Lab 5: LCD Display

Reference

Hitachi HD44780 Datasheet.

Introduction

In the previous labs we have displayed constant messages, but we have not been able to display variables calculated by the program. Also the EnterKey gave us only two options. Today's lab will display not only "Press Enter", and Name and Affiliation, but also will serve the practical purpose of displaying and scanning through the entire character set of the LCD display on the Qik-Start board. In the present lab we will see how to display variables with a new subroutine called DisplayV. In addition we will develop a generalized routine routine that accesses different modes of operation. We will reactivate the RPG interrupt handler and use the incr and decre keys to control access to the character codes of the LCD display.

Main

Modify the main() subroutine such that initially the message "press enter " is displayed. When the Enter key is pressed again, your name and affiliation is displayed, but unlike last lab, when released the name and affiliation should stay. Another press puts you in the "character mode". In the top row, display the two digit hex vector, a space, and the associated character in the character set of the PIC, starting with H'30' or 0. For example, initially the upper row should read:

30 0

Use the memory locations designated Row1 to store this data.

In the second row, display the entire 16 characters in the column in which the character in row one resides. For example your initial second row should be:

0 1 2 3 4 5 6 7 8 9 : ; < = > ?

Use the sequence of variables Row2 to store this data for display. When the Enter key is pressed again, you will go back to display of your name and affiliation, NOT 'Press Enter'. Thus the sequence of events controlled by pressing Enter goes Press Enter-> Name -> Character -> Name.... . You can accomplish this by defining a state variable that in this context is simply a variable that controls the state or mode of operation of the microcontroller during a particular pass through a segment of code.

RPG

Restore the interrupt handler from lab 1 that responded to the INCR and DECR buttons.
The simulated rotary pulse generator is based on a 12 series PIC on your board that senses pulses from the INC and DEC push buttons and generates a pulse to portb,0 and a high or low to portc,0 [low for DEC, high for INC]. Since RB0 is the external interrupt port, the rising edge causes an interrupt. In lab 1, we used this code to set bits in the LEDs. Now, we will use the RC0 flag to tell us whether the INCR or DECR button was pressed. We can use that information to set the CharCode variable which is used in the DisplayV() routine described below. Be sure to initialize CHARCODE to 0x30 in Initial(). Also note that keeping INC or DEC depressed causes a train of pulses to B0, causing it to increment or decrement rapidly.

**DisplayV**
Develop a new subroutine named DisplayV that should be called when you are in character mode. This routine will display the value of CharCode in hex on the top row as well as the character representation. On the second row, you will display the entire set of characters corresponding to the column comprising CharCode. You may wish to add two new subroutines to lcd2.c – DisplayCharacter() and DisplayHex(). DisplayCharacter() will output a single character to the display instead of a full string like DisplayC. DisplayHex() will take an eight bit number and output a two-character hex number to the display. This routine will convert the top four bits and the lower four bits of the input number to hex and then call DisplayCharacter() to display it. A simple formula to convert a four bit number to a hex character is to add “0x30” if the number is less than 9 and to add “0x37” if the number is from 10-15.

**Conclusion**
Keep all other functions operational. Thus BarChart and BarIntensity should continue to operate. If your code exhibits difficulties, set up a watch window and use the ICD to debug. Demonstrate your project and hand in your well-commented C code by end of class next week.