EE-3306 HC6811 Lab #2
Modular Programming

The purpose of this lab is to gain further experience in writing modular assembly programs and
debugging programs. This lab builds on the knowledge gained from the previous lab to read the
keypad and output characters to the LCD screen. This lab aims at help students build modular
programming skills writing code to increment a number entered through the keypad.

Task for the Lab:
The main purpose of this lab is to learn how to code and assemble an entire program to
increment a given number. The entire program can be broken down into 4 small sections.
- Display a message on the LCD screen, requesting the user to give the input (in this
case, a single digit number).
- Convert the input from ASCII format to decimal.
- Increment the given number.
- Convert the decimal number to ASCII format and send it to the display.

Using the following hints the students are required to write a program and assemble it to
show the output (incremented number) successfully. The TA will also assist you in writing
the code. The programs provided in the previous lab should be able to assist you in writing
this program. You can use any subroutine given in those programs in your code.

Step 1:
Before starting off with the main program, there are a few steps to be followed as is seen in
the previous programs. They are listed below:
- Assign labels to the various LCD command codes, addresses of the different lines
within the display, PORTD and PORTE. These will prove to be very convenient
while writing a big program.
- Assign a starting memory location from where the main code is stored. This can be
done using the assembler directive ‘ORG’ (usually the memory location used is
$2000).
- Create a lookup table in the memory with all the characters available on the keypad
for reference in the program.
- Assign variables for the different message strings that you will need to use in the
program. Note that the messages should be within single quotes (‘ ‘) and end with a
‘0’.

Step 2:
First thing you can do is assemble all the subroutines required for the program. These
include subroutines meant for accepting keys from the keypad, identifying their row and
column, shifting the cursor to the required line on the display, initializing the ports and the
LCD, displaying characters and strings and finally creating delays.

Prelab: Complete Step 1 and 2 before coming to the lab.
Main program:
1. Disable all the maskable interrupts and initialize the stack location.
2. Initialize the LCD and also PORTD.
3. Display the first message asking the user to enter the input.
4. Position the cursor wherever required on the LCD screen.
5. Load the address of the memory location, where the string is stored, into the x register.
6. Call the subroutine ‘cputs’ which ultimately sends the message string to the LCD data register to display the message.
7. The next step is to identify if a key has been depressed. This is detected using the ‘anykey’ subroutine, which sends out a signal if a key has been pressed.
8. Identify what row and column the key pressed belongs to.
9. Use the look up table as reference and identify the key pressed.
10. Once the key has been identified it should be displayed on the screen. This can be done using the ‘putc’ subroutine (which sends out the character onto the LCD data register).
11. The key identified should be converted from ASCII format (H30+d number) to the decimal format. To get the actual decimal number, subtract H30 from the given number.
12. Place the number in the accumulator and increment it.
13. Add H30 to the resulting number again and store it in a separate register.
14. Clear the LCD again and display the result by sending it to the LCD data register.
15. Jump back to the main program after a slight delay.

Here is an example to show how the program works. It should clear the LCD and then prompt the user to enter a single digit number and wait for a response. After the user enters the number it should clear the LCD again, display the incremented number.

Clear LCD
Enter a single digit number: 7
Clear LCD
Answer: 8