AIN SHAMS UNIVERSITY
FACULTY OF ENGINEERING
Credit Hours Programs

Summer Semester, 2021

CSE 345/CSE 347: Embedded Systems

Lab 1: Manual Tasks Switch

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 learn how to manually switch between two tasks.

Step 1: Create the tasks:

- Create two infinite loops functions; one toggles the red LED each 500 ms on Port F and the other toggles the blue LED each 1000 ms.
- The tasks code should be written with the help of the TivaWare™ Peripheral Driver Library.
- The structure of a infinite loop function should be:

```
void function_name (void)
{
    //initializing code to be executed only once.
    for (;;)
    {
      // infinite loop code to be executed repeatedly
    }
}
```

Step 2: Write the Systick handler:

- Initialize the Systick to fire an interrupt each 100 ms.
- The Systick interrupt handler should increment a counter variable to keep track of the time passed. PS. This counter could be used in toggling the LEDs.

Step 3: Write the program main:

- Write the program main such that it:
    - Calls the functions that initializes the Port F and the Systick.
    - Calls the task that toggle the red LED.
    - Contains an infinite loop that should never be reached.

Step 4: Run the code in debugging mode and manually switches between the tasks:

1. Run the code in debugging mode.
2. Find the memory address of each of the two tasks with the help of the disassembly window.
3. Put a breakpoint in the Systick handler.
4. Find the return stack from the Systick handler with the help of the memory window and the stack pointer in the register window.
5. Change the value of program counter in the return stack to the address of the blue LED task.
6. Witness what happens.

Step 5: Repeat step 4 to switch back to red LED task.