A Variable-Speed Sensorless System for BLDC Motor

V1.1 – Dec 8, 2005

<u>English Version</u>

19, Innovation First Road • Science Park • Hsin-Chu • Taiwan 300 • R.O.C.

Tel: 886-3-578-6005 Fax: 886-3-578-4418 E-mail: mcu@sunplus.com

http://www.sunplusmcu.com http://mcu.sunplus.com

Important Notice

SUNPLUS TECHNOLOGY CO. reserves the right to change this documentation without prior notice. Information provided by SUNPLUS TECHNOLOGY CO. is believed to be accurate and reliable. However, SUNPLUS TECHNOLOGY CO. makes no warranty for any errors which may appear in this document. Contact SUNPLUS TECHNOLOGY CO. to obtain the latest version of device specifications before placing your order. No responsibility is assumed by SUNPLUS TECHNOLOGY CO. for any infringement of patent or other rights of third parties which may result from its use. In addition, SUNPLUS products are not authorized for use as critical components in life support systems or aviation systems, where a malfunction or failure of the product may reasonably be expected to result in significant injury to the user, without the express written approval of Sunplus.

Table of Content

PAGE

Revision History

Revision	Date	By	Remark	Page Number(s)
V 1.1	2005/12/8		Update	
V1.0	2004/10/26		First	

1 SPMC75F2413A BLDC controller

1.1 Introduction

The SPMC75F2413A digital motor controller developed by SUNPLUS is well suitable for wide range of variable voltage and variable frequency motor drives. It supports up to two channels of six-independent PWM waveform modulated output and others peripherals that need to be implemented in real product. This application note describes the fundamental theory and control topology of BLDC motor, and the position sensorless detection scheme based on position detection control (PDC) timer module on SPMC75F2413A. Finally, it shows the sensorless is cost and reliability effective in comparison with the hall sensored solution through the experimental results.

1.2 SUNPLUS SPMC75F2413A Controller Features

The SUNPLUS SPMC75F2413A is well suitable for digital motor control product, supports up twenty channels of PWM waveform output and eight channels of duration calculation by input capture function. It equips the u'nSP kernel authored from SUNPLUS with few DSP instructions which can easily do multiply-accumulate and filter operation. This MCU offers lots of dedicated peripherals such as General-Purpose IO (GPIO), Position Detection Control (PDC) Timer, TPM Timer, Motor Control PWM (MCP) Timer, Compare Match Timer (CMT), Standard Peripheral Interface (SPI), UART, Analog-to-Digital Converter (ADC) and Watchdog Timer (WDT). The memory interface includes 32K word Flash memory and 2K words SRAM memory. Following describes the detail features of SPMC75F2413A.

- SUNPLUS 16-bit u'nSP processor (ISA 1.2)
- Operating voltage:
  — Core: 4.5V~ 5.5V
- Operating speed: Maximum 24MHz
- Operating temperature: -40 ℃~85 ℃
- On-chip Memory
  — 32KW (32K x 16) Flash
  — 2KW (2K x 16) SRAM
- On-chip PLL based clock generation
- Watchdog timer
- 10-bit analog-to-digital converter
  — 8 multiplexed input channels
  — 10us (100kHz) conversion time
- Serial communication interface
  — UART

— SPI

- Up to 64 GPIO pins
- Power management

— 2 power-down modes: Wait/Standby

— Each peripheral can be powered down independently

- Two Compare Match Timers
- Five 16-bit general-purpose timers

— 2 for PWM, 2 for rotor speed capturing, 1 for speed loop

— MCP Timer channel 3 supports TIO3A-TIO3F, MCP Timer channel 4 supports TIO4A-TIO4F

— PWM timers support up/down count, compare match output function

— PDC Timer channel 0/1 each supports 3-channel capture input TIO0A-TIO0C and TIO1A-TIO1C

— TPM Timer channel 2 supports Capture/PWM compare match output function

- Twelve 16-bit PWM outputs

— 2-ch Motor drive PWM outputs (3-phase 6-pin complementary PWM outputs)

— TIO3A-TIO3F work with MCP Timer channel 3, TIO4A-TIO4F work with MCP Timer channel 4

— Center- or Edge-aligned PWM outputs

— Temporary PWM outputs shutdown with external overload protection pins

— Emergency PWM outputs shutdown with external fault protection pins

— Programmable Dead time control

— PWM service and fault interrupt generation

— Capable of driving AC induction and BLDC motors

- Embedded In-Circuit-Emulation Circuit

The position sensorless control greatly benefits from the PDC Timer module, MCP Timer module, and speed closed loop timer. The following describes the PDC and MCP Timers:

A. PDC Timers

❑ Capability to process up to six inputs for capture or hall position input. Or six output for PWM operation.

❑ Six timer general registers (TGRAx/TGRBx/TGRCx, x = 0, 1): three registers for each channel independently assignable PWM or input capture functions.

❑ Six timer buffer registers (TBRAx/TBRBx/TBRCx, x = 0, 1) : three registers for each channel used for PWM buffering and capture operation.

❑ Selection of eight programmable clock source: six internal clocks (FCK/1, FCK/4, FCK/16, FCK/64, FCK/256, FCK/1024), two external clocks (TCLKA and TCLKB).

❑ Programmable of timer operating modes:

1. Timer counting modes:

– Normal counting mode: continuous up counting.

– PWM function: selection of 1 output, 0 output at compare match and output hold.

– Input capture function: selection of issue capture at clock rising, falling, both edge, and position detection change event.

2. PWM mode:

– Three independent PWM output for each channel and can be provided with desired duty ratio.

3. Edge-aligned PWM generation mode:

– PWM output for normal and up counting operation.

4. Center-aligned PWM generation mode:

– PWM output for normal and directional up-down counting operation.

5. Position detection change (PDC) module:

– Programmable sample clock source for position signals input: four internal clocks source (FCK/4, FCK/8, FCK32 and FCK/128).

– Programmable position sampling mode: selection of sample position when PWM on, regularly, and lower phase are conducting current.

– Programmable sampling count for position signal to avoid glitch affects the position data.

❑ Totally 14 interrupt sources:

– One timer period compare match interrupt and counter overflow/underflow interrupt sources.

– Three TGR register compare match interrupt sources.

– Position detection changes interrupt sources.

❑ Timer buffer operation:

– The input capture register can be consisted of double buffers. The PWM timer general register can automatically be modified.

B. MCP Timers

❑ Capability to generate up to twelve programmable PWM waveform.

❑ Six timer general registers (TGRAx/TGRBx/TGRCx, x = 3, 4): three registers for each channel independently assignable PWM compare match output functions.

❑ Six timer buffer registers (TBRAx/TBRBx/TBRCx, x = 3, 4) : three registers for each channel used for PWM buffering operation.

❑ Partial load of duty value of PWM generation problem prevention: provide a load control register to make the duty value for PWM could be load simultaneously.

❑ Selection of eight programmable clock source: six internal clocks (FCK/1, FCK/4, FCK/16, FCK/64, FCK/256, FCK/1024), two external clocks (TCLKA and TCLKB).

❑ Programmable of timer operating modes:

1. Timer counting modes:

– Normal counting mode: continuous up counting.

– PWM compare match output function: selection of 1 output, 0 output at compare match and output hold.

2. PWM mode:

– Two independent set of six-phase PWM output can be provided with desired duty ratio.

3. Edge-aligned PWM generation mode:

– PWM output for normal and up counting operation.

4. Center-aligned PWM generation mode:

❑ Totally 10 interrupt sources, five for each channel:

– Timer period compare match interrupt.

– TGRD register compare match interrupt sources.

– External fault input interrupt.

– External overload input interrupt.

– PWM output short protection interrupt.

❑ UVW phases output synchronization source select: synchronized to position data register P_POSx_DectData (x = 0, 1) change, TGRB or TGRC register compare match.

❑ PWM duty mode selection: use TGRA or three phase are independent.

❑ Timer buffer operation:

– The input capture register can be consisted of double buffers. The PWM compare match register can automatically be modified.

2 Position Sensorless Control of BLDC Motor

2.1 BLDC Motor and PWM

The behavior of Brushless-DC (BLDC) motor is something like the DC motor but with help of electrical commutating phase with transistor devices. There is nearly zero wiring impedance on the rotor due to the permanent magnet, so it provides good motor efficiency in comparison with DC and AC induction motor. The rotor flux generated from permanent magnet can be divided two kinds of waveform: trapezoidal or sinusoidal back-EMF wave shape. In this application note, the controlled motor is chosen as a trapezoidal waveform and main control methodology is 120-degree square wave PWM modulation. The figure 3-1 depicts the back-EMF waveform with the rotor position signals.

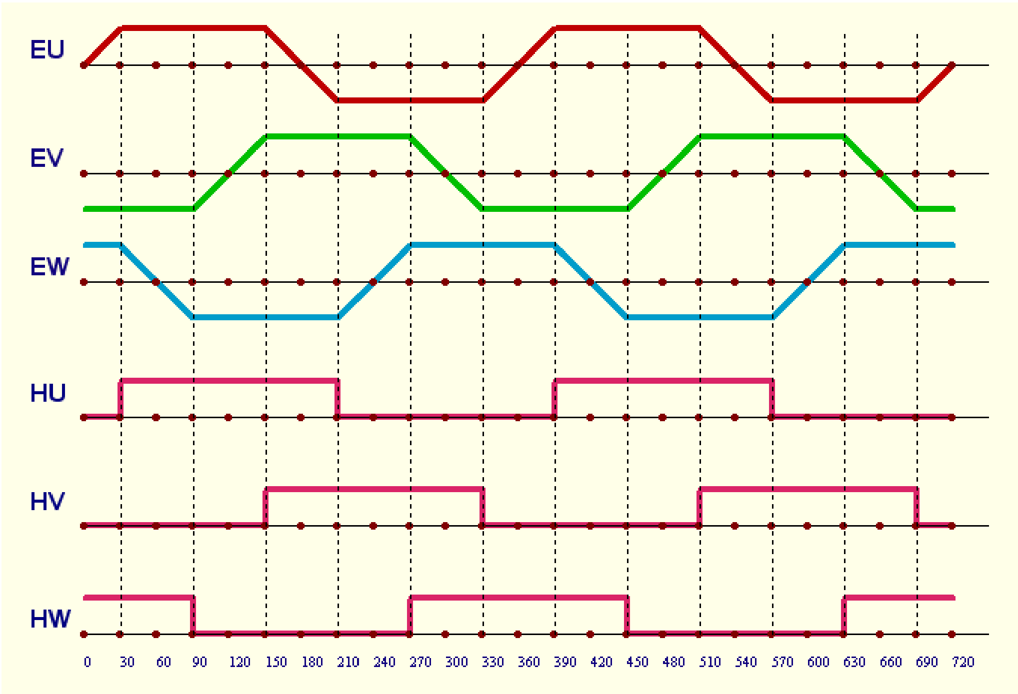

Figure 3-1 Back-EMF vs. hall signals

In above figure, the back-EMF zero crossing point and hall position edge signal phase displacement can be 30 degree lead or lag. The EU, EV and EW signals are the three-phase back-EMF and HU, HV, and HW are the three-phase rotor position signals. The x-axis unit is the electrical degree and every one electrical revolution is 360 degree. The pole-pair of the BLDC motor defines the ratio between the electrical revolution, F and the mechanical revolution, N where F is frequency in Hz and N is the speed in rpm.

This is the necessary condition need for 120-degree PWM. There are totally 4 combination of 120-degree PWM waveform: upper-leg switching, lower-leg switching, pre-sixty degree switching and post-sixty degree switching [3]. In this application note, we select the upper-leg switching PWM for waveform generation.

The three phase voltage of BLDC motor is generated from the inverter circuit in figure 3-2, it behaves as the DC-to-AC conversion. The voltage and frequency of three phases can be modulated by MCP Timers on SPMC75F2413A, commutate phase according the rotor position. In the 120 degree PWM scheme, there are only two transistors device in active state and others are completely shut-down during the same commutation phase. In this example, the ST1 in upper phase performing PWM switching and ST2 in lower phase do turn-on action. The motor current flow is depicts as the blue line. Every wiring phase contains the resistance ***R***, inductance ***L***, and back-EMF ***e***.

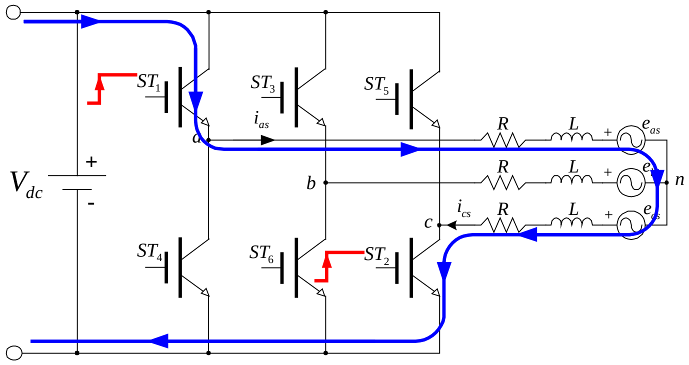

Figure 3-2 Inverter stage circuit for 120-degree PWM

The formula below represents the motor terminal model shown in figure 3-2. The Vn is the star connection neutral voltage reference to power ground and Vp is the phase voltage reference to power ground.

$$V_p = RI_p + L\frac{dI_p}{dt} + e_p + V_n \qquad (1)$$

For any kind of three phase motor, the sum of stator current and back-EMF are both equal to zero. The amplitude of neutral voltage of start connection is one-third of per-phase and frequency of neutral voltage is three times faster than per-phase frequency. The following formula depicts the calculation of neutral voltage.

$$V_n = \frac{1}{3}\sum_{p=1}^{3} V_p \qquad (2)$$

When the specified phase is in non-conducting state, it means no current flow and this phase voltage can be simplified as follows:

$$e_p = V_p - \frac{1}{3}\sum_{p=1}^{3} V_p \qquad (3)$$

Thus, with the comparison between non-conducting phase and neutral voltage, we can obtain the back-EMF signal of the phase, and the rotor signal can be directly derived through the figure 3-1.

2.2 Position Sensorless Control

The rotor position signals are the necessary condition for achieving the closed-speed control and well electrical phase commutation. Without these signals feedback or incorrect, the motor current gets much larger and unstable. But in some applications field, the mounting of hall sensor is not allowed because of the high temperature or pressure or limited space. So the position sensorless control methodology [4] is need for such application.

Figure 3-3 shows the dedicated position sensorless detection hardware, it is obviously the system cost is highly saving in comparison with hall sensors, and it consist a few of low cost resistance, capacitor and one comparator device. The circuit parameters of this circuit should be well-tuned to suit the specified motor characteristics.

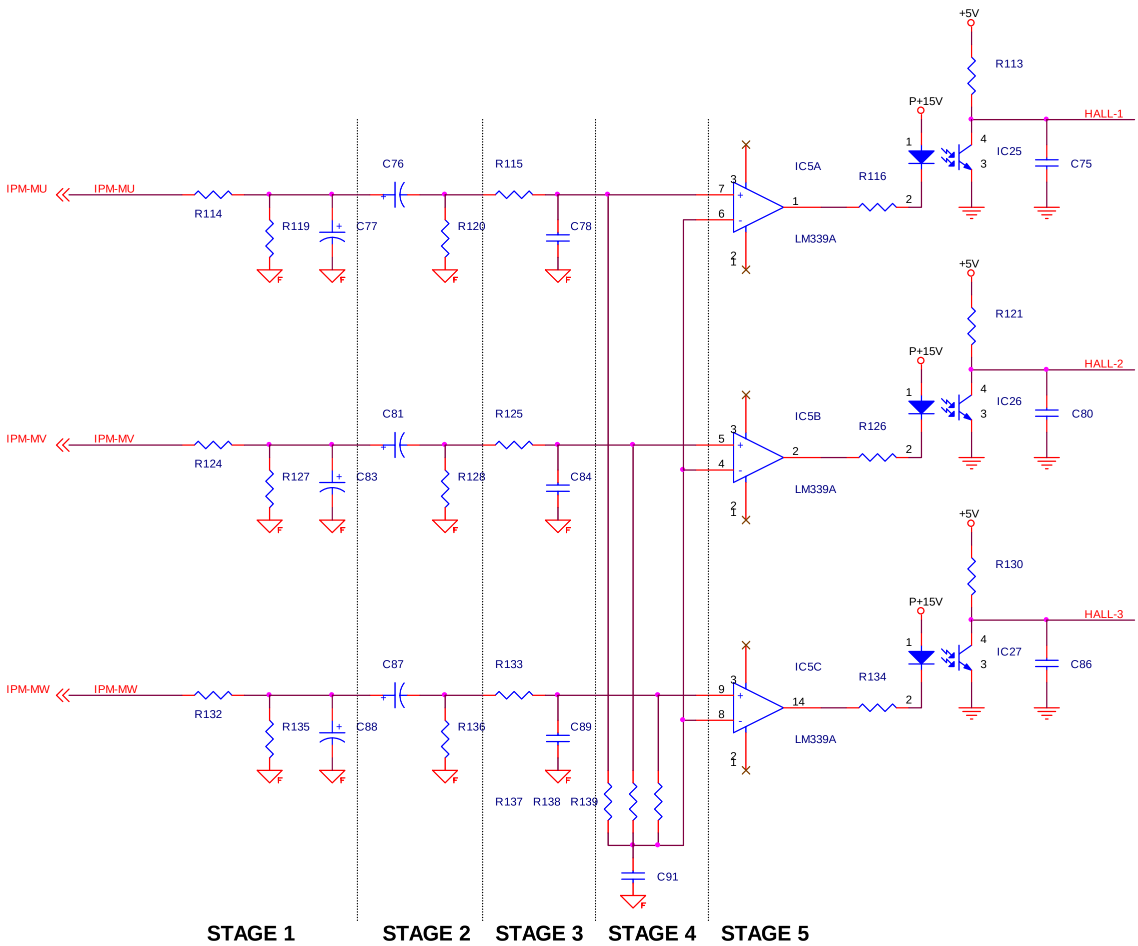

Figure 3-4 Position detection circuit for sensorless

The above circuit can be divided into five stage and input signals are connected to the intelligent power module (IPM) device; they are IPM_MU, IPM_MV and IPM_MW. The position detection circuit is three phase independent and parameters are the same, thus we only take care one of three phases, phase U as the example. The stage 1 circuit forms the low pass filter of the incoming high frequency signal and performs the voltage divider. The amplitude of phase voltage is range from positive half and negative half of DC-bus voltage. The resistors R114 and R119 reduce the voltage level to acceptable level. The equivalent resistance of R114 and R119 can be formed as a simple low pass filter in combine with C77. This filter also gets rid of high frequency signal due to the PWM switching signals and the output signal of stage 1 is trapezoidal shape reference to power ground. According to the Bode Diagram, the phase of output waveform will be lag 90 degree in comparison with IPM_MU.

The circuit of stage 2 forms a high pass filter of filtering the DC component of stage 1 output signal. There is no PWM carrier signal at the output of this stage and signal is reference to a dc-offset not the power ground. The stage 3 performs a further simple low pass filter of stage 2 output signal. The waveform shape is basically the same as output of stage 2.

To get the neutral voltage signal of star-connection, stage 4 circuit do this task. The pseudo star connection is formed by paralleling the three phase resistor R137, R138 and R139, and the neutral voltage is feed into the negative input of comparator device. The capacitor is C91 is used to filter the noise of neutral voltage signal. The amplitude and frequency of neutral voltage is one-third and three times faster in comparison of one phase voltage signal, respectively. The phase relationship between two inputs of comparator is depicts as figure 3-5.

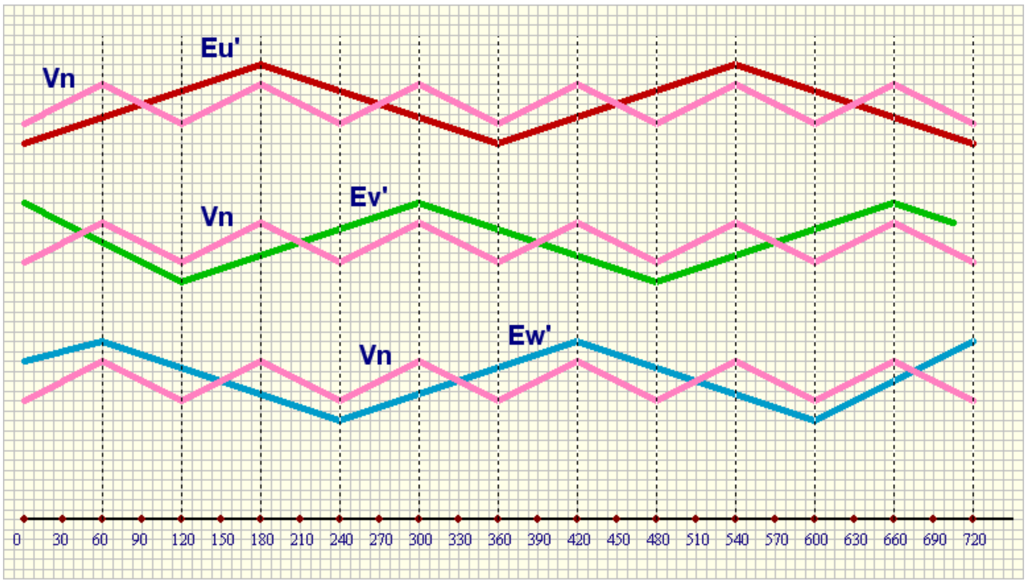

Figure 3-5 Phase relationship between comparator inputs

The signal Vn in figure 3-5 is the neutral voltage, and Eu', Ev' and Ew' are the three phase pseudo back-EMF signals. Through the output of comparator device, we can obtain emulated position signal of BLDC motor. The output signal of stage 5 also includes the transfer the comparator signal through an isolated opto-coupler. The signals HALL-1, HALL-2, and HALL-3 do not actually match the position signal of BLDC motor. It has the inverse polarity and different phase match characteristics [3]. Also note that, this circuit is based on the back-EMF information to detect the rotor position. Thus, it is not possible to get the emulated rotor position at stalling state. The back-EMF voltage is induced when motor is in rotating state, the faster of speed will result in larger back-EMF depend on coefficient Ke, and vice versa. So we need to let the motor running without position signals feedback and called in forced-commutating phase. After the electrical speed of rotor is ten times faster than the cut-off frequency of stage 1, there will be 90 degree shift and we should able to get the emulated position signal at opto-coupler output. The PDC Timers module can detect the position signal and select different position sampling condition to suit ones application [1].

2.3 Alignment for Absolute Position

Before the BLDC motor start to enter the forced-commutating phase, there should be a short period to let the rotor align to the user defined position. It is because we do not know the rotor position when motor is completely stopped. This methodology is achieved by applying PWM signals from MCP Timer onto one upper phase and one lower phase with certain time. The motor current will rise from zero and saturated at a level depend on applied voltage amplitude and motor's electrical time constant. It is obviously the motor current will be controlled fine at defined level when we implement a current loop controller. Different motor characteristics and load condition should be mapping to different alignment time and voltage in order to produce enough electrical torque to align at specified location.

2.4 Position Sensorless Control Flow

Figure 3-6 shows the state transition when motor is from stalling to running. First, the system driver needs to charge the bootstrap voltage in order to supply the upper phase a gate source. It is implemented by turn on all the transistor device at lower phase simultaneously with a certain time to meet the charging parameters of bootstrap circuit. Then make the rotor align to a user defined location. After all these action done successfully, one needs to make the motor start to run to an specified open loop speed without position feedback. This is achieved by increasing the PWM duty ratio value and change the pseudo position signal in system driver. When the motor rotates and able to sample correct position signal for several times, ones can switch from non-position feedback condition to position feedback rotating state. At this stage the speed can be well controlled and also the commutating phase.

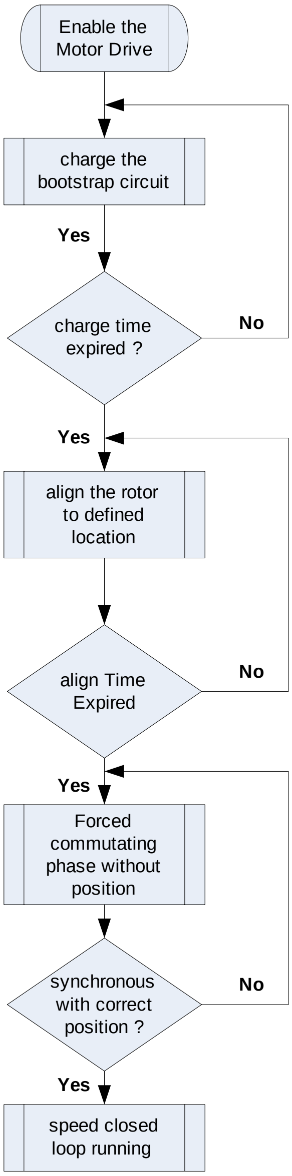

Figure3-6 Position sensorless control flow

2.5 Sensorless BLDC Drive System Implementation

The position sensorless scheme is based on detection of back-EMF zero crossing point. The point of back-EMF zero crossing is generated when one of the three phase windings is not powered. When such point is detected, the corresponding commutating phase action should be happened. The isolated position signals from figure 3-4 are connected to TIO0A, TIO0C and TIO0C pins of PDC Timer 0. When a position detection change interrupt occurred, one can calculate the duration of input position to make sure if enter the synchronous state. The MCP Timer 3 output PWM waveform according to the position signal fetch from PDC Timer 0. The PWM waveform is programmable through P_TMR3_DeadTime and P_TMR3_OutputCtrl registers. There are totally eight waveform combinations in 120-degree PWM: upper-leg switching, lower-leg switching, pre-sixty switching, post-sixty switching, upper-leg switching with dead-time, lower-leg switching with dead-time, pre-sixty switching with dead-time and post-sixty switching with dead-time. Figure 3-7 shows the system implementation block diagram.

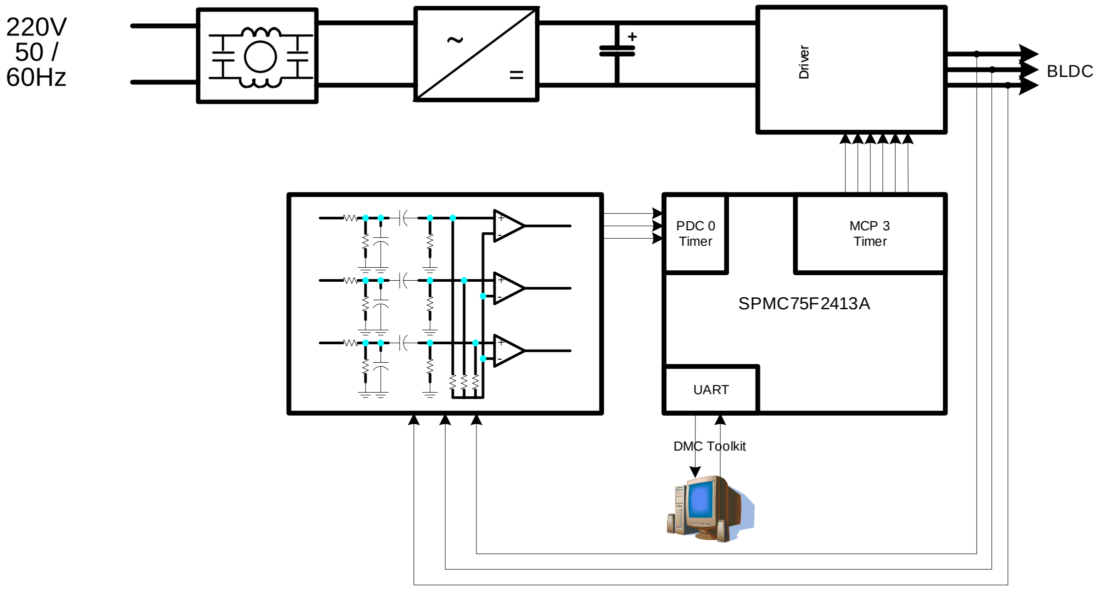

Figure 3-7 Position sensorless control implementation

3 Experimental Results

The control terminal is done through the DMC Toolkit [5][6] running on PC. When user presses the “motor 1 start” icon, the BLDC motor will start to move. In figure 4-1, the first action of motor drive is output a PWM signal to let the rotor align to the specified current. The phase current starts to rise and saturated at about 1.8A. When rotor is stalled at pre-defined location, motor drive inhibits all PWM signals output to release the motor current. If longer alignment time and without release phase current action, it may damage the motor drive board someday. After release phase, the motor drive board enters the forced commutating phase. The motor current will become larger when speed increases. After motor is rotating above the certain speed, fetch the position signals fetched by back-EMF circuit to determine when to enter normal running phase.

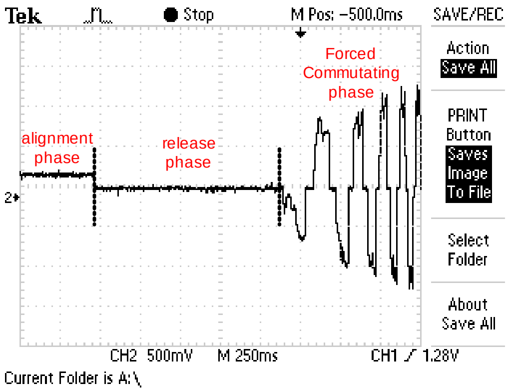

Figure 4-1 Phase current when motor starts to move (5A/div)

The figure 4-2 shows the motor current waveform versus the position signal generated from figure 3-4 circuit. The rising edge and falling edge of hall signal is the action point of commutating current of specified phase. The motor speed is 3000 rpm and from the position and current response, it shows good performance compared to hall sensor signals feedback condition.

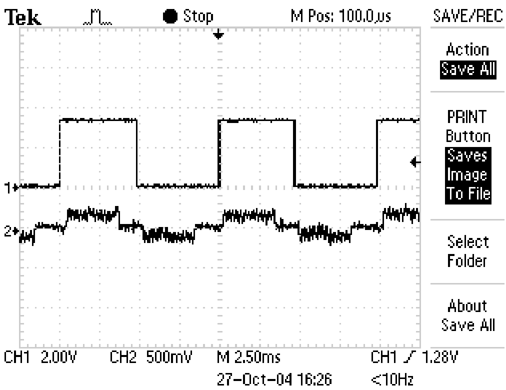

Figure 4-2 Motor current and emulated position signal

The emulated position signal from sensorless circuit, there may exist certain noise on the rising and falling edge of the position signal. In figure 4-3, the glitch interval is about 144us. If there is no filter circuit to remove the glitch, the high-low transition may result in wrong phase commutation and incorrect speed information. In the PDC timer module of SPMC75F2413A, the position sampling clock, counter and condition are all programmable. In such condition, user only needs to setup the SPLCK, SPLMOD, and SPLCNT to be able to remove this kind of glitch.

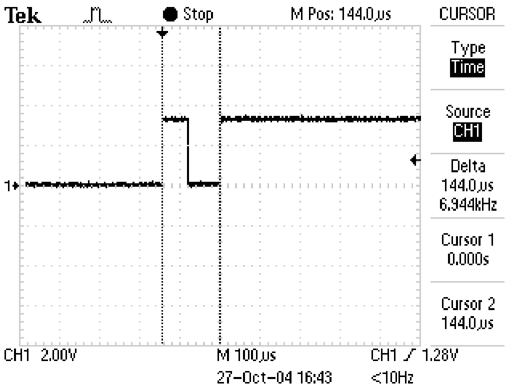

Figure 4-3 Glitch on the position signal

The eight possible PWM waveform combination of 120-degree conduction topology are all programmable from the MCP timer module on SPMC75F2413A. The figure 4-4 shows the pre-sixty 120-degree PWM waveform without dead-time. Different waveform output is achieved by setting the DUTYMODE, POLP, WPWM, VPWM, UPWM, UOC, VOC and WOC bits in P_TMRx_OutputCtrl (x = 3, 4) register.

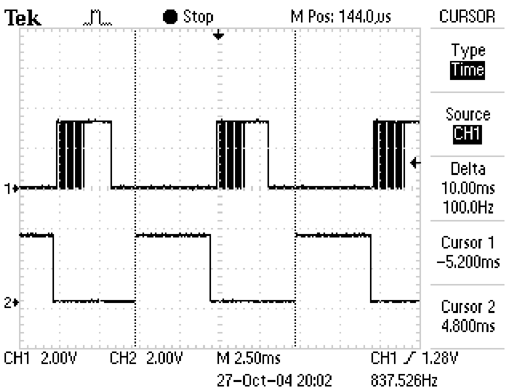

Figure 4-4 Pre-sixty PWM vs. position signal at 3000 rpm

4 Application Example Code

The following sample codes are developed by SUNPLUS and the control terminal is done through DMC Toolkit. The motor drive kernel API function can be reference to document [7]. In Spmc75_irq3.c, the function `MC75F_MCP3_OptoMotorDrv` function is used to do the position sensorless control. And in Spmc75_irq1.c, the function `MC75F_TPM_PDCFilter` captures the interval duration and calculation the motor speed.

```
/* ========================================================================= */
/* File Name   : Spmc75_main.c
*/
/* Project     : AN_SPMC75_0009
*/
/* Description : main entry function of C programming
*/
/* Processor   : SPMC75F2413A
*/
/* Author      : Chih ming Huang
*/
/* Date        : October 2004                                          */
/* Tools       : u'nSP IDE tools v1.16.1
*/
/* Version     : 1.00
*/
/* Security    : Confidential Proprietary
*/
/* E-Mail      : MaxHuang@sunplus.com.tw
*/
/* ========================================================================= */
#include "Spmc75_regs.h"
#include "unspmacro.h"
#include "Spmc75_control_ext.h"
#include "Spmc75_dmc_uart_ext.h"

extern void DMCAPI_bTask(void);

int main(void)
{
```

```
 Disable_FIQ_IRQ();
//
 MC75F_SFR_Init();                              /* initialize IO and timer */
 MC75F_Setup_DrvEn(P_IOB_Data_ADDR, 14);   /* IOB.14 is driver enable pin */
 MC75_DMC_UART_Setup(14400);

 P_TMR1_INT->B.PDCIE = 0;
//
 INT_FIQ_IRQ();                          /* enable FIQ/IRQ channel interrupt */
 MC75F_TPM_TimerStart(PDC0_TMR_ID);                         /* start TMR0 */
 MC75F_TPM_TimerStart(MCP3_TMR_ID);                         /* start TMR3 */
 MC75F_TPM_TimerStart(TPM2_TMR_ID);                         /* start TMR2 */

   MC75F_BLDC_Waveform_Set(BLDC1_ID, BLDC_WAVE_1, PWM_ACTIVE_HI);
    /* setup for MCP3, pre-60 120 degree PWM, PWM is active high */

 MC75F_Set_SpdACC(BLDC1_ID, 500);                  /* acc/dec 500 rpm/sec */

 for(;;)
 {
     MC75_DMC_UART_Service();
     DMCAPI_bTask();
    }
    return (1);
}

/*****************************************************************************/
/* DMCAPI_bTask() : DMC tools API background task                         */
/*****************************************************************************/
void DMCAPI_bTask(void)
{
 Int16 n;
 UInt16 m1, m2 = 0;

 /* check motor run/stop signal command */
 if(SPMC_DMC_Load_MotorSig(1))           /* polling motor 1 run or stop */
     MC75F_TPM_Motor_Start(BLDC1_ID);               /* start the motor */
```

```
    else
    {
         MC75F_TPM_Motor_Stop(BLDC1_ID);                 /* stop the motor */
         MC75F_ClearAlarm();
        }

    n = MC75F_TPM_GetSpeed(BLDC1_ID);
    if(n > 0)
        SPMC_DMC_Save_SpdNow(1, n);   /* display current speed to Speed1_Now */

    n = SPMC_DMC_Load_SpdCmd(1);
    if(n > 0)
        MC75F_TPM_SetSpeed(BLDC1_ID, n);              /* setup speed command */

    n = MC75F_TPM_GetSpeedRef(BLDC1_ID);
    if(n > 0)
        SPMC_DMC_Save_Aux(0, n);                  /* save speed command to R0 */

    /* setup Kp and Ki */
    n = SPMC_DMC_Load_Aux(0);                                    /* W0 = Kp */
    if(n > 0)
        MC75F_Set_SpdKp(BLDC1_ID, n);

    n = SPMC_DMC_Load_Aux(1);                                      /* W1 = Ki */
    if(n > 0)
        MC75F_Set_SpdKi(BLDC1_ID, n);

    m1 = P_TMR3_TPR->W;
    m2 = P_TMR3_TGRA->W;

    SPMC_DMC_Save_Aux(1, 100 - EASYDIVU_32_16((100L * m2), m1));
                                              /* display duty ratio on R1 */

     SPMC_DMC_Save_SpdNow(2,(MC75F_TPM_GetSpeedRef(BLDC1_ID)-
     MC75F_TPM_GetSpeed(BLDC1_ID)));
                                     /* display speed error on Speed2_Now */
  } /* DMCAPI_bTask() */
```

Listing 51 Spmc75_main.c

In Listing 5-1, the communication speed is setup through `MC75_DMC_UART_Setup` function and is 14400 bps. The waveform for driving the BLDC motor is programmed as pre-60 120-degree PWM conduction type and in motor development board chooses a high active IPM (Intelligent Power Module) device. All of these requirements in software are achieved by `MC75F_BLDC_Waveform_Set` function. In `DMCAPI_bTask` function, it performs the background task for DMC Toolkit. This function is polling the motor run or stops status command from the DMC Toolkit, and always receives the speed command from the monitoring window of DMC Toolkit. The speed command will be used for the reference rotating speed of the BLDC motor, and will be controlled well by calling `MC75F_TPM_SpdCalcCtrl` in Spmc75_irq4.c interrupt service routine. The speed command and actual speed value will be display on User_R0 and Speed1_Now control variables, respectively. Developer can well tune the speed performance from observing the charting windows.

```
   /* ========================================================================= */
   /* File Name   : Spmc75_irq6.c
*/
   /* Project     : AN_SPMC75_0009
*/
   /* Description : IRQ6/FIQ definition of C programming
*/
   /* Processor   : SPMC75F series
*/
   /* Author      : Chih ming Huang
*/
   /* Date       : October 2004                                              */
   /* Tools     : u'nSP IDE tools v1.16.1
*/
   /* Version     : 1.00
*/
   /* Security    : Confidential Proprietary
*/
   /* E-Mail      : MaxHuang@sunplus.com.tw
*/
   /* ========================================================================= */
   #include "Spmc75_dmc_uart_ext.h"

   /***************************************************************************/
```

```
    /* UART_RXINT_ISR() : UART receive buffer full interrupt
*/
    /*****************************************************************************/
    void UART_RXINT_ISR(void)
    {
     MC75_DMC_RcvStream();
    } /* UART_RXINT_ISR() */
```

Listing 5-2 Spmc75_irq6.c

In Listing 5-2, the function `MC75_DMC_RcvStream` does the receiving message from the DMC Toolkit. It requires the IRQ6 interrupt hardware resource on the SPMC75F2413A chip and is enabled from the `MC75_DMC_UART_Setup` function.

```
    /* ========================================================================= */
    /* File Name   : Spmc75_irq1.c
*/
    /* Project     : AN_SPMC75_0009
*/
    /* Description : IRQ1/FIQ definition of C programming
*/
    /* Processor   : SPMC75F series
*/
    /* Author      : Chih ming Huang
*/
    /* Date        : October 2004                                          */
    /* Tools      : u'nSP IDE tools v1.16.1
*/
    /* Version     : 1.00
*/
    /* Security    : Confidential Proprietary                              */
    /* E-Mail      : MaxHuang@sunplus.com.tw
*/
    /* ========================================================================= */
    #include "Spmc75_regs.h"
    #include "Spmc75_dcspeed_ext.h"
    #include "Spmc75_control_ext.h"
```

```
/***************************************************************************/
/* PDC0_PDCINT_ISR() : PDC0 position detection change interrupt          */
/***************************************************************************/
void PDC0_PDCINT_ISR(void)
{
 MC75F_TPM_PDCFilter(PDC0_PDC_ID);
} /* PDC0_PDCINT_ISR */
```

Listing 5-3 Spmc75_irq1.c

The Spmc75_irq1.c does the duration calculation of three-phase hall signals and this information is used for speed control of BLDC motor.

```
/* ======================================================================== */
/* File Name   : Spmc75_irq3.c
*/
/* Project     : AN_SPMC75_0009
*/
/* Description : IRQ3/FIQ definition of C programming
*/
/* Processor   : SPMC75F series
*/
/* Author      : Chih ming Huang
*/
/* Date        : October 2004                                               */
/* Tools     : u'nSP IDE tools v1.16.1
*/
/* Version     : 1.00
*/
/* Security    : Confidential Proprietary
*/
/* E-Mail      : MaxHuang@sunplus.com.tw
*/
/* ======================================================================== */
#include "Spmc75_regs.h"
#include "Spmc75_control_ext.h"

unsigned int  ph_slice = 0;
```

```
/******************************************************************************/
/* MCP3_TPRINT_ISR() : MCP3 period interrupt
*/
/******************************************************************************/
void MCP3_TPRINT_ISR(void)
{
 MC75F_MCP3_OptoMotorDrv();                      /* sensorless MCP3 drive */

     if(ph_slice == 0)
     MC75F_TPM_Monitor();                            /* motor drive monitor */
 else if(ph_slice == 1)
     MC75F_TPM_Update();                        /* command filter function */

   ph_slice = ++ph_slice & 0x1;            /* phase slice counter from 0 to 1 */

} /* MCP3_TPRINT_ISR() */
```

Listing 5-4 Spmc75_irq3.c

In actual motor drive condition, the PWM waveform according to rotor position signal is generated by `MC75F_MCP3_OptoMotorDrv` function. The active or inactive the output command to IPM device is monitored from `MC75F_TPM_Monitor` and `MC75F_TPM_Update` functions. When the motor run command is active from calling the `MC75F_TPM_MotorStart` function, the duty ratio of PWM waveform is determined by `MC75F_MCP3_OptoMotorDrv` function.

```
/* ==========================================================================
*/
/* File Name   : Spmc75_irq4.c
*/
/* Project     : AN_SPMC75_0009
*/
/* Description : IRQ4/FIQ definition of C programming
*/
/* Processor   : SPMC75F series
*/
/* Author      : Chih ming Huang
*/
/* Date        : October 2004
```

```
*/
    /* Tools      : u'nSP IDE tools v1.16.1
*/
    /* Version     : 1.00
*/
    /* Security    : Confidential Proprietary
*/
    /* E-Mail      : MaxHuang@sunplus.com.tw
*/
    /* =========================================================================
*/
    #include "Spmc75_regs.h"
    #include "Spmc75_dcspeed_ext.h"
    #include "Spmc75_control_ext.h"

    Unsigned int iii = 0;
    /***************************************************************************
***/
    /* TPM2_TPRINT_ISR() : TPM2 period interrupt
      */
    /***************************************************************************
***/
    void TPM2_TPRINT_ISR(void)
    {
     if(iii == 0)
         MC75F_TPM_SpdCalcCtrl(BLDC1_ID);
        iii = ++iii & 0x01;
    } /* TPM2_TPRINT_ISR() */
```

Listing 5-5 Spmc75_irq4.c

The speed control algorithm is done through `MC75F_TPM_SpdCalcCtrl` function. The sampling period of TPM2 timer in this listing is 1.0 KHz. The function `MC75F_TPM_GetSpeed` will return the actual speed of BLDC motor with rotating direction.

5 Reference

[1] SUNPLUS, "SPMC75F2413A Programming Guide V1.40".

[2] SUNPLUS, "SPMC75F2413A 16-bit Microcontroller with 2-channel Motor Driver Data Sheet".

[3] Hitachi, "Technology of Small Motor".

[4] Satoshi Ogasawara and Hirofumi Akagi, "An Approach to Position Sensorless Drive for Brushless dc Motors", IEEE Trans., Indus., Appli., vol. 27, no. 5, pp.928-933, Sep./Oct. 1991.

[5] SUNPLUS, "Real Time Development of Motor Control Application Using the Visualization DMC Toolkit".

[6] SUNPLUS, "DMC Toolkit Library APIs Usage"

[7] SUNPLUS, "BLDC Motor Drive API".