Requirements Specification for EELE465 Lab Project 3: Keypad and Time-Varying Patterns on Eight LEDs

Academic Dishonesty Clause: It is expected that you, along with your partner, write this code uniquely for this project during this semester. It is acceptable to use code YOU have previously written as examples and guides. It is not acceptable to copy/paste code written by other individuals nor by you during previous semesters or for other classes.

Lab project goal: Enter values using 16button keypad and use them to control LEDs.

A keypad will be used to select from different time-varying patterns displayed on eight LEDs using your MSP430 and a breadboard. Patterns will be selected based on which button is pressed on your keypad. Three patterns will be time-varying and repeating sequences, one will be static. An unlock code will be required to begin the display.

Outcomes:

After this lab you should be able to:

- Poll a 16button keypad and determine which button was pressed
- Develop functional user interface code more complicated than simply polling whether a single switch is high/low
- Interface with an 10digit LEDbar (we'll only be using 8 of the 10 digits)

Background:

The most complicated component of this lab is to understand how the keypad functions. Refer to the lecture notes for details on this.

Because of the complexity of the keypad, you won't be able to simply check a single input status and then make a decision based on that individual sample. Instead, you'll need to initialize the I/O port connected to the keypad, save some status information, modify the I/O port initialization, and collect more status information. Then, you will concatenate the first sample with the second, and finally make an analysis as to which button was pressed. While this won't be particularly difficult, you will need to have very precise ideas of how the flow will work before starting to code. It will be very important to develop a useable flow chart and keep it up-to-date.

Be sure to complete the workload distribution form with your partner before starting on this project. Upload the distribution form before the end of the first project day dedicated to this lab.

Acknowledgements: Randy Larimer

Requirements for lab project completion:

Part 1: Getting started with the keypad – all partners are responsible for this step
Connect the keypad to the MSP430 dev-board, you might want to jumper this to a breadboard. You need to implement a polling routine that will check the status of the buttons. Before doing so, write a sample-code to verify that you are able to observe the changing I/O pins. Assuming you can see the port-values change, implement a polling routine to determine which button is pushed. It might be wise to save the keypad results for future use.

Part 2: Control the LED lightbar
Review the following datasheets: 08DC10IDA and 4100R. Determine the orientation of the LED lightbar. Explain the need for the 4100R. If you are not sure what to do with this part and why – ASK before connecting anything on this step. Determine the tolerance of the 4100R. Connect these parts to your MSP430 dev board and verify these hardware connections are correctly configured. Perhaps you should manually verify the operation in hardware, then write a simple test-script to test the software compatibility. Create some sort of indication pattern to be used in part 3. This pattern should be a static-symbol, similar to pattern A. When initialized, your final version of code should initial the lightbar to display this pattern.

Part 3: Keypad unlock code
Once you are able to use the keypad, you can start developing the project code. The code should sit in an infinite loop until it is "unlocked." Use the static pattern created for Part 2 to indicate you are "locked."

Use your 16button keypad to enter an unlock code of your choosing. The code must be at least 3 digits long, but can be longer.

Once the unlock code is correctly entered, turn the entire lightbar off and wait for an input from the keypad. After the unlock-code has been correctly entered, the keypad should freely control any of the patterns described below. The code should not immediately relock after the patterns complete – however, if you would like to add some sort of timeout or unlock feature that is acceptable, so long as you are able to run multiple patterns prior to reaching an unlock state.

Part 4: User control over the LED lightbar
Use your 16-button keypad as select-switches for the desired patterns (a through d) described below. All patterns should commence and continue scrolling after a single press (ie: you should not have to hold the button down). Standard teams of two will need to complete three patterns (A, B, and C) to receive full credit on this lab, pattern D is extra credit.

Acknowledgements: Randy Larimer

(X indicates LED is ON, 0 indictes LED is OFF)

a. *Select this pattern with switch "A" (static)*
4 LED's on, 4 off in this pattern X0X0X0X0.

b. *Select this pattern with switch "B" (counting up)*
Pattern displays a binary counter with the following pattern. Value should update 1x/sec. Polling for a new button press should continue and switch if/when that happens.

A representation of the sequence of this pattern is:

left	0	1	2	3	4	5	6	7	right
0	0	0	0	0	0	0	0	0	
1	0	0	0	0	0	0	0	X	
2	0	0	0	0	0	0	X	0	
3	0	0	0	0	0	0	X	X	
4	0	0	0	0	0	X	0	0	
				continue					
254	X	X	X	X	X	X	X	0	
255	X	X	X	X	X	X	X	X	
				repeat					

c. *Select this pattern with switch "C" (scrolling)*
Pattern rotates to the right. Pattern is always one LED off, seven on. Pattern should start with the all LEDs lit except left-most LED. Exactly one LED in the string should be off at any one time. Each LED should stay off for 2.0 seconds, then the next LED to the right should turn off and the previous LED should light.

A representation of the sequence of this pattern is:

left	0	1	2	3	4	5	6	7	right
i.	0	X	X	X	X	X	X	X	
ii.	X	0	X	X	X	X	X	X	
iii.	X	X	0	X	X	X	X	X	
iv.	X	X	X	0	X	X	X	X	
v.	X	X	X	X	0	X	X	X	
vi..	X	X	X	X	X	0	X	X	
vii.	X	X	X	X	X	X	0	X	
viii.	X	X	X	X	X	X	X	0	
ix.	0	X	X	X	X	X	X	X	(start sequence over again)

d. *Select this pattern with switch "D" (in and out)*
Two LEDs are on at any one time. LEDs are on for 2.0 second followed by a dwell time where all LEDs are off for 0.25 seconds.This pattern starts in the middle, works to both ends, then works back to the middle.

A representation of the sequence of this pattern is:

left	01234567	right
i.	000XX000	
ii.	00X00X00	
iii.	0X0000X0	
iv.	X000000X	
v.	0X0000X0	
vi.	00X00X00	
vii.	000XX000	(start sequence over again)

Acknowledgements: Randy Larimer

After power-on reset, all LEDs should be off. When the proper switch (A, B, C, or D) is pressed on your keypad, the displayed changes to the appropriate pattern and runs that pattern until another button is pressed.

If the 'next' button pressed in the same selection, the pattern should restart from the beginning.

If the 'next' button pressed is a different selection, the should immediately stop the current pattern and start the new pattern. If a previously selected pattern is re-selected, the pattern shodul restart where it left off.

For example:

User selects B	-	Counting Pattern starts counting **0, 1, 2, 3,**
User selects B	-	Counting Pattern restarts at count **0, 1, 2, 3, 4, 5,**
User selects C	-	Display switches to Scrolling Pattern displaying **i, ii, iii,**
User selects B	-	Counting Patern resumes **6, 7, 8….**

The display should react immedately to a button-press. It should not wait until the end of a pattern-step to switch patterns.

Project stretch goal #1(worth +0.5 EC): Complete pattern D.

Project stretch goal#2 (worth +0.5 EC): The three digit code should be completed within 2 seconds of the first correct button pressed. If 2 seconds passes before the 3 digits are entered, the user should be required to start the code over. If you choose to have a longer code (ie: more digits), you may make this requirement longer, so long as it is a known/predictable duration.

Double stretch goal (worth a high five from your favorite microcontrollers instructor): Create the 2 second duration above using the *same* timer as the 1 second LED flash. Hint: You will only ever use the unlock code before the LED flash code, and you can reconfigure anything (ie: Timers) on the fly.

Triple stretch goal (worth a high five from *all* your microcontrollers instructors): Create a signal of your choosing to indicate that the time required for the passcode has expired.

Acknowledgements: Randy Larimer

Assessment for project completion:

Start the demo with the code running, the microcontroller in the "lock" state, and lock-indicator from part 2 visible.

a. Introduce your project (both partners should be involved)

    1. Use circuit diagram to explain how the circuit was constructed

    2. Use flowchart to describe how the keypad was set-up/verified to operate before you started rigerously developing the project. Summarize whether the keypad is at all functional and, if so, whether there are any problems/non-ideal performance.

Present the aspects of the project you were primarily responsible for. If you share the workload, state that while presenting each step.

b. Demonstration

    1. Smoothly transition from "introduction" to "demo" by directing attention to the lock-indicator. Whoever was responsible for the initial set-up explain the 4100R.

    2. Prove the password capability by:
        - Use the Keypad to attempt to select a display to prove that the system is locked.
        - IF you completed Project Stretch #1, enter the unlock code slowly and demonstrate that you cannot unlock if you took longer than 2 seconds.
        - Use the Keypad to unlock the MSP (all LEDs should turn off).

    3. Select Display A.

    4. Select Display B, allow it to run until the instructor asks you to continue.
        - Note exactly which step of Display B you are on, and Select Display A.
        - Reselect B (it should start where Display B left off).
        - Reselect B again (now the count should restart at 0)

    5. Repeat all steps of #4 for Display C

    6. If you completed any project stretch goals beside #2 (which was demonstrated already), state what you did (one goal at a time) & demonstrate that function.

c. Review of results
    1. The instructor should review
    2. Review each item and explain how the points should be divided according to the final version of the workload distribution form.
    3. Be sure to review any extra credit you expect to receive (per individual).

Acknowledgements: Randy Larimer