# LAB #7: PARALLEL I/O PORTS AND INTERFACING

## OBJECTIVE

Students will learn how to read from and write to a peripheral device. Students will write to a 7-segment LED display connected to output Port B of the M68HC11 and read from a keypad connected to the bi-directional Port C.

## MATERIALS

- Axiom 68HC11 EVBU board
- Protoboard from EEL3701L Kit
- One 16-button keypad (will be provided by instructor).
- Eight 360Ω resistors for LED drivers
- Four 10KΩ resistors for keypad (may need more depending on your design)
- Two 74xx04 hex inverters (for LED drivers, will be provided by instructor)
- One 7 segment LED (will be provided by instructor)

## INTRODUCTION

The M68HC11 has five ports. All of these serve multiple functions depending on the operating mode and the value of several internal control registers. Port B is used for general purpose output. Ports C and D are bi-directional ports and can be configured for input or output using their respective Data Direction Register (DDR). For more details refer to Chapter 7 of your textbook or the M68HC11 reference manual.

## PROGRAM REQUIREMENTS

For this lab you will be writing one program that meets the following two requirements:

### Program Requirement 1

Display the following characters one at a time on the 7-segment display: 0123456789. You can initialize memory with the character string above. Each character should be displayed for approximately 1/3 second by writing to Port B. The 74xx04 will invert these bits before they reach the seven-segment display. The above sequence of characters should be displayed twice then the display should be cleared.

### Program Requirement 2

Continuously scan the keypad connected to Port C. Identify the key pressed. Display the character corresponding to the key pressed. That character should be displayed until another key is pressed.

## PRE-LAB

Read the entire lab handout and complete the pre-lab requirements before coming to lab. Procedures 1-9 listed below must be completed prior to lab time. You must have a syntax-error-free assembly program assembled and ready to be downloaded and executed on the EVBU board upon entering the lab. It is not the lab instructor's responsibility to trouble shoot your code or circuits. Prior to lab, connect the 16-button keypad circuit using the 10k resistors on the protoboard. Also, connect the 7-segment display circuit using the 360Ω resistors and the 74xx04s on the same protoboard. Both circuits should be correctly wired to the 68HC11 EVBU board. See figures 1 and 2 below for the keypad and 7-segment display pinouts.

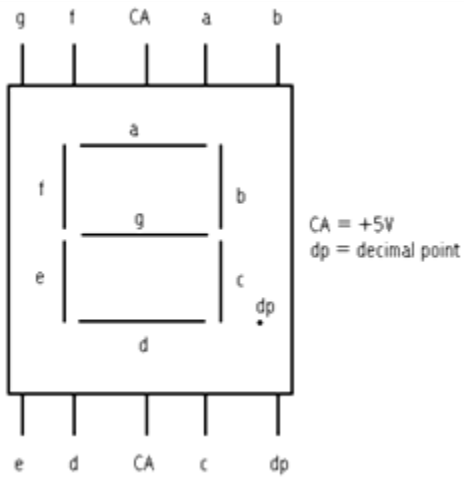


**Figure 1.** Seven-segment LED display pinout.

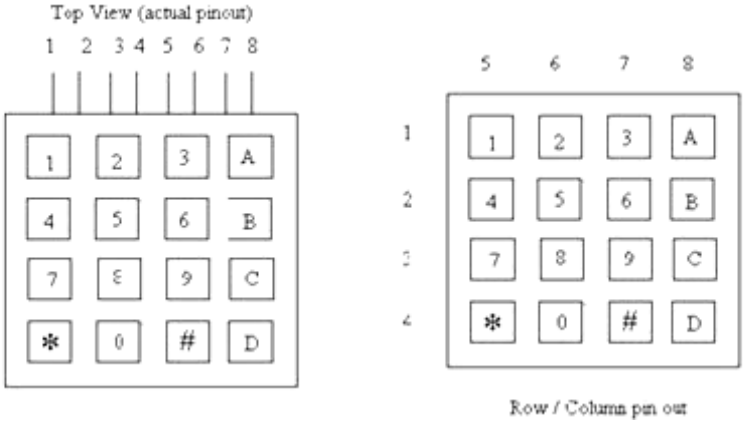


**Figure 2.** Keypad pinout.

## PROCEDURE

1) Make sure to read and understand the material relevant to this lab in chapter 7 of your textbook and the M68HC11 reference.
2) Write a delay subroutine called Delay which delays for 1/3 second.

3) Write a display subroutine called Display which will output a character stored at the location Data+offset (indexed mode addressing) to Port B. Port B is connected to the 7-segment LED through a driver (74xx04).
4) Using the subroutines Delay and Display, write a subroutine called Disp_data to meet Program Requirement 1.
5) Write a subroutine called Scan_key which scans the Keypad. If any key is pressed, identify the row and column of the pressed key.
6) Write a subroutine called Which_key that identifies the key location, which will give the offset address for the code to be sent to Port B (i.e., a number 0 to 15, a total of 16 keys).
7) The Keypad has 4 rows and 4 columns, a total of 16 keys. The 4 columns are connected to Port C pins PC0-PC3 and the 4 rows are connected to Port C pins PC4–PC7. Configure Port C pins PC0–PC3 for output and PC4–PC7 for input. One method that may be used to determine which key is pressed is to output known voltage levels on the columns and then read the row pins to determine which key was pressed. The 10KΩ resistors would be used on the row circuitry. Of course, you will have to design the circuitry and algorithm needed to perform this task.
8) Use the subroutines Scan_key, Which_key, and Display to meet Program Requirement 2.
9) Assemble the program and show your list file to the lab instructor.
10) Draw a neat schematic of all your hardware connections. No hand-drawn schematics, please.
11) Download the assembly-error-free program to the EVBU board.
12) Debug the program using BUFFALO.
13) Run the debugged program. (The lab instructor will verify that your program works).

## NOTES

1) As the code segment is going to be large use EEPROM (addresses $B600–$B7FF) for main program and subroutines.
2) Complete program requirement 1 first, this will familiarize you with using a port. Then, complete Program Requirement 2.
3) When you write to Port C, the data is latched.
4) Make your circuits as neat as possible. It may take you longer to build, but you will save lots of time when troubleshooting.
5) The * and # symbols cannot be displayed on the 7-segment display. Display '-' (the middle segment) when the * is pressed and 'P' (for pound) when the # key is pressed.