EGR 262

Fundamental Circuits Lab

File: N262L4

**Lab # 4**

**Digital-to-Analog Conversion**

A. **Objectives**

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

1. Digital to analog conversion using an R-2R ladder network

B. **Materials**

Breadboard with Arduino UNO

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

Seven 220 Ω resistors

Ten 2.2 kΩ resistors

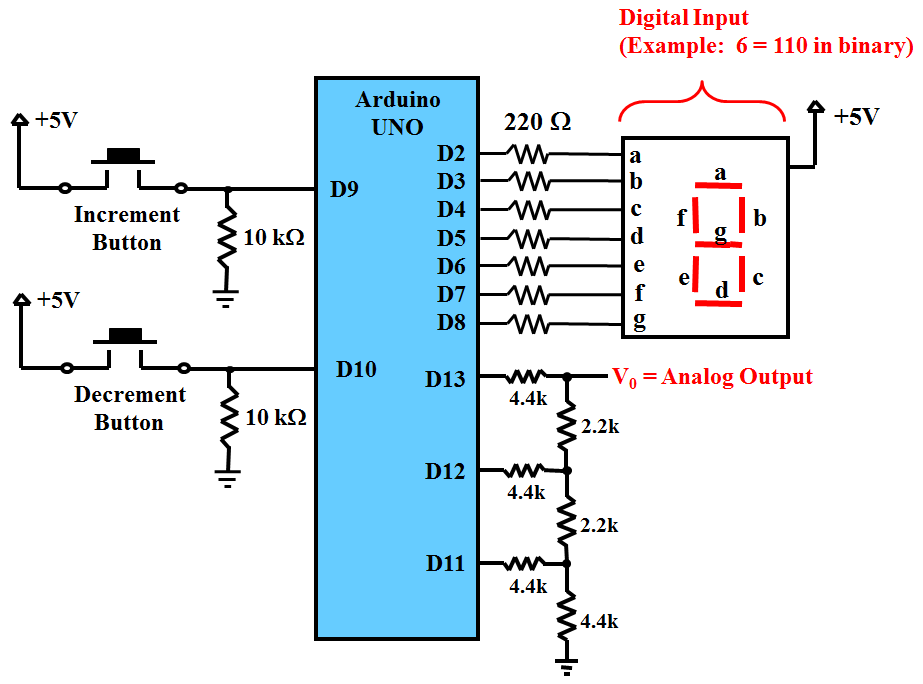
Two 10 kΩ resistors

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

Two momentary pushbutton switches (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 #4*** for more detailed background information.



Circuit 1

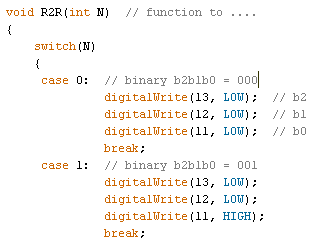
D. **Pre-Lab Tasks**

1. Use ***Fritzing*** to draw a breadboard layout and schematic for Circuit 1.
   * You should be able to simply add on to your Lab #3 breadboard layout.
   * Save your breadboard layout and schematic for Lab 4 (.fzz file)
   * Export the breadboard as a jpg image to insert in your lab report.
   * Export the schematic as a jpg image to insert in your lab report.
2. Perform a ***detailed analysis*** of the 3-bit R-2R ladder network to be used in the lab and shown below.

Use superposition to analyze the circuit as shown in the lab presentation. In particular:

* + Analyze the circuit with b2 = 5V, b1 = 0V, and b0 = 0V. Show all details of the analysis.
  + Analyze the circuit with b2 = 0V, b1 = 5V, and b0 = 0V. Show all details of the analysis.
  + Analyze the circuit with b2 = 0V, b1 = 0V, and b0 = 5V. Show all details of the analysis.
  + Create a table of analog outputs versus digital inputs for all possible combinations of b2, b1, and b0 (see slide 14)
  + Use Excel to form a stairstep graph of Analog Outputs versus Digital Inputs (see slide 16)

1. What is the resolution (as a voltage) of the 3-bit DAC to be used in lab? What is the resolution as a percentage?
2. Write a function ***R2R( )*** to control the R-2R DAC as follows:
   * Input: Count (should have value 0 to 7 represention binary 000 to 111)
   * Outputs: D13, D12, and D11 should be set HIGH or LOW according to the binary value of the input.
   * The function might look as follows:



1. Write a C++ program for Lab #4 as follows: to control the R-2R DAC as follows:
   * This program builds on the program for Lab #3, so begin with that program.
   * Display brief instructions on the computer screen.
   * A second button switch will be added. One will be used to increment the counter and one to decrement the counter.
   * The count should be initialized to 0 and shown on the 7-segment display and the computer screen.
   * The count should be mod-8.
   * When the user presses a button:
     + The program should determine which button was pressed.
     + The count should increment or decrement based on the button pressed.
     + Two versions of the readButton( ) function will be needed with different variable names in each function (except for local variables): one for incrementing and one for decrementing. This is necessary because the readButton( ) function uses the last button state, last time the function was called, etc.
     + The count should be displayed on the 7-segment display. Do not make any changes to the ***display\_digit( )*** function from Lab #3.
     + The count (in decimal and binary form) should be displayed on the computer screen.

Hint: ***Serial.print(78, BIN)*** gives "1001110" (it is OK that it doesn’t show leading zeros)

* + - The function ***R2R( )*** should be called to set or clear the appropriate digital pins
  + The program might look something like the following:

**// Initial block of comments**

**// Global variables for incReadButton( ) function**

**// Global variables for decReadButton( ) function**

**// 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**

**// Call R2R( ) to send initial value (000) to output pins**

**}**

**void loop( )**

**{ // if(incReadButton(incrementPin) == 1 && decReadButton(decrementPin) = 00)**

**{**

**// Increment counter**

**// Adjust counter for mod-8 operation**

**// Display count on computer screen**

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

**// Call R2R( ) to send count to output pins**

**}**

**elseif( ……..)**

**}**

**void incReadButton(int buttonPin)**

**{**

**// Modified code from Lab #3 (incReadButton( ) and decReadButton( ) must use different variable names)**

**}**

**void decReadButton(int buttonPin)**

**{**

**// Modified code from Lab #3 (incReadButton( ) and decReadButton( ) must use different variable names)**

**}**

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

**{**

**// Use code from Lab #3 (do not make any changes)**

**}**

**void R2R(int N)**

**{**

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

**}**

E. **In-Lab Tasks**

1. **Build Circuit 1**
   * Most of the circuit should already be in place from Lab #3. Add the additional parts.
   * The circuit should match your Fritzing breadboard layout.
   * Use neat wiring practices.
2. **Testing the Program**
   * Download and run the program for Lab 4.
   * Verify that the counter properly increments and decrements (mod-8) both on the 7-segment display and on the computer screen.
   * Connect a voltmeter to output V0 on the R-2R ladder network. Measure and record the voltage for all 8 possible inputs. Compare the voltages to the predicted analog voltages in the Pre-Lab section. If they are not close, check the circuit and/or program for errors.
   * 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 .
   * If any changes were made to the program, save the program under a different name (L4v2, perhaps) and include a copy in the Post-Lab section.
   * Record comments for any problems encountered or lessons learned.

F. **Post-Lab Tasks**

1. Discuss the performance of the program.
2. Compare the measured analog outputs with the predicted analog outputs for each digital (binary) input. Include % error.
3. If you wanted to replace the 3-bit R-2R ladder network with a 10-bit R-2R ladder network:
   1. How many digital output pins would be needed for the R-2R circuit?
   2. How many 2.2 kΩ resistors would be needed?
   3. How many different analong voltages would be available?
   4. What would be the increment between voltages (the resolution as a voltage)?

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.