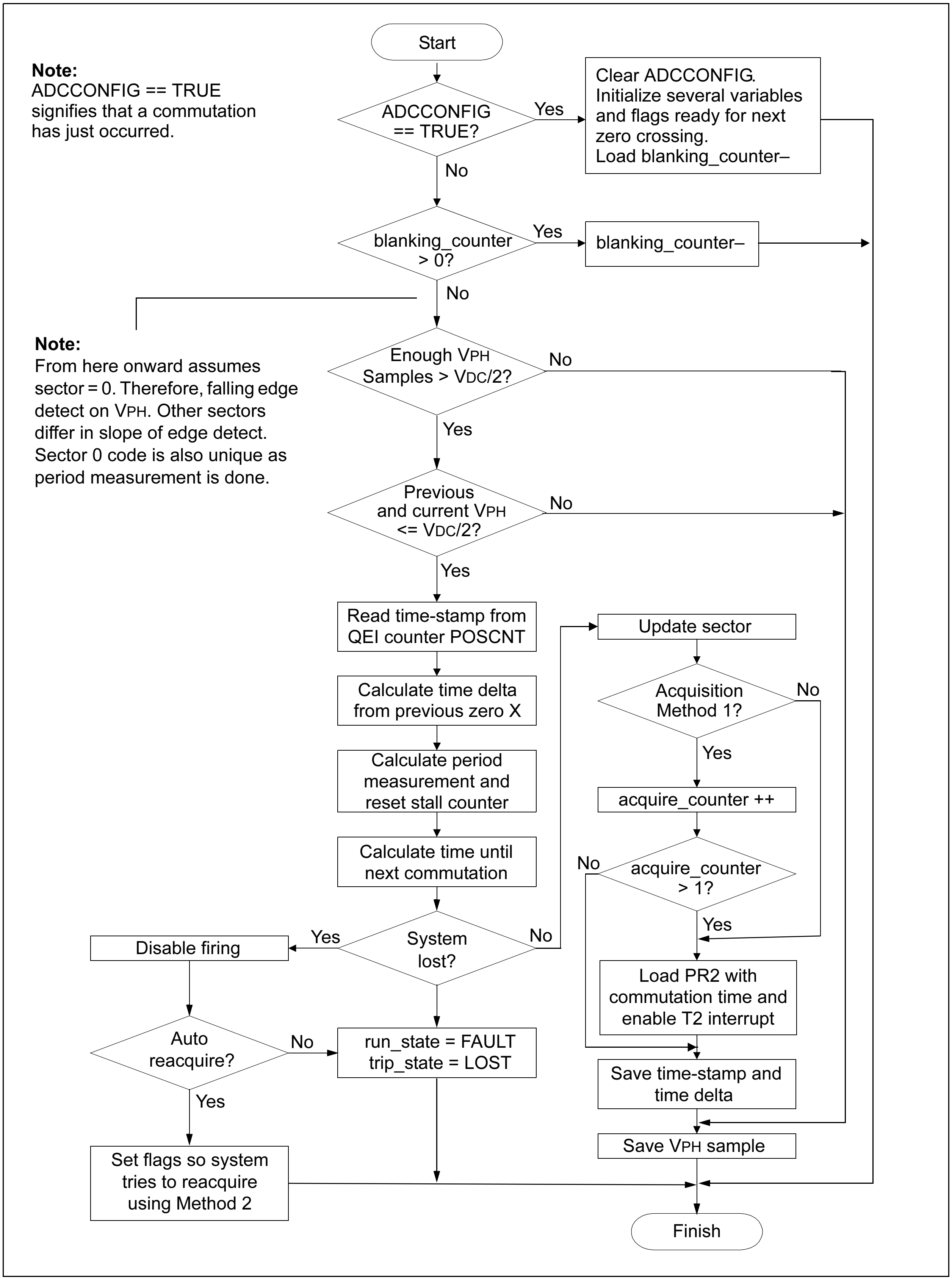
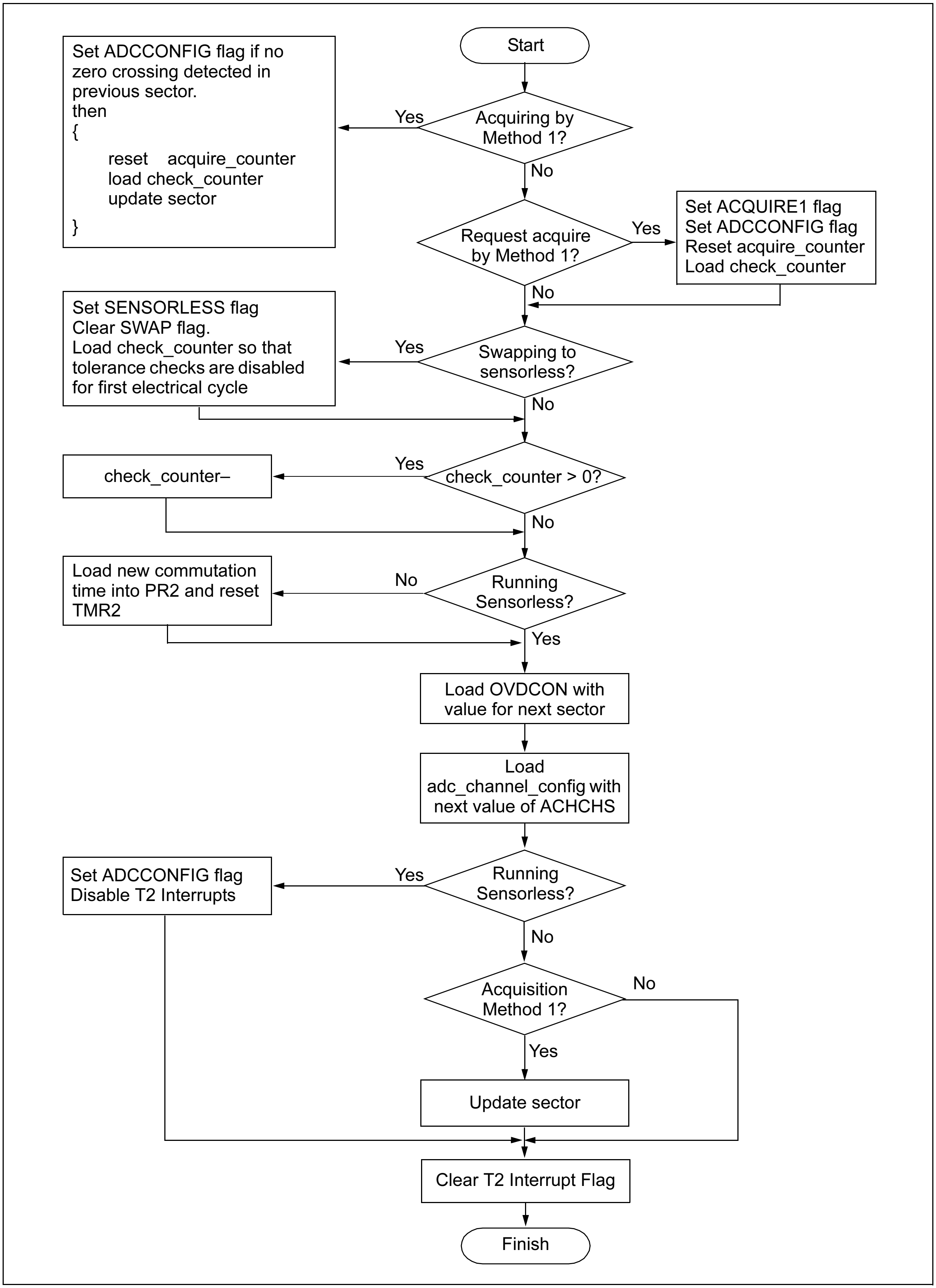


FIGURE C-9: TIMER 2 INTERRUPT SERVICE ROUTINE



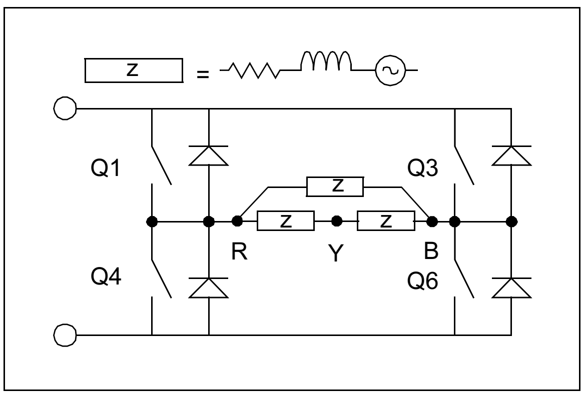
APPENDIX D: DELTA CONNECTED MOTORS

In a Y connected three phase circuit, the 3 phase currents must sum to zero at all times with the voltages being unconstrained. The delta connected circuit is the "dual" of this. The 3 phase voltages must sum to zero and the currents are unconstrained. If the voltages are not balanced in a delta connected source, a current will circulate within the delta that will cause an additional voltage drop across the windings. Clearly this is undesirable as it will cause additional resistive losses. In the case of a motor, any circulating current will cause the generation of an additional braking torque to supply the additional resistive losses. The magnitude of the circulating current will depend on the magnitude of the voltage imbalance and the impedance of the phase windings.

With reference to the BLDC motor, the restriction on phase vector imbalance has a direct influence on the shape of the BEMF waveform that the delta connected motor may have. This is because any triple-n or "3n" (where n = 1, 2, 3 etc.) harmonics will be in-phase and will therefore sum thus creating a voltage imbalance. Given that a trapezoidal waveform will contain substantial 3rd harmonic by its very nature, clearly a delta connected BLDC will not normally have a trapezoidal BEMF. The only time when this is not true is in very small motors where the per-unit phase impedance is very high, which will limit the extent of the circulating current and the loss that results. Assuming that this is not the case, it can be assumed that the Delta connected BLDC BEMF will have a sinusoidal shape. It can still be operated with conventional BLDC energization, but the torque ripple will be increased.

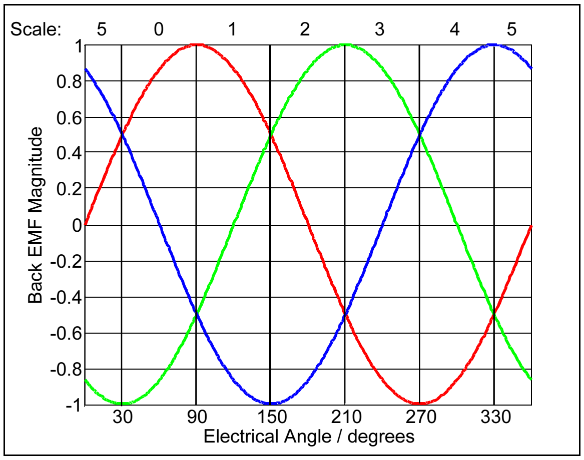
With the above in mind, the influence of the delta motor on the zero crossing BEMF algorithm will now be considered. For delta operation, all three phases have current flowing in them, but only two of the three wires are driven at any one time. This is shown in Figure D-1, where the R and B phase outputs are being driven by the inverter.

FIGURE D-1: DELTA MOTOR OPERATION



If the BEMF RY is positive and the BEMF YB is negative and Q1/Q6 conduct, this is the equivalent to sector 1 of the standard Y connection commutation diagrams given previously. The equivalent diagram for a sinusoidal BEMF is given below.

FIGURE D-2: SINUSOIDAL BEMF



Bearing in mind that the three BEMF voltages sum to zero at all times, half way through a sector, the measured voltage at the inactive terminal will be ½ VDC. This is because at this unique point in the sector, one BEMF is zero and the other two are equal and opposite in magnitude. This is just the same as for the Y connected motor. Again, this corresponds to the correct point to commutate to the next energization pattern.

In summary, despite substantial differences in the BEMF shape and the way the phases conduct, the delta connected motor should work with no changes to the code.

APPENDIX E: PID CONTROLLER BACKGROUND

A complete discussion of Proportional Integral Derivative (PID) controllers is beyond the scope of this discussion, but this section will try to provide you with some guidelines for tuning the controllers.

A PID controller responds to an error signal in a closed control loop and attempts to adjust the controlled quantity in order to achieve the desired system response. The controlled parameter can be any measurable system quantity, such as speed, voltage, current or stock price. The output of the PID controller can control one or more system parameters that will affect the controlled system quantity. For example, the speed control loop in this application can control the PWM duty cycle directly or it can set the current demand for an inner control loop that regulates the motor currents. The benefit of the PID controller is that it can be adjusted empirically by adjusting one or more gain values and observing the change in system response.

A digital PID controller is executed at a periodic sampling interval and it is assumed that the controller is executed frequently enough so that the system can be properly controlled. For example, the current controller in this application is executed every PWM cycle, since the motor can change very rapidly. The speed controller in this application is executed at the medium event rate (100 Hz), because motor speed changes will occur relatively slowly due to mechanical time constants.

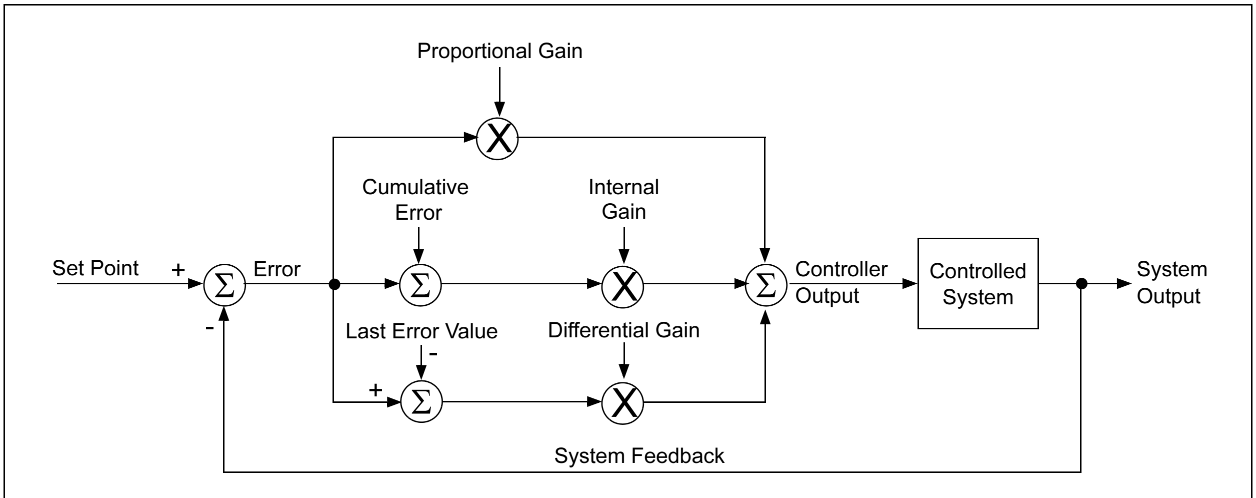
A PID Algorithm block diagram is shown in Figure 13 The error signal is formed by subtracting the desired setting of the parameter to be controlled from the actual measured value of that parameter. This sign of the error indicates the direction of change required by the control input.The Proportional (P) term of the controller is formed by multiplying the error signal by a P gain. This will cause the PID controller to produce a control response that is a function of the error magnitude. As the error signal becomes larger, the P term of the controller becomes larger to provide more correction.

The effect of the P term will tend to reduce the overall error as time elapses. However, the effect of the P term will reduce as the error approaches zero. In most systems, the error of the controlled parameter will get very close to zero, but will not converge. The result is a small remaining steady state error. The Integral (I) term of the controller is used to fix small steady state errors. The I term takes a continuous running total of the error signal. Therefore, a small steady state error will accumulate into a large error value over time. This accumulated error signal is multiplied by an I gain factor and becomes the I output term of the PID controller.

The Differential (D) term of the PID controller is used to enhance the speed of the controller and responds to the rate of change of the error signal. The D term input is calculated by subtracting the present error value from a prior value. This delta error value is multiplied by a D gain factor that becomes the D output term of the PID controller. The D term of the controller produces more control output the faster the system error is changing.

It should be noted that not all PID controllers will implement the D or, less commonly, the I terms. For example, the speed controller in this application does not have a D term due to the relatively slow response time of motor speed changes. In this case, the D term could cause excessive changes in PWM duty cycle that could affect the operation of the sensorless algorithm and produce over current trips.

FIGURE 13: THREE TERM CONTROLLER



E.1 Adjusting the PID Gains

The P gain of a PID controller will set the overall system response. When first tuning a controller, the I and D gains should be set to zero. The P gain can then be increased until the system responds well to set-point changes without excessive overshoot or oscillations. Using lower values of P gain will ‘loosely’ control the system, while higher values will give ‘tighter’ control. At this point, the system will probably not converge to the set-point.

After a reasonable P gain is selected, the I gain can be slowly increased to force the system error to zero. Only a small amount of I gain is required in most systems. Note that the effect of the I gain, if large enough, can overcome the action of the P term, slow the overall control response, and cause the system to oscillate around the set-point. If this occurs, reducing the I gain and increasing the P gain will usually solve the problem.

After the P and I gains are set, the D gain can be set. The D term will speed up the response of control changes, but it should be used sparingly because it can cause very rapid changes in the controller output. This behavior is called ‘set-point kick’. The set-point kick occurs because the difference in system error becomes instantaneously very large when the control set-point is changed. In some cases, damage to system hardware can occur. If the system response is acceptable with the D gain set to zero, you can probably omit the D term.

NOTES:

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.
- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as "unbreakable."

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break microchip's code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Trademarks

QUALITY MANAGEMENT SYSTEM
CERTIFIED BY DNV
ISO/TS 16949:2002

Microchip received ISO/TS-16949:2002 quality system certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona and Mountain View, California in October 2003. The Company's quality system processes and procedures are for its PICmicro® 8-bit MCUs, KEELOQ® code hopping devices, Serial EEPROMs, microperipherals, non-volatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS

Corporate Office
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7200
Fax: 480-792-7277
Technical Support: 480-792-7627
Web Address: http://www.microchip.com

Atlanta
3780 Mansell Road, Suite 130
Alpharetta, GA 30022
Tel: 770-640-0034
Fax: 770-640-0307

Boston
2 Lan Drive, Suite 120
Westford, MA 01886
Tel: 978-692-3848
Fax: 978-692-3821

Chicago
333 Pierce Road, Suite 180
Itasca, IL 60143
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
4570 Westgrove Drive, Suite 160
Addison, TX 75001
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Tri-Atria Office Building
32255 Northwestern Highway, Suite 190
Farmington Hills, MI 48334
Tel: 248-538-2250
Fax: 248-538-2260

Kokomo
2767 S. Albright Road
Kokomo, IN 46902
Tel: 765-864-8360
Fax: 765-864-8387

Los Angeles
18201 Von Karman, Suite 1090
Irvine, CA 92612
Tel: 949-263-1888
Fax: 949-263-1338

Phoenix
2355 West Chandler Blvd.
Chandler, AZ 85224-6199
Tel: 480-792-7966
Fax: 480-792-4338

San Jose
1300 Terra Bella Avenue
Mountain View, CA 94043
Tel: 650-215-1444

Toronto
6285 Northam Drive, Suite 108
Mississauga, Ontario L4V 1X5, Canada
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC

Australia
Suite 22, 41 Rawson Street
Epping 2121, NSW
Australia
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Unit 706B
Wan Tai Bei Hai Bldg.
No. 6 Chaoyangmen Bei Str.
Beijing, 100027, China
Tel: 86-10-85282100
Fax: 86-10-85282104

China - Chengdu
Rm. 2401-2402, 24th Floor,
Ming Xing Financial Tower
No. 88 TIDU Street
Chengdu 610016, China
Tel: 86-28-86766200
Fax: 86-28-86766599

China - Fuzhou
Unit 28F, World Trade Plaza
No. 71 Wusi Road
Fuzhou 350001, China
Tel: 86-591-7503506
Fax: 86-591-7503521

China - Hong Kong SAR
Unit 901-6, Tower 2, Metroplaza
223 Hing Fong Road
Kwai Fong, N.T., Hong Kong
Tel: 852-2401-1200
Fax: 852-2401-3431

China - Shanghai
Room 701, Bldg. B
Far East International Plaza
No. 317 Xian Xia Road
Shanghai, 200051
Tel: 86-21-6275-5700
Fax: 86-21-6275-5060

China - Shenzhen
Rm. 1812, 18/F, Building A, United Plaza
No. 5022 Binhe Road, Futian District
Shenzhen 518033, China
Tel: 86-755-82901380
Fax: 86-755-8295-1393

China - Shunde
Room 401, Hongjian Building, No. 2
Fengxiangnan Road, Ronggui Town, Shunde District, Foshan City, Guangdong 528303, China
Tel: 86-757-28395507 Fax: 86-757-28395571

China - Qingdao
Rm. B505A, Fullhope Plaza,
No. 12 Hong Kong Central Rd.
Qingdao 266071, China
Tel: 86-532-5027355 Fax: 86-532-5027205

India
Divyasree Chambers
1 Floor, Wing A (A3/A4)
No. 11, O'Shaugnessey Road
Bangalore, 560 025, India
Tel: 91-80-2290061 Fax: 91-80-2290062

Japan
Benex S-1 6F
3-18-20, Shinyokohama
Kohoku-Ku, Yokohama-shi
Kanagawa, 222-0033, Japan
Tel: 81-45-471- 6166 Fax: 81-45-471-6122

Korea
168-1, Youngbo Bldg. 3 Floor
Samsung-Dong, Kangnam-Ku
Seoul, Korea 135-882
Tel: 82-2-554-7200 Fax: 82-2-558-5932 or 82-2-558-5934

Singapore
200 Middle Road
#07-02 Prime Centre
Singapore, 188980
Tel: 65-6334-8870 Fax: 65-6334-8850

Taiwan
Kaohsiung Branch
30F - 1 No. 8
Min Chuan 2nd Road
Kaohsiung 806, Taiwan
Tel: 886-7-536-4818
Fax: 886-7-536-4803

Taiwan
Taiwan Branch
11F-3, No. 207
Tung Hua North Road
Taipei, 105, Taiwan
Tel: 886-2-2717-7175 Fax: 886-2-2545-0139

EUROPE

Austria
Durisolstrasse 2
A-4600 Wels
Austria
Tel: 43-7242-2244-399
Fax: 43-7242-2244-393

Denmark
Regus Business Centre
Lautrup hoj 1-3
Ballerup DK-2750 Denmark
Tel: 45-4420-9895 Fax: 45-4420-9910

France
Parc d'Activite du Moulin de Massy
43 Rue du Saule Trapu
Batiment A - ler Etage
91300 Massy, France
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany
Steinheilstrasse 10
D-85737 Ismaning, Germany
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy
Via Quasimodo, 12
20025 Legnano (MI)
Milan, Italy
Tel: 39-0331-742611
Fax: 39-0331-466781

Netherlands
P. A. De Biesbosch 14
NL-5152 SC Drunen, Netherlands
Tel: 31-416-690399
Fax: 31-416-690340

United Kingdom
505 Eskdale Road
Winnersh Triangle
Wokingham
Berkshire, England RG41 5TU
Tel: 44-118-921-5869
Fax: 44-118-921-5820

01/08/04