Work Completed:
After encountering several problems with the configuration and initialization of the UART functionality of the PIC24 chip, I was finally able to determine the settings necessary for establishing successful communication with the LCD screen, via its serial port, (see Fig. 1).

Figure 1 – Communication between PIC24 and LCD Screen

I can now send commands and text to the screen using a program that executes directly on the microprocessor. The main issues that needed to be addressed were the proper configuration of an oscillator for the processor, and the determination of the correct settings for the PIC24’s UART baud rate, mode register, and status register. With assistance from Dave Price, I was able to connect a 6 MHz oscillator to the microprocessor and enable it for use by setting configuration bits programmatically. I was able conduct initial tests of UART communication by connecting the processor to a computer’s serial port and using HyperTerminal to display received data transmissions. The initial tests were only partially successful; characters were received incorrectly. In stepping through the entire range of possible characters, it appeared that several letters and numbers were missing. Believing that the baud rate had been incorrectly set, I conducted numerous tests in an attempt to determine the correct setting to use. These tests failed to fix the problem; they only appeared to result in greater deterioration of the transmission signal.

I attempted to use delays between character transmissions and parity bit error checking, in hopes of successfully being able to send data. Unfortunately those methods also failed solve the problem.
After searching for possible suggestions on Microchip’s developer forums, I found reference to an alternative to the HyperTerminal program, called RealTerm (see Fig. 2), which is also capable of monitoring serial port communication.

RealTerm provides many more options for serial port configuration and data display than the HyperTerminal program. One of the more useful settings was an option that let you view incoming characters as binary blocks. Using this display mode I was able to determine that data being set was being inverted and shifted by one bit. Some further investigation revealed that the settings for the PIC24’s Tx and Rx functions could be configured to make them Active Low (output 0’s) or Active High (output 1’s) when interpreting character data. Switching the Tx setting to Active High resulted in data being sent successfully.

Once the UART configuration was properly set, I began looking at the commands that would be needed to perform basic functions on the LCD screen; such as that needed for clearing the display. I then configured a little demo program to clear the screen and display a simple multi-line message.

Future Work:
I will work on developing the menu navigation and prompts that will be require for the syringe loading device. I will set up simple switches to provide inputs to the menu, to mimic the selection of various options, and trigger conditional code execution. I will look at using the PIC’s built-in Real Time Clock and Calendar to retrieve the current date and time. I will also experiment with writing and retrieving information to the microprocessor’s data memory.
Project Review:
Work continues on the syringe cartridge. The plunger claw has been welded together; some adjustments need to be made to allow the plunger shaft to more easily fall & lock into position during loading. Several parts have recently arrived, (the case, the stepper motor control IC’s, assorted gears, tilt sensors, the sound output module); work needs to begin on configuring these components and incorporating them into the over all design of the device.

Hours Worked: 14 hrs