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 and multiply two numbers entered through the keypad.

**Objectives**
- Gain further experience in modular programming and documentation.
- Understand how to implement user input from the keyboard from both menus and strings.
- Use error detection to display error messages for invalid control entries.

**Prelab:**
1. Where is the ‘reset’ vector of the 68HC11 located? Suppose your program is located at address $675f$, what do you need to do to get the 68HC11 to execute your program?
2. What are the functions of the instructions tab and tba?
3. What is the status of registers D and X after executing the idiv instruction?
4. What is the status of registers A and B after the execution of the mul instruction?
5. How is the hex to decimal and dec to hex conversion performed?

**References**

**Equipment for this lab:**
- 68HC11 trainer kit, to include 68HC11 EVBU and prototyping strips
- IBM compatible PC to connect to the trainer kit via an RS-232 serial cable
- Floppy disk provided by the student

**Laboratory Exercise**

**Notes**
- In this lab, all numbers such as addresses and data are given in hexadecimal format (“hex”) unless otherwise indicated. In completing the lab, record all information in hexadecimal unless directed differently.
- Make sure your development EVBU board is connected to power (green LED on board ON) and the serial port of the EVBU is connected to the serial port of your development PC containing the AXIDE software, configured to the correct port at the correct baud rate, etc.
Task for the Lab:
In this lab you will write a code to first display a message, increment a given variable and also accept a number (1 to 9) from the keypad to multiply it by a multiplication factor. The TA will have a working example of one implementation.

A step by step procedure for the program is given below:

- The following messages should be displayed initially when you run the program.
  - “Welcome to EE3306 Lab#2”
  - Enter A for increment.
  - Enter B for multiplication.

- When you enter the option 'A' the message given below should be displayed:
  - “Enter any positive integer.”

- The incremented number is displayed next.

- When option 'B' is entered the following message is displayed:
  - “Enter the multiplication factor:”

- Then again a message is displayed which asks you to enter any positive integer.

- The product is displayed next.

Hints to write the code:

- The following subroutines are provided to the students to make use of them in their main code:
  - Accepting a key from the keypad - 'anykey'.
  - Get the row and column number for the accepted character – 'get_column' and 'get-row-column'.
  - Identifying it from a given table of characters – 'key_lookup'.
  - Bringing the cursors to the beginning of each line on the display – 'csr_to_ln1, 2, 3 & 4'.
  - Display the messages used in the code – 'cputs' (which includes the subroutine to display each character on the display after checking the position of the cursor on the display).
  - Displaying a decimal number after converting it to an ASCII character – 'dec_to_lcd'.
  - Initialize the LCD (clear the screen) – 'INIT_LCD'.
  - Initialize PORTD which is the output port – 'INIT_PORTD'.

The following guidelines should be considered while writing the code with the above given requirements:

1. Set global variables for the LCD register addresses and the ports (which has already been done for the students).
2. Set the location of the main program ($2000).
3. Set up a table lookup for the characters on the keypad.
4. Set variables for the messages to be displayed on the display.
5. Start the main program by initializing the stack and masking all other interrupts.
6. Write code to display the welcome message in the beginning and enter the options after a long delay.
7. Accept a key from the keypad and check if it matches with 'A' or 'B' and keep reading the key until it is released.
8. If it matches with 'A' increment option is chosen and hence the following steps should be considered:
   1. Initialize the LCD and PORTD again to accept a new number.
   2. Display a message on the LCD to accept a number between 1 and 9.
   3. Accept a number from the keypad which is in ASCII form.
   4. Convert it to decimal format by subtracting 30(48 in decimal) from it.
   5. Increment the number and store the number in a separate register.
   6. Display the message to show the result of incrementation.
   7. Retrieve the incremented number.
   8. Convert it to a decimal number by running a loop to divide the number by 10 until 0 is obtained as the remainder. The quotient obtained in this division process will be the final decimal number.
   9. Convert each digit in the final result to ASCII by adding 30 and display the digits one by one.
9. If the key matches with 'B' multiplication option is chosen and the following steps should be followed:
   1. Initialize the LCD and PORTD again to accept a new number.
   2. Display a message to accept the multiplication factor.
   3. Accept a key from the keypad and convert it to decimal format by subtracting 30 from it. Keep reading the key until it is released.
   4. Store it in a separate register for future use.
   5. Initialize the LCD and PORTD again.
   6. Display a message to accept a number between 1 and 9.
   7. Accept the number and convert it to decimal again and store it in Reg A.
   8. Get the other number into reg B and multiply both the numbers and the result is stored in D.
   9. Transfer the contents from reg B to reg A since the product will not exceed 8 bits.
   10. Push the contents of reg A into stack and clear the screen again.
   11. Display the message to display the product.
   12. Display the product after converting each decimal digit to ASCII format.
   13. The procedure for converting the hex number to decimal is given in the previous steps.
   14. Create a long delay and repeat the process from the beginning.

Demonstrate to the T. A. that you program is running. T. A.
Initials: ______________________

When you are done, do the Following to document your program (Note: comments, structure, and readability counts!!!)
1. Hand-in a complete Flowchart and for the entire program (interface_students.RTF).
2. Hand in the listing of the complete, well-commented source-code that matches the flowchart. Remember, if the TA can not read your code with “reasonable” effort, then it is wrong.