

# EECE 5141C/6041C

## Introduction to Mechatronics

### Lab 1: Basic I/O

Name: \_\_\_\_\_

In this lab, you will design, program, and construct a simple microcontroller system capable of monitoring four inputs produced by pull-up switch circuits and then displaying the hexadecimal representation of the binary input on a seven-segment display in real-time. This circuit will form the core component for your pinball machines score display.

#### DELIVERABLES

1. Completed Worksheet
2. Annotated Source Code

#### SYSTEM SPECIFICATIONS

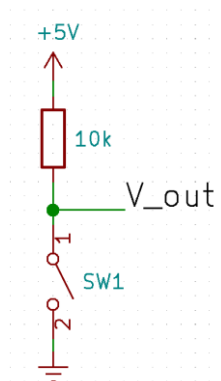


Figure 1: Pull-Up Switch Circuit

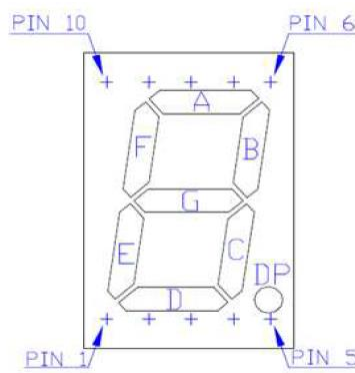


Figure 2: Seven-Segment Pinout

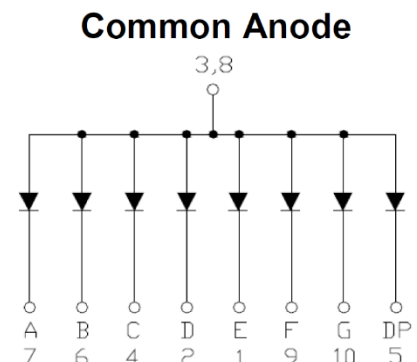


Figure 3: Seven-Segment Schematic

The microcontroller must monitor four digital inputs supplied by four independent pull-up resistor, switch circuits. A schematic of this switch circuit is shown in Figure . The microcontroller will then read this binary value and display the hexadecimal equivalent value on a common anode seven-segment display. The pinout diagram for the seven-segment display is provided in Figure 2, and the internal schematic for the is provided in Figure 3. Additional information can be found in the datasheet uploaded to Catalyst. Be sure to include current limiting resistors in series with each segment LED when interfacing the microcontroller to the seven-segment display. Select a resistor value such that the current is limited to 10mA per segment.

## SYSTEM DIAGRAM

---

Draw a system diagram for your implemented system. Be sure to label the microcontroller pins (PB0, PD1, etc.).

## DESIGN QUESTIONS

---

1. Suppose that your inputs were split between ports. For example, you had to use PD1, PD0, PB1, and PB0 as the input pins with PD1 being the most significant bit and PB0 being the least significant bit. How could you read these pins and combine the information into one variable such that binary values had a range from 0 to 15? For example, inputs PD1=1, PD0=0, PB1=0, and PB0=1 would result in a value of 9 stored in a variable called *input*. Write out the C-code that reads the pins values, manipulates those values, and stores the result in *input*.
2. Suppose you were provided common cathode seven-segment displays instead of common anode. How would this change your hardware (draw a circuit diagram) and software?

3. Suppose you needed to drive three seven-segment displays. Would this be possible with a single microcontroller? How?