

CG2271 Realtime Operating Systems

Lab 3 – Timers

1. Introduction

In this lab we will program the timer TPM0 on the MCXC444 to flash the RGB LEDs on the FRDM-MCXC444 board in sequence.

2. Submission

Please fill your answers in the provided answer book, and rename it to LAB3-GROUPBx_SUBGPy.docx. So if you are in sub-group 8 of the Tuesday group (B01), your answer book should be named LAB3-GROUPB1-SUBGP08.docx.

Submit only one copy of the answer book per sub—group, and only in DOCX format. **Submit to Canvas by 2359 hours on Friday 19 September 2025.**

3. Lab Assignment

You are given a file called timers.c which contains the skeleton code:

- i. Create a new project called CG2271Lab3.
- ii. Open the CG2271Lab3.c file, and delete all its contents.
- iii. Copy over the contents of timers.c to CG2271Lab3.c

Answer the questions that follow.

- a. The low frequency internal clock for MCG-Lite has a 2 MHz clock and an 8 MHz clock. For this lab we will use the **2 MHz clock**.
- b. We will clock timer TPM0 using the MCG Internal Reference Clock (MCGIRCLK), but we reduce the clock speed to **1 MHz**.

Question 1. (2 MARKS)

When reducing the internal clock from 2 MHz to 1 MHz, we should set a division factor of 2 for LIRC DIV2, and set the division factor for LIRC DIV1 at 1. Why can't we do it the other way around? (I.e. set division factor of LIRC DIV1 to 2, and LIRC DIV2 to 1). **Hint:** Look at Figure 27-1 in the Reference Manual or page 13 of Lab Lecture 4.

- c. You are given the empty code for setMCGIRCLK, fill in the body of this function to configure MCGIRCLK appropriately:
- Use the 2 MHz internal reference clock.
 - Use LIRC DIV2 to scale it down to 1 MHz.

Question 2. (4 MARKS)

Cut and paste your code for setMCGIRCLK here.

```
}
```

- d. We will now configure TPM0 to trigger an interrupt every 500ms. Choose the the smallest prescaler and modulo (TPM0->MOD) values that gives the best accuracy.

Question 3. (2 MARKS)

State your prescaler and modulo values here (**NOTE:** Choose the SMALLEST possible prescaler and the corresponding modulo value.

Prescaler:

Modulo:

We will now set up TPM0 timer. We will set the TPM0_IRQn interrupt to the **highest** priority level.

Question 4. (5 MARKS)

Cut and paste your code for initTimer below:

```
void initTimer() {
```

```
}
```

- e. Complete the code for the ISR. Fill in the correct vector name, and the test condition for the TOF. Also remember to zero TPM0->CNT and clear the TOF flag.

Question 5. (2 MARKS)

Fill in the code for the IRQ Handler

- f. Finally, complete the functions to start and stop the timer.

Question 6. (2 MARKS)

Fill in the code for startTimer and stopTimer:

```
void startTimer() {  
    // Use TPM counter clock to increment.  
}  
  
// Turn off the timer.  
void stopTimer() {  
}
```

Demo (3 MARKS)

Demo your completed code to your TA.