

ASSEMBLY LANGUAGE PROGRAMMING AND 9S12 PORTS (PART 3)

Introduction and Objectives

Pre-Lab

We want to write a program which will display four different patterns on the seven segment LED display connected to Port H. We want to use two input lines to select which of the four patterns to display. To do this, we will use bits 1 and 0 of Port B.

Write a program to set up Port H as an 8-bit output port, and to implement (i) a binary up counter, (ii) a decimal down counter, (iii) a flasher, (iv) a message scrolling banner. For the binary up counter, just have Port H count 0, 1, 2, 3, 4 ... This will generate a random looking pattern on the LEDs. It will take 128 counts from the time all LEDs are off until the next time all are off. Samples from the other sequences that you should generate are shown in Fig. 1. Include an appropriate delay (about 250 ms) between changing the pattern so that you can easily and comfortably see them flash. Use a subroutine to implement the delay.

Set up Port B as an input port, and use bits 1 and 0 to control which of the Port H functions are performed as shown in Fig. 2. You will need to connect DIP switches to bits 1 and 0 of Port B. (Look at the pin out diagram for the 112 pin chip on page 54 of the [MC9S12 Core Users Guide](#) to determine which pins to use.)

When you switch between functions, the new function should start up where it ended when it was last activated, so set aside variables to save the states of the various patterns.

Start writing the program before coming to lab. You should at least have an outline (or a flowchart) of the flow control of the program, and pseudocode for how you plan to implement your functions. Be sure to write the program using structured, easy-to-read code.