

Homework #2

Out: April 25, 2023
Outline Due: April 29, 2023
Due: May 6, 2023
Value: Outline: 60 points
Correctness: 210 points

Hexer Switch Routines

Write the `GetSwitches`, `SwitchAvailable`, and switch debouncing functions for the Hexer game. The functions to write are:

[`GetSwitches\(\)`](#) returns a debounced switch pattern
[`SwitchAvailable\(\)`](#) returns TRUE if a debounced switch pattern is available

These functions are described in more detail below.

`GetSwitches()`

This function is called with no arguments and returns with the code (you define the codes) for the switch pattern pressed and debounced in register R16. The procedure should only return fully debounced switch presses. The routine returns with the switch pattern code after the switch has been pressed and debounced, but **not** necessarily released. The procedure does not return until a switch has been pressed and debounced (this is called a blocking function).

`SwitchAvailable()`

The function is called with no arguments and returns with the zero flag **reset** if there is a debounced switch pattern available and the zero flag **set** otherwise. In other words, if `SwitchAvailable` returns with the zero flag reset that means `GetSwitches` would return immediately if called because there is a debounced switch pattern available (it won't need to wait for a switch to be pressed). This function should **not** affect whether or not a switch is available.

Additionally you will need to write the switch debouncing function (which is called by the timer event handler). This function should be called approximately every millisecond. It either checks for a new switch pattern being pressed if none is currently pressed or debounces the currently pressed switch pattern. Once it has a debounced switch it should set a shared flag (used by [`SwitchAvailable`](#) and [`GetSwitches`](#)) and store the

switch pattern code in a shared variable (used by [GetSwitches](#)). You will most likely also need an initialization function that initializes these shared variables.

To test and demonstrate your code you **must** use the procedure `SwitchTest`. This procedure is in the segment `cseg`. It is defined in the file [HW2TEST.ASM](#) (in the directory U:\EE10b). The procedure stores the sequence of switch presses in an array in memory. You **cannot** modify this program.

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.

Notes

The main loop is supplied but you will need to write functions to initialize the I/O ports and timers.

The switch functions should be in their own file, 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 switch routines (including interrupt driven debouncing); a file with the timer and I/O port initialization; and **at least** one include file with your symbol definitions.

Resources

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

Last updated April 14, 2023 01:19 AM by 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.