Freescale Semiconductor
Application Note

AN3729
Rev. 0, 06/2008

Using FlexTimer in ACIM/PMSM Motor Control Applications

by: Pavel Grasblum, Ph.D.
System Applications
Freescale Czech System Laboratories
Rožnov pod Radhoštem, Czech Republic

This application note deals with the correct usage of FlexTimer, a new Freescale timer module, in ACIM/PMSM motor control applications. Motor control applications place specific demands on timer modules with regard to signal consistency, the safety of complementary signal generation, and the handling of application fault conditions.

Freescale offers a wide range of microcontrollers with PWM modules dedicated to motor control applications. These PWM modules reflect the specific requirements of a motor control application, to ensure safe PWM signal generation with minimal MCU intervention.

Freescale announces a new timer module, the FlexTimer (FTM), an evolution of the TPM module. The FTM targets 8-bit and 32-bit products (primarily high-end S08 and ColdFire V1 families). From a motor control point of view, it brings new features such as complementary signal generation, dead time generation, mask, polarity and fault control, etc., which are features typical for dedicated PWM modules. On the other hand, if not used

Contents

freescale semiconductor

for motor control applications, the FTM retains standard timer features such as output compare or input capture functions. This makes the FlexTimer more flexible, covering a wider range of applications.

This application note reflects FlexTimer version 1.0, implemented on the MC51AC256. The FlexTimer module is currently available on Freescale's MCF51AC256/128 devices, and will be offered on selected others in the future.

1 FlexTimer Overview

The FTM is a timer with up to eight channels. It supports input capture, output compare, and the generation of PWM signals to control electric motor and power management applications. The FTM time reference is a 16-bit counter that can be used as an unsigned or signed counter. The FTM is backwards-compatible with the TPM for simple configuration and operation, and a high degree of reconfigurability at the design phase.

1.1 FlexTimer Features

FTM features:

- Up to eight channels
- Selectable FTM source clock
  - Source clock can be the system clock, the fixed system clock, or a clock from an external pin
  - The fixed system clock source is synchronized to the system clock by an external synchronization circuit to the FTM
  - An external clock pin may be shared with any FTM channel pin or a separated input pin
  - An external clock source is synchronized to the system clock by the FTM
- Prescaler can divide by the values 1, 2, 4, 8, 16, 32, 64, or 128
- FTM has a 16-bit counter
  - Can be a free-running counter or a counter with initial and final values
  - Counting can be up or up/down
- Each channel can be configured for input capture, output compare, or edge-aligned PWM mode
- In input capture mode:
  - Capture can occur on rising edges, falling edges, or both edges
  - An input filter can be selected for some channels
- In output compare mode, the output signal can be set, cleared, or toggled on match
- All channels can be configured for center-aligned PWM mode
- Each pair of channels can be combined to generate a PWM signal (with independent control of both edges of the PWM signal)
- FTM channels can operate as pairs with equal outputs, as pairs with complementary outputs, or as independent channels (with independent outputs)
- Deadtime insertion is available for each complementary pair
- Generation of triggers to ADC (hardware trigger)

- Software control of PWM outputs
- A fault input for global fault control
- Polarity of each channel is configurable
- Loading of FTM registers that have a write buffer can be synchronized
- Write protection for critical registers
- Generation of one interrupt per channel
- Generation of one interrupt at the end of counting
- Backwards compatibility with TPM

2 New FTM Features Description

This section will describe new FTM features useful for ACIM/PMSM motor control applications. To enable all new features, the FTMEN bit has to be set to 1 in the FTMxMODE register.

2.1 Polarity Control

The polarity control allows setting the active level for each FTM channel. This polarity can be set in the FTMxPOL register. Each POLn bit corresponds to an FTMxCHn channel in this way:

- If POLn is set to 0, the polarity is high and the logical level 1 is active (the signal is not inverted).
- If POLn is set to 1, the polarity is low and the logical level 0 is active (the signal is inverted).

This feature is used during initialization to adjust the polarity of the PWM signal to an input of the IGBT or MOSFET driver. An example can be seen in Figure 1. In this example the bottom channel, FTM1CH1, is set to high polarity (POL1=0) and the top channel, FTM1CH0, is set to low polarity (POL0=1).

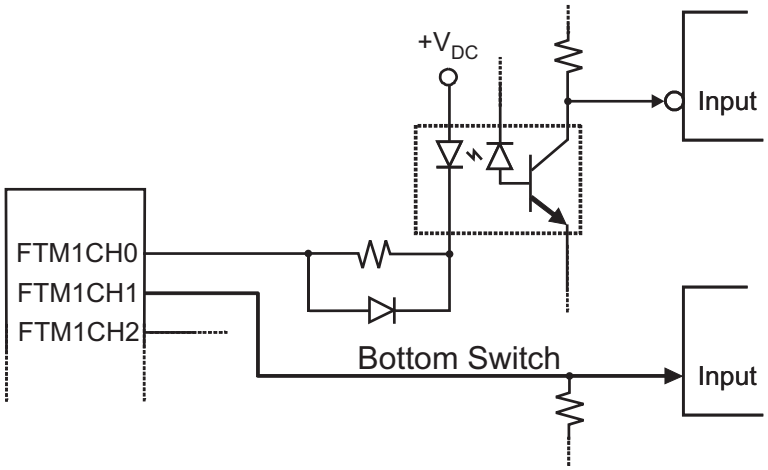

Figure 1. Polarity Control Example

NOTE

After an MCU reset, the FTM is disabled and the pins of the FTM channels are set to GPIO input mode. Therefore the proper external pullup or pulldown resistor has to be used to ensure the correct voltage level on the input of the IGBT or MOSFET driver.

2.2 Complementary Signals and Dead Time Generation

The FTM supports complementary PWM generation. If complementary generation is enabled by the COMP bit in the FTMxCOMBINEm register, the output signal is generated by an even FTM channel only. The odd output signal is generated by complementary logic as a complement to the even FTM channel. The complementary signal generation can be set individually for each pair of FlexTimer outputs.

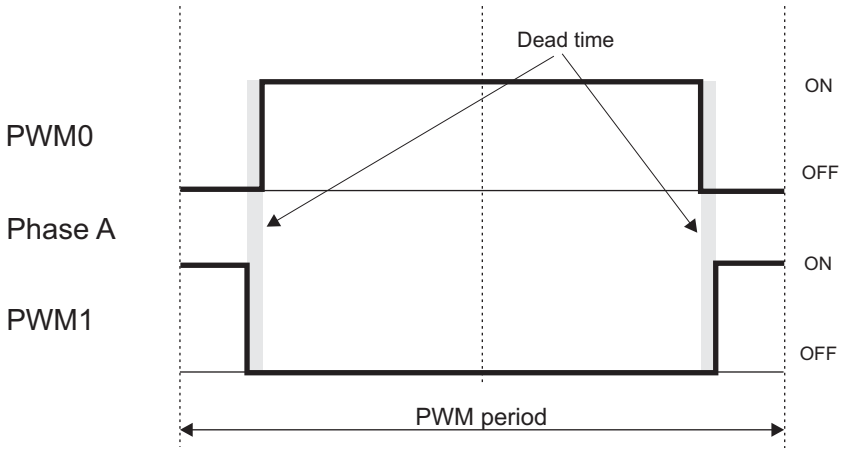

Figure 2. Complemetary Signals with Dead Time of a Single Phase

To avoid a short circuit, dead time has to be inserted into complementary signals. The dead time insertion is ensured by dead time logic, following complementary logic. This feature can be enabled by the DTEN bit in the FTMxCOMBINEm register. The dead time logic delays every rising edge by a time set in the FTMxDEADTIME register.

The deadtime consists of two parts. The first two most significant bits DTPS[1:0] define the prescaler of the system clock. Next, bits DTVAL[5:0] define the duty cycle value using the prescaled clock.

Example 1. Example of Dead Time Calculation

```
Input clock = 50.33 MHZ
Dead time = 1µs

Since 1/50.33MHz * 64 = 1.27 µs, DTPS[1:0]=0:x
Then DTVAL[5:0] = 1µs*50.33MHz = 50 = 0x32
```

2.3 Fault Control

The fault control logic ensures the shut down of FlexTimer outputs in case of any fault condition. The FlexTimer is equipped with a single fault pin. If the signal is active on this pin (set to logic 1), the

FlexTimer outputs are set to an inactive level. The inactive level is defined by polarity control (opposite level to the active level set by polarity control).

2.3.1 Enabling Fault Control

To enable fault control, the following setting must be made:

- Set fault mode in the FTMxMODE register (FAULTM[1:0] bits). For FAULTM[1:0] bit settings, see Table 1.
- Enable fault control for each pair of FTM outputs by setting the FAULTEN bit in FTMxCOMBINEm.

Table 1. Fault Control Mode Selection

FAULTM	Fault Control Mode
00	Fault control is disabled.
01	Fault control is enabled for even channels only (channels 0, 2, 4, and 6) and manual fault clearing is selected.
10	Fault control is enabled for all channels and manual fault clearing is selected.
11	Fault control is enabled for all channels and automatic fault clearing is selected.

2.3.2 Fault Control Modes

The fault control can operate in two modes:

- Automatic Fault Clearing
- Manual Fault Clearing

2.3.2.1 Automatic Fault Clearing

If automatic fault clearing is selected (FAULTM[1:0] = 11), then the disabled channel outputs are re-enabled at the beginning of each PWM pulse, and the fault input signal returns to zero. This setting is usually used for implementing cycle-by-cycle current limitation.

2.3.2.2 Manual Fault Clearing

If manual fault clearing is selected (FAULTM[1:0] = 01 or 10), then disabled channel outputs are enabled when the FAULTF bit is cleared and a new PWM cycle begins. This mode is used for fault protection. Because the fault input is edge-sensitive only, the actual state of the fault input pin must be checked by testing the FAULTIN bit in the FTMxFMS register before clearing the FAULTF bit.

2.3.3 Fault Status and Interrupt Control

The fault event can generate an interrupt if enabled by the FAULTIE bit in the FTMxMODE register. The fault event is indicated by the FAULTF bit in the FTMxFMS register. To clear the FAULTF bit:

1. While the FAULTF bit is set, read the FTMxFMS register.
2. Test the FAULTIN bit in the FTMxFMS register.

NOTE

Clearing bit FAULTF re-enables FTM outputs under all conditions, because the fault input is edge-sensitive only. Therefore, before clearing the FAULTF bit, the actual state of the fault input pin must be checked by testing the FAULTIN bit in the FTMxFMS register.

3. Write a logic 0 to the FAULTF bit.

2.3.4 Fault Input Pin Filter

The fault signal can be filtered to avoid glitches. The filter delay is set by the FFVAL[3:0] bits in the FTMxFLTFILTER register. The filter delay is equal to 4 + FFVAL[3:0] system clocks. If the fault signal is shorter then the delay defined by the FFVAL[3:0] bits, the input signal is ignored and a fault event is not triggered. If the input filter is disabled, the fault signal is always delayed by four system clocks.

2.4 Enable/Disable FTM Outputs

The FTM doesn't directly support the function to enable/disable FTM outputs. Nevertheless, there are two ways to enable/disable FTM:

- Using ELSnB:ELSnA bits
- Using output mask control

2.4.1 Using ELSnB:ELSnA bits

The ELSnB:ELSnA bits are available for each FTM channel in the FTMxCnSC registers. If ELSnB:ELSnA=00, the FTM channel output reverts to a general-purpose I/O pin. If the I/O pin is set as an input pin, the FTM outputs are tightened to an inactive level by an external resistor. If the I/O pin is set as an output pin, the FTM output can be controlled by software as a standard GPIO pin. In that case, the GPIO pin has to be set to an inactive value before the ELSnB:ELSnA bits are set to zero.

2.4.2 Using Output Mask Control

Another way to enable/disable FTM outputs is to use output mask control. The mask control sets the FTM channel output to an inactive level when the corresponding bit in the FTMxOUTMASK register is set. To use output mask control for this purpose, SYNCHOM bit in the FTMxSYNC register must be set to zero. In this case, any change in the FTMxOUTMASK register takes effect immediately.

2.5 Channel Duty Cycle Update Synchronization

To generate consistent PWM signals on multiple FTM channels, the update of duty cycles for all channels must be synchronized, especially when the duty cycle update is done asynchronously. Channel duty cycle update synchronization is available in the FTMxSYNC register. Before synchronization is used, it has to be enabled individually for each pair of FTM outputs by setting the SYNCEN bit to 1 in the FTMxCOMBINEm register.

The duty cycle update synchronization is controlled by CNTMAX and CNTMIN. These bits define where the duty cycle update takes place. If CNTMAX is set to 1, the duty cycle update is done when the FTM counter reaches its maximum value. If CNTMIN is set to 1, the duty cycle update is done when the FTM counter reaches its minimum value.

The four most significant bits (SWSYNC, TRIG2:0) trigger the duty cycle update. The SWSYNC bit allows triggering a duty cycle update by software, and the next three bits (TRIG2:0) allow triggering a duty cycle update by an external event. The source signal for the TRIG2:0 bits can be found in the datasheet for each microcontroller.

When the SWSYNC bit is set to 1, the duty cycle update occurs on the next event, corresponding to the CNTMAX and CNTMIN settings, after the SWSYNC bit is set. So if CNTMAX is set to 1, the duty cycle update occurs when the FTM counter reaches its maximum value after the SWSYNC bit is set. Similarly, if CNTMIN is set to 1, the duty cycle update occurs when the FTM counter reaches its minimum value after the SWSYNC bit was set.

The SWSYNC bit is cleared automatically after the duty cycle update. The SWSYNC bit corresponds to the LDOK bit on other PWM modules used by Freescale's microcontrollers.

When the TRIG2:0 bits are set to 1, the duty cycle update occurs on the next event, corresponding to the CNTMAX and CNTMIN settings, after the trigger event occurs. The TRIG2:0 bits remain set until cleared by the user. However, this feature is not needed in ACIM/PMSM motor control applications. The software duty cycle update is typically used in these applications.

NOTE

If both CNTMIN and CNTMAX bits are set to 1, the duty cycle update occurs when the FTM reaches its maximum value only.

2.6 Combine PWM Mode

The FTM offers a completely new way to generate PWM signals — a combine mode. The combine mode merges two FTM channels to generate a single PWM signal. This mode allows individual control of both edges of a PWM signal. The even channel controls the first edge, and the odd channel controls the second edge in the PWM period. The combine mode is enabled by setting the COMBINE bit to 1 in the FTMxCOMBINE register. The period of the PWM signal is defined by FTMxMODH:L – FTMxCNTINH:L. The FTMxCNTINH:L value can even be negative, as shown in Section 2.6.2, "Generation of Center PWM in Combine Mode." To ensure correct PWM generation, the value of FTMxMODH:L must be greater than the value of FTMxCNTINH:L.

The PWM duty cycle is defined by the FTMxC(n)VH:L and FTMxC(n+1)VH:L registers in this way:

When ELSnB:ELSnA=1:0 (Figure 3):

- FTM channel output is cleared to 0 at the beginning of a PWM period, except if FTMxC(n)VH:L = FTMxCNTINH:L.
- When the FTM counter reaches the FTMxC(n)VH:L value (even channel), the FTM channel output is set to 1.
- When the FTM counter reaches the FTMxC(n+1)VH:L value (odd channel), the FTM channel output is cleared to 0.
- When FTMxC(n)VH:L = FTMxC(n+1)VH:L, the FTM channel output is cleared to 0.
- When FTMxC(n)VH:L > FTMxC(n+1)VH:L, the FTM channel output is cleared to 0.
- When FTMxC(n)VH:L > FTMxMODH:L, the compare event is missed.
- When FTMxC(n+1)VH:L < FTMxCNTINH:L, the compare event is missed.

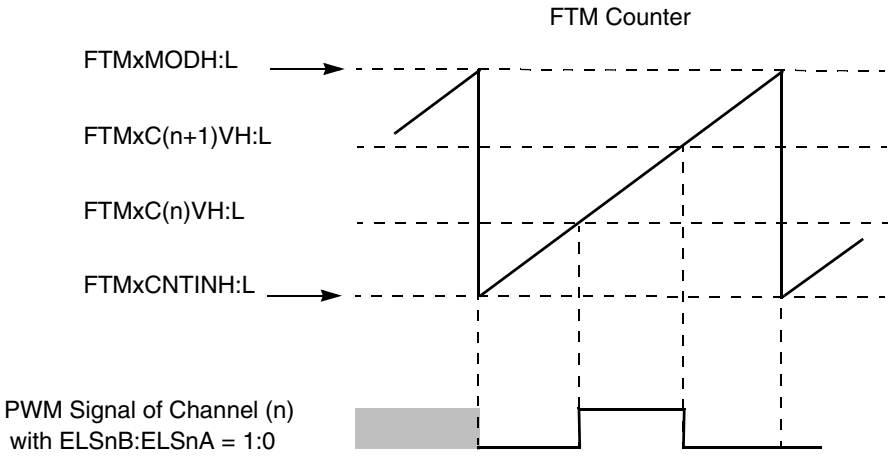

Figure 3. Combine Mode with ELSnB:ELSnA=1:0

To generate a 0% duty cycle, FTMxC(n)VH:L must be equal to FTMxC(n+1)VH:L. To generate a 100% duty cycle, FTMxC(n)VH:L must be equal to FTMxCNTINH:L and FTMxC(n+1)VH:L must be greater than FTMxMODH:L.

When ELSnB:ELSnA=x:1 (Figure 4):

- FTM channel output is set to 1 at the beginning of a PWM period, except if FTMxC(n)VH:L = FTMxCNTINH:L.
- When the FTM counter reaches the FTMxC(n)VH:L value (even channel), the FTM channel output is cleared to 0.
- When the FTM counter reaches the FTMxC(n+1)VH:L value (odd channel), the FTM channel output is cleared to 1.
- When FTMxC(n)VH:L = FTMxC(n+1)VH:L, the FTM channel output is set to 1.
- When FTMxC(n)VH:L > FTMxC(n+1)VH:L, the FTM channel output is set to 1.
- When FTMxC(n)VH:L > FTMxMODH:L, the compare event is missed.
- When FTMxC(n+1)VH:L < FTMxCNTINH:L, the compare event is missed.

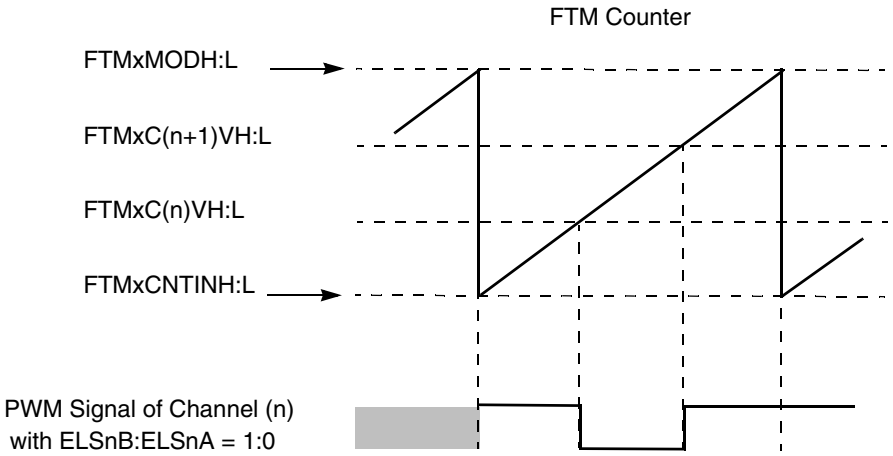

Figure 4. Combine Mode with ELSnB:ELSnA=x:1

To generate a 0% duty cycle, FTMxC(n)VH:L must be equal to FTMxCNTINH:L and FTMxC(n+1)VH:L must be greater than FTMxMODH:L. To generate a 100% duty cycle, FTMxC(n)VH:L must be equal to FTMxC(n+1)VH:L.

2.6.1 Generation of Edge-Aligned PWM in Combine Mode

An edge-aligned PWM has the rising edge aligned with the beginning of a period, and the falling edge changed according to the required duty cycle (Figure 5). To generate edge-aligned PWM, the FTM should be set in this way (for combine mode):

- TMxCNTINH:L = 0
- FTMxMODH:L = MODULO – 1, where

$$MODULO = \frac{FTM_{CLK}}{PWM_{FREQ}} \qquad \textit{Eqn. 1}$$

and where

FTM_{CLK} — FTM input clock frequency [Hz]

PWM_{FREQ} — required PWM frequency [Hz]

- FTMxC(n)VH:L = 0
- FTMxC(n+1)VH:L = <0; MODULO> according to the required duty cycle

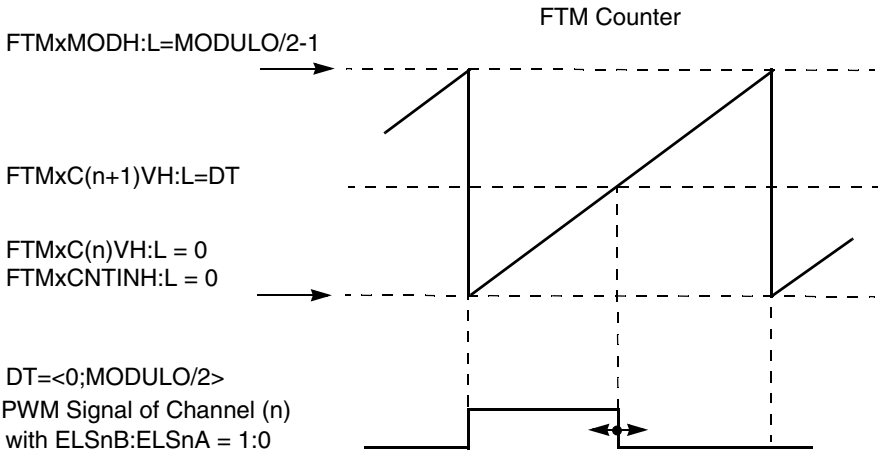

Figure 5. Edge-Aligned PWM in Combine Mode

2.6.2 Generation of Center PWM in Combine Mode

Center-aligned PWM has a pulse centered on the middle of a PWM period (Figure 6). Both rising and falling edges are updated when the duty cycle is changed. To generate center-aligned PWM, the FTM should be set as follows (for combine mode):

- TMxCNTINH:L = – MODULO/2
- FTMxMODH:L = MODULO/2 – 1, where

$$MODULO = \frac{FTM_{CLK}}{PWM_{FREQ}} \qquad \textit{Eqn. 2}$$

and where

FTM_{CLK} – FTM input clock frequency [Hz]

PWM_{FREQ} – required PWM frequency [Hz]

- FTMxC(n+1)VH:L = <0; MODULO/2> according to required duty cycle
- FTMxC(n)VH:L = – FTMxC(n+1)VH:L

NOTE

As can be seen, the center-aligned mode uses signed counting for the FTM counter. This allows simplifying the calculation of FTMxC(n)VH:L where it is a complement of the FTMxC(n+1)VH:L value.

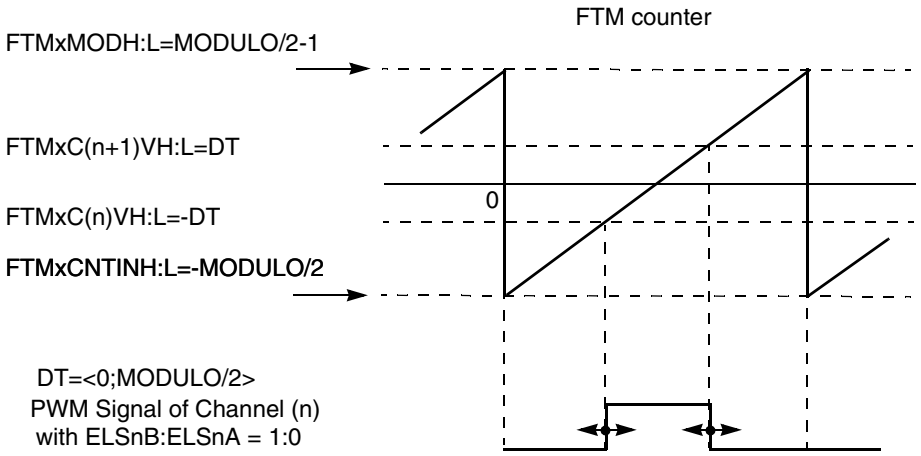

Figure 6. Center Aligned PWM in Combine Mode

The center-aligned PWM is typically used in ACIM/PMSM applications. For sine wave modulated signals, each transistor turns ON/OFF at different times, and the noise generated by power device switching is reduced.

3 Example of FTM Use in ACIM/PMSM Applications

This chapter describes a typical use of the FlexTimer in ACIM/PMSM applications. The typical setup is shown in Figure 7. All six FTM channels are configured to generate a PWM signal for the IGBT/MOSFET transistors. The fault pin is usually used to shut down the PWM outputs in case of an over-current.

The example code generates three-phase complementary center-aligned PWM signals with a frequency of 16 kHz and a dead-time of 1 µs (for a system clock of 40 MHz). The initialization routine sets all phases to 50% of the duty cycle. The fault input pin is enabled, so in the case of any fault condition, the PWM outputs are disabled. The example does not consider any interrupt events. The interrupt event can be generated on an FTM counter overflow, any compare event, and a fault event. If any interrupt is required, the appropriate bit must be set in the initialization routine.

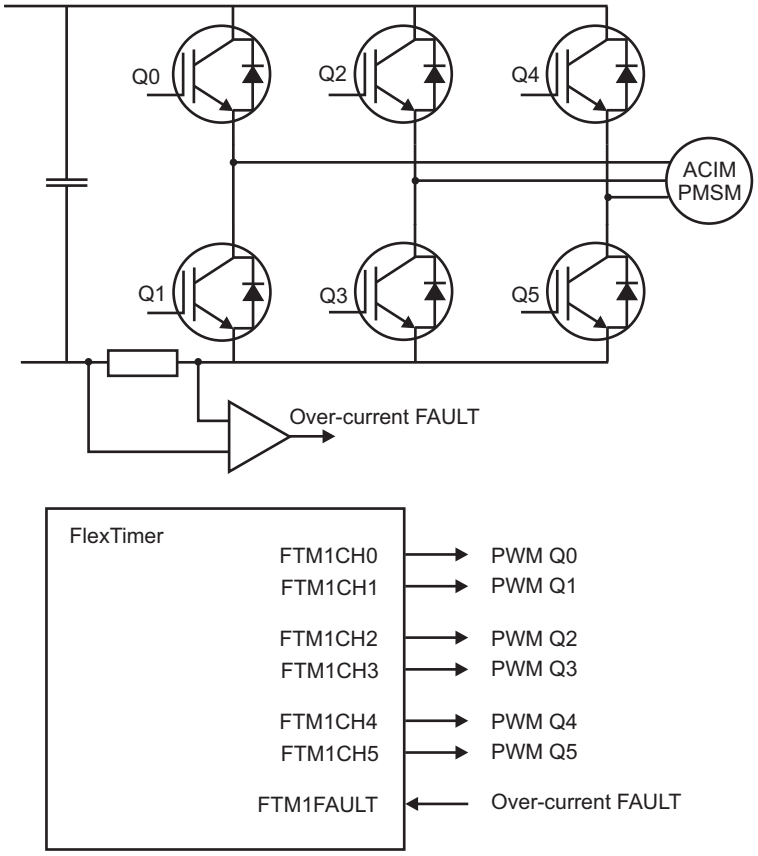

Figure 7. FlexTimer Configuration for ACIM/PMSM Applications

3.1 FTM Initialization

To generate PWM properly without any glitches on the PWM outputs, the following initialization procedure must be used:

1. Set the polarity of the PWM outputs.
2. Disable PWM outputs using the OUTPUT MASK feature.
3. Set FTM according to requirements.
4. Enable PWM outputs using the OUTPUT MASK feature.

An example of FTM initialization to generate 6-channel center-aligned PWM in combine mode is shown below.

Example 2. FTM Setting for Center-Aligned PWM in Combine Mode

```
/* Output Polarity Setting
If positive polarity is required for all channels (RESET state) , the
following line can be removed from the code */
FTM1POL = 0;

/* Disable all channels outputs using the OUTPUT MASK feature.
However, the output pins are still driven as GPIO since the
channel mode is set to FTM channel disabled after RESET */
```

```
FTM1OUTMASK = FTM1OUTMASK_CH5OM_MASK | FTM1OUTMASK_CH4OM_MASK | FTM1OUTMASK_CH3OM_MASK |
              FTM1OUTMASK_CH2OM_MASK | FTM1OUTMASK_CH1OM_MASK | FTM1OUTMASK_CH0OM_MASK;

/* Set system clock as source for FTM1 (CLKS[1:0] = 01) */
FTM1SC = FTM1SC_CLKSA_MASK;
/* Set PWM frequency; MODULO = Fclk/Fpwm
for center aligned PWM using combine mode
MODULO = 40 MHz/16kHz=2500 */
FTM1MOD = MODULO/2-1;
FTM1CNTIN = -MODULO/2;

/* COMBINE = 1 - combine mode set
   COMP = 1    - complementary PWM set
   DTEN = 1    - deadtime enabled
   SYNCEN = 1  - PWM update synchronization enabled
   FAULTEN = 1 - fault control enabled */
FTM1COMBINE0 = FTM1COMBINE0_FAULTEN_MASK | FTM1COMBINE0_SYNCEN_MASK | FTM1COMBINE0_DTEN_MASK
               | FTM1COMBINE0_COMP_MASK | FTM1COMBINE0_COMBINE_MASK;
FTM1COMBINE1 = FTM1COMBINE1_FAULTEN_MASK | FTM1COMBINE1_SYNCEN_MASK | FTM1COMBINE1_DTEN_MASK
               | FTM1COMBINE1_COMP_MASK | FTM1COMBINE1_COMBINE_MASK;
FTM1COMBINE2 = FTM1COMBINE2_FAULTEN_MASK | FTM1COMBINE2_SYNCEN_MASK | FTM1COMBINE2_DTEN_MASK
               | FTM1COMBINE2_COMP_MASK | FTM1COMBINE2_COMBINE_MASK;

/* FAULTM[1:0] = 10 - manual fault clearing
FTMEN = 1   - FTM particular features enabled */
FTM1MODE = FTM1MODE_FAULTM1_MASK | FTM1MODE_FTMEN_MASK;

/* CTNMAX = 1  - PWM update at counter in max. value */
FTM1SYNC = FTM1SYNC_CNTMAX_MASK;

/* Dead  time = 1 us for 40 MHz core clock 1us/25ns */
FTM1DEADTIME = 40;

/* Initial setting of value registers to 50 % of duty cycle */
FTM1C0V = -MODULO/4;
FTM1C1V = MODULO/4;
FTM1C2V = -MODULO/4;
FTM1C3V = MODULO/4;
FTM1C4V = -MODULO/4;
FTM1C5V = MODULO/4;

/* SWSYNC = 1 - set PWM value update. This bit is cleared automatically */
FTM1SYNC |= FTM1SYNC_SWSYNC_MASK;

/* ELSnB:ELSnA = 1:0 Set channel mode to generate positive PWM
Note:
1. From this moment the output pins are under FTM control. Since the PWM output is disabled by
the FTM1OUTMASK register, there is no change on PWM outputs. Before the channel mode is set,
the correct output pin polarity has to be defined.
2. Even if the odd channels are generated automatically by complementary logic, these channels
have to set to be in the same channel mode. */
FTM1C0SC = FTM1C0SC_ELS0B_MASK;
FTM1C1SC = FTM1C1SC_ELS1B_MASK;
FTM1C2SC = FTM1C2SC_ELS2B_MASK;
FTM1C3SC = FTM1C3SC_ELS3B_MASK;
FTM1C4SC = FTM1C4SC_ELS4B_MASK;
FTM1C5SC = FTM1C5SC_ELS5B_MASK;
```

3.2 Sine Wave Generation

After initialization, all channels are disabled and the duty cycle is set to 50%. To enable/disable PWM according to application needs, the OUTPUT MASK feature can be used:

```
/* Enable PWM outputs */
FTM1OUTMASK = 0;
```

or

```
/* Disable PWM outputs (0-5)*/
FTM1OUTMASK = 0x3F;
```

The sine wave drivers require the generation of three sine wave modulated signals, shifted by 120° or using space vector modulation. The PWM duty cycle update is usually done every PWM cycle in the FTM overflow interrupt. The regular update of the duty cycle channels can be done this way:

```
duty_cycle_phaseA = MODULO/4 + MODULO/4*sin(angle);
duty_cycle_phaseB = MODULO/4 + MODULO/4*sin(angle + OFFSET120);
duty_cycle_phaseC = MODULO/4 + MODULO/4*sin(angle + OFFSET240);

FTM1C0V = -duty_cycle_phaseA;
FTM1C1V = duty_cycle_phaseA;
FTM1C2V = -duty_cycle_phaseB;
FTM1C3V = duty_cycle_phaseB;
FTM1C4V = -duty_cycle_phaseC;
FTM1C5V = duty_cycle_phaseC;
FTM1SYNC |= FTM1SYNC_SWSYNC_MASK;
```

4 Conclusion

This application note describes typical uses of the FlexTimer in ACIM/PMSM applications. The FlexTimer simplifies calculation of PWM signals (automatic complementary signal generation and dead time insertion) and significantly increases the safety of PWM generation and of the whole application (automatic complementary signal generation, dead time insertion and fault control). If the FlexTimer is not used for motor control purposes, the standard timer features such as capture and compare are still available.

THIS PAGE IS INTENTIONALLY BLANK

How to Reach Us:

Home Page:
www.freescale.com

Web Support:
http://www.freescale.com/support

USA/Europe or Locations Not Listed:
Freescale Semiconductor, Inc.
Technical Information Center, EL516
2100 East Elliot Road
Tempe, Arizona 85284
1-800-521-6274 or +1-480-768-2130
www.freescale.com/support

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

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

Asia/Pacific:
Freescale Semiconductor China Ltd.
Exchange Building 23F
No. 118 Jianguo Road
Chaoyang District
Beijing 100022
China
+86 10 5879 8000
support.asia@freescale.com

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

Document Number: AN3729
Rev. 0
06/2008

freescale™
semiconductor