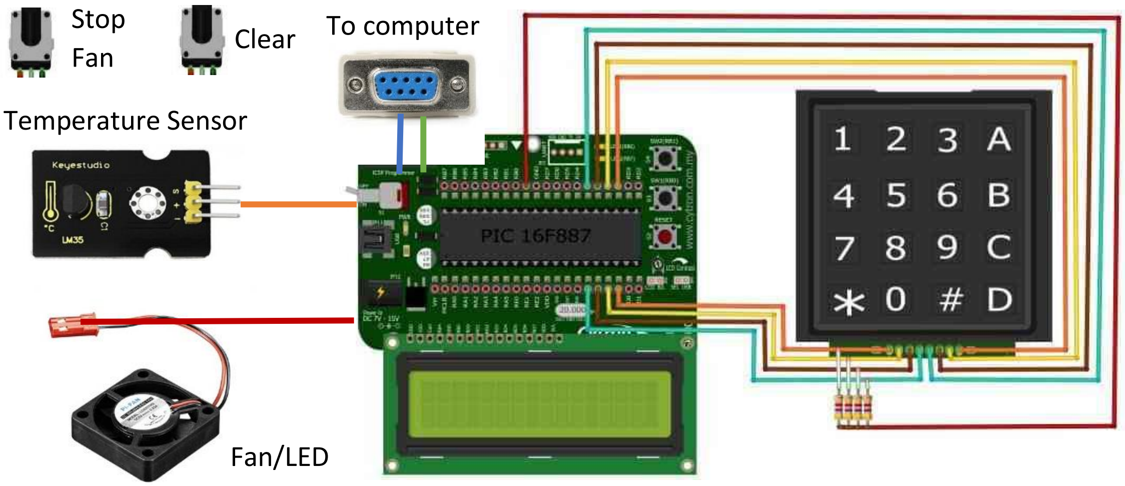
Microcontrollers Lab

Week 12 – LCD, Keyboard, Timer, Interrupt and ADC integration

May 18th – Due May 27th 2020

In this lab the idea is to integrate aspects of the four modules we have seen so far: LCD display, 4x4 matrix keyboard and ADC (and UART). 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 four parts, described as follows:

Part 1. ADC Part 1. In this part, you should integrate the ADC and an internal sensor. We saw that the ADC has an internal temperature sensor attached to one of the channels of the ADC. The idea in this part of the lab is to retrieve that value through a program running in your MCU. Instead of using the LEDs as it was the case of the example seen in class, you need to convert the integer obtained from the ADC to display the temperature in the LCD screen through polling.

Part 2. ADC Part 2: Simple thermostat. As with the previous part, we will be using the internal ADC, but this time connecting a sensor. It will be optimal if you use something similar to the LM35 but any other sensor is fine (even a potentiometer). We will be using the keyboard, LCD, timers, interrupts and the UART. The code in this part should proceed as follows:

1. When you start your application, the message next message should be displayed.

Set the desired temperature

2. As with the previous code you should be able to read the temperature from the keyboard and display in the LCD. Then, “set” the “air conditioner” (a LED, but if you have a simple fan it would be nice). In order to start sensing the temperature (using the ADC) you should press the # or * button. The fan is activated for a predefined time (let say 10 seconds, for demonstration purposes) using a counter, and re-activated every minute to “maintain the temperature” (this is how these systems usually work)

3. The message should be displayed it for the normal execution of the program (only changing the values of the ADC), and the system should keep activating the fan (or LED). However, if the value surpasses the desired temperature, the system should activate another output (“alarm”) and activate the fan for a longer time (2 mins). After this time, if the temperature is still higher than the set temperature, it should go again for two mints. Otherwise, it moves to normal mode.

4. The fan can be put in idle mode at any moment through a pushbutton via an interrupt, as in the previous lab. The idea is that you still show the temperature but you don’t activate the fan in this mode. Whenever you want, press the # button in the keyboard to resume.

5. Reset the system with a second button and an interrupt, in which case the code should go to the first step and ask again for the temperature.

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 second 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.