EEE 212-Microprocessors Lab Assignment 5

Due Date: 14.12.2022 - 13.30

1 Introduction

For the Lab 5 assignment, you will implement **Pulse Width Modulated (PWM)** signals using the timers of the FRDM-KL25Z board to light up the external **RGB LED**. You can find an external RGB LED in your component pack (we are not using the RGB LED of FRM-KL25Z) and only one external RGB LED is needed.

RGB LED has 3 pins which corresponds to Red, Green and Blue channels and you can produce different shades of colour with controlling these pins. (In addition these pins, there is one additional pin that you should connect to Vcc). In order to generate different colors, you have to use 3 different PWM signals. In addition, you can adjust the blinking time of RGB LED with controlling the frequency of the signal.

Some pins of FRDM-KL25Z board are capable of generating PWM signals. Indeed, PWM signals are nothing but periodic square waves with two different parameters: duty cycle (%) and periodic frequency (Hz). By changing the duty cycle of the PWM signals, you can adjust the color of the RGB LED. (i.e, in order to produce different kinds of colors, we need to set the intensity of each internal LED and combine the three color outputs). By changing the frequency of signal, you adjust the blinking frequency of LED.

You should be able to change the color and blinking frequency of LED by giving different input values via keypad. Please observe how you can generate different colors via changing the duty cycle of signals. You will be asked to generate a specific color and a frequency on RGB LED.

2 Assignment Details

1. Only 1 frequency (Hz), 3 duty cycle (%) values (3 different duty cycle values corresponding to each PWM signal) are to be read from the keypad (i.e, each PWM signal has same frequency but different duty cycle value). Note that frequency interval is in between 1Hz and 100 Hz and duty cycle interval is in between %0 and %100. If you enter 1Hz, you are able to see RGB LED blinks in 1 seconds. To observe blinking, choose low frequency values.

2. Then, using timers of FRDM-KL25 board (TPM timers), generate PWM signals on three-different pins of FRDM-KL25 (with same frequency for both and three different duty cycle value respectively). For this step, you have to configure your timers properly (Please refer to guidelines for this part). You can check whether you generate correct signals (given input values) on oscilloscope screen and be sure that you get correct signals from these pins before you proceed to light up the RGB LED.

3. Connect these ports to channel pins on RGB LED and experiment with different duty cycle values on for each signal. By changing duty cycle values of PWM's, you should produce different colours and by changing the frequency, you should different blinking times on RGB LED.

4. We will ask you to adjust the specific shade of color and frequency during your demo. For example, you will be asked to generate yellow color by changing the duty cycle values of your PWM signals. You should be able to change the color and the blinking frequency of LED by giving different input values via keypad. Also note that, you should connect a proper resistor for channel pins in order not to burn your RGB LED.

3 Guidelines

Follow the guidelines given below for implementation:

(a) Please refer to **Module 4 of Lecture Notes** and additionally **Mazidi**'s book *Freescale ARM Cortex-M Embedded Programming.* Chapter 5 - Section 2 is dedicated for general configuration of TPM timers, Chapter 11- Section 2 is dedicated for generating PWM signals, Chapter 2 is dedicated to GPIO.

(b) If you are to use PWM signals generated by the board, you should choose pins which are capable of generating these signals (For example PORTE Pin 20 or PORTE Pin 31 are not used by any of the LCD and keypad connection and may be good option for generating PWM signals). Please refer to the manual in this case. This figure in manual shows which of these pins can be used as a PWM signal generating pin.

(c) Be careful about configuring your timers and ports; if you correctly configure your timers and ports, then most of the lab assignment is done. Try to thoroughly understand all steps of reference examples of PWM in your lecture notes.

(d) You can use mathematical library for this lab. For example, rounding may be good option for calculating values for timers. So include ”math.h ” at the beginning of your code. Also, be careful about data types when you make mathematical operations. Although, you can see some warnings when you make mathematical operations with two different data types (for example, operations involving two different data types: float and int), it does not make such difference; it still works.