

# Homework #3

**Out:** May 9, 2023  
**Outline Due:** May 13, 2023  
**Due:** May 20, 2023  
**Value:** Outline: 60 points  
Correctness: 210 points

## Hexer Display Routines

Write the display functions for the Hexer game. The functions to write are:

[ClearDisplay\(\)](#) clears the display, no LEDs are on  
[DisplayHex\(\*n\*\)](#) outputs the number *n* to the 7-segment LED display in hexadecimal  
[DisplayGameLEDs\(\*m\*\)](#) outputs the passed pattern/mask *m* to the game board LEDs

These functions are described in more detail below.

`ClearDisplay()`

The function clears the display. After it is called all LEDs should be off.

`DisplayHex(n)`

The function is passed a 16-bit unsigned value to output (*n*) in hexadecimal (at most 4 digits) to the 7-segment LED display. The number (*n*) is passed in R17 | R16 by value.

`DisplayGameLEDs(m)`

The function is passed a 16-bit mask (*m*) that indicates the game board LEDs to turn on and off. If a bit in the mask is set, the corresponding game board LED is turned on and if it is reset the LED is turned off.

The bits are defined as follows:

Bit															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LED 9	LED 1	LED 5		LED 10	LED 4	LED 3	LED 2	LED 13	LED 11	LED 8	LED 15	LED 14	LED 12	LED 7	LED 6

The mask (*m*) is passed in R17 | R16 by value. Unused bits have no effect on the display.

Additionally you will need to write the functions necessary to multiplex the LEDs under interrupt control. (This includes the multiplexing function to call from the timer event handler and the initialization routine for that function).

To test and demonstrate your code you **must** use the procedure `DisplayTest`. This procedure is in the segment `cseg`. It is defined in the file [HW3TEST.ASM](#) (in the directory U:\EE10b). The procedure loops calling the procedures listed above with time delays between each call. You **cannot** modify this program. You can also do your own testing by setting breakpoints at the start of each procedure and putting appropriate values in R16 and possibly R17.

When you demonstrate your code you must turn in a printout of all new or changed code and you must **also** submit your program electronically via the [website](#). The file submitted electronically should be a zip file containing all files needed to build your program, including any make or batch files.

To cut down on "busy work" you are being supplied with tables of segment patterns for 7-segment displays. One table contains patterns for each ASCII character for which a "reasonable" representation exists. The table has 128 entries (many of which are the blank pattern for non-representable characters), one for each of the 7-bit ASCII codes. The second table contains patterns for each hex digit value (16 entries). Like all fixed tables, the tables are located in the segment `cseg`. The tables are in the file [SEGTABLE.ASM](#) (in the directory U:\EE10b).

## Notes

You will need to write a main loop to test your code. This just initializes the I/O ports, timers, and switch code then calls the `DisplayTest` procedure.

The display functions should be in their own files, separate from the initialization functions and the main loop and test functions. You should create **at least** 4 files: a file with the main loop; a file with the display routines (including interrupt driven multiplexing); a file with the timer and I/O port initialization functions; and **at least** one include file with your symbol definitions.

You do not need to modify the supplied file containing the character tables, just link it with your code.

## Resources

- Hexer board [schematic](#)
- [segment tables](#)
- [test code](#)
- [test code description](#)
- [Homework Q&A](#)
- [Electronic Submission](#)

---

*Last updated May 18, 2023 04:05 AM by [glen@caltech.edu](mailto:glen@caltech.edu)*

copyright © 2023, [Glen George](#). All rights reserved. Reproduction of all or part of this work is permitted for educational or research use only, provided that this copyright notice is included in any copy.