Embedded Systems

Postlab 8 TPM-DAC

By Rajendra Singh (111601017), final year, CSE, IIT PALAKKAD

4 Oct 2019

Objective

To learn how to control the brightness of the LED (blue here) in freedom board using PWM signals generated from the TPM module.

Abstract

- **What is PWM?**

  Pulse Width Modulation, or PWM, is a technique for getting analog results with digital means. Digital control is used to create a square wave, a signal switched between on and off. This on-off pattern can simulate voltages in between full-on (5 Volts) and off (0 Volts) by changing the portion of the time the signal spends on versus the time that the signal spends off. The duration of "on time" is called the pulse width. To get varying analog values, you change or modulate, that pulse width.

- Period of the PWM i.e ***Period = (MOD + 1) cycles.***
- Pulse Width i.e ***Pulse width = (CnV) cycles***
- TPM counter clock = 21 MHz
- MOD value = 21000 - 1 (so as the PWM signal period =1KHz)

Steps

- **TPM Enabling** i.e enable TPM0 (using SCGC6) and select MCGFLLCLK clock.
- **TPM Configuring** i.e disable LPTPM counter, centre aligned PWM and edge-aligned PWM
- **Set MOD & C1V** i.e Set the MOD value, set LPTPM counter = 0 (in the up-counting mode when it equals MOD value) and Set PWM signal period equal to 1KHz and set the duty cycle
- **Counter enabling i.e** Prescale factor =16 (i.e clk/=16) and enable the counter
- Use PWM signal to for the blue LED by selecting ALT4 for PORTD1 register.

Kindly look the code for more information regarding the port and pin selection.

EXERCISE

1 : Control the blue led brightness based on the TPM-DAC.

=> In below is detailed code with comments for each steps for controlling the blue led brightness is presented.

CODE:

Below code can also be found at github.

```
/*

*   Author : Rajendra singh

*   Roll no: 111601017

*/




#include<MKL25Z4.h> //INCLUDING LIBRARY
```

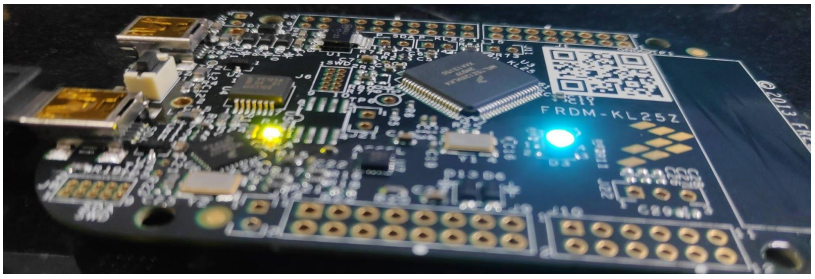
```
//============================ADC0_init============================//
void init_TPM(){
   //disable clock gating (enabling clock) of
   SIM->SCGC6 |= (1<<24);

   //setting 24th,25th = 01 for enabling
   SIM->SOPT2 |= (1<<24); //
   SIM->SOPT2 &= ~(1<<25);

   //disable tpm0 counter
   TPM0->SC &= ~(3<<3);

   //timer clock freq
   TPM0->MOD = 20999;

   //configure channel for edge align
   //use tpm0_s
   int temp = TPM0->CONTROLS[1].CnSC;
```

```
temp |= (1<<5);

temp &= ~(1<<4);

temp |= (1<<3);

temp &= ~(1<<2);

TPM0->CONTROLS[1].CnSC = temp;



//TPM0->CONTROLS[1].CnSC = (0x08 | 0x20);



//set duty cycle 75%

//TPM0->CONTROLS[1].CnV = 52500; //20999/4

//TPM0->CONTROLS[1].CnV = (3/4)*20999;

//TPM0->CONTROLS[1].CnV = (1/2)*20999;

//TPM0->CONTROLS[1].CnV = (0)*20999;

TPM0->CONTROLS[1].CnV = (1)*20999;



//disable LPTPM counter

TPM0->SC = 0b01000;



//configure pddr for blue led
```

```
   SIM->SCGC5 |=(1<<12); //TO ACTIVATE PORT B

    //PORTD->PCR[1]|=(1<<8); //SETTING 8TH BIT TO 1

    //PORTD->PCR[1]&=0xFFFFF9FF; //SETTING 9TH, 10TH BIT TO 0, OTHER UNCHANGED



   //[10:8] =100

   PORTD->PCR[1]|=(1<<10);

   PORTD->PCR[1]&=~(3<<8);

}



void decreaseBrightness(){

  if(TPM0->CONTROLS[1].CnV == 1){

       TPM0->CONTROLS[1].CnV = 20999;

  }

  TPM0->CONTROLS[1].CnV-=5;;

}



//==============================blue D1==============================//

//=======init========//

void led_blue_init(){
```

```
    SIM->SCGC5 |=(1<<12); //TO ACTIVATE PORT B

    PORTD->PCR[1]|=(1<<8); //SETTING 8TH BIT TO 1

    PORTD->PCR[1]&=0xFFFFF9FF; //SETTING 9TH, 10TH BIT TO 0, OTHER UNCHANGED

    PTD->PDDR |= (1<<1); //18TH BIT = 1

}

//=======ON========//

void led_blue_on(){

    PTD->PCOR |= (1<<1 ); //CLEAR 18PIN VALUE

}

//=======OFF========//

void led_blue_off(){

    PTD->PDOR |= (1<<1); //CLEAR 18PIN VALUE

}

//=======TOGGLE========//

void led_blue_toggle(){

    PTD->PTOR |= (1<<1); //CLEAR 18PIN VALUE

}



//============================DELAY old============================//
```

```
void delay_old(long long int d){

    while(d--);

}




//==============================MAIN==============================//

int main(void){



   SystemCoreClockUpdate(); //updating clock from PLL



   init_TPM();

   //led_blue_init();



   //led_blue_off();

   while(1){

       decreaseBrightness();

       delay_old(1);

   }

}
```



Conclusion

- In this exercise, we learnt more about TPM-DAC.
- Learn to use TPM mode
- Hence then generated PWM signal
- Feeded it to PORTD1
- Hence, controlled brightness of the blue LED
- Overall it was nice exercise.

Reference

1) Google
2) **Cortex M0+ Generic User's Guide**
3) **Cortex M0+ Technical Reference Manual**