CG2271: Real-Time Operating Systems

Lab 4: PWM & Audio

In this lab, you will be exploring the PWM capabilities of the FRDM-KL25Z microcontroller. As learnt in the Lectures, we make use of the TPM (Timer/PWM) module to generate the PWM signals. We will first be learning how to generate the signals to control some LED's. From there we will proceed to interface it with the Motor Driver Chip to see how it controls the motors.

Part 1: PWM

In Part 1, we will be exploring the PWM module of the KL25Z microcontroller.

A. Initializing the PWM Module

The following screenshot shows the InitPWM() code to initialize the PWM module.

```
#define PTB0_Pin  0
#define PTB1_Pin  1

/* intiPWM() */
void initPWM(void)
{
    SIM_SCGC5 |= SIM_SCGC5_PORTB_MASK;

    PORTB->PCR[PTB0_Pin] &= ~PORT_PCR_MUX_MASK;
    PORTB->PCR[PTB0_Pin] |= PORT_PCR_MUX(3);

    PORTB->PCR[PTB1_Pin] &= ~PORT_PCR_MUX_MASK;
    PORTB->PCR[PTB1_Pin] |= PORT_PCR_MUX(3);

    SIM->SCGC6 |= SIM_SCGC6_TPM1_MASK;

    SIM->SOPT2 &= ~SIM_SOPT2_TPMSRC_MASK;
    SIM->SOPT2 |= SIM_SOPT2_TPMSRC(1);


    TPM1->MOD = ?? ;

    TPM1->SC &= ~((TPM_SC_CMOD_MASK) | (TPM_SC_PS_MASK));
    TPM1->SC |= (TPM_SC_CMOD(1) | TPM_SC_PS(7));
    TPM1->SC &= ~(TPM_SC_CPWMS_MASK);

    TPM1_C0SC &= ~((TPM_CnSC_ELSB_MASK) | (TPM_CnSC_ELSA_MASK) | (TPM_CnSC_MSB_MASK) | (TPM_CnSC_MSA_MASK));
    TPM1_C0SC |= (TPM_CnSC_ELSB(1) | TPM_CnSC_MSB(1));
}
```

CODE REVIEW:

Q1. What is the purpose of the following line?

```
SIM_SCGC5 |= SIM_SCGC5_PORTB_MASK;
```

CODE REVIEW *(This is for your own understanding. It is not a Demo Requirement)*:

Q2. What is the purpose of the following FOUR lines of code?

```
PORTB->PCR[PTB0_Pin] &= ~PORT_PCR_MUX_MASK;
PORTB->PCR[PTB0_Pin] |= PORT_PCR_MUX(3);

PORTB->PCR[PTB1_Pin] &= ~PORT_PCR_MUX_MASK;
PORTB->PCR[PTB1_Pin] |= PORT_PCR_MUX(3);
```

Q3. Why is a value of 3 used in the above code? Which part of the datasheet provides this information? Provide a screenshot of that section in the datasheet.

Q4. What is the purpose of the following line?

```
SIM->SCGC6 |= SIM_SCGC6_TPM1_MASK;
```

Q5. What is the purpose of the following TWO lines of code?

```
SIM->SOPT2 &= ~SIM_SOPT2_TPMSRC_MASK;
SIM->SOPT2 |= SIM_SOPT2_TPMSRC(1);
```

Q6. What is the purpose of the following THREE lines of code?

```
TPM1->SC &= ~((TPM_SC_CMOD_MASK) | (TPM_SC_PS_MASK));
TPM1->SC |= (TPM_SC_CMOD(1) | TPM_SC_PS(7));
TPM1->SC &= ~(TPM_SC_CPWMS_MASK);
```

Q7. What is the purpose of the following TWO lines of code?

```
TPM1_C0SC &= ~((TPM_CnSC_ELSB_MASK) | (TPM_CnSC_ELSA_MASK) | (TPM_CnSC_MSB_MASK) | (TPM_CnSC_MSA_MASK));
TPM1_C0SC |= (TPM_CnSC_ELSB(1) | TPM_CnSC_MSB(1));
```

Change the Core Clk to 48MHz.

B. Generate a 50Hz PWM output

Calculate the appropriate values in order to generate a 50Hz signal on the Timer 1 Channel 0.
You must update TPM1-> MOD and the TPM1_C0V registers.

CODE REVIEW *(This is for your own understanding. It is not a Demo Requirement)*:

Q8. Show how you arrived at the values for the TPM1-> MOD and the TPM1_C0V registers.

LAB DEMO: Using the oscilloscope, show the 50Hz signal from your board to the Lab TA.

Part 2: Audio

In Part 2, we will be exploring how we can make use of the PWM signal to generate musical tones in our code. There are multiple scales for the notes. Once possible scale that you can use is as shown below:

Note	Frequency
C	262
D	294
E	330
F	349
G	392
A	440
B	494

Using the same PWM output from Part A, modify your code to generate the required frequencies. You can use the buzzer to hear the sound.

LAB DEMO: Demonstrate the ability to generate different musical notes to your TA.

Conclusion

In this lab, you have learnt how to generate PWM signals and use them for Audio Generation. For the project, you will need to use PWM signals to also control the Motors.
Good Luck! ☺