AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
MECHATRONICS ENGINEERING DEPARTMENT
CREDIT HOURS PROGRAM

Spring Semester, 2024

CSE411: Embedded Systems

Lab (3): Creating Simple Scheduler.

Goals of this Lab:

Real time operating systems are composed of various tasks. Tasks are infinite loop functions that serve certain functionalities. The operating system Kernel switches the processor resources (processing time, hardware peripherals, etc.) between tasks. The goal of this lab is to implement a simple scheduler that switches between two tasks non preemptively.

Introduction:

A scheduler is the core of RTOS. You will have to create a non-preemptive scheduler that schedules different tasks and call them whenever their periodicities occur. The scheduler will be called in the main and will schedule according to the counter that increments in the Systick handler.

Task 1:

Use the SysTick timer to create a simple non-preemptive scheduler with the requirements below.The scheduler will call 3 tasks.

1. Task 1 will run every 5 ms.
2. Task 2 will run every 10 ms.
3. Task 3 will run every 20 ms.

Task 2:

The user needs an **API** to create the tasks from the **main()**. Implement a function called "**Create task** "The function takes two arguments:

1. A pointer to the periodic task created.
2. The periodicity of the task.

Note:

The scheduler implementation must be in separate files (e.g., **scheduler.c** and **scheduler.h**) and the application is in the main file.

Your Scheduler supports multiple tasks using a macro "**NUM_OF_TASKS**" that is predefined in the **scheduler.h** file.

***Hint:* That should cause build errors and you have to fix them. Also, you have to get the syntax of the pointer to function or the syntax of the array of structs you will use yourself.**