EGR 262

Fundamental Circuits Lab

File: N262L3

**Lab # 3**

**Switches and 7-Segment Displays**

A. **Objectives**

The objectives of this laboratory are to introduce the student to:

1. Current-limiting resistors and pull-down resistors
2. Switch debouncing
3. Active-LOW versus active-HIGH outputs
4. 7-segment displays (common anode and common cathode)
5. Writing Arduino programs to read pushbutton switches and control 7-segment displays

B. **Materials**

Breadboard with Arduino UNO

Adaptor (120V AC to 12 VDC, 1000 mA)

Seven 220 Ω resistors

10 kΩ resistor

Common-anode 7-segment display (GNS 3011 or similar)

Momentary pushbutton switch (BTS-1102B-2 or similar)

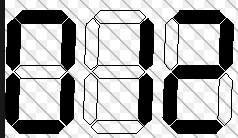
C. **Introduction**

Circuit 1 below will be constructed and tested in lab using programs written for the Arduino UNO. See the ***Presentation for Lab #3*** for more detailed background information.

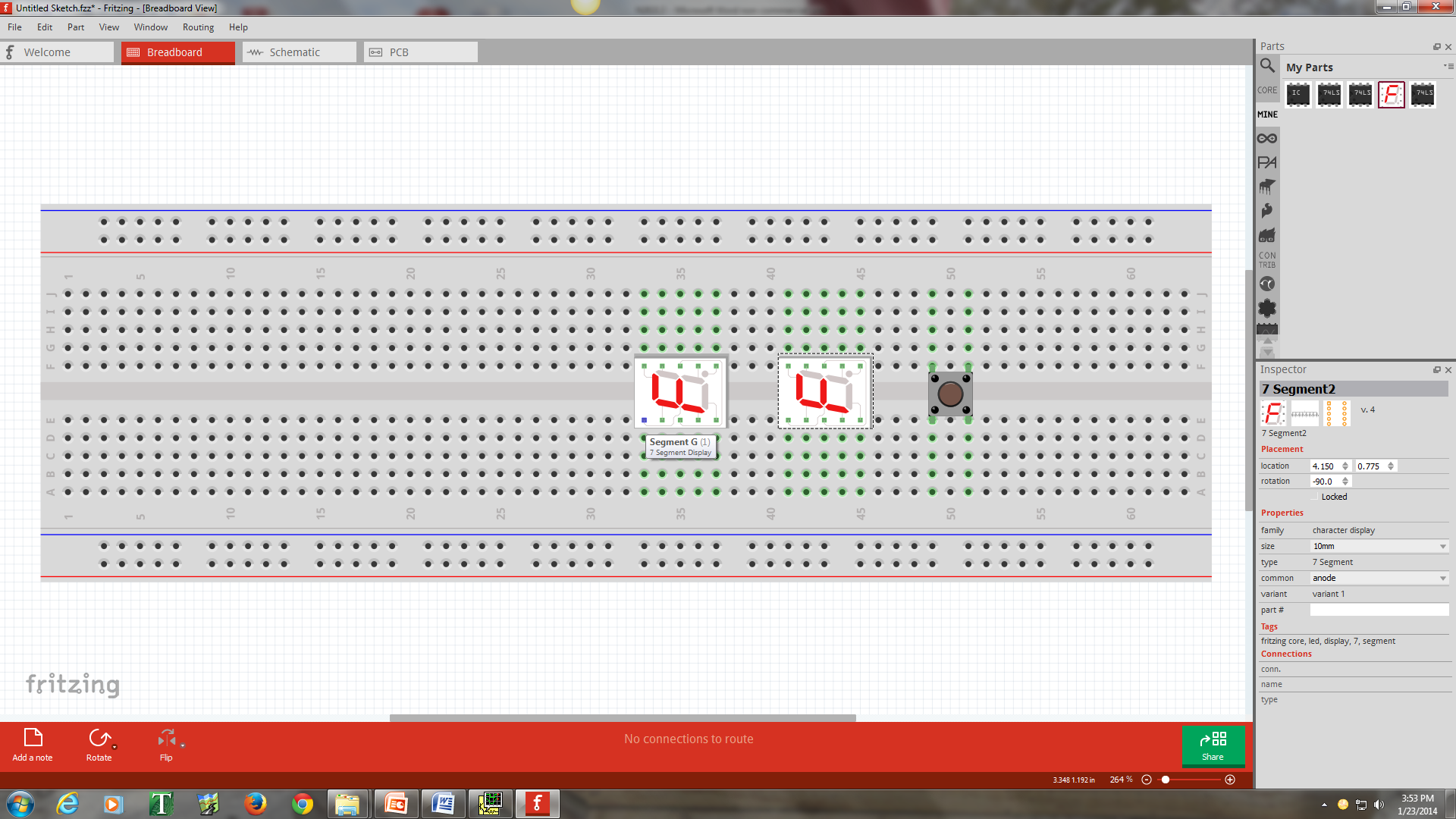
Circuit 1

D. **Pre-Lab Tasks**

1. Show the pinout for the GNS-3011 common anode 7-segment display.
2. Explain the difference between common anode and common cathode 7-segment displays.
3. Show a truth table for a common-anode display (see related lecture notes).
4. Show a diagram illustrating which segments are lit for each BCD input. For example:



1. Show the pinout for the BTS- 1102B-2 pushbutton switch. Which pins are connected when the button is pushed? Which pins are always connected together?
2. Use ***Fritzing*** to draw a breadboard layout and schematic for Circuit 1.
   1. Reference: See the file ***FritzingEGR262*** on the course website.
   2. Follow the **Basic Rules for Breadboards Produced for EGR 262 labs** listed in reference above.
   3. Begin with the standard EGR 262 breadboard layout in the file ***EGR262Breadboard.fzz*** (available on the course website).
   4. Use the 4-pin button switch shown below for the switch. It is similar to the BTS-1102B-2.
   5. The 7-segment display in Fritzing has a slightly different pinout than the GNS-3011. You can use the 7-segment display in Fritzing (and ignore the pin labels) or edit the part or import the part GNS-3011 from the course website. In any case, wire the 7-segment display in Fritzing using the pinout for the GNS-3011 as you will in lab.
   6. Be sure to include a current-limiting resistor each each LED segment and a pull-down resistor for the switch.



* 1. Save your breadboard layout and schematic for Lab 3 (.fzz file)
  2. Export the breadboard as a jpg image to insert in your lab report.
  3. Export the schematic as a jpg image to insert in your lab report.

1. Write a C++ program for the Arduino UNO as described below. Be sure to include an initial block of comments at the start of each program as well as comments throughout. Compile each program before lab and correct any errors that occur. Include a copy of each program in the Pre-Lab section of the report (if changes are made during lab, the final version will be included in the Post-Lab section).
   * Display brief instructions on the computer screen.
   * The count should be initialized to 0 and shown on the 7-segment display and the computer screen.
   * When the user presses the button the count should advance and the new value should be displayed on the 7-segment display and the computer screen.
   * The count should be mod-10 (0 to 9 and repeat).
   * Use the ***display\_digit( )*** function described presentation for this lab.
   * Use the ***readButton( )*** function provided in the presentation for this lab to allow for switch debounce.
   * See the following page for more details on the code.

**// Global variables for readButton( ) function (see Lab #3 Lecture)**

**// Other global variables**

**void setup( )**

**{ // Define input and output pins**

**// Set up serial communication**

**// Display instructions on the computer screen**

**// Display initial value (0) on 7-segment display**

**// Display initial value (0) on computer screen**

**}**

**void loop( )**

**{ // If button is pressed:**

**// Increment counter**

**// Adjust counter for mod-10 operation**

**// Display count on computer screen**

**// Display count on 7-segment display**

**}**

**void readButton(int buttonPin)**

**{**

**// Use code provided in Lab #3 Lecture**

**}**

**void display\_digit(int N)**

**{**

**// See Lab #3 Lecture for details**

**}**

E. **In-Lab Tasks**

1. **Build Circuit 1**
   * The circuit should match your Fritzing breadboard layout.
   * Build the circuit somewhat compactly and close to the Arduino to leave plenty of room on the breadboard for later circuits. The 7-segment display circuit will be used in several labs, so do not remove it after the lab.
   * Use neat wiring practices.
2. **Testing the Program**
   * Download and run the program for Lab 3.
   * Once the program is running correctly, use PrintScreen to obtain a copy of the output to the computer screen and include it in the report.
   * Demonstrate the program to the instructor once it is working correctly and also record the results (state specifically what was observed in lab).
   * If any changes were made to the program, save the program under a different name (L3v2, perhaps) and include a copy in the Post-Lab section.
   * Record comments for any problems encountered or lessons learned.
3. **Testing Program without allowing for switch bounce**
   * Modify your program to ignore switch bounce by removing the readButton( ) function and related global variables. Also replace ***if (readButton(ButtonPin == 1)*** with

***if (digitalRead(ButtonPin) == 1)***

* + Test the program above. You should find that when you press the input switch the 7-segment display advances several counts.
  + Press the button 10 times and record how many numbers the display advanced each time. Also calculate the average number of bounces. You can see this more clearly by watching the computer output from the Arduino.
  + Demonstrate the program to the instructor .
  + Include a copy of the program in the Post-Lab section.
  + Record comments for any problems encountered or lessons learned.

F. **Post-Lab Tasks**

1. Discuss the performance of the original program (which allowed for switch bounce).
2. Discuss the performance of the modified program (which ignored switch bounce).
3. Explain (in your own words) the meaning of “switch bounce.” Why is this a problem when using input switches with microprocessors?
4. Explain (in your own words) the reason for using current-limiting resistors.
5. Explain (in your own words) the reason for using a pull-down resistor with a pushbutton switch.

G. **Report**

A lab report is due 1 week after the date of the experiment.

* The lab report must be your own work. Copying data, tables, graphs, circuits, etc., from other students is not allowed and will result in grades of 0 on the lab.
* Be sure to follow good practices for presenting all tables and graphs. See the Presentation for Lab #1 for examples.
* The lab report should consist of the following sections:

1. Title Page (include course number & title, lab number & title, date, and your name)
2. Pre-Lab Tasks
   * Include instructions or headings for all items to make the report easy to follow.
3. In-Lab Tasks

* Include instructions or headings for all items to make the report easy to follow.
* Be sure to include comments from lab as well as measured data.

1. Post-Lab Tasks

* Include instructions or headings for all items to make the report easy to follow.