A new subroutine has been implemented in the code for the microcontroller. This subroutine is responsible for the liquid crystal display (LCD) and “enter key” interruption. The purpose of this subroutine is to introduce the use of liquid crystal display (LCD) and code of operation behind enter key. The use of “enter key” can be used in all microcontroller applications where there is a need to enter the data or have any subroutines being called to perform a specific function with a push of button. The use of liquid crystal display (LCD) provides a cost effective means to display information or a means of communication between user and microcontroller.

In order to implement the “LCD and enter key” code, the work is broken down into three different parts performing three tasks. First of all, the code for LCD is written and then this code is tested by single stepping through shift registers and observing the change in LCD_TEMP and LCD_TEMP2 variables. Also, code from previous weeks is then enhanced by the addition of new code LCD2.ASM.

For the second task, new subroutines from LCD2 are installed into the previous week’s programs. In addition to that, there are few statements added to test if new code and old code are implemented correctly. At the end of task 2, the new program should be able to display the characters on the screen.

For the third task, a new subroutine is to be added called “Enter Key”. This subroutine will allow the control of messages displayed on the screen. Also, once this...
The subroutine that allows sensing “enter key” is inserted in the main loop. Once implemented, Port E bit 2 is tested for being “Low”; being “low” will enable it to display name on the LCD (for testing purposes). However, if the enter key isn’t pressed, the value from Port E bit 2 will result in “High” and it will just read “Enter” instead of showing name. The sole purpose of having to test the LCD is to enable the use of enter key/keys while watching for errors that might happen. In case of errors, the LCD won’t display the expected result allowing us to go back and debug the program.
Besides the LCD and enter key subroutine, a motor control subroutine is added to previous week’s program code. This subroutine is responsible for lighting up the LEDs indicating the pulse width modulation controlled by the microcontroller. Pulse width modulation is way of controlling the motors using software instead of having an outside voltage control. These PWM will control LED’s intensity (dim/bright) and all will regulate power (on/off). Upon installation into the circuitry, these PWM will be responsible for running the motors in a similar fashion.

Figure 3: Picture of a quick start board used for programming the microcontroller
Future work:

PWM will be extensively tested using a motor once mechanical parts are completed. Having the motors will allow us to test the other constraints of the program as well. Instead of having an IR receiver for the keypad, a radio frequency based keypad will be installed. Once the grippers are made, the force sensors can be put on a protoboard and tested. These force sensors can be easily placed under the rubber cover of the PVC for the grippers.

Project Overview:

So far, mechanical parts have been in the process of updating on a daily basis. The programming aspect of the microcontroller is well under way. The LEDs on the quick start have been extensively tested to check the pulse width modulation so that we won’t run into any problems once motors are connected to the microcontroller. As of now, it is a matter of receiving motors and power source to get started with electrical components.

Hours worked: 11