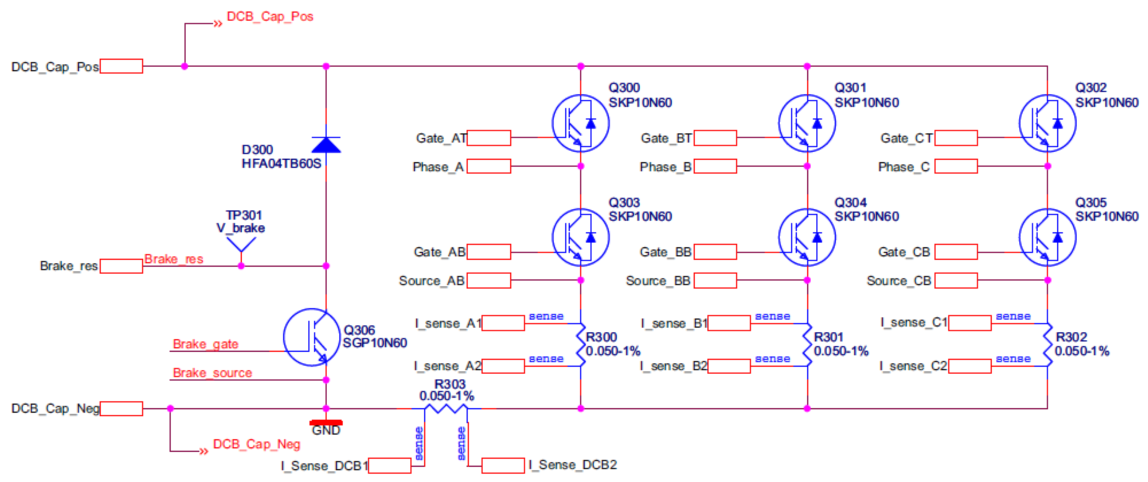
PWM signal trigger ADC for KEA family

For motor application, it is required to sample 3 phase current so that the FOC algorithms can be used for ACIM and PMSM. Generally, from hardware perspective, 3 shunt resistors are connected at the three branches as the following power stage schematics. The R300,R301 and R302 are the three shunt resistors, only when the bottom IGBT turns on can the current flows over the shunt resistor, it makes sense for the ADC samples the voltage over the shunt resistor.



For the high level Kinetis family such as K1x,K2x….K6x, there is PDB module, the PDB module can be used to trigger ADC after a programmable delay from triggering signal event, but the KEA family does not have PDB module, so the KEA family provides a mechanism so that the FTM signal can trigger ADC with a programmable delay.

Firstly, the FTM module can generate the triggering signal, the FTM_EXTTRIG register sets up the triggering source, setting bit 6 INITTRIGEN bit in FTM_EXTTRIG register Enables the generation of the trigger when the FTM counter is equal to the CNTIN register. Setting the CHnTRIG bit Enables the generation of the channel trigger when the FTM counter is equal to the CnV register.

Secondly, the SIM_SOPT0 register specifies the ADC triggering source and the programmable delay tick number. The bits24~31 DELAY bits set up the delay time, Specifies the delay from FTM2 initial or match trigger to ADC hardware trigger when 1 is written to ADHWT. The 8-bit modulo value allows the delay from 0 to 255 upon the BUSREF clock settings. This is a one-shot counter that starts ticking when the trigger arrives and stops ticking when the counter value reaches the modulo value that is defined. The bits 20~22 specify the ADC triggering sources, if you select 3b’010, you select that FTM2 init triggers with 8-bit programmable counter delay. The bits 16~18 BUSREF bits specify the BUS REFERENCE clock frequency, which is a bus clock divider.

The example is developed under TRK-KEA128 board, the tools is Kinetis Design Studio version 3.2.0
Parameter configuration:

The FTM2 module is used to output PWM signals, because only FTM2 of KEA128 can output 6 channels PWM signals to control a motor. For the example, FTM2_CH0 pin(PTC0 pin) is the PWM0 signal, which is connected to bottom IGBT, FTM2_CH1 pin(PTC1 pin) is the PWM1 signal, which is connected to the top IGBT for the same IGBT pair. The PWM0/PWM1 signals are complementary PWM signals in center-alignment mode and the PWM0 is set up in the High-true pulses mode(clear Output on match-up) by setting the bits **ELSnB:ELSnA** as 2b'10 or set up the FTM2_C0SC as 0x28. For the example, PWM0 signal is the PWM signal to control bottom IGBT, the PWM1 signal connected to top IGBT is the inverter of PWM0 signal. In the High-true pulse mode, the PWM0 signal is high from beginning(when FTM counter is the FTM_CNTIN value), when the FTM counter reaches up to FTM_C0V register value, the PWM0 signal becomes low. If you set the INITTRIGEN bit in FTM2_EXTTRIG register, the delay time is zero, because PWM0 signal starting instant is the center of PWM0 signal in center-alignment mode.

The SIM_SOPT0 register controls the ADC triggering sources and the delay time. The ADHWT bits in SIM_SOPT0 register specify the ADC hardware triggering sources, in the example, it is set as 3b'010, which means that FTM2 init trigger with 8-bit programmable counter delay. The DELAY bits specify the delay time, it's tick is BUSREF clock number, BUSREF clock is a divided clock from bus clock, the divider is defined by BUSREF bits. In the example, because the delay time is zero, so only the ADHWT is set up.

For the ADC module, the ADTRG bit is set in ADC_SC2 register so that ADC is triggered by hardware source.

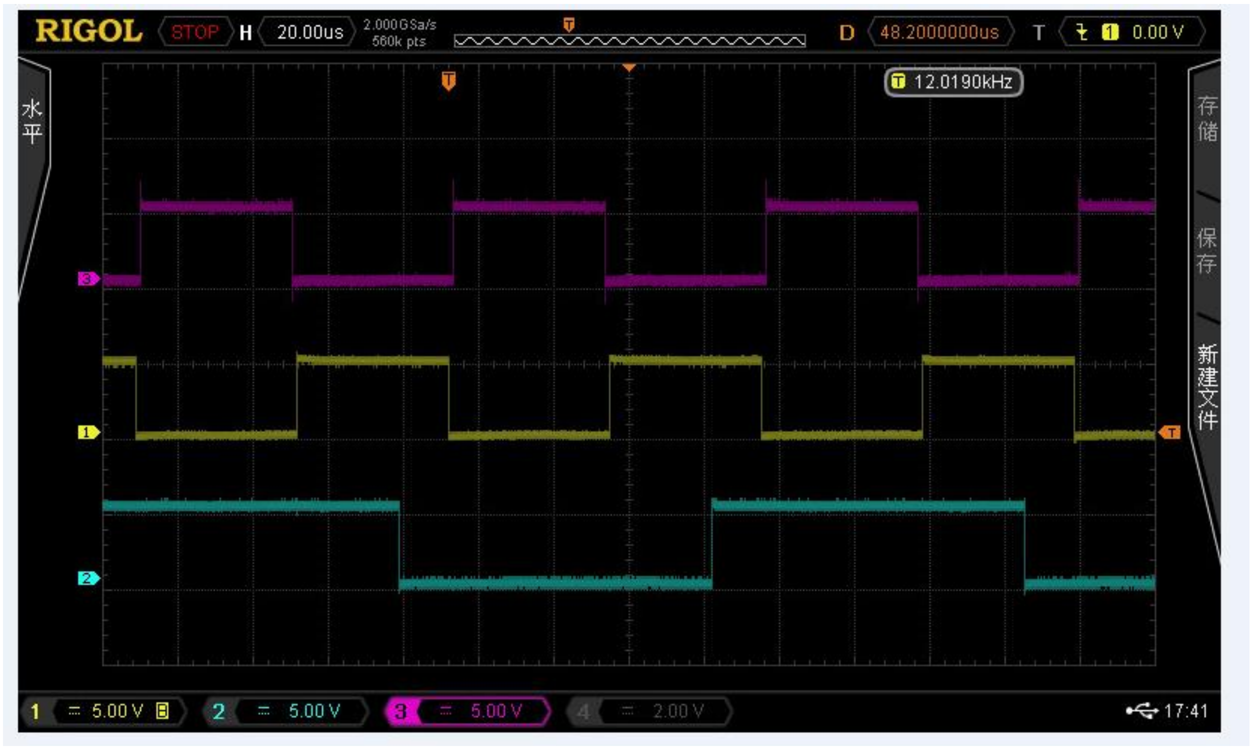
After the ADC is triggered, the ADC will sample the analog channel, after the conversion is finished, the ADC Interrupt service routine(ISR) is triggered and executed, in the ADC ISR, PTB2 is toggled so that user can observe the timing relationship.

Conclusion:

The PWM0(from FTM2_CH0 pin) and PWM1(from FTM2_CH1 pin) and PTB2 are connected to scope, the scope screenshot is copied here.

Channel1 is the PWM0 signal from FTM2_CH0(PTC0), on the TRK-KEA128 board, it is the pin1 of J9.
Channel3 is the PWM1 signal from FTM2_CH1(PTC1), on the TRK-KEA128 board, it is the pin2 of J9.
Channel2 is the PTB2 signal from GPIOB module, on the TRK-KEA128 board, it is the pin3 of J3.

From the scope, the PWM0 and PWM1 are complementary signals, because the PTB2 is toggled in the ADC ISR, we can see that the sampling instant is in the center of PWM0 high logic, there is some delay from the timing relationship because of delay incurred by the software.



APPENDIX:

```
//The example is developed under KDS3.2.0 and TRK-KEA128 board by XiangJun Rong
//The example code demonstrates how to trigger ADC at the center of high logic of
bottom PWM0 signal
//the PWM signal is generated by FTM2
//The KEA128 is designed to enable the FTM2 to generate signal to trigger ADC so that
the sampling instant is at the center of PWM0 high logic.
//Conclusion: connect the PTB2 and PTC0/PTC1 pin to scope you can see that the PTB2
toggle at the center of  high logic of PTC0 PWM signal
//SIM_SOPT0 register provide the mechanism to control the sampling instant of ADC.
#include "SKEAZ1284.h"

void PWMOutput_CenterAlignment(void);
void enableADCFTMClock(void);
void AdcInit(void);
void ADC_IRQHandler(void);
void FRM2_TriggerSignal(void);
void GPIOInit(void);
static int i = 0;
unsigned int sample[8];

void enableADCFTMClock(void)
{
	//enable ADC, FTM0/1/2 clock
	SIM_SCGC|=0x200000E0;

}
```

```
void PWMOutput_CenterAlignment(void)
{
	FTM2_CONF=0xC0; //set up BDM in 11
	FTM2_FMS=0x00; //clear the WPEN so that WPDIS is set in FTM0_MODE reg
	FTM2_MODE|=0x05; //enable write the FTM CnV register
	FTM2_MOD=1000;
	FTM2_C0SC=0x28; ////center-alignment, PWM begins with High
	FTM2_C1SC=0x28; //PWM waveform is high-low-high
	FTM2_COMBINE=0x02; //complementary mode for CH0&CH1 of FTM0
	FTM2_COMBINE|=0x10; // dead timer insertion enabled in complementary mode for
//CH0&CH1 of FTM0
	FTM2_DEADTIME=0x1F; //dead time is 16 system clock cycles
	FTM2_C1V=500;
	FTM2_C0V=500;
	FTM2_CNTIN=0x00;
	FTM2_C2SC=0x28;
	FTM2_C3SC=0x28;
	FTM2_COMBINE|=0x0200;
	FTM2_COMBINE|=0x1000;
	FTM2_C3V=250;
	FTM2_C2V=250;
	FTM2_SC=0x68;
}
//Pin assignment
//PTC0		FTM2_CH0 ALT2
//PTC1		FTM2_CH1 ALT2
//PTC2		FTM2_CH2 ALT2
//PTC3		FTM2_CH3 ALT2
//PTB4		FTM2_CH4 ALT2
//PTB5		FTM2_CH5 ALT2


void FRM2_TriggerSignal(void)
{
	FTM2_EXTTRIG=0x40; //enable FTM2_CNTIN register triggering
	SIM_SOPT0|=0x00200000; //set the ADHWT as 3b'010
	__asm("nop");
}
//ADC interrupt index is 15
void AdcInit(void)
{

	ADC_SC2=0x40; //hardware triggered,
	ADC_SC3=0x03; //not continuous conversion
	ADC_SC1=0x54; //interrupt enable and select ADC1_DM1 channel
//set that FTM0 to tri>gger ADC1
	NVIC->IP[3]&=0x3FFFFFFF; //setting interrupt of ADC1
	NVIC->ICPR[0]|=1<<15; //clear the pending register of interrupt source 58
	NVIC->ISER[0]|=1<<15; //set the interrupt source of ADC1
	__asm("cpsie i"); //interrupt enable
}


void ADC_IRQHandler(void)
```

```
{
	GPIOA_PTOR|=0x0400; //toggle indicator PTB2 pin
	sample[0]=ADC_R;
	__asm("nop");
//ADC1_SC1A=0x14; //interrupt enable and select ADC1_DM1 channel
}
//set PTB2 as GPIO output mode
void GPIOInit(void)
{
	FGPIOA_PDDR|=0x0400;
	FGPIOA_PIDR|=0x0400;
	FGPIOA_PTOR|=0x0400; //toggle indicator
	FGPIOA_PTOR=0x0400; //toggle indicator
	FGPIOA_PTOR=0x0400; //toggle indicator
	FGPIOA_PTOR=0x0400; //toggle indicator
	FGPIOA_PTOR=0x0400; //toggle indicator
	FGPIOA_PTOR=0x0400; //toggle indicator
	FGPIOA_PTOR=0x0400; //toggle indicator
}

int main(void)
{

    /* Write your code here */
	//PTC0/1/2/3 are PWM output pins
	SIM_PINSEL1&=0xFF00;
	enableADCFTMClock();
	GPIOInit();
	PWMOutput_CenterAlignment();
	AdcInit();
	FRM2_TriggerSignal();
    /* This for loop should be replaced. By default this loop allows a single
stepping. */
    for (;;) {
        i++;
    }
    /* Never leave main */
    return 0;
}
```