

CMPEN 472, The Pennsylvania State University

Homework 8

Due: Oct. 25, 2023 11:30pm

Objective

To learn interrupts and timing: Real Time Interrupt (RTI).

Textbook Reading (for next homework):

1. Chapter 9
2. MC9S12C Family Reference Manual/Data Sheet: Chapters 1, 5, 9, and 13

Instruction

1. Write a program to make a **digital clock**, displayed on the 7-segment displays attached to PORTB and the HyperTerminal connected to the HCS12 board simulation. Full 24 hour digital clock in hh:mm:ss format are accepted and displayed.
2. The digital clock rules are:
 1. 24 hour clock, display on the terminal and the two 7-segment LED displays
 2. "t" for 'set time' command, "q" for 'quit' command, "h" for 'hour display' command,
 3. "m" for 'minute display' command, and "s" for 'second display' command
 4. Display two digit hour, minute, or second on the LED display
 5. Two digit display on the two 7-segment LED displays attached to PORTB
 6. Update the time on the terminal and LED displays every second
 7. Clock commands ("t", "q", "h", "m", and "s" commands) entered on the terminal screen
 8. Use only one line on the terminal for the clock, separate columns for "Clock", "CMD>", and "Error>" display
 9. Show "Clock> " prompt for time display, "CMD> " prompt for command, and "Error> " prompt for error message
 10. In case of an invalid input, print error message on the same line: "Error> Invalid input"
 11. Clock time command display continue updated every second, while a "t" command being entered
 12. Use Real Time Interrupt feature to keep the time
3. The example Terminal time display line should look something like the following:

Clock> 12:34:58	CMD>	Error>
Clock> 12:34:59	CMD>	Error>
Clock> 12:35:00	CMD> t 11:0	Error>
Clock> 12:35:01	CMD> t 11:09:35	Error>
Clock> 11:09:35	CMD>	Error>
Clock> 11:09:36	CMD>	Error>
Clock> 11:09:37	CMD> t 76:5	Error>

```
Clock> 11:09:38      CMD> t 76:54:32      Error>
Clock> 11:09:39      CMD> t 76:54:32      Error> Invalid input
Clock> 11:09:40      CMD>                  Error>
Clock> 11:09:41      CMD> m                Error>
Clock> 11:09:42      CMD>                  Error>
Clock> 11:09:43      CMD> s                Error>
Clock> 11:09:44      CMD>                  Error>
Clock> 11:09:45      CMD> a                Error>
Clock> 11:09:46      CMD> a                Error> Invalid input
Clock> 11:09:47      CMD>                  Error>

Clock> 12:34:54
Clock> 12:34:55
Clock> 12:34:56      CMD> t 20:0
Clock> 12:34:57      CMD> t 20:09:35
Clock> 20:09:35
Clock> 20:09:36
Clock> 20:09:37
Clock> 20:09:38      CMD> t 18:0
Clock> 20:09:39      CMD> t 18:094:35
Clock> 20:09:40      CMD> t 18:094:35      Error> Invalid input
Clock> 20:09:41
Clock> 20:09:42

Clock> 12:34:55
Clock> 12:34:56      CMD> t 78:0
Clock> 12:34:57      CMD> t 78:09:35
Clock> 12:34:58      CMD> t 78:09:35      Error> Invalid input
Clock> 12:34:59
Clock> 12:35:00      CMD> q
Clock> 12:35:01
Clock stopped and Typewrite program started.
You may type below.
```

4. Run and study the following sample programs that use Real Time Interrupt:

Simple LED blinking example program - [hw8sample1e.asm](#) file.

For the additional CodeWarrior simulation, please follow the Homework 8 [Full Chip Simulator](#) guide. And be sure to set the **interrupt count number** so that your clock display is accurate within 1 second over two minutes, running CodeWarrior on your PC.

5. Make your program user friendly by giving simple directions as to how to correctly use your program.
6. Also, make your program 'fool-proof', never crash or stop based on wrong user response.
7. You may add other features or decorations.
8. Use as many re-usable subroutines as possible, and make your overall program to be small. So you may re-visit your Homework 7 and digital clock, and identify the tasks in your main program that can be made to be subroutines. Once you made those subroutines, your main program becomes much more simpler and your overall program be smaller. In many cases, your program may be run faster too.
9. Design the program to start at \$3100 and data to start at \$3000.
10. Be sure to put much comments so that grader and others can clearly and quickly understand your program. Comments are very important in assembly (and any) language programs.
11. You may want to see and check the [Sample Grading Sheet](#) for this homework.
12. Copy your 'main.asm' file to 'cmpen472hw8_YourLastName.asm'. For example, mine will be 'cmpen472hw8_choi.asm' Do not ZIP your 'cmpen472hw8_YourLastName.asm' file.
13. Turn-in your project source code file through [Penn State CANVAS](#). Upload your source code file into the CANVAS Assignment's Homework submission. Be sure to select CMPEN 472 class and correct Homework number, and with correct file name.

Congratulations on your eighth CMPEN 472 homework completion!
