

Department of Electrical and Computer Engineering

ECE 5710/6710 Lab 3

Title: Cyclic Scheduling

Objective: The student should be able to implement a cyclic scheduler.

Parts: 1-EFM32GG Evaluation Board

Software: Silicon Laboratories Simplicity Studio.

Preparation: Write the title and a short description of this laboratory exercise in your lab book. Make sure the page is numbered and make a corresponding entry in the table of contents.

You are to write a cyclic scheduler to generate a feasible schedule for the following six tasks:

Task	Period	Execution Time	Relative Deadline
A	3 ms	0.5 ms	3 ms
B	5 ms	0.9 ms	5 ms
C	6 ms	1.1 ms	6 ms
D	10 ms	1.4 ms	5 ms
E	15 ms	1.9 ms	15 ms
F	15 ms	1.2 ms	6 ms

Determine a suitable frame size and find a feasible schedule for the hyperperiod. Document your calculations and the schedule you obtain in your lab book.

Create a new (empty C) project using Simplicity Studio and rename it to something more suitable, like Lab3.

You will need to generate an interrupt once per frame, so configure the SysTick timer accordingly. (The core system clock defaults to 14MHz, so call SysTick_Config with a parameter equal to the frame period you want times 14,000,000. Make sure your parameter is an integer.) For example, if your frame period is 2 ms, you might write:

```
SystemCoreClock = 14000000; // 14 MHz for this device
SysTick_Config(2 * SystemCoreClock / 1000);
```

Configuring this timer will cause the function SysTick_Handler() to be called (as an interrupt) at the rate specified in SysTick_Config(). Initially declare SysTick_Handler() as follows:

```
void SysTick_Handler(void)
{
}
```

Implement your cyclic scheduler in this function. To do so, you will need to call predefined functions task_A(), task_B()...task_F() according to your schedule. Include the line below to declare prototypes for these functions:

```
extern void task_A(), task_B(), task_C(), task_D(), task_E(), task_F();
```

Go to the Lab 3 assignment page in canvas and download tasks3.o. Move it into your project folder. Right-click on your project name and select "Properties". Under "C/C++ Build" select Settings. Then under Tool Settings, select Miscellaneous from the GNU ARM Linker. Add "\${ProjDirPath}/tasks3.o" to the "Other objects" box and click both "Apply" and "Ok". Build your code and make sure there are no errors.

Procedure: Download your code to the EFM32GG and run it. If you have scheduled the tasks correctly, the two amber LEDs will light and remain lit. If not, you will need to debug your code. The same debugging functions and hooks you had in Lab 2 are available to you in this lab.

When your scheduler works, demonstrate it to your lab instructor.

Print a copy of your code and affix it into your lab book. Write a short summary or conclusion and submit your lab book to your lab instructor for grading. (Remember to initial and date!)

Points will be assigned according to the rubric below:

Criterion	Points
Lab book properly bound and kept in ink	1
Lab book contains a title and short description	1
Each page initialed and dated, large areas crossed out	1
Pages are legible with no obliterations	1
Lab book contains schedulability calculations	1
Lab book contains cyclic scheduler code	1
Program works	4

Late work is penalized 2 points (20%).