EECE 5141C/6041C
Introduction to Mechatronics
Lab 2: Time Multiplexed I/O

Name: ___________________________

In this lab, you will design, program, and construct a simple microcontroller system capable of monitoring eight digital inputs and displaying their hexadecimal equivalent representation on two seven segment displays.

DELIVERABLES

1. Completed Worksheet
2. Annotated Source Code

SYSTEM DESCRIPTION

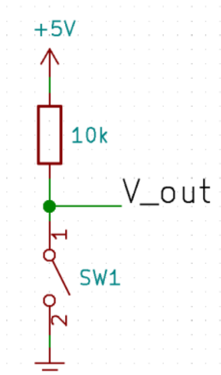

Figure 1: Pull-Up Switch Circuit

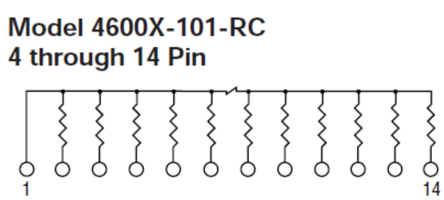

Figure 2: SIP Resistor Array

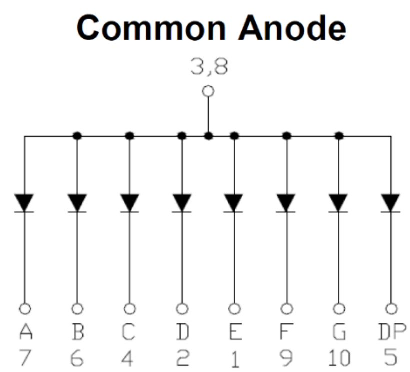

Figure 3: Seven-Segment Schematic

The microcontroller must monitor eight digital inputs supplied by eight independent pull-up switch circuits. A schematic of this switch circuit is shown in Figure . It is recommended that these switch circuits are constructed using the provided DIP (Dual Inline Package) switches and SIP (Single Inline Package) resistor array. The internal schematic of the SIP resistor array is shown in Figure 2. Pin 1 is indicated by the dot on the physical package.

In the previous lab, the seven-segment display was driven directly by the microcontroller I/O pins. However, we will now need two hexadecimal digits to display the 8-bit input value. This would require fourteen separate I/O lines for each of the individual segments. Instead of driving each segment individually, we will time multiplex the outputs for each digit's set of segments and turn on each digit in sequence. Therefore, we will only need nine outputs (2 digit control signals + 7 segment control signals). We will utilize Timer 1 to generate interrupts at regular intervals in order to update the segment signals as well as the activating the appropriate digit. A schematic diagram of the LED matrix is shown in Figure 4. We control each digit by applying or removing 5V to the common anode of the 7-segment display. Although we could attempt to use one of the general I/O pins to drive each digit, that pin would need to source the total current for all seven segments. This would likely exceed the current limitations of the pin. Therefore, we will utilize a P-channel MOSFET to drive the anode of each display. A "high" signal turns off the P-channel MOSFET, and a "low" signal turns the MOSFET on. Since each digit will only be driven 50% of the total period, we may need to pulse the LEDs with higher current while activated in order to

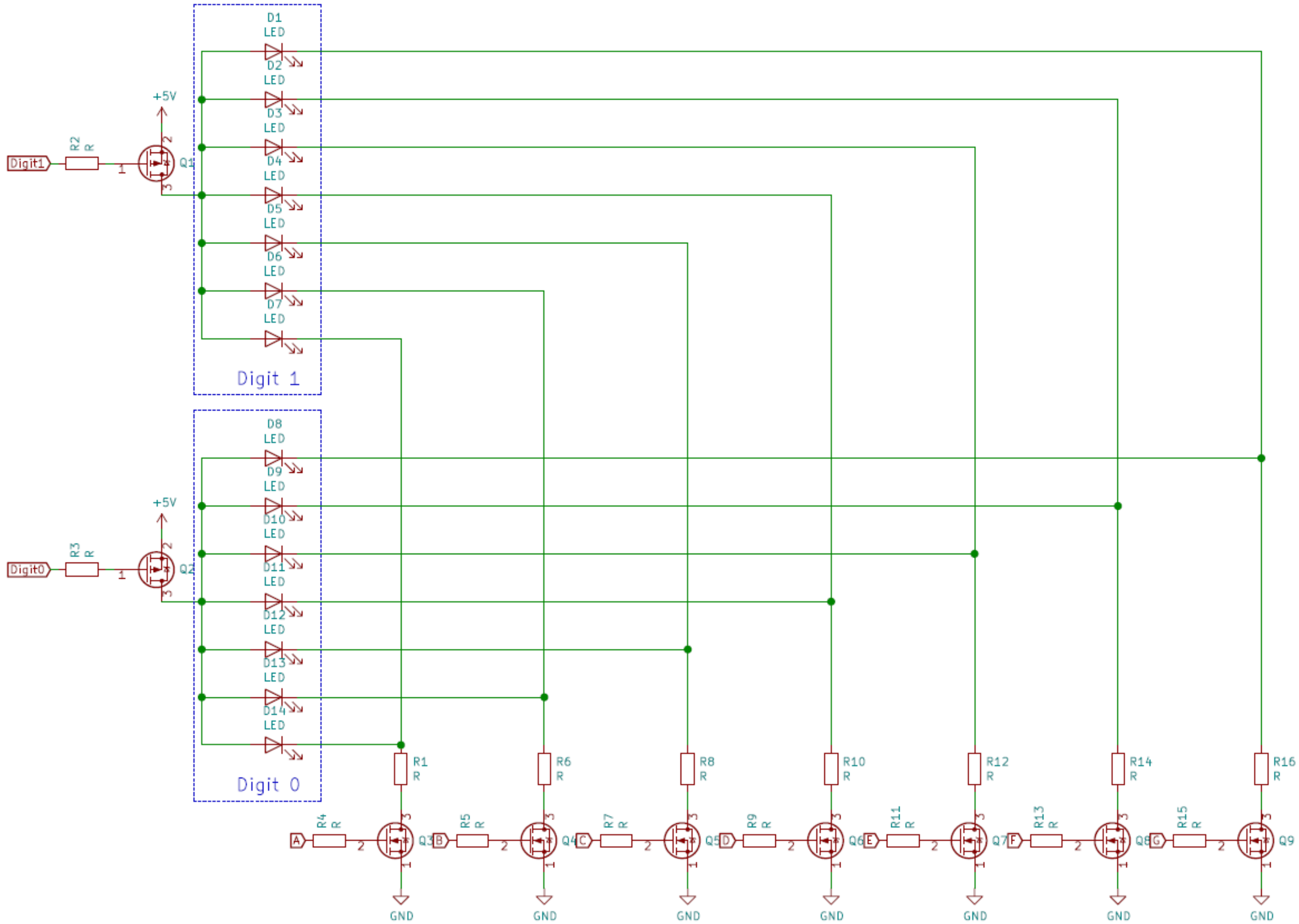

Figure 4: LED Matrix

maintain a suitable brightness level. This higher current level may exceed the current limitations of the I/O pins of the microcontroller. Therefore, we will drive each segment with an N-channel MOSFET that is controlled by applying a high or low signal to the MOSFET gate. A “high” signal turns on the MOSFET and a “low” signal turns off the MOSFET. Both the N-channel and P-channel MOSFETs have a series resistor connected to their gates in order to limit in rush current. For this particular application, the value is not critical and any value that is “relatively low” will be okay. For example, the 330 Ohm in your kits will be okay, but any lower value will work as well. We will discuss MOSFET drivers in more detail later in the course. You will also need to include current limiting resistors in series with each of the N-channel MOSFETs. You can use your 330 Ohm resistors for these resistors as well.

In order to cycle through activation of each segment, we will utilize Timer 1 in CTC mode to generate interrupts at a defined interval. Within the *Timer1_CompA* ISR, the program will read the eight switch states, activate the next digit of the display, and update the segment control signals accordingly. This process will repeat each time Timer 1 generates a match of the output compare register A (OCR1A). For this lab, calculate the appropriate clock source and match register values for periods of 1 s, 250 ms, 100 ms, 30 ms, and 1 ms. Record these values in the design question section below. You will also implement these values and record your observations in the design question section.

SYSTEM DIAGRAM

Draw a system diagram for your implemented system. Be sure to label the microcontroller pins (PB0, PD1, etc.). You do not need to redraw the LED matrix components; simply label the pins of the microcontroller which connect to the labels (Digit1, Digit0, A, B, C, D, E, F, and G) in Figure 4.

DESIGN QUESTIONS

1. Calculate clock source and match register values for each of the following interrupt periods.
    a. 1 s
    b. 250 ms
    c. 100 ms
    d. 30 ms
    e. 1 ms

2. Implement each of the interrupt periods from Question 1 and record your observations below. In particular, was their noticeable blinking or blurring of the digits?

3. Why do we place the resistor at the drain of the N-channel MOSFET? Why don’t we place resistors in series with the P-channel MOSFETs instead? After all, we would only need to resistors then instead of seven.

4. Suppose you needed to drive additional seven-segment displays. What modifications would need to be made to the hardware and software?