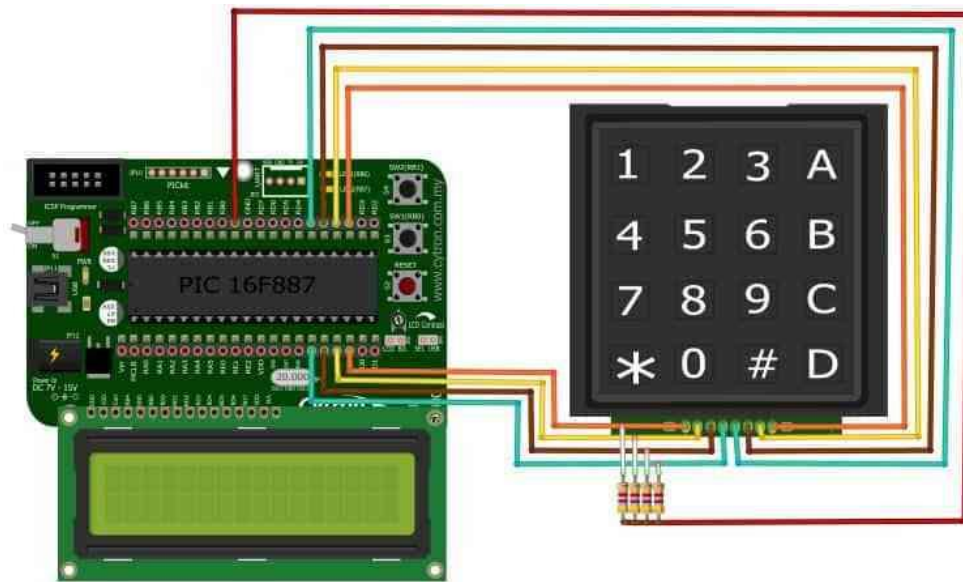


Microcontrollers Lab

Week 11 – LCD, Keyboard, Timer and Interrupt integration

May 4th – Due May 13th 2020

In this lab the idea is to integrate aspects of the four modules we have seen so far: LCD display, 4x4 matrix keyboard, timers and interrupts. Thus, before proceeding, be sure that your code for the LCD and the keyboard works properly. You can use either the 8- or 4-bit option for the LCD code.



The lab will be divided into three parts, described as follows:

Part 1. Simple GPIO Interrupt. Implement the example one seen in class: the main program toggles the red LED continuously and can be interrupted by a push button connected to port PTA1. The interrupt routine service (SIR) for this button is simply toggle the green LED for a short time, then moves back to thread mode to toggle the red LED again.

Part 2. Distinguishing interrupts from different pins. In this part, you should implement the second example seen in class: two buttons are connected to the KL25Z board through the port A (PTA1 and PTA2), and both can interrupt the main process running in thread mode is the processor (again, just the red LED being toggled continuously).

As there is just one interrupt for the port A, we need to implement a mechanism for differentiating between the two ports; this can be done through flags in the ISFR register. Please remember to properly disable interrupts before the initialization code in your program. Also, it is important to clear the interrupt after having served it, otherwise we might never go back to the main program!!!

Part 3. Event counter. The goal of this last part of the lab is to integrate concepts of interrupts and timers, and put together a simple application involving the LCD and the 4x4 matrix keyboard. The main idea is to reuse the code of part 1 of this lab, but instead of toggling LEDs, we will integrate into the second part of the previous lab (ascending timer): when the counter is counting, if interrupt from PTA1 is activated, the code should execute an ISR that halts the counter and shows a message such as "PAUSED", the main program can resume if you press the * key.

Requirements for the report

1. Include the code for each of the functions of your code. The code should be commented and the report should include a short description of each function and how it works, including images of the registers that have been configured for enabling the different functionalities in your code (you should include the image of the registers as seen in class and in your commented code put which bits have been configured for certain purposes)
2. Provide a state machine or a flow diagram for the entire code (only applies for the last part)
3. For the last part, provide a schematic view of the connections of your design.
4. Attach a short video demonstrating the system working. Alternatively, you can share the link to the video in your google drive for me to evaluate it.