

ECEN 215 Lab Manual

Wei Trinh

In Collaboration With

John Tyler
Samuel Villareal
Mina Rahimian
Nathan Gober

Contents

Acknowledgments	iii
How To Use This Manual	iii
Lab Report Guidelines	iii
Notation Conventions	iii
I Lab Assignments	1
1 Ohms Law and Kirchoffs Laws	2
1.1 PreLab Calculations	2
1.2 MultiSim Simulations	2
1.3 Laboratory Procedure	3
1.4 Additional Lab Measurements	3
2 Lab # 2: Node Voltages and Equivalent Circuits	5
2.1 PreLab Calculations	5
2.2 MultiSim Simulations	6
2.3 Laboratory Procedure	6
3 Lab # 3: Introduction to Op Amps	7
3.1 PreLab Calculations	7
3.2 MultiSim Simulations	7
3.3 Laboratory Procedures	8
4 Lab # 4: First Order RL and RC Transients	9
4.1 PreLab Calculations	9
4.2 MultiSim Simulations	9
4.3 Laboratory Procedures	10
5 Lab # 5: Introduction to AC Signals	12
5.1 PreLab Calculations	12
5.2 MultiSim Simulations	12
5.3 Laboratory Procedure	13
6 Lab # 6: Frequency Response	14
6.1 PreLab Calculations	14
6.2 MultiSim Simulations	14
6.3 Laboratory Procedure	15
7 Lab # 7: Operational Amplifier Integrator and Active Filter	17
7.1 PreLab Calculations	17
7.2 MultiSim Simulations	17
7.3 Laboratory Procedures	18
8 Lab # 8: Introduction to Logic Gates	19
8.1 PreLab Calculations	19
8.2 MultiSim Simulations	19
8.3 Laboratory Procedure	20

9 Lab # 9: Introduction to Microcontrollers	21
9.1 PreLab Calculations	21
9.2 MultiSim Simulations	22
9.3 Laboratory Procedure	22
II Reference Materials	27
A Appendix A	28
A.1 Main Window	28
A.2 Components Toolbar	28
A.3 Measurement Toolbar	29
A.4 AC and DC Circuits	30
A.5 Making Measurements	33
A.6 Presentation of Results	37
B Appendix B: Breadboard Basics	39
B.1 How to use the Breadboard	39
B.2 Passive Components	40
B.3 Operational Amplifiers	44
B.4 Logic Gates	45
B.5 Guidelines for Building Circuits	45
C Appendix C: Waveforms 2015 Basics	47
C.1 Contents of the Analog Discovery 2	47
C.2 Main Window	48
C.3 Power Sources	49
C.4 Making Measurements	51
C.5 Example Circuit Measurements	54
C.6 Digital Multimeter	56
C.7 Presentation of Results	58
D Appendix D: Introduction to Microcontrollers	60
D.1 Computers, Processors, and Microcontrollers	60
D.2 MSP432 LaunchPad - Hardware Overview	61
D.3 Energia - Software Overview	62

Acknowledgments

I would like to first and foremost thank all the faculty and students whom have provided their input and their support towards the final product of this lab manual. To Mr. Tyler, who was the author of the original lab manual, for supporting this idea from day 1, and constantly providing insight and feedback on the entire lab manual as a whole. To Dr. Villareal, who provided the novel lab material for both Labs #9 and #10, several of the appendices, and who provided a myriad of interesting ideas to improve the quality of various labs. To Dr. Rahimian, who continually gave valuable insight on how to communicate ideas better to the students, and how to make the content easily digestible to all. To Nathan Gober, who provided a base file and many other smaller sections in this lab manual, from guidelines to conventions. And finally, to all my friends and family who consistently provided feedback from a third person perspective, and supported me through this endeavor.

How To Use This Manual

This manual is split into two parts: the laboratory assignments and the reference materials. A lab assignment contains three parts: a pre-lab assignment, a simulation exercise, and an experimental procedure. Each part of a lab assignment will help you complete the next part, so it is best if you perform them in order.

In general, the pre-lab assignment will produce an analytical solution for the lab assignment. The simulation will rapidly generate many data points, which will later be compared to the measured results, that should confirm your understanding of the analytical solution. The lab procedure will require you to collect data and present it. The required data collection will be shown in **bold text**. It should, to within a reasonable degree of error, match the predictions of the previous two parts.

In the reference section, you will find guides and datasheets describing how to use the laboratory equipment. These documents are provided to equip you to complete the assignments easily. You may wish to read through the reference section before the first lab to familiarize yourself with it and make later references faster. However, any time the lab manual references a certain appendix, figure, or table, you may click on the associated number, and it will bring you to the page with the relevant information. ie: Clicking on the A in "Appendix A" will bring you to the start of the associated appendix.

Lab Report Guidelines

- Mark all materials that you turn in with your name (printed legibly) and section.
- Pre-lab calculations, analyses, and simulations are individual work and are due at the start of the lab period for that lab. Keep a copy of your results for reference during the lab.
- Simulation printouts are required to document your simulations.
- Lab reports are due at the time specified by your lab instructor. Due dates are typically one week after the date of the lab period.
- Use a professional, technical writing style in your reports. Support all claims with observations and data.

Notation Conventions

In general, this manual will adhere to widely accepted conventions for naming quantities and drawing circuit diagrams. You are encouraged to follow these conventions in your own work. If a circuit diagram deviates from these conventions, follow its conventions where they conflict with these.

- Wherever possible, quantities with identical units are given the same letter, and are distinguished by a subscript (e.g. R_1 , R_2). When only one such quantity is present, it is generally given without a subscript. In general,

- V denotes a voltage,
 - I denotes a current,
 - R denotes a resistance,
 - C denotes a capacitance, and
 - L denotes an inductance.
- Time-constant quantities are notated with capital letters (e.g. R , C).
 - Time-variant quantities are notated with lowercase letters (e.g. i , v). Where significant, they may also be notated as functions of time, like $i(t)$.
 - The voltage across a component and the current through it will share a subscript with the component's name (that is, V_1 is the voltage across C_1 , and $i_1(t)$ is the current through it). If the component's name does not have a subscript, the name is used instead (thus, V_S is the voltage across source S).
 - Current is assumed to flow from high voltages to low voltages (sometimes known as positive current flow).
 - Time-constant voltages are assumed to be greater than zero.

The following physical constants are also used in this manual:

- $\pi \approx 3.1416$ is the ratio of a circle's circumference to its diameter.
- $j = \sqrt{-1}$ is the imaginary unit.
- $e \approx 2.7183$ is the base of the natural logarithm.

Part I

Lab Assignments

1 Ohms Law and Kirchoffs Laws

1.1 PreLab Calculations. Due at the beginning of lab.

Do not use MultiSim for this part. Read through the entire lab before the start of the lab period.

- A. Read through the entire lab before the start of the lab period.
- B. By hand, using Kirchoff's and Ohm's Laws, calculate all indicated quantities in the circuits of Figures 1.1a- 1.1d.

1.2 MultiSim Simulations. Due at the beginning of lab.

Refer to Appendix C for information on how use MultiSim and how to submit your results.

- A. Use MultiSim to simulate each of the circuits in Figures 1.1a- 1.1d. A separate meter should be used for each simulated measurement.
- B. Print out your schematics showing all the meter's readings.
- C. Tabulate your results, comparing theoretical predictions from Section 1.1. Your tabulated results should numerically demonstrate the validity of Ohm's Law, KVL, KCL, and the voltage divider.
- D. Write a sentence for each figure describing how each demonstrates a specific circuit principle. Refer to Ohm's Law for Figure 1.1a, KCL for Figure 1.1b, KVL for Figure 1.1c, and voltage dividers for Figure 1.1d.

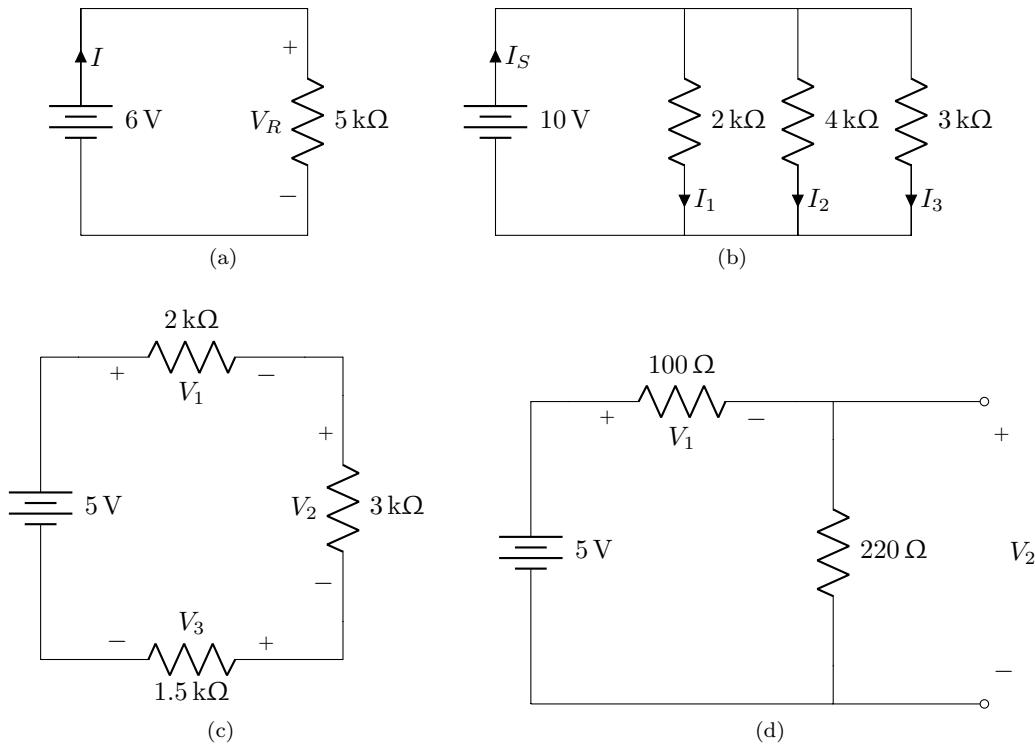


Figure 1.1: Lab #1 Figures

1.3 Laboratory Procedure

- A. In your component kit, find the two resistors you will need to construct Figure 1.1d. Use the battery-powered Digital Multimeter (DMM) to measure both resistor values, and refer to Appendix C.6.1 for information on how to measure resistance. They are nominally $100\ \Omega$ and $220\ \Omega$, but may vary. **Record the resistor values.**
- B. Build the circuit seen in Figure 1.1d. Use the positive power supply provided by the AD2. Refer to Appendix C.3.1 for more information on how to use the power supply from the AD2.
- C. Measure V_S using the AD2. Note that V_S is the nominal 5 V power supply on the circuit board, but it, like the resistors, has a tolerance. **Be sure to record the voltage.**

NOTE: From now on, whenever you are using a voltage source, be sure to measure it, as you cannot assume its value is exactly equal to its nominal value. Refer to Appendix C.3.1 for information on how to properly measure voltage.
- D. Use your measured values of R_1 , R_2 , and V_S to calculate V_1 and V_2 as you did in the pre-lab assignment. **Record the calculated values of V_1 and V_2 .**
- E. Measure V_1 and V_2 using the AD2. **Record the measured values of V_1 and V_2 .**
- F. Compare your calculated voltage values from 1.3D with your measured values from 1.3E. **Calculate a percent difference for both V_1 and V_2 .**

1.4 Additional Lab Measurements

1.4.1 Photoresistor Measurements

- A. What is the measured resistance of the photoresistor when exposed to ambient light? Cover the surface of the photoresistor with cardstock at about .5 inches above the surface. What is the resistance now? What about when you shine a flashlight onto the surface? **Record these results in Table 1.**

Table 1: Resistance of the Photoresistor

Light Level	Resistance (Ω)
Covered with Cardstock (Dark)	
Ambient Light (Neutral)	
Flashlight (High)	

- B. Using the results from Table 1, **Give a one to two sentence statement on what happens to the resistance as the light level changes.**
- C. Modify the circuit in Figure 1.1d by replacing R_1 with a $10\text{ k}\Omega$ resistor, and replace R_2 with the photoresistor. Repeat the procedure from 1.4.1A, but measure the voltage across the photoresistor instead. **Record these results in Table 2.**

Table 2: Voltage across the Photoresistor

Light Level	Voltage (V)
Covered with Cardstock (Dark)	
Ambient Light (Neutral)	
Flashlight (High)	

- D. Again using the results from Table 2, **give a one to two sentence statement on what happens to the voltage as the light level changes.**

1.4.2 Thermistor Measurements

- A. What is the measured resistance of the thermistor when exposed to room temperature? Rub the thermistor between your palms to generate some heat, and measure the resistance. **Record your results in Table 3.**

Table 3: Resistance of the Thermistor

Temperature	Resistance (Ω)
Ambient Temperature (Neutral)	
Rubbed Between Palms (Warm)	

- B. Using the results from Table 3, **give a one or two sentence statement on what happens to the resistance as the temperature changes.**
- C. Modify the circuit in Figure 1.1d by replacing R_1 with a $10\text{k}\Omega$ resistor, and replace R_2 with the thermistor. Repeat the procedure from 1.4.2A, but measure the voltage across the thermistor instead. **Record your results in Table 4.**

Table 4: Voltage across the Thermistor

Temperature	Voltage (V)
Ambient Temperature (Neutral)	
Rubbed Between Palms (Warm)	

- D. Again using your results from Table 4, **give a one to two sentence statement on what happens to the voltage as the temperature changes.**
- E. This circuit, containing either of the variable resistors, could be included as the first stage in the design of a control instrument. **Write a brief (two or three sentence) description of a practical device which would make use of such a circuit.**

2 Lab # 2: Node Voltages and Equivalent Circuits

Refer to the circuit seen in Figure 2.1. Resistor values have been selected from available components. Nominal values should be used for calculations and simulations, but as soon as measured values are available, they should be used to improve comparisons among calculations, simulations and measurements.

Nominal	Measured
$R_1 = 47 \Omega$	$R_1 = \underline{\hspace{2cm}}$
$R_2 = 47 \Omega$	$R_2 = \underline{\hspace{2cm}}$
$R_3 = 100 \Omega$	$R_3 = \underline{\hspace{2cm}}$
$R_4 = 220 \Omega$	$R_4 = \underline{\hspace{2cm}}$
$R_L = 100 \Omega$	$R_L = \underline{\hspace{2cm}}$
$V_S = 5 \text{ V}$	$V_S = \underline{\hspace{2cm}}$

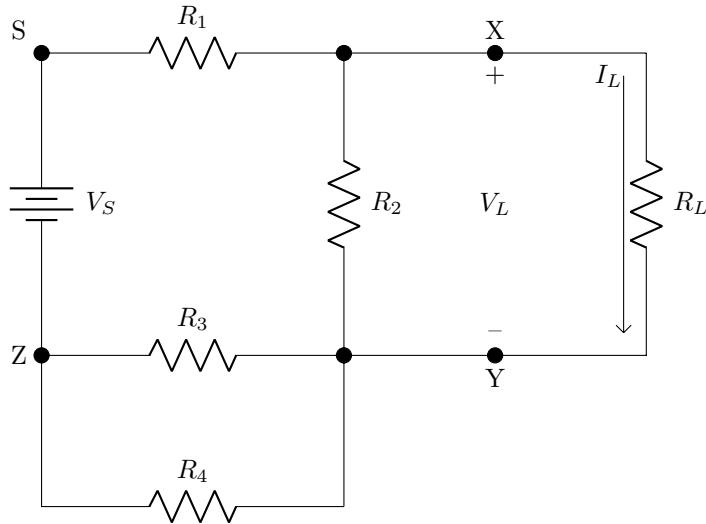


Figure 2.1: Lab #2 Figure

2.1 PreLab Calculations. Due at the beginning of lab.

Do not use MultiSim for this part. Read through the entire lab before the start of the lab period.

- Determine I_L by the node voltage technique. Choose the ground node wisely.
- Determine the Thevenin equivalent circuit at nodes x and y. (Remember to disconnect R_L first.) Use your results to form a Thevenin equivalent network at nodes x and y. What is the Thevenin voltage and resistance?
- Reconnect R_L at nodes x and y, and use the equivalent network to determine I_L . Compare your results with the node voltage method.
- Now consider R_L to be a variable resistor. What value of R_L would be required to achieve the maximum power transfer from the active network at x and y to the load resistor R_L ?

2.2 MultiSim Simulations. Due at the beginning of lab.

- A. Simulate the original circuit using the given values of R_L . Measure I_L with an ammeter. Remember, ammeters are connected in series with the current to be measured.
- B. Define node z to be ground. Using a voltmeter, measure the node voltages at nodes s, x, and y. Tabulate your results. Use the measured node voltages at x and y to calculate I_L .
- C. Now delete R_L . Measure V_{OC} and I_{SC} at nodes x and y (with R_L removed). Use these measurements to design a Norton equivalent network. Build your Norton equivalent network in MultiSim, re-attach R_L , and measure I_L .
- D. Restore the original circuit. Prepare a table of V_L , I_L , and P_L (the load resistor power). Demonstrate that P_L is maximized when $R_L = R_{Th}$. Include at least three values of R_L less than R_{Th} , one value equal to R_{Th} , and at least three values of R_L greater than R_{Th} .

2.3 Laboratory Procedure

- A. Locate all required resistors and measure their resistances with the DMM. Build the original circuit on circuit board using the nominal 5 V power supply for V_S . **Measure I_L with the DMM, and record the value.** Remember that ammeters are connected in series with the current to be measured, with positive current flow into the positive terminal of the meter.
- B. Measure $V_L = V_{xy}$ using the AD2. Calculate the product $I_L \times R_L$, where I_L is the measured current from the previous circuit. **Record both the measured and calculated values, and compare them to each other.**
- C. Now remove R_L and measure V_{OC} and I_{SC} with the necessary tools. **Record the values of V_{OC} and I_{SC} .**
- D. Calculate the Thevenin equivalent circuit at nodes x and y with the measured values of all the components. **Draw the Thevenin equivalent circuit, and use it to calculate I_L .**
- E. Simulate the circuit from Figure 2.1 in Multisim using the measured component values. Measure I_L using an ammeter. **Record I_L .**
- F. Compare your calculated, measured, and simulated load currents I_L . **What is the percentage difference between the calculated, simulated, and measured values?**

3 Lab # 3: Introduction to Op Amps

3.1 PreLab Calculations. Due at the beginning of lab.

Do not use MultiSim for this part. Read through the entire lab before the start of the lab period.

- Review the basics of the ideal op amp model.
- Apply the ideal op amp model to the schematic of the inverting amplifier shown in Figure 3.1. Write the equation relating output voltage to input voltage for this circuit. Use this equation to calculate the output voltage when:

$$V_{in} = .5 \text{ V} \quad R_1 = 1 \text{ k}\Omega \quad R_2 = 5 \text{ k}\Omega$$

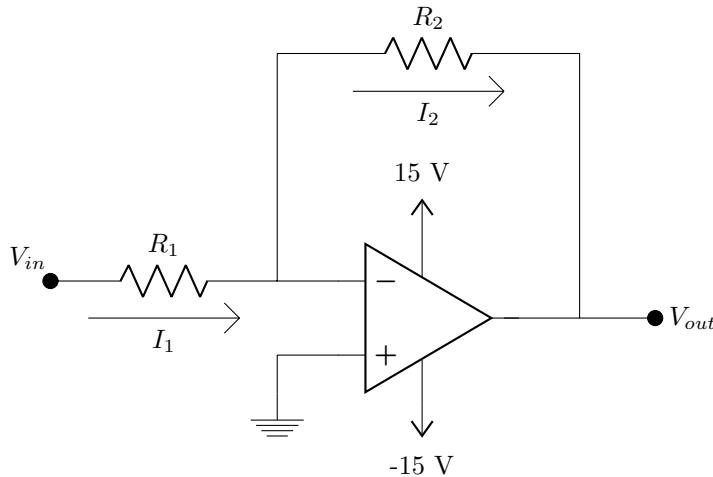
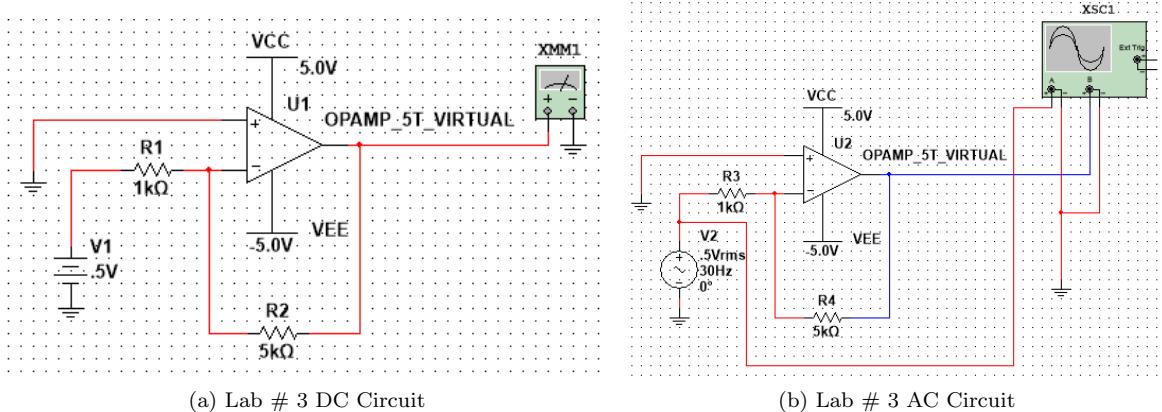


Figure 3.1: Lab # 3 Inverting Amplifier

3.2 MultiSim Simulations. Due at the beginning of lab.



- DC Operation. Simulate the circuit shown in Figure 3.2a. Nominal values of R_1 and R_2 are given in Section 3.1. Use the five terminal virtual op amp as described in Appendix A.4.3, and use the V_{CC} and V_{EE} power sources as your voltage rails. What is the value of V_{out} ? **Be careful of the signs of the voltage inputs.**

- B. AC Operation. Simulate the circuit shown in Figure 3.2b. The AC source has a .5 V magnitude at 30 Hz. Measure the amplitudes of both V_{in} and V_{out} .
- C. AC Operation. Measure the time shift Δt using cursors. **Remember that the time shift is the difference in time between consecutive peaks of two different waves.** Refer to Appendix C.4.3 for further details on how to use cursors. Using this time shift, calculate the phase shift $\Delta\phi$ using the equation below.

$$\Delta\phi = 360 \times \Delta t \times f$$

3.3 Laboratory Procedures

- A. Collect all the resistors you will need to construct the circuit of Figure 3.2a and Figure 3.2b. For R_2 , note that you should use 10 k Ω resistors in parallel, which will make an equivalent resistor of 5 k Ω . See Appendix B.2.1 for further details on how to properly wire parallel components. **Measure and record all resistor values with the DMM.**
- B. Build the DC circuit in Figure 3.2a. The input voltage should still be a nominal .5 V DC source provided by the AD2. Refer to Appendix C.3.2 for information on how to generate a DC signal. **Pay close attention to power supply connections, and ensure that the +5 V and -5 V sources are in the correct locations.**
- C. **Use the AD2 to measure V_{in} and V_{out} . Compare these measurements with Section 3.2.** Refer to Appendix C.4 for information on what are the appropriate tools to use.
- D. What is the scaling factor for the DC measurements? **Report this value and state whether or not it matches what you expect, and why or why not.**
- E. Now construct the AC circuit seen in Figure 3.2b. Note that this circuit is extremely similar to the DC circuit, but you only need to switch the signal from a DC one to an AC signal. **Save the waveforms from the AD2, and present them in your lab report. Refer to Appendix C.7.2 for information on how to submit graphs.**
- F. Using the cursors on the oscilloscope, **measure the amplitude of V_{in} and V_{out} . Using these results, what is the scaling factor for the AC measurements? Does this match what you expect?**
- G. Using the cursors on the oscilloscope, **measure the time shift Δt , and use the same procedure as 3.2C to calculate the phase shift $\Delta\phi$.**

4 Lab # 4: First Order RC and RL Transients

4.1 PreLab Calculations. Due at the beginning of lab.

A. For the circuit in Figure 4.1:

- Use KVL and the linear models to set up the differential equation for $i(t)$.
- Solve the differential equation for $i(t)$.
- Use this result to calculate $v_c(t)$, the voltage across the capacitor for $t > 0$.

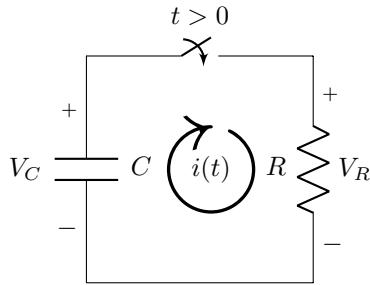


Figure 4.1: RC Circuit

B. For the circuit in Figure 4.2, given $i(0) = I$:

- Use KVL and the linear models to set up the differential equation for $i(t)$.
- Solve the differential equation for $i(t)$.
- Use this result to calculate $v_R(t)$, the voltage across the resistor for $t > 0$.

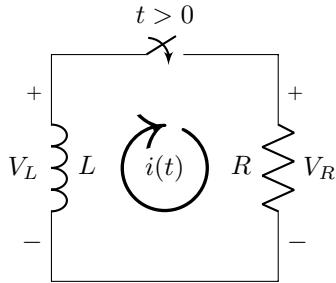


Figure 4.2: RL Circuit

4.2 MultiSim Simulations. Due at the beginning of lab.

- Simulate the circuit in Figure 4.3. Record three to four cycles of both the input and output. Refer to Appendix A.4.5 for information on how to change wire colors, and Appendix A.6 for information on how to present the results. The function generator should have the following settings:
 - Square Wave • f = 20 Hz • Duty Cycle = 50 % • Offset = 1 V • Amplitude = 1 V
- Repeat 4.2A, but change the capacitor to a nominal 10 μF one.
- Repeat 4.2A, but change the capacitor to a nominal 100 μF one.
- Calculate $\tau = RC$ for all three capacitor values. Use these values to explain the differences among the three output values.

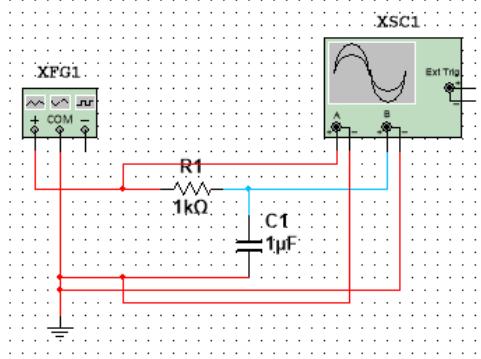


Figure 4.3: Lab #4 RC Simulation

E. Simulate the circuit in Figure 4.4. Record three to four cycles of both the input and output.

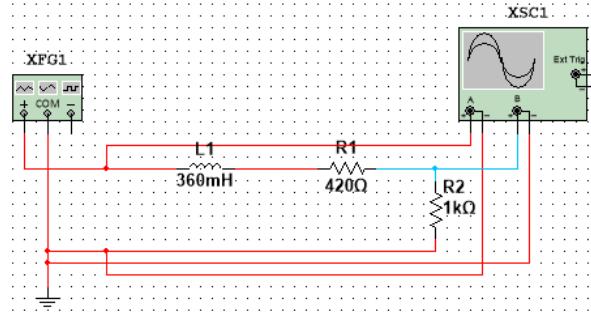


Figure 4.4: Lab #4 RL Simulation

F. Use the MultiSim scope cursors to measure τ . See Appendix A.5.4 for more details on how to use cursors. When measuring τ , there are two equations that must be considered:

$$V_{charge}(t) = V_{max}(1 - e^{-\frac{t}{\tau}}) \quad (1)$$

When $t = \tau$ for Equation 1, we get that $V(\tau) = .63V_{max}$. This means that the time constant, τ , is the time elapsed after 63% of V_{max} has been reached from 0%.

$$V_{discharge}(t) = V_{max}e^{-\frac{t}{\tau}} \quad (2)$$

Conversely, when $t = \tau$ for the Equation 2, we get that $V(\tau) = .37V_{max}$. This means that the time constant, τ , is the time elapsed after 37% of V_{max} is reached. Examples of these measurements are seen in Figure 4.5.

4.3 Laboratory Procedures

4.3.1 RC Circuit

A. Construct Figure 4.3 on the breadboard. Use the wavegen tool from the AD2 as the function generator, and use the same settings as used in Section 4.2:

- Square Wave • f = 20 Hz • Duty Cycle = 50 % • Offset = 1 V • Amplitude = 1 V

Refer to Appendix C.3.2 for details on how to use the wavegen tool. **Measure all component values that you will be using in the lab.**

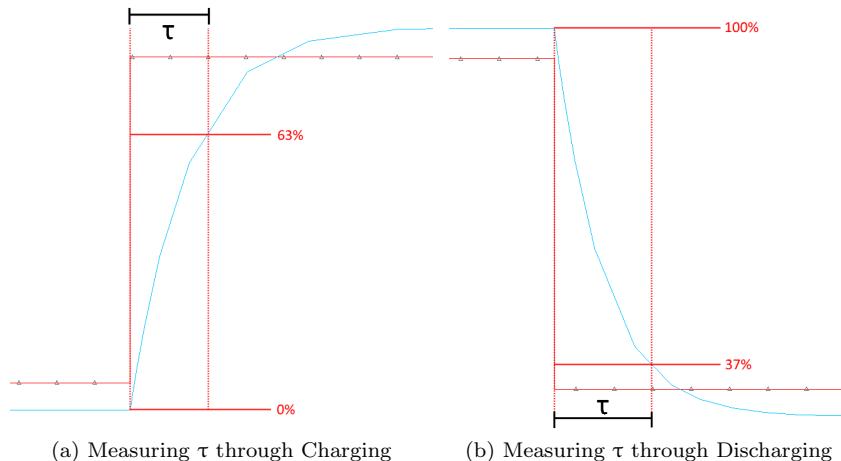


Figure 4.5: Measuring τ Examples

- B. Using the scope tool from the AD2, **present the input and output waveforms using the scope tool**. Refer to Appendix C.4.3 for further details on how to use the scope tool.
- C. Repeat 4.3.1B for a 10 μF capacitor. **Remember to record the input and output waveforms**.
- D. Repeat 4.3.1B for a 100 μF capacitor. **Remember to record the input and output waveforms**.
- E. Calculate τ using measured component values. **Report these τ values**.
- F. Using the τ values from 4.3.1E, **explain the differences in the plots from 4.3.1B , 4.3.1C, and 4.3.1D**.

4.3.2 RL Circuit

- A. Measure both the inductance and the internal resistance of the inductors, and the resistance of the resistor. **Record these values in your lab report**.
- B. Construct Figure 4.4 on the breadboard. Keep in mind two things when constructing the circuit:
 - Use the three inductors in series to recreate a 360 mH inductor.
 - The 420 Ω resistor is the sum of the internal resistances of the inductors; NOT an external resistor.
- , Use the wogen tool from the AD2 as the function generator, and use the same settings as used in Section 4.2:
 - Square Wave • f = 20 Hz • Duty Cycle = 50 % • Offset = 1 V • Amplitude = 1 V
- C. Using the scope tool from the AD2, **present the input and output waveforms**.
- D. Using the cursors from the scope tool, **measure τ using the same method as 4.2F**.
- E. Calculate τ using Equation 3. **Record this value, and compare it to the simulated and calculated values of τ** .

$$\tau = \frac{L}{R + R_L} \quad (3)$$

5 Lab # 5: Introduction to AC Signals

5.1 PreLab Calculations. Due at the beginning of lab.

- A. Given the following equation for a general AC waveform, label V , ω , and ϕ .

$$A = V \cos(\omega t + \phi)$$

- B. Start the Waveforms 2015 software. You do not need the AD2 to be plugged in. Using the demo Discovery 2, simulate various waveforms using the wavegen tool. Refer to Appendix C for further details on how to use Waveforms 2015.

- Make a comment (1-2 sentences) on what happens to the waveform as you adjust the amplitude.
- Make a comment (1-2 sentences) on what happens to the waveform as you adjust the frequency.
- Make a comment (1-2 sentences) on what happens to the waveform as you adjust the phase shift.

- C. Simulate three signals of varying amplitudes and phase shifts. Maintain the frequency at 30 Hz.

- D. Express your three signals in both polar and rectangular form.

- E. Plot the rectangular form of the equation on the complex plane.

5.2 MultiSim Simulations. Due at the beginning of lab.

- A. Simulate the circuit seen in Figure 5.1. Use the following settings for the function generator:

- AC Sine Wave
- $f = 30$ Hz
- Phase Shift = 50°
- Offset = 0 V
- Amplitude = 2 V

Verify KVL for the circuit by printing out the outputs of each voltmeter and proving KVL.

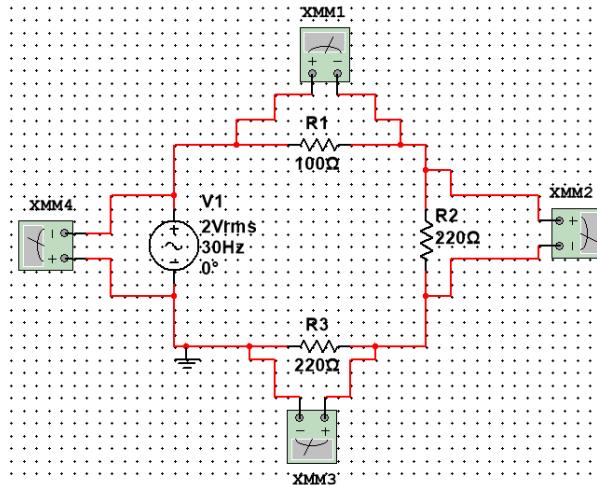


Figure 5.1: Lab #5 Basic AC KVL Circuit

- B. Simulate the circuit seen in Figure 5.2. Use the same function generator settings as 5.2A. Record three to four cycles of both the input and output. Using the cursors on the oscilloscope, measuring the time shift between the maxima of the two waves, Δt , and use it to calculate the phase shift $\Delta\phi$ using the following equation:

$$\Delta\phi = 360 \times \Delta t \times f$$

- C. Using the cursors in the oscilloscope, measure the period T for both the input and output waveforms of Figure 5.2. Use this period to calculate the frequency, $f = \frac{1}{T}$. Are the frequencies the same for the input and output waveforms?

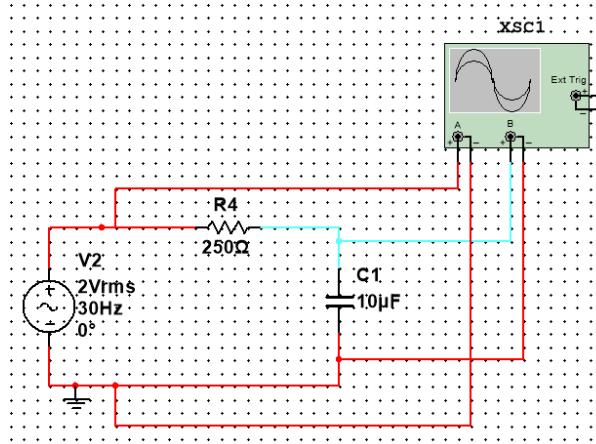


Figure 5.2: Lab #5 Basic AC RC Circuit

5.3 Laboratory Procedure

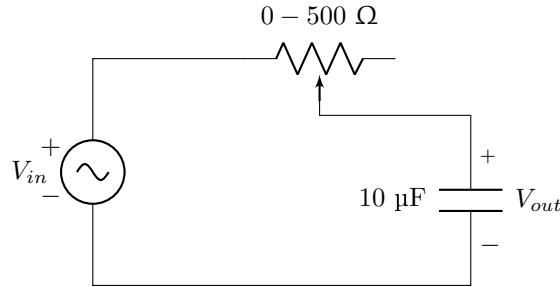


Figure 5.3: Lab # 5 Breadboard Connection

- Connect the potentiometer to an empty portion of the breadboard. **Measure and record the minimum and maximum resistance that it can achieve.** Refer to Appendix B.2.4 for further details on the potentiometer.
- Build the circuit in Figure 5.3. **Using the waveforms of both V_{in} and V_{out} , record what happens to the phase shift, $\Delta\phi$, as the resistance of the potentiometer changes.**
- Set the resistance of the potentiometer to its maximum resistance. **Measure and record the time shift, Δt , and use it to calculate the phase shift $\Delta\phi$.**
- Save a screenshot of the input and output waveforms when the resistance of the potentiometer is at a maximum.** Refer to Appendix C.7.2 on how to appropriately save the screenshot for submission.
- Measure the RMS voltage of source, potentiometer, and capacitor when the resistance of the potentiometer is at a maximum.** Use the AC RMS measurement option of the logger tool.
- 5.2A has you confirm KVL for Figure 5.1. **Does KVL apply for the measurements made in 5.3E? If KVL fails, what information are you missing that should be included?**

6 Lab # 6: Frequency Response

6.1 PreLab Calculations: Due at the beginning of lab.

- A. For the circuit in Figure 6.1, derive the complex valued transfer function, V_{out}/V_{in} , in terms of R, L, and the angular frequency ω .

- From the transfer function, derive the real valued function for the phase response. Use the variable ω_0 for the cutoff frequency, which replaces the ratio R/L .
- From the transfer function, derive the real valued function for the magnitude response. Use the variable ω_0 for the cutoff frequency, which replaces the ratio R/L .

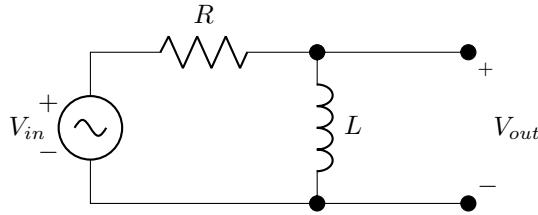


Figure 6.1: Lab # 6 High Pass Filter

- B. For the circuit in Figure 6.2, derive the complex valued transfer function, V_{out}/V_{in} , in terms of R, L, and the angular frequency ω .

- From the transfer function, derive the real valued function for the phase response. Use the variable ω_0 for the cutoff frequency, which replaces the ratio R/L .
- From the transfer function, derive the real valued function for the magnitude response. Use the variable ω_0 for the cutoff frequency, which replaces the ratio R/L .

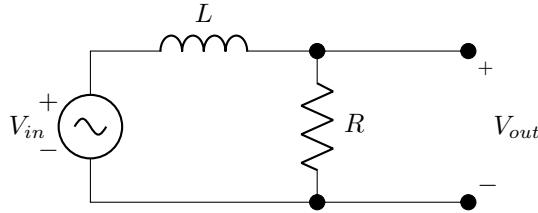


Figure 6.2: Lab # 6 Low Pass Filter

- C. Using the magnitude and phase response from 6.1A, determine the inductance L that satisfies the follow conditions, given $R = 1 \text{ k}\Omega$.

$$V_{in} = 10\angle 0^\circ \text{ V} \quad V_{out} = 7.07\angle 45^\circ \text{ V} \quad f = 1.3 \text{ kHz}$$

- D. Using the magnitude and phase response from 6.1B, determine the inductance L that satisfies the follow conditions, given $R = 1 \text{ k}\Omega$.

$$V_{in} = 20\angle 0^\circ \text{ V} \quad V_{out} = 11.2\angle -56^\circ \text{ V} \quad f = 4.8 \text{ kHz}$$

6.2 MultiSim Simulations: Due at the beginning of lab.

- A. Simulate the circuit seen in Figure 6.3a. Use the values for R and L calculated in 6.1C. See Appendix A.5.5 for details on how to use the Bode Plotter.

- Print out the magnitude and phase plots. See Appendix A.6 on how to properly present results.
 - What is the magnitude and phase at $f = 1.3$ kHz? Is that what you expected? Why or why not?
- B. Simulate the circuit seen in Figure 6.3b. Use the values for R and L calculated in 6.1D.
- Print out the magnitude and phase plots. See Appendix A.6 on how to properly present results.
 - What is the magnitude and phase at $f = 4.8$ kHz? Is that what you expected? Why or why not?

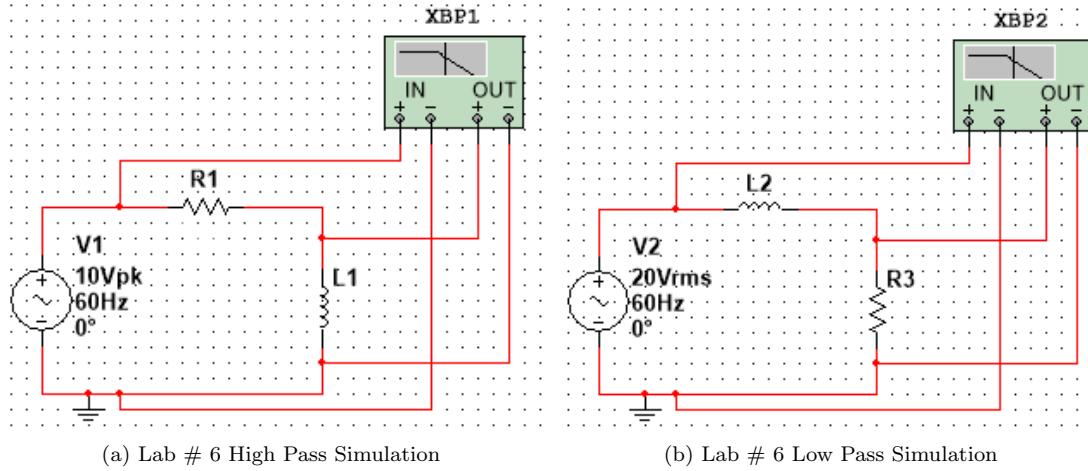


Figure 6.3: fig:Lab # 6 Simulations

6.3 Laboratory Procedure

- A. Construct the circuit in Figure 6.4. Set the initial settings of the AC source to be:
- AC Sine Wave
 - $f = 1$ Hz
 - Phase Shift = 0°
 - Offset = 0 V
 - Amplitude = 2 V

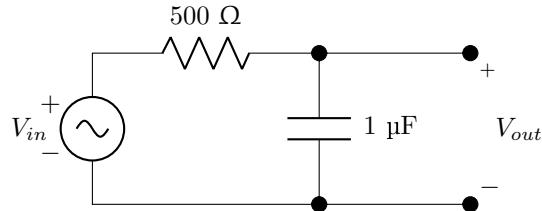


Figure 6.4: Lab # 6 Breadboard Connection

- B. Measure both the V_{in} RMS voltage and the V_{out} RMS voltage. **Record these results in Table 5.**
- C. Repeat the process for all other frequencies in Table 5. **Calculate the ratio V_{out}/V_{in} for each frequency.**
- D. **Using the data from Table 5, create a Bode Plot of the magnitude response.** Remember that the horizontal axis of the plot will be $\log(f)$, and the vertical axis will be decibels, which can be calculated through the following equation:

$$dB = 20 \log(V_{out}/V_{in})$$

E. Using your Bode Plot, what is the approximate frequency f when the gain is -3 dB? **Record this value, and compare it to the theoretical value calculated using the following equation:**

$$f_0 = \frac{1}{2\pi RC}$$

F. Is the circuit in Figure 6.4 a high pass or low pass filter? How do you know?

Table 5: Lab #6 Measurements

f (Hz)	V_{in} (mV)	V_{out} (mV)	V_{out}/V_{in}
1			
2			
10			
20			
50			
100			
200			
250			
300			
350			
400			
450			
500			
800			
1000			
1200			

7 Lab # 7: Operational Amplifier Integrator and Active Filter

7.1 PreLab Calculations. Due at the beginning of lab.

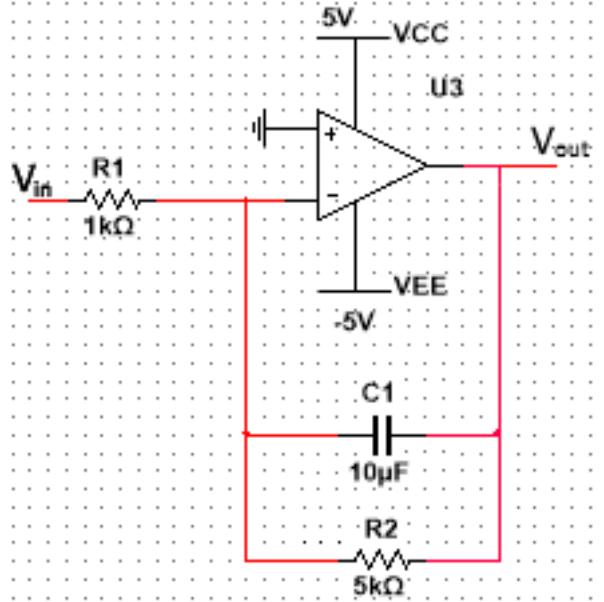


Figure 7.1: Circuit for Lab # 7

- A. Consider the circuit in Figure 7.1 to be an op amp integrator. Using the ideal op amp model and KCL, show that (note: do not consider R_2 as part of the circuit):

$$V_{out} = \frac{-1}{R_1 C} \int V_{in}(t) dt$$

- B. If V_{in} is a square wave, what will the shape of V_{out} be? Note that you only need to state the shape, and do not need to provide a numeric solution.
- C. If V_{in} is a triangle wave, what will the shape of V_{out} be? Note that you only need to state the shape, and do not need to provide a numeric solution.
- D. If V_{in} is a sine wave, what will the shape of V_{out} be? Note that you only need to state the shape, and do not need to provide a numeric solution.
- E. Now consider the circuit in Figure 7.1 to be an active low-pass filter. Show that the amplitude response and phase response are:

$$A(\omega) = \frac{R_2/R_1}{\sqrt{1 + (\omega R_2 C)^2}} \quad \phi(\omega) = \tan^{-1}(-\omega R_2 C)$$

F.

7.2 MultiSim Simulations. Due at the beginning of lab.

- A. Simulate the circuit seen in Figure 7.2a. Use the following settings for the function generator:
 • AC Sine Wave • f = 30 Hz • Phase Shift = 0° • Offset = 0 V • Amplitude = .5 V
 Note that the circuit acts as a low pass filter. Measure the low frequency gain (dB) and the frequency (Hz) at this gain.

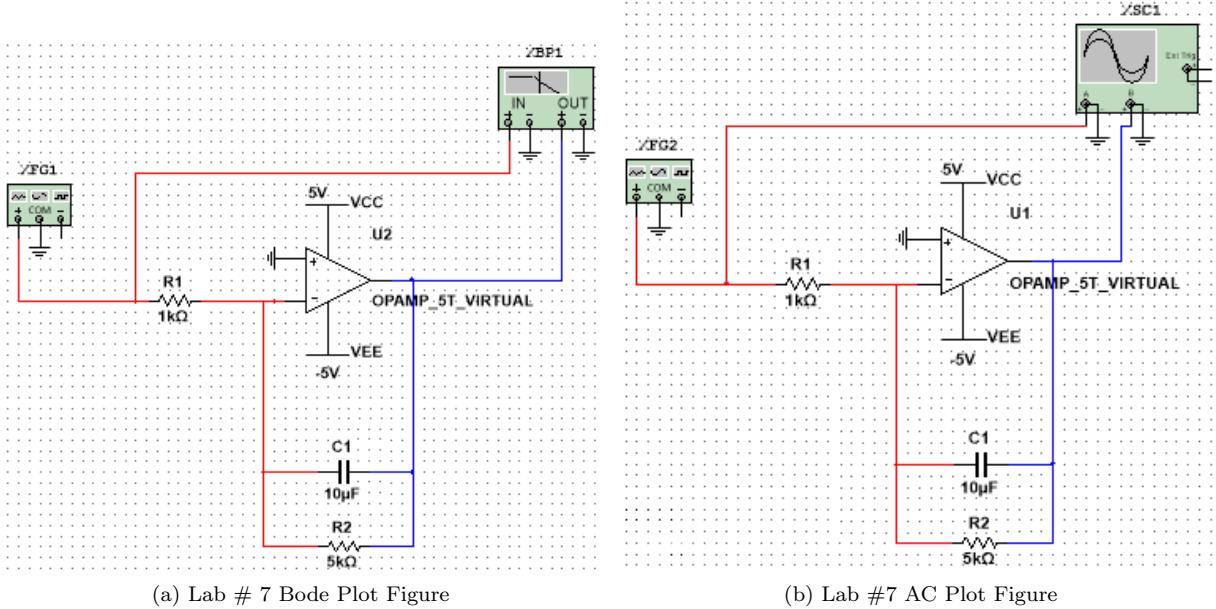


Figure 7.2: Lab # 7 Simulation Figures

- B. Print out the schematic and the Bode amplitude plot for the circuit in Figure 7.2a.
- C. Switch the input waveform type between AC Sine Wave, Square Wave, and Triangle Wave for Figure 7.2b. Print out the associated oscilloscope plots for all three different inputs. Do these match your results in Section 7.1?

7.3 Laboratory Procedures

- A. Build the circuit seen in Figure 7.2b. Remember to measure all the component values. Note that the $5\text{ k}\Omega$ resistor is constructed using two $10\text{ k}\Omega$ resistors.
- B. Set the wave generator to the following settings:
 - AC Sine Wave • $f = 30\text{ Hz}$ • Phase Shift = 0° • Offset = 0 V • Amplitude = $.5\text{ V}$
 Record the input and output waveforms using the scope tool. Measure the amplitudes of both V_{in} and V_{out} , and the phase shift between the waves. Does this match your expected results?
- C. Repeat the procedure of 7.3B, except use a square wave instead of a sine wave.
- D. Repeat the procedure of 7.3B, except use a triangle wave instead of a sine wave.

8 Lab # 8: Introduction to Logic Gates

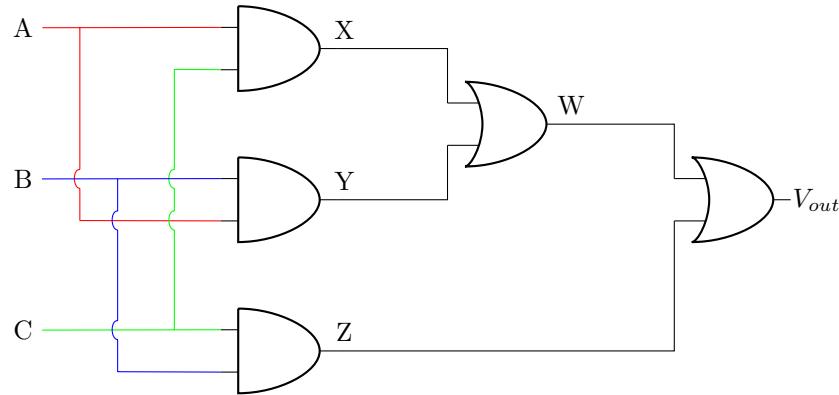


Figure 8.1: Logic Gate Circuit

8.1 PreLab Calculations: Due at the beginning of lab.

- Fill out the truth table in Table 6 using Figure 8.1.
- Looking at the results, what condition for the inputs must hold in order for the output to be 1?
(Hint: Look at the number of 1's versus the number of 0's that correspond to each output of 1.)

8.2 MultiSim Simulations: Due at the beginning of lab.

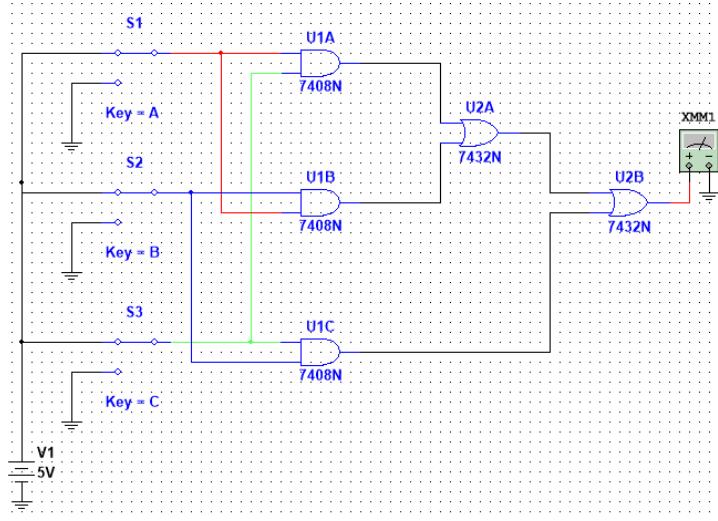


Figure 8.2: Lab # 8 Simulation

- Construct the circuit seen in Figure 8.2. Refer to Appendix A for more information on how to use logic gates.
- Using the circuit you constructed, fill out the truth table seen in Table 7. Note that V_{out} is the voltage measured by the multimeter.
- Compare your results for V_{out} between Tables 6 and 7. Do the results agree with each other?

Table 6: Truth Table for Figure 8.1

A	B	C	X	Y	Z	W	V_{out}
0	0	0					
0	0	1					
0	1	0					
0	1	1					
1	0	0					
1	0	1					
1	1	0					
1	1	1					

Table 7: Truth Table for Figure 8.2

A (V)	B (V)	C (V)	V_{out} (V)
0	0	0	
0	0	5	
0	5	0	
0	5	5	
5	0	0	
5	0	5	
5	5	0	
5	5	5	

8.3 Laboratory Procedure

A. Test the inputs and outputs of an AND (7408) gate. Refer to Appendix B for the pin diagram. Don't forget to supply the chip with 5 V. **Fill out the simple truth table seen in Table 8.** There are a few important notes to remember when making measurements:

- When attempting to get inputs of 1 and 0, you **must** use either 5 V for 1, or a ground for 0. Leaving an input without anything plugged in will not work.
- When making measurements on the output:
 - If $V_{out} \geq 3.5$ V, the output may be considered a 1.
 - if $V_{out} \leq 100$ mV, the output may be considered a 0.
 - If V_{out} does not fall within these ranges, double check all inputs and outputs are functioning correctly.

Table 8: Basic 2-Input Truth Table

A	B	V_{out}
0	0	
0	1	
1	0	
1	1	

- B. Test the inputs and outputs of an OR (7432) gate. Refer to Appendix B for the pin diagram. Don't forget to supply the chip with 5 V. **Fill out the simple truth table seen in Table 8.**
- C. Construct the circuit seen in Figure 8.1. You may use any number of AND and OR chips to build your circuit. **Remember to keep track of your inputs and outputs carefully, as to avoid confusion.**
- D. Using 5 V as an input of 1 and a ground as an input of 0, **fill out the truth table seen in Table 7. Confirm these results with both your prelab and your simulations.**

9 Lab # 9: Introduction to Microcontrollers

9.1 PreLab Calculations. Due at the beginning of lab.

- A. Read Appendix D. This appendix contains all the relevant information on the hardware and software associated with microcontrollers.
- B. Consider Figure 9.1, which is a simple flowchart representing a blink program. Note that flowcharts are a visual representation of algorithms, which are nothing more than a finite set of precise instructions to solve a problem or to carry out a computation.

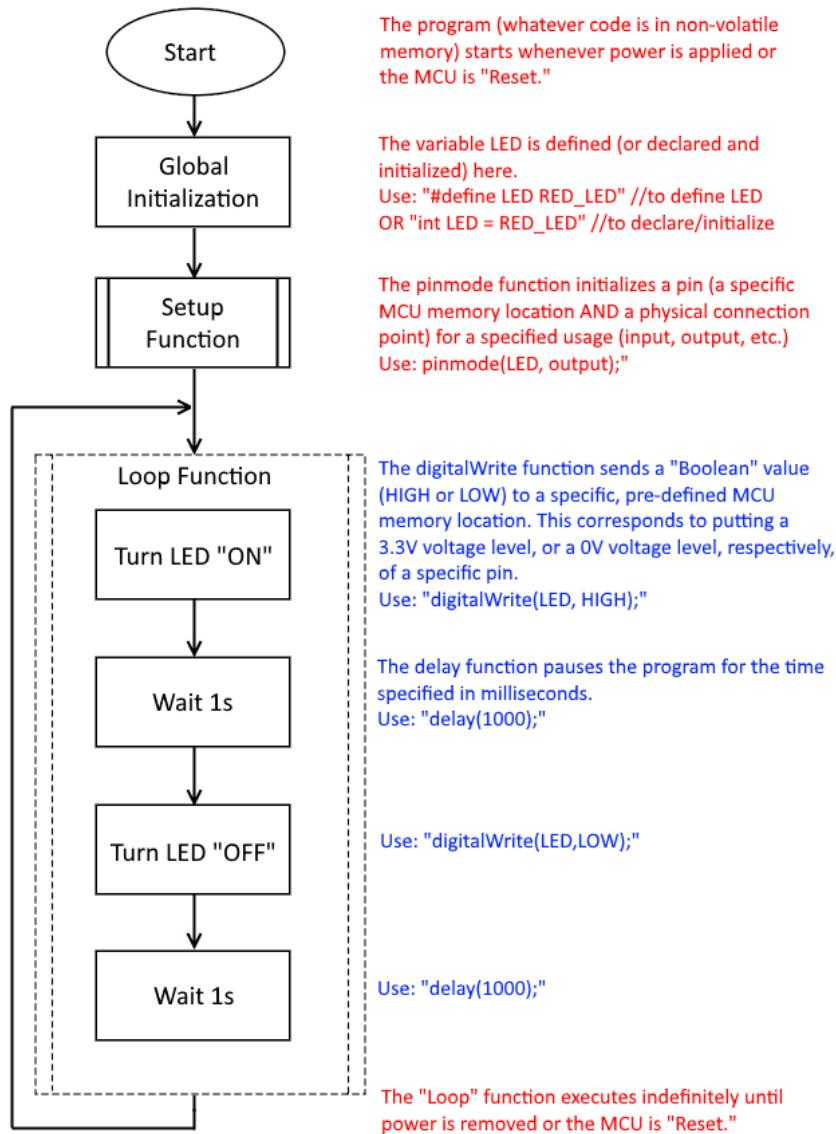


Figure 9.1: Blink Program Flowchart

- C. Modify this flowchart to represent an algorithm which blinks LED1 ON → OFF → ON → OFF, and simultaneously blinks LED2 in the opposite manner, OFF → ON → OFF → ON.
 - Focus primarily on the loop part of the flowchart.
 - Details about the other portions of the flowchart will be discussed later in lab.
 - You need only submit the modified flowchart. Additional comments are not necessary.

9.2 MultiSim Simulations: Due at the beginning of lab.

There are **NO** MultiSim simulations for Lab 9.

9.3 Laboratory Procedure

- A. Ensure that your LaunchPad and Energia software are running correctly. For detailed instructions on how to set up the devices, see Appendix D.3.1.
- B. Run the "Blink" sketch available in Energia.
 - This is found in: File → Examples → 01.Basics → Blink
 - You should also enable line numbers. This is available through: File → Preferences, where you can select "Display Line Numbers."



The screenshot shows the Energia IDE interface with a red title bar labeled 'Blink'. The main window displays the following C code:

```
12 // most launchpads have a red LED
13 // see pins_energia.h for more LED definitions
14 #define LED RED_LED
15
16 // the setup routine runs once when you press reset:
17 void setup() {
18     // initialize the digital pin as an output.
19     pinMode(LED, OUTPUT);
20 }
21
22 // the loop routine runs over and over again forever:
23 void loop() {
24     digitalWrite(LED, HIGH);    // turn the LED on (HIGH is the voltage level)
25     delay(1000);              // wait for a second
26     digitalWrite(LED, LOW);   // turn the LED off by making the voltage LOW
27     delay(1000);              // wait for a second
28 }
```

Figure 9.2: Blink Program Sketch

Figure 9.2 shows the code in Energia. Comparing this to Figure 9.1, you'll see that:

- Global Initialization occurs from Line 1 to Line 17.
- The "Setup Function" occurs from Line 19 to Line 23.
- The "Loop Function" occurs from Line 25 to Line 31.

In order to properly run the program, follow the instructions in Appendix D.3.1.

- C. Does the Energia sketch do what you expect it to? **Discuss whether or not it does, and give a thorough comparison of the flowchart in Figure 9.1 to the code.**
- D. Modify the "Blink" sketch to fit the following requirements:
 - Change the LED from Red to Green.
 - Change the OFF time to 250 ms, and the ON time to 500 ms.

Run the sketch, and show the working LaunchPad to your TA. **Include the modified sketch in your report, either through a screenshot of the code, or the code itself copied and pasted into a word processor.**

- E. Download and modify the "Blink2LED_student.ino" file to fit the following requirements:
 - Use an integer counter to track iterations.
 - For every iteration, turn LED1 ON for 0.5s, and OFF for 0.5s.
 - Every fifth iteration, also turn LED2 ON for 0.5s, and OFF for 0.5s.

Figure 9.3 shows a flowchart of the modified sketch, with comments that should help you with modifying the code. **Show the working LaunchPad to your TA, and include the modified sketch in your report.**

F. Download and modify the "Analog_LED_student.ino" file to fit the following requirements:

- Use an integer counter to track PB1 and PB2 actuation (presses).
- For PB1 actuation, increase x if $x \geq 10$.
- for PB2 actuation, decrease x if $x \leq 0$.
- Set the LED intensity that is directly proportional to the value of x .

Figure 9.4 shows a flowchart of the modified sketch, with comments that should help you with modifying the code. **Show the working LaunchPad to your TA, and include the modified sketch in your report.**

G. Download and modify the "Binary_LED_student.ino" file to fit the following requirements:

- Use an integer count to track the binary state of this 2-bit binary system.
 - LED1 represents the Most Significant Bit (MSB)
 - LED2 represents the Least Significant Bit (LSB)
- Use PB1 to move this binary system from state to state.

Figure 9.5 shows a flowchart of the modified sketch, with comments that should help you with modifying the code. **Show the working LaunchPad to your TA, and include the modified sketch in your report.**

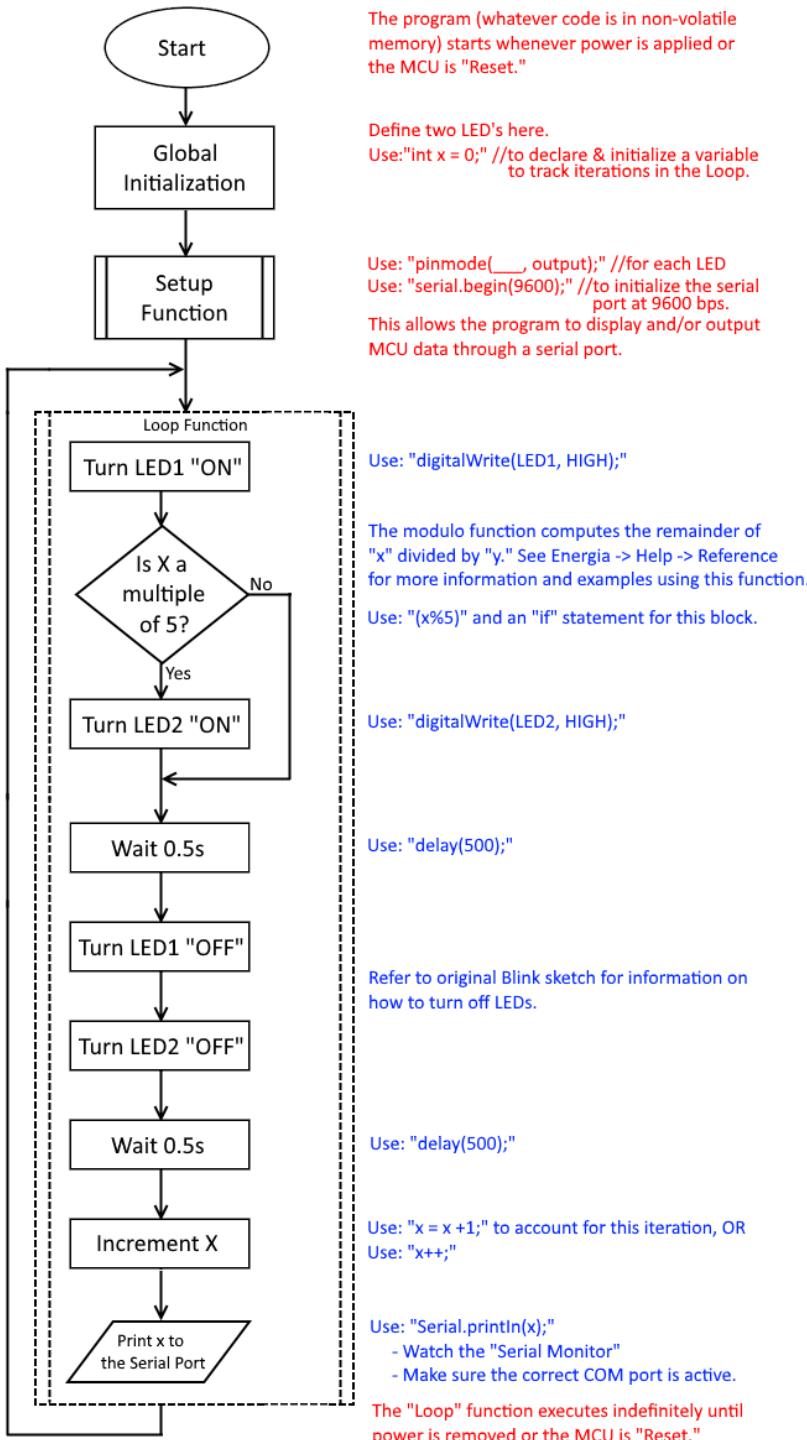


Figure 9.3: Blink2 Program Flowchart

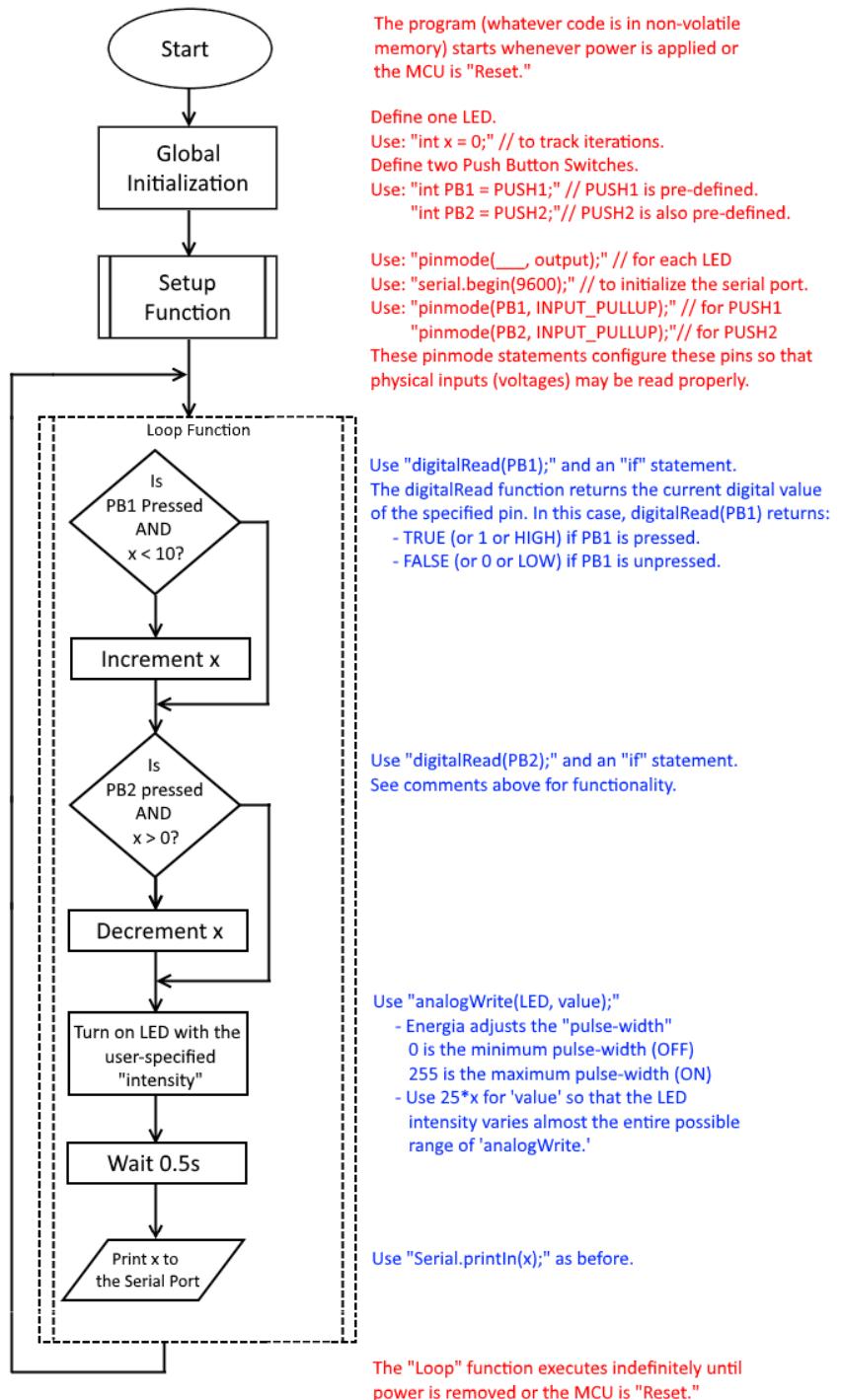


Figure 9.4: Analog Program Flowchart

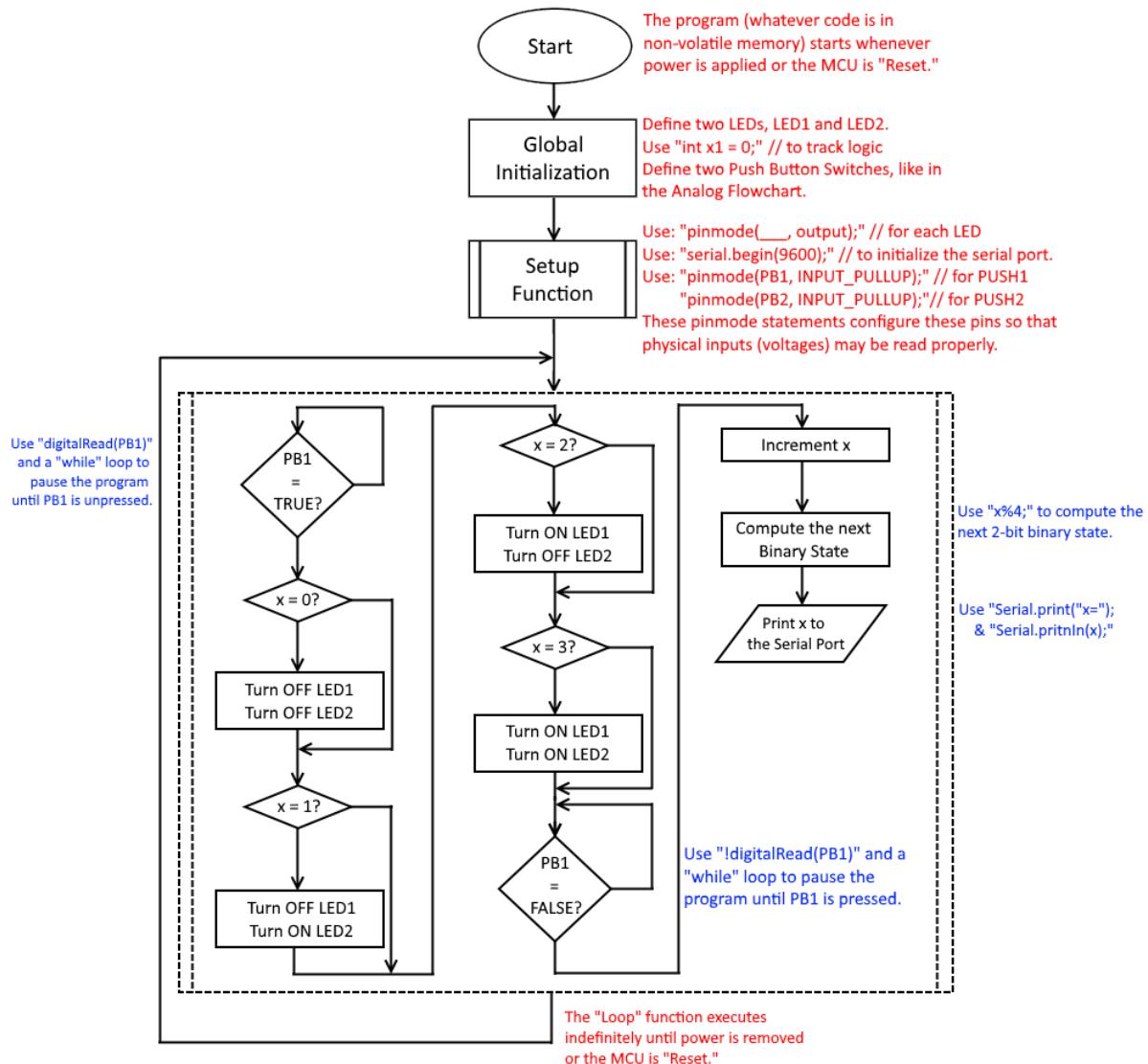


Figure 9.5: Binary Program Flowchart

Part II

Reference Materials

A Appendix A

A.1 Main Window

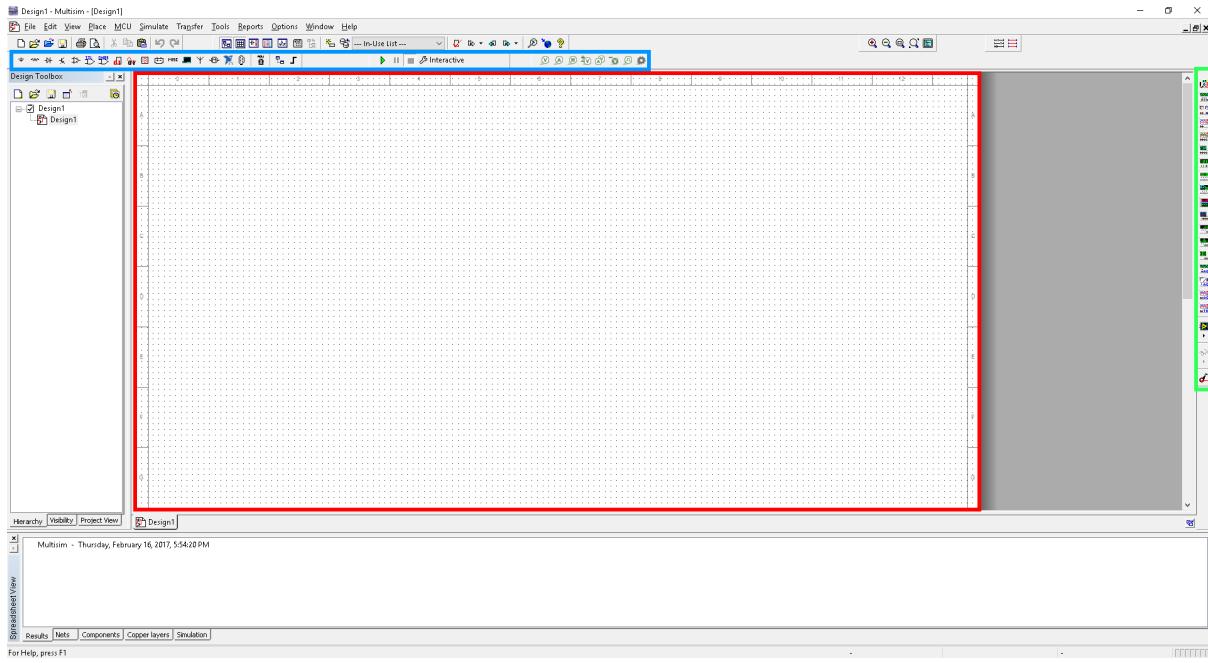


Figure A.1: Main Window of MultiSim

Looking at Figure A.1, we see the main window of MultiSim that shows up when loading the program. There are three main components that will be focused on, whose descriptions are seen in Table 9. We will explore the different toolbars in subsequent subsections.

Table 9: Descriptions of MultiSim Parts in Figure A.1

Color	Label	Description
Red	Design Window	This is where you will place all the components and measuring devices, and where a majority of your work will be.
Blue	Component Toolbar	This is where you will access components and sources that you will put on the design window.
Green	Measurement Toolbar	This is where you will find measurement tools like multimeters and oscilloscopes.

A.2 Components Toolbar

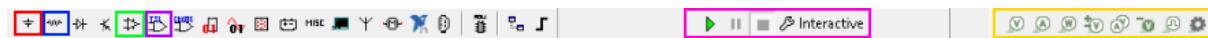


Figure A.2: Component Toolbar of MultiSim

Figure A.2 is a zoomed in version of the components toolbar, one of the primary toolbars that will be used when doing any simulations. While there is a lot of different options here, there are only a few that you need to keep mind of, which are seen in Table 10.

Table 10: Descriptions of Component Toolbar Options

Color	Label	Description
Red	Sources	This is where you will access all power sources and grounds.
Blue	Basic Components	This is where you will access components like resistors and capacitors.
Green	Analog	This is where you will find Operational Amplifiers.
Purple	Transistor-Transistor Logic (TTL)	This is where you will find TTL components like AND/OR gates.
Pink	Simulations	This is where you will play, pause or stop your simulations.
Orange	Node Measurements	This is where you will access node voltage and current measurement devices.

A.3 Measurement Toolbar



Figure A.3 is a zoomed in version of the measurement toolbar, the other primary toolbar that will be used during simulations. Here, instead of picking components like resistors and operational amplifiers, you will find primarily measurement tools.

Table 11: Descriptions of Measurement Toolbar Options

Color	Label	Description
Red	Multimeter	Primarily used to measure DC current, voltage, and resistance.
Blue	Function Generator	Not used for measurement. Used to generate AC Signals for AC circuits.
Green	Oscilloscope	Primarily used to measure AC signals like voltage current.
Purple	Bode Plotter	Used to generate Bode plots for frequency response.

While there are a variety of measurement tools here, it is not enough to just use the tools and print the results for submission. Instead, you must be conscientious of how to properly present the results. This will be discussed further in the appendix, but is important to note now.

Figure A.3:
Measurement
Toolbar of MultiSim

A.4 AC and DC Circuits

A vast majority of circuits you will be using this semester fall into one of two specific categories; alternating current, also known as AC, circuits, or direct current, also known as DC, circuits. We will begin by closely examining power sources and components, and then expanding into how to actually construct circuits.

A.4.1 Power Sources

Looking at Figure A.2, clicking on the sources button results in a window popping up, seen in Figure A.5. Under the Power Sources tab, the primary three components you need to concern yourself with are the DC_POWER, AC_POWER, and GROUND components. Double clicking any of the components will close the sources window and return you to the design window.

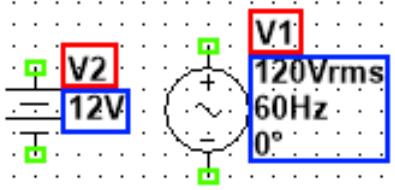


Figure A.4: AC and DC Power Sources

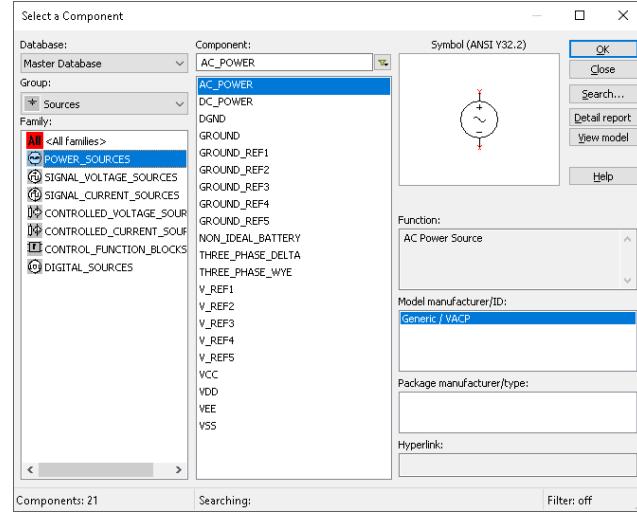
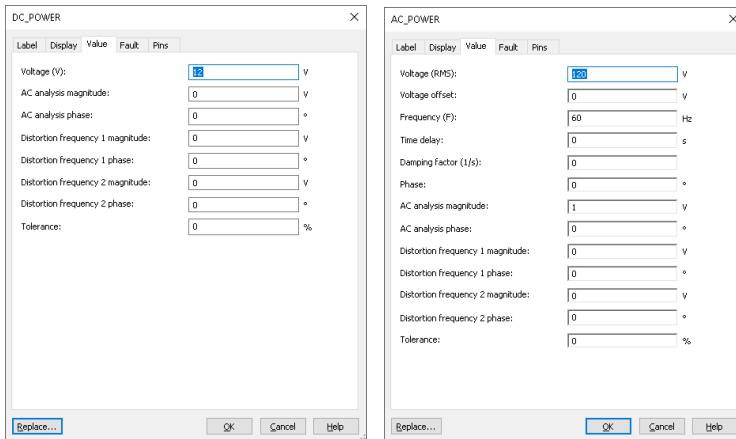


Figure A.5: Power Sources Window

Table 12: Descriptions of Power Source Parts

Color	Label	Description
Red	Name	Names each power source. Automatically updates as you place more sources.
Blue	Value	Displays important values associated with the power source
Green	Nodes	Connection points for other components.



(a) DC Power Source

(b) AC Power Source

From the main window, double clicking on any source on the design window will pull up a second window that allows you to change any of the parameters of the power source. The windows for DC power sources and AC power sources are seen in Figures A.6a and A.6b, respectively.

A.4.2 Basic Components

Looking back at Figure A.2, clicking on the components button results in a window popping up, seen in Figure A.7. The three primary types of components we will be focused on are resistors, capacitors, and inductors, known as passive components. They are highlighted in Figure A.7.

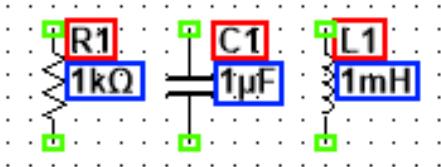


Figure A.6: Passive Components

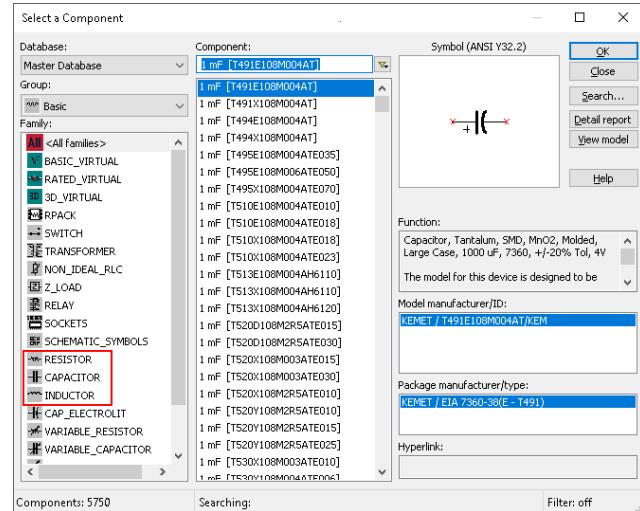


Figure A.7: Components Window

Table 13: Descriptions of Passive Component Parts

Color	Label	Description
Red	Name	Names each component. Automatically updates as you place more components.
Blue	Value	Displays important values associated with the component.
Green	Nodes	Connection points for other components.

Like the power sources, clicking on any of these components will bring up a window that allows you to change the associated value of the passive component, like the resistance of a resistor. Examples of these windows are seen in Figure A.8.

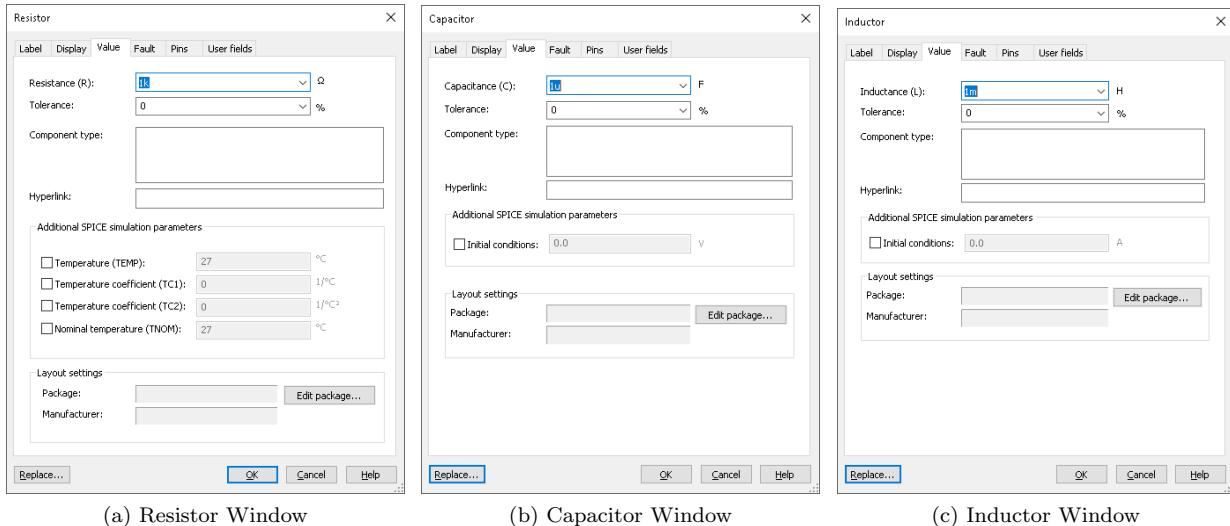


Figure A.8: Passive Component Windows

A.4.3 Operational Amplifiers

The primary use of the analog tab in Figure A.2 is to access operational amplifiers. This results in the window seen in Figure A.9. From here, there are two options we are concerned with; virtual op amps and real op amps. Virtual op amps can be accessed through the ANALOG_VIRTUAL tab and results in perfectly accurate results, where the output of the op amp obeys theoretical calculations. This differs from real op amps, accessed through the OPAMP tab, whose internal structure will result in minute differences in output compared to theoretical results.

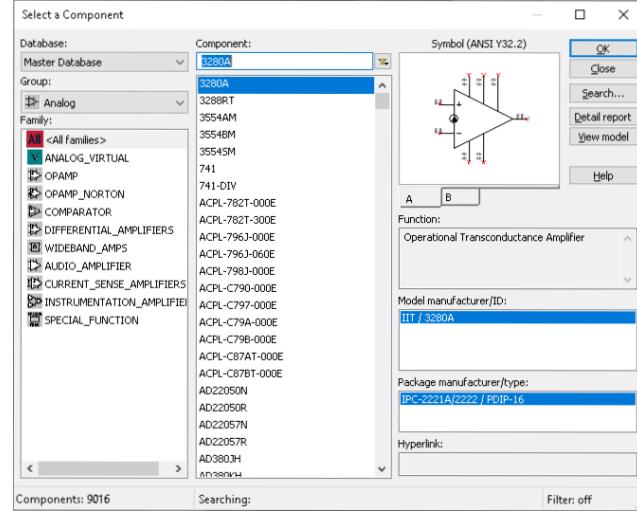


Figure A.9: Analog Components Window

Table 14: Operational Amplifiers Terminals

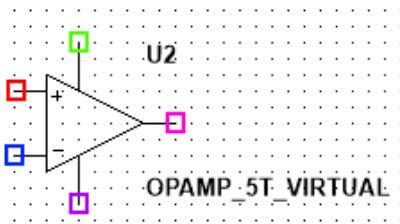


Figure A.10: Virtual Op Amp

Color	Label	Description
Red	Positive Input	Maintains the same sign of the voltage entering this terminal.
Blue	Negative Input	Inverts the sign of the voltage entering this terminal.
Green	Positive Voltage Rail	Used with the Negative Voltage Rail to account for offset voltage.
Purple	Negative Voltage Rail	Used with the Positive Voltage Rail to account for offset voltage.
Pink	Output	Output of the Op Amp.

Virtual Op Amps are primarily used to compare theoretical values from calculations to simulated values. As such, real op amps in MultiSim will be discussed when examining the integrated circuit and how to use them in the lab.

A.4.4 Digital Logic Components

The TTL tab of Figure A.2 results in the window seen in Figure A.11. All logic gates are identified by not by their purpose, but by a number. There are also many duplicate numbers, just with different letter endings. For the time being, you may choose to only focus on logic gates that end with the letter N. Ex: 7400N.

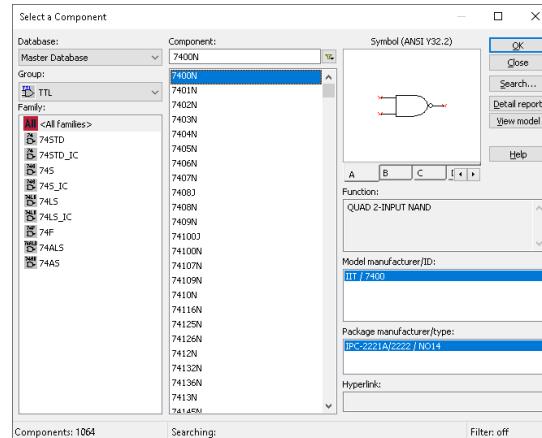


Figure A.11: Analog Components Window

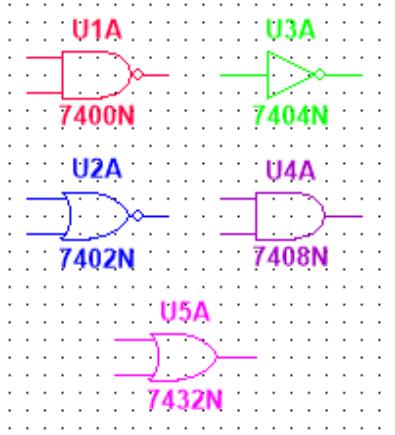


Table 15: Logic Gate Labels

Color	Number	Purpose
Red	7400N	NAND Gate
Blue	7402N	NOR Gate
Green	7404N	NOT Gate
Purple	7408N	AND Gate
Pink	7432N	OR Gate

Figure A.12: Logic Gates

A.4.5 Basic Circuit Examples

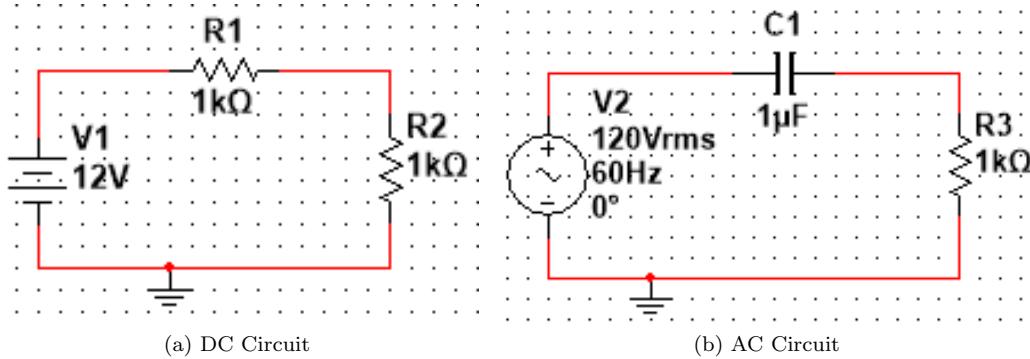


Figure A.13: Basic Circuit Examples

Figures A.13a and A.13b are examples of simple DC and AC circuits, respectively. While there are not many differences between the two, there are still some important features to note:

- There is a ground, seen by the component beneath the power source. These must be included in all circuits built, and are generally placed at the negative node of a power source.
- In order to rotate components for easier wire drawing, simply select the component and press **CTRL + R**.
- In order to draw wires, simply click on one node of the first component and MultiSim will automatically draw the wire to the second node you click.

A.5 Making Measurements

Now we must focus on how to make measurements and find the results that are deemed necessary. There are a variety of values you will measure in this lab, and as such, each result requires a different method of measurement.

A.5.1 Multimeter Basics

Multimeters are a unique tool with many different measurement capabilities. However, we will primarily be using them for measuring DC voltages and currents. Multimeters are accessed using the multimeter

tab seen in Figure A.3. Clicking on the tab will give you a multimeter to place anywhere on the design window, with a positive and negative terminal to connect to. Double clicking on the multimeter will result in the window seen in Figure A.14.

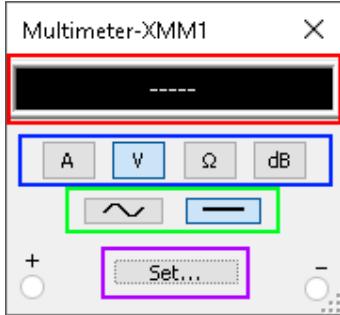


Figure A.14: Multimeter Window

Table 16: Descriptions of Multimeter Window

Color	Label	Description
Red	Display	Displays selected value.
Blue	Units	Chooses what value you are measuring.
Green	Current Flow	Switches between AC and DC measurements.
Purple	Settings	Access settings of the multimeter.

A.5.2 Measuring DC Voltage

Using the basic DC circuit example, we will measure the voltage across the resistor R2. In order to accurately measure voltage, there are two criteria to meet:

- The multimeter must be **parallel** to the component. This means that one terminal of the multimeter is connected to one node of the component you are measuring across. This is because ideal voltmeters have infinite resistance, and if this is placed in series where current flows through, you will end with inaccurate measurements.
- The current flowing through the system flows through the positive terminal of the multimeter first, and then the negative terminal. It is important to have this orientation correct in order to ensure the sign of the measurement is accurate. Examples of the two different orientations are seen in Figure A.15.

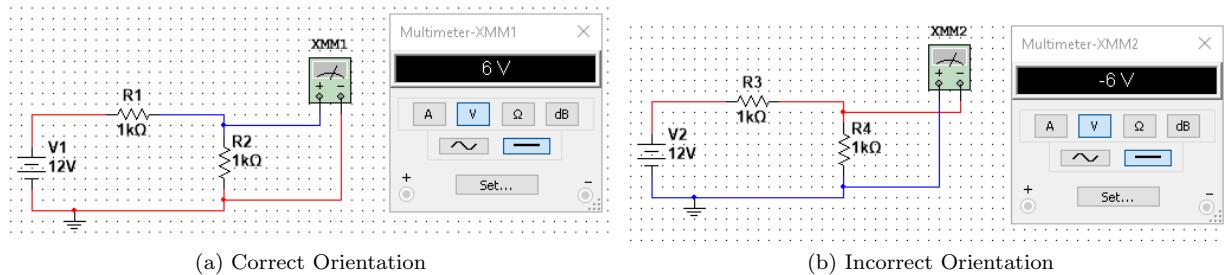


Figure A.15: Orientations for Voltmeter

A.5.3 Measuring DC Current

Again using the basic DC circuit example, we will measure the current flowing through the circuit. In order to accurately measure voltage, there are two criteria to meet:

- The multimeter must be **in series** with all other components. This means that the current flows in one terminal of the multimeter and out the other terminal. This is because ideal ammeters have zero resistance, and as such, if placed in parallel, would not read any results.
- The current flowing through the multimeter enters the positive terminal and exists the negative terminal. This is for the same reason as the voltmeter. Examples of the two different cases are seen in Figure A.16

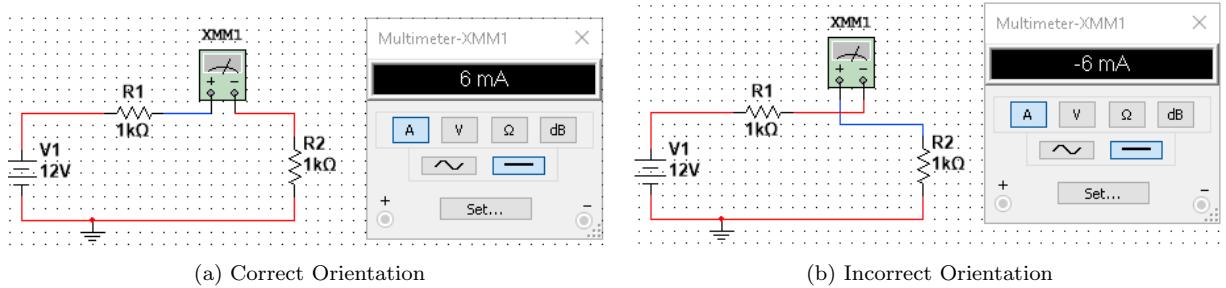


Figure A.16: Orientations for Ammeter

A.5.4 Using the Oscilloscope

The oscilloscope is the primary tool in MultiSim that will be used for measuring AC signals. We will be using the basic AC circuit as an example and show how to measure different values and manipulate the display. There are a few things to note about the example seen in Figure A.17:

- There are two channels in the oscilloscope, channel A and channel B. This means you may make two different measurements at a time.
- The color of the wires entering the positive terminal of the channel determines the color of the signal on the oscilloscope.
- There are some unused ports on the right of the oscilloscope. These are known as trigger ports, and will not be used during your simulations.

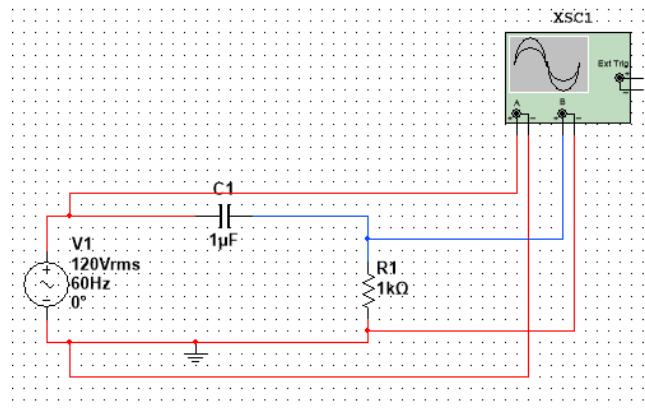


Figure A.17: Oscilloscope Attached to Basic AC Circuit

When you click on the oscilloscope after starting a simulation, you will notice that the oscilloscope is outputting hundreds of waveforms a second. This is due to the fact that the oscilloscope is operating in real time. As such, pausing the simulation will also pause the oscilloscope. This will result in the window seen in Figure A.18.

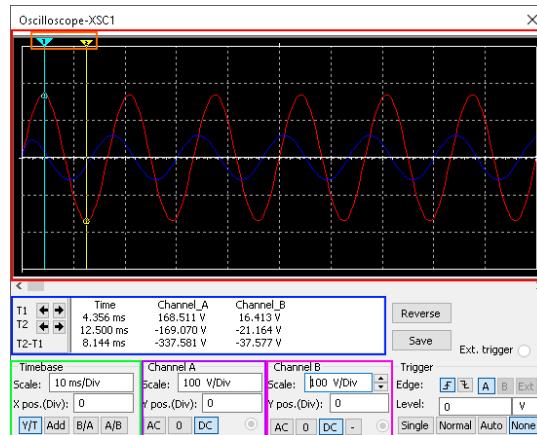


Figure A.18: Oscilloscope Window

Table 17: Oscilloscope Window Descriptions

Color	Label	Description
Red	Display	Displays measured signals.
Blue	Cursor Values	Displays the current value the cursors are measuring.
Green	Timebase	Allows you to change the time scale. Increasing the scale will compress the plot, while decreasing the scale expands it.
Purple	Channel A	Changes the settings of the Channel A signal. The voltage scale refers to the height of the signal. Increasing the scale shrinks the vertical height of the signal, while decreasing the scale increases the height of the signal.
Violet	Channel B	Changes the settings of the Channel B signal. The voltage scale refers to the height of the signal. Increasing the scale shrinks the vertical height of the signal, while decreasing the scale increases the height of the signal.
Orange	Cursors	Allows you to measure specific values at any time in the signal. They are initially on the far left of the display window and can be dragged anywhere along the signal.

A.5.5 Using the Bode Plotter

Bode Plotters are used when examining the output of a filter. We will again be using the basic AC circuit example. The Bode Plotter is accessed through the Bode Plotter tab from the measurement toolbar in Figure A.3. The Bode Plotter has two channels; an input and an output channel. Both must be used correctly in order to create accurate Bode plots. This can be seen in Figure A.19.

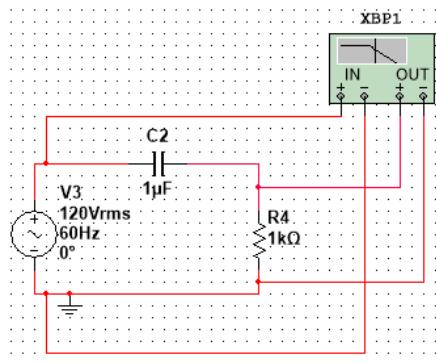


Figure A.19: Bode Plotter Example

Double clicking on the Bode Plotter results in the windows seen in Figure A.20. There are two types of Bode plots; a magnitude plot that displays the gain of the filter in decibels (dB), and a phase plot that displays the phase of the filter in degrees (°). You may change the final and initial values of the horizontal and vertical access to zoom in or out of the plots.

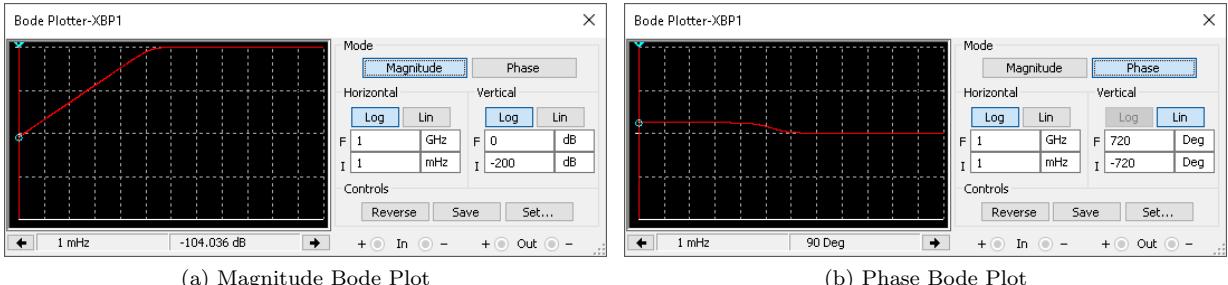


Figure A.20: Bode Plotter Results

A.5.6 Interpreting TTL Results

TTL is a relatively short topic in this course, and as such, does not need extensive explanation on how to build and understand results in MultiSim. However, it does require some niche components that are not used anywhere else.

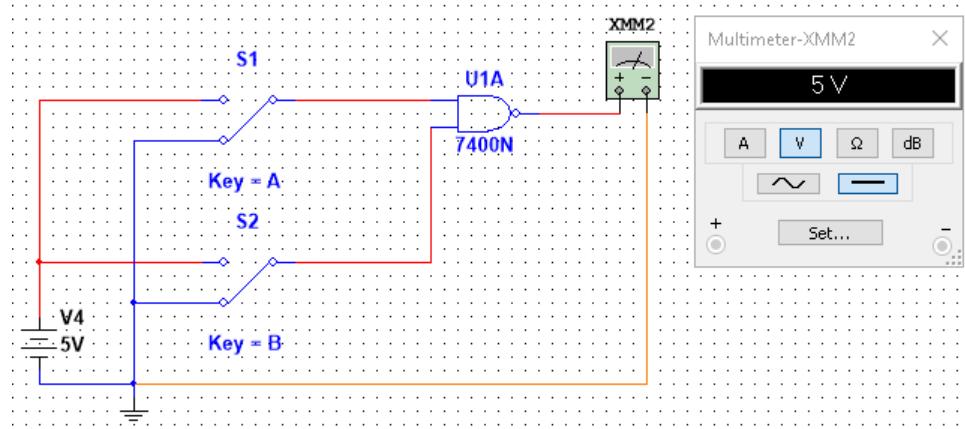


Figure A.21: TTL Example

Looking at Figure A.21, there are a few things to note:

- a) TTL requires the usage of binary for inputs and outputs. In MultiSim, a 1 corresponds to a 5 V power source, and a 0 corresponds to ground. This means that if 5 V is being sent into a gate, it is receiving a 1 as an input, whereas an input being connected to ground is equivalent to 0.
- b) TTL uses switches to flip between 1's and 0's. Switches can be found under the Basic components tab, where other passive components are found. There is a "SWITCH" family, and you will be using the SPDT switch. This can be rotated with CTRL + R, and the key flips the state of the switch. You can change the key at any time by clicking on the switch.
- c) Multimeters are used to measure the output of logic gates. The output of the gate goes into the positive terminal of the multimeter, and the negative terminal is connected to ground. Likewise with the input, a 5 V output corresponds to a 1, and a 0 V output corresponds to a 0.

A.6 Presentation of Results

When submitting results, there are some general guidelines to help improve clarity. This is important because results that are clear and easy to read are easier to comprehend for the reader, and will benefit you.

A.6.1 Presenting Numerical Results

Numerical results are relatively easy to present from MultiSim. The quickest method to present results is to open any results windows, like multimeters, and screenshot the main design window using tools like the Snipping Tool in Windows, and COMMAND + SHIFT + 4 in Mac.

A.6.2 Presenting Graphs

Graphs are a more complex result to submit, as by default, the background is black and there is much of the graph interface that does not need to be submitted. As such, there is a separate grapher tool in MultiSim that allows for adjustments. This can be accessed through the grapher button seen in Figure A.22.



Figure A.22: Grapher Button in MultiSim

We will be using the basic AC circuit for this example. Clicking on this button results in the window seen in Figure A.23. This is a zoomed out display of what is seen in the oscilloscope. It should be noted that the Grapher should be used when the simulation is either paused or stopped.

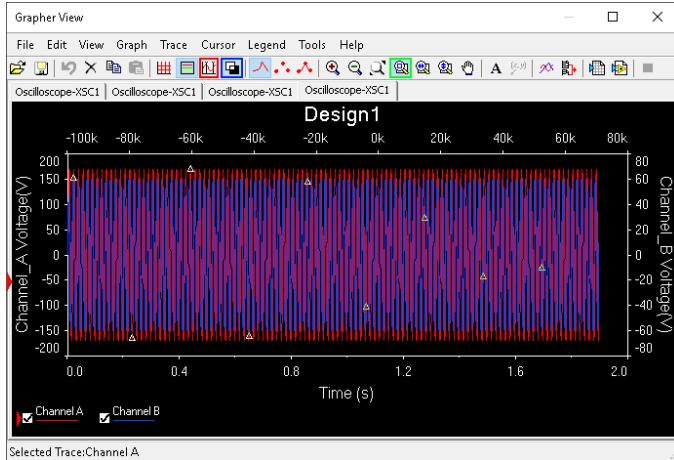


Table 18: Descriptions of Graph Window

Color	Label	Description
Red	Cursors	Opens up cursors on the display.
Blue	Background	Changes the background color between black and white.
Green	Square Zoom	Allows you to draw a box on the plot and zoom in on it.

Figure A.23: Grapher Window

An example graph output for submission with cursors should look something like Figure A.24. This format eliminates the black background and zooms in on only four or five waveforms. Cursor values are also easily seen.

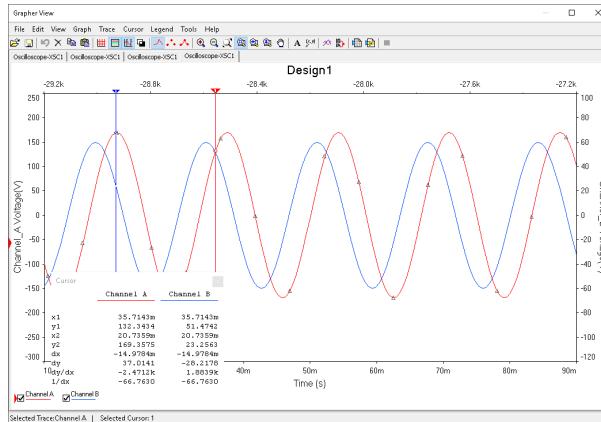


Figure A.24: Example Output for Grapher

B Appendix B: Breadboard Basics

B.1 How to use the Breadboard

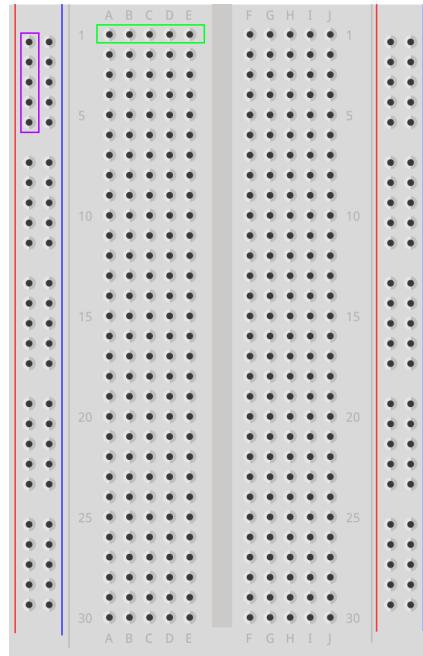


Figure B.1: Breadboard Layout

Figure B.1 shows a small example of a breadboard, one of the main tools you'll be using throughout the semester. These breadboards are where all of your components will go, and as such, learning to use them is of vital importance. While they may look somewhat intimidating at first, simple guidelines on the breadboard are available in Table 19. Examples of how to use the breadboard are also available in Appendix B.5.

Table 19: Descriptions of Breadboard Parts in Figure B.1

Color	Label	Description
Red	Positive Vertical	The red lines on the breadboard indicate a continuous vertical connection along the entire right side lines. If there are any breaks in the line, the vertical connection breaks as well.
Blue	Negative Vertical	The blue lines on the breadboard indicate a continuous vertical connection along the entire left side lines. If there are any breaks in the line, the vertical connection breaks as well.
Green	Horizontal Connection	The 5 holes in a horizontal row indicate a single node. That is, all five holes inside the green box are connected to each other, but not connected to anything else.
Purple	Vertical Connection	The 5 holes in a vertical row indicate a single node. That is, all five holes inside the purple box are connected to each other, but not connected to anything else.

B.2 Passive Components

In lab, there are many different components you will deal with. The first set of components are called passive components, and consist of resistors, capacitors, and inductors.

B.2.1 Resistors

The first component dealt with also happens to be the most common component you will see, and is called a resistor. Resistors are described by their resistance, which has units of Ohms (Ω). There are a variety of resistors that are available, some examples of which can be seen in Figure B.2.

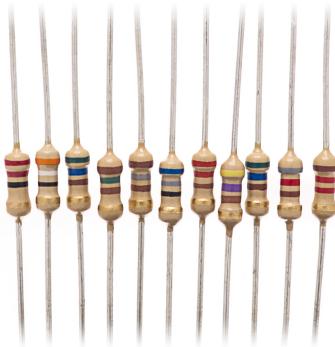


Figure B.2: Different Resistor Examples

It should be noted that resistors are physically different from each other by the color schemes on the components. These correspond to different numbers, and each combination represents a different resistance value. All possible combinations may be read off of a resistance color code chart, much like the one seen in Figure B.3. Some examples of common resistors and their associated color code are seen in Table 20.

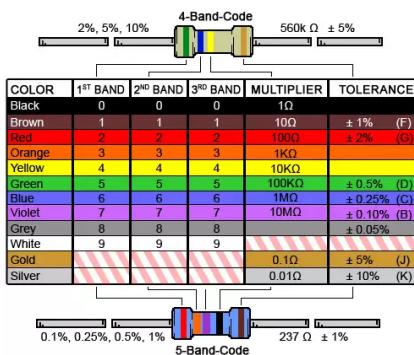
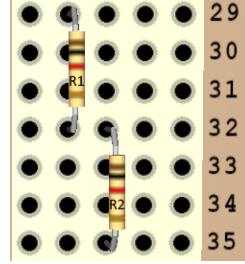
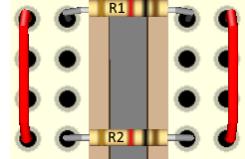
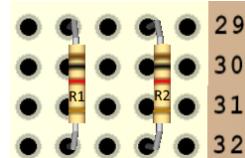


Table 20: Example Resistor Colors

Figure B.3: Resistor Color Code Chart

However, not all resistors are available to you in lab. In the event you need a specific resistance value but do not have a resistor with that exact value, you must utilize the series and parallel properties of resistors. These are described in Table 21.

Table 21: Resistor Connections

Image	Type	Description
	Horizontal Series	$R_{tot} = R_1 + R_2$
	Vertical Series	$R_{tot} = R_1 + R_2$
	Horizontal Parallel	$\frac{1}{R_{tot}} = \frac{1}{R_1} + \frac{1}{R_2}$
	Vertical Parallel	$\frac{1}{R_{tot}} = \frac{1}{R_1} + \frac{1}{R_2}$

B.2.2 Capacitors

The second most common passive component is the capacitor. Capacitors store energy using electric fields, and are defined by their capacitance, which has units of Farads (F). Capacitors are much more easily differentiated from each other, as can be seen in Figure B.4.



Figure B.4: Different Capacitor Examples

Like resistors, putting capacitors in series or parallel will result in components that have different capacitances. However, capacitors in series and parallel obey different rules than resistors. This is seen in Table 22.

Table 22: Capacitor Connections

Image	Type	Description
	Horizontal Series	$\frac{1}{C_{tot}} = \frac{1}{C_1} + \frac{1}{C_2}$
	Vertical Series	$\frac{1}{C_{tot}} = \frac{1}{C_1} + \frac{1}{C_2}$
	Horizontal Parallel	$C_{tot} = C_1 + C_2$
	Vertical Parallel	$C_{tot} = C_1 + C_2$

B.2.3 Inductors

The final passive component you will deal with in lab is the inductor. Like capacitors, inductors store energy, but in the form of magnetic fields instead of electric fields. Inductors are identified by their inductance, which has units of Henries (H).

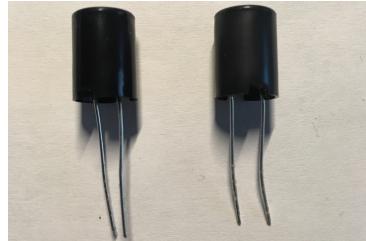


Figure B.5: Inductor Examples

Inductors take many different forms, but the primary ones you will be using are seen in Figure B.5. Putting inductors in series or parallel results in the same effect as in resistors, and can be seen in Table 23.

Table 23: Inductor Connections

Image	Type	Description
	Horizontal Series	$L_{tot} = L_1 + L_2$
	Vertical Series	$L_{tot} = L_1 + L_2$
	Horizontal Parallel	$\frac{1}{L_{tot}} = \frac{1}{L_1} + \frac{1}{L_2}$
	Vertical Parallel	$\frac{1}{L_{tot}} = \frac{1}{L_1} + \frac{1}{L_2}$

B.2.4 Potentiometers

Potentiometers are variable resistors; that is, you may tune potentiometers to have any resistance value ranging from 0Ω to a set maximum resistance, generally either 500 or 5000 Ω . The two types of potentiometers you will potentially see are either horizontally or vertically oriented, as seen in Figures B.6a and B.6b, respectively.

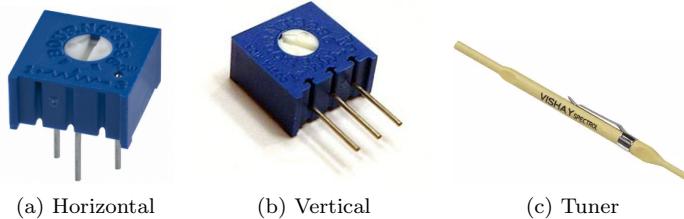


Figure B.6: Types of Potentiometers and Tools

It should be noted that each potentiometer has three prongs, which differentiates it from the other two terminal components that were previously looked at. Each prong of the potentiometer must go into its own node. In order to tune the potentiometer to a specific resistance, a tuner must be used, as seen in Figure B.6c. This device has a metal tip that fits into the slot of a potentiometer, allowing you to adjust the resistance.

B.3 Operational Amplifiers

Operational Amplifiers, commonly known as Op Amps, are the first of the integrated circuits you will encounter in lab. Op amps are significantly more complex than the two-terminal components, with 8 pins, and require careful attention when using them. It should be noted that each pin of an op amp must have its own row on a breadboard. An example of this is seen in Figure B.7. Each pin of the operational amplifier corresponds to a different purpose, which can be seen in the pin diagram in Figure B.8.

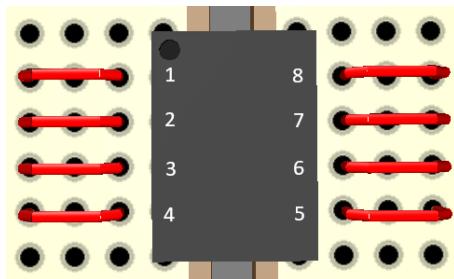


Figure B.7: Op Amp on Breadboard

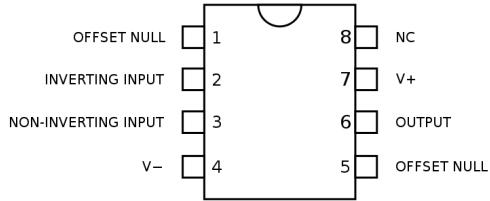


Figure B.8: Op Amp Pin Diagram

B.4 Logic Gates

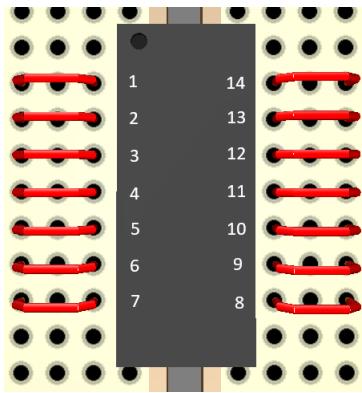


Figure B.9: Logic Gate on Breadboard

Logic gates are another integrated circuit you will be using in lab. They are more complex than op amps when wiring, simply due to the number of gates and the amount of pins. Like op amps, each pin must have its own row on the breadboard, as demonstrated in Figure B.9. The different types of logic gates and their corresponding pin layouts are seen in Table 24.

Table 24: Logic Gate Pin Diagrams

Image	Number	Type	Image	Number	Type
	7400	NAND Gate		7408	AND Gate
	7402	NOR Gate		7432	OR Gate
	7404	Inverter			

B.5 Guidelines for Building Circuits

There are several conventions you may find useful when building circuits. These conventions will help you to maintain a relatively organized breadboard, and streamline the circuit building progress.

- Maintain the same shape and component orientation as the circuit diagram. That is, when a component is vertically oriented in a drawing, ensure that the component on the breadboard is also vertically oriented. This should result in you being able to quickly and easily track currents flowing through your circuit. When the circuit is complete, you should be able to look at the circuit from a bird's eye view and see the same circuit that is in the drawing.
- Do not hesitate to use as much of the board as you need. While it may seem simpler at first to use smaller portions of the board to keep everything neat, more often than not you will run into problems trying to fit all the components. There is also an inherent danger in using such little space, as current flowing through tightly coiled wires may overheat.
- Use as many colors as you need. Each kit has been equipped with a myriad of wires in various colors and lengths. Each wire is designed to span a specific distance and fits snugly on the board. Using different colors for different connections allows you to easily trace certain connections during

debugging. Two general conventions are that the positive voltage source is designated by a red wire, and the grounds are designated by a black wire.

- When handling power sources, if you only need one, simply use the rows for the source. If you need more than one power source, such as when using Op Amps, designate one column on the board for positive voltage, and another for ground. This gives you a single location from which all your voltages and grounds span from, since the AD2 only has one positive DC voltage source.

C Appendix C: Waveforms 2015 Basics

C.1 Contents of the Analog Discovery 2

The Analog Discovery 2 (AD2) Box contains four main components you need to be concerned with:

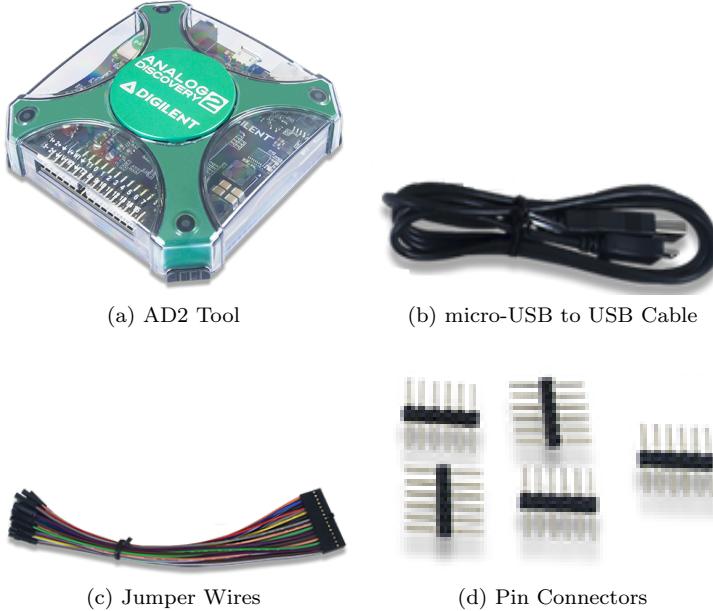


Figure C.1: AD2 Components

- a) The actual Analog Discovery tool. It is a clear gray square with the Digilent logo on it. It is seen in Figure C.1a, and is the brain of the system. This will be referred to as the AD2 for the rest of the appendix.
- b) A micro-USB to USB cable. This will power the Analog Discovery tool. The micro-USB side attaches to the AD2, while the USB is plugged into the computer that you will be using. This is seen in Figure C.1b.
- c) A set of flexible breadboard jumper wires. They are all connected in one bundle, and attached to the bottom of the AD2. Each wire has a specific purpose that is will be explained. These wires are seen in Figure C.1c.
- d) A set of pin connectors. These are used to attach the jumper wires to the breadboard, although you may use normal wires as well. The pin connectors are seen in Figure C.1d.

C.2 Main Window

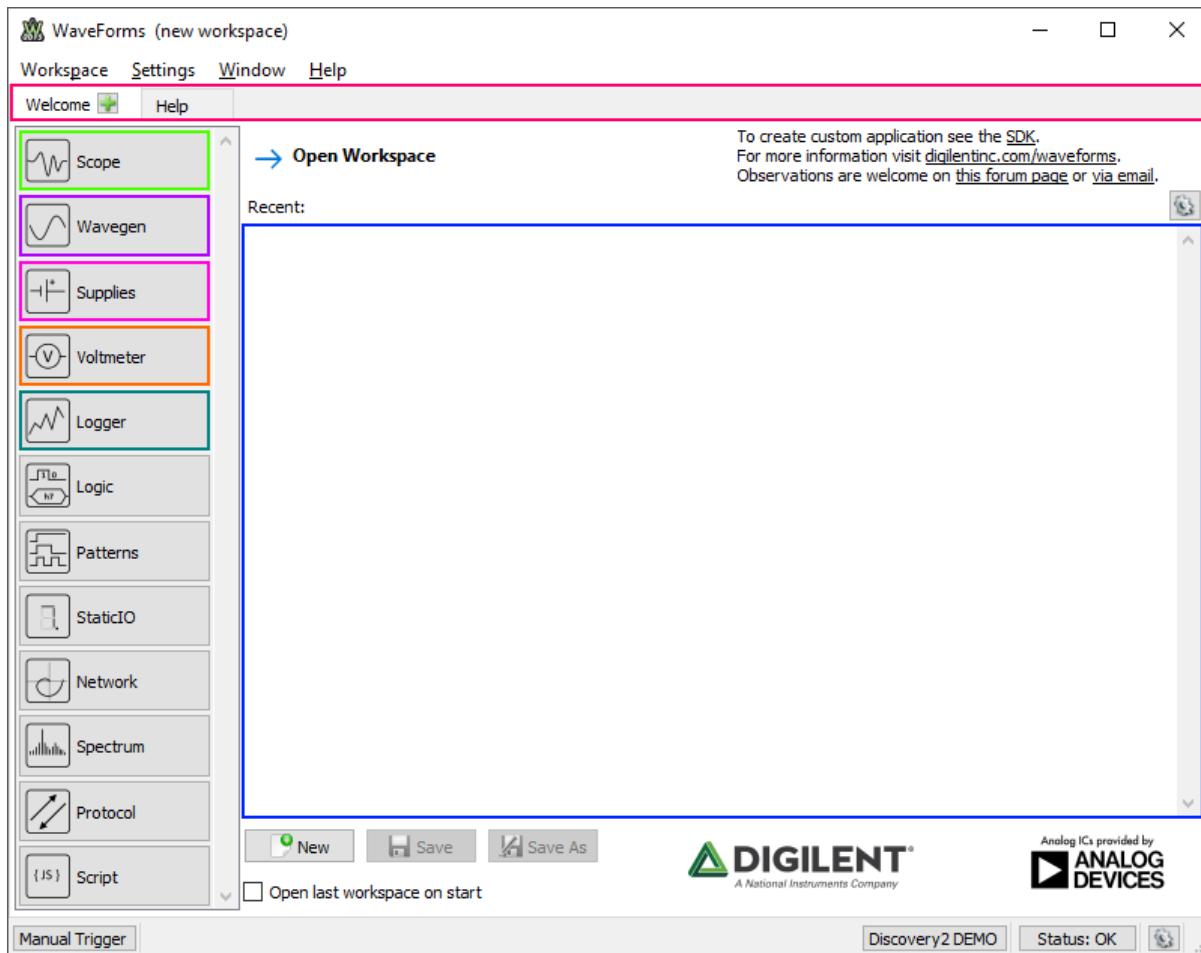


Figure C.2: Main Window of Waveforms 2015

The main window of Waveforms 2015 is seen in Figure C.2. There are a lot of options here, but you are only concerned with a few of them for the purpose of this lab.

Table 25: Waveforms 2015 Window Description

Color	Label	Description
Red	Tabs	Where you will be able to see tabs of all your tools that are currently in use.
Blue	Previous Workspaces	Where you can access previous workspaces for other labs easily.
Green	Scope Tab	Used to access the oscilloscope tool.
Purple	Wave Generator	Used to access the wave generator tool.
Pink	Voltage Supplies	Used to access DC voltage supplies.
Orange	Voltmeter	Used to measure access the voltmeter tool.
Teal	Logger	Used to access the logger tool.

C.3 Power Sources

The two primary types of power sources used in this lab are AC and DC voltages.

C.3.1 DC Voltage

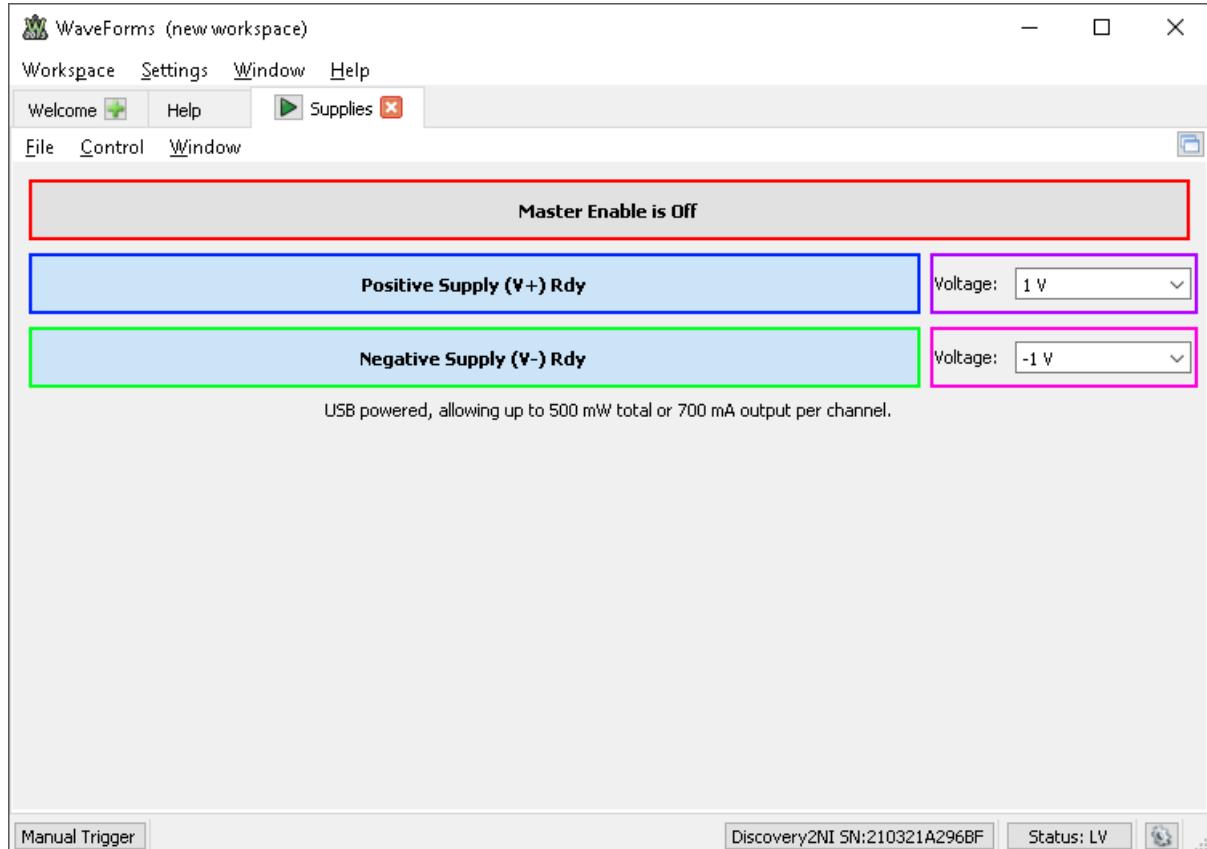


Figure C.3: DC Voltage Window

In order to generate DC voltages, you must use the Supplies tab on the main window, as seen in Figure C.2. This pulls up the screen seen in Figure C.3, which contains three buttons; a positive supply, a negative supply, and a master enable. You will also notice that a tab appears at the top of the screen, where you may start and stop the tool.

Table 26: DC Voltage Window Description

Color	Label	Description
Red	Master Enable	Toggles all voltage sources on and off.
Blue	Positive Voltage Toggle	Toggles the positive voltage source on and off.
Green	Negative Voltage Toggle	Toggles the negative voltage source on and off.
Purple	Positive Voltage Value	Changes the value of the positive voltage source.
Pink	Negative Voltage Value	Changes the value of the negative voltage source.

C.3.2 AC Voltage

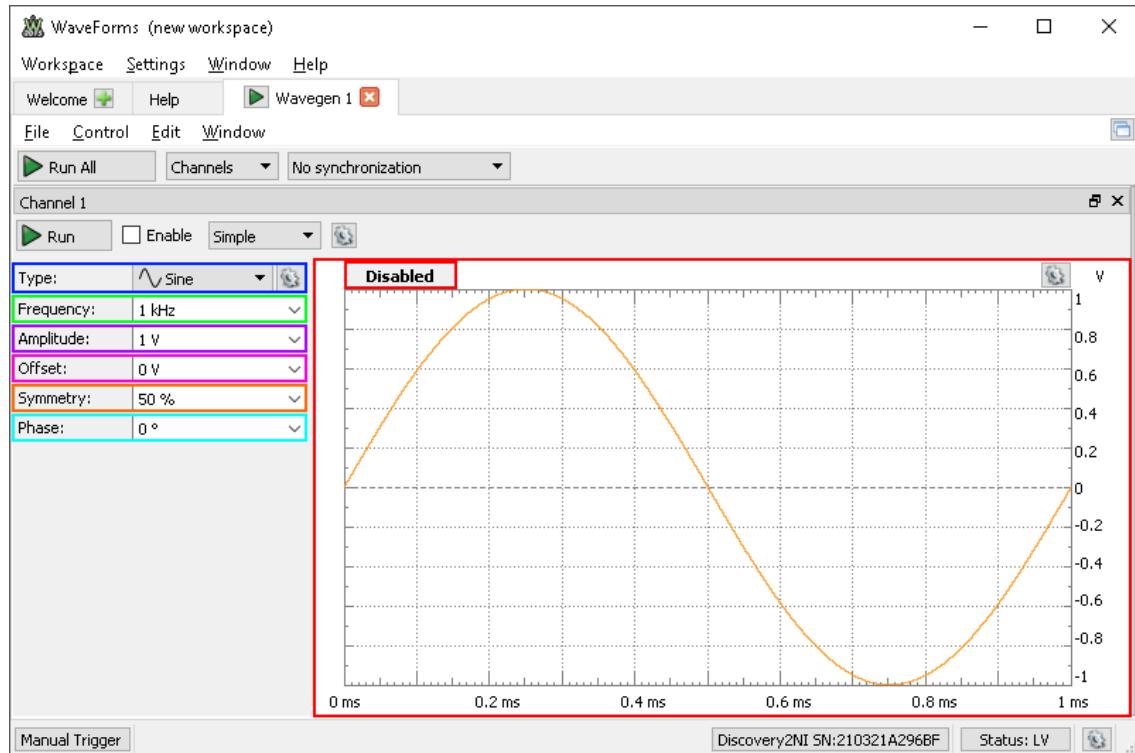


Figure C.4: AC Voltage Window

In order to generate AC voltages, you must use the Wavegen tab on the main window, as seen in Figure C.2. This pulls up the screen seen in Figure C.4, which has many different options for AC signals. It should be noted you can also use this tool to create DC signals, in the event you need more than the two provided. You will also notice that a tab appears at the top of the screen, where you may start and stop the tool.

Table 27: AC Voltage Window Description

Color	Label	Description
Red	Main Window	Space where you will observe the signal shape. You can change the color of the screen by right clicking on it and changing the color.
Blue	Wave Type	Allows you to switch the wave type between many different choices.
Green	Frequency	Allows you to change the frequency of the wave, or how often it oscillates in one period.
Purple	Amplitude	Adjusts the amplitude of the wave, which is the height of both the negative and positive sides of the wave.
Pink	Offset	Adjusts the vertical offset of the wave, or where the zero of the wave occurs on the y-axis.
Orange	Symmetry	Adjusts the symmetry of the wave. 50% symmetry means the first half of the wave is the exactly the opposite of the second half, and vice versa.
Cyan	Phase	Adjusts the phase of the wave, which is the location of the first zero of the wave.

C.4 Making Measurements

The AD2 is capable of measuring both DC and AC values, both of which require different tools.

C.4.1 Voltmeter Tool

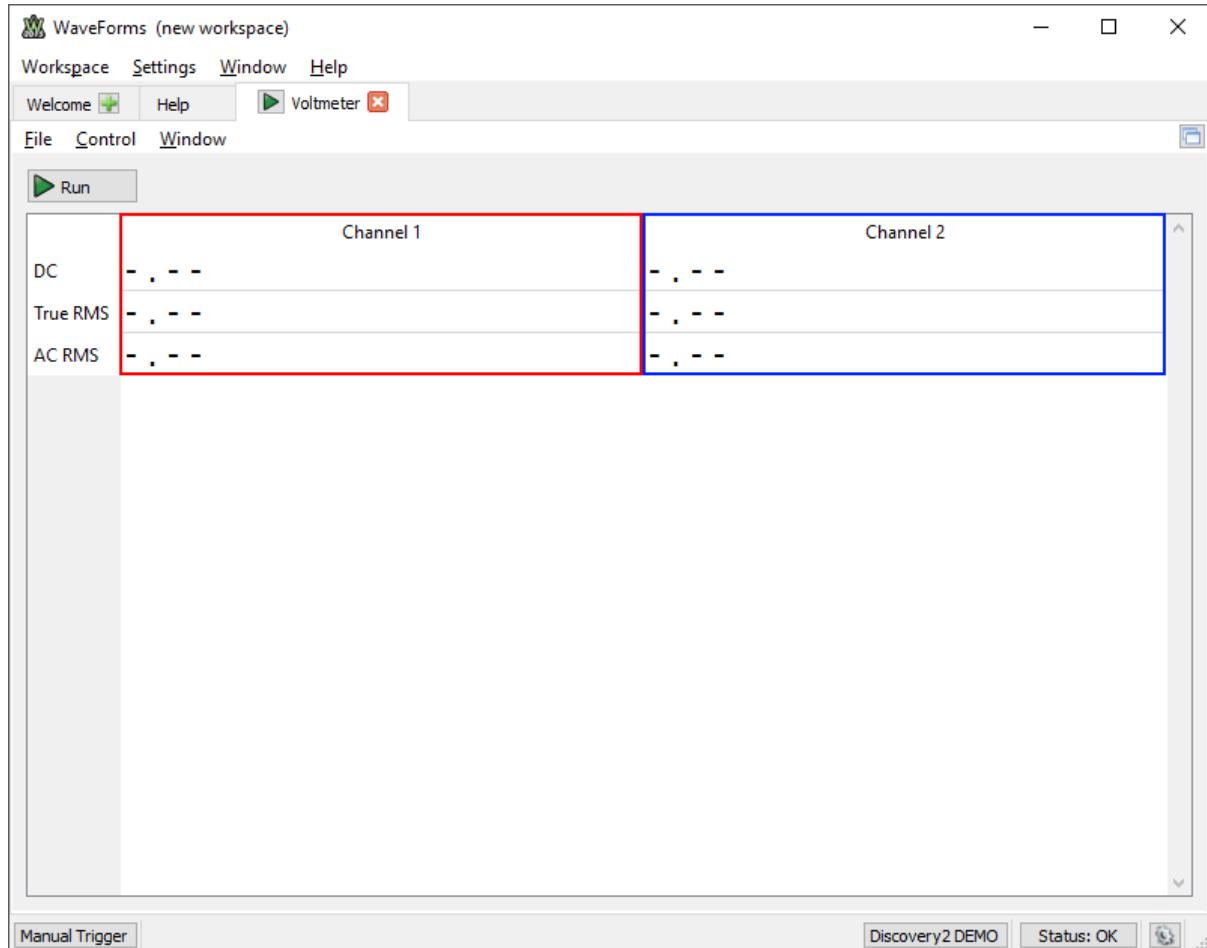


Figure C.5: Voltmeter Window

The AD2 uses the Voltmeter tool to measure DC and AC voltages, which can be accessed through the tab seen in Figure C.2. Clicking on this tab brings up the Logger window, seen in Figure C.5. You will also notice that a tab appears at the top of the screen, where you may start and stop the tool.

Table 28: Voltmeter Window Description

Color	Label	Description
Red	Channel 1 Values	Displays the DC, True RMS, and AC RMS values measured by Channel 1.
Blue	Channel 2 Values	Displays the DC, True RMS, and AC RMS values measured by Channel 2.

C.4.2 Logger Tool

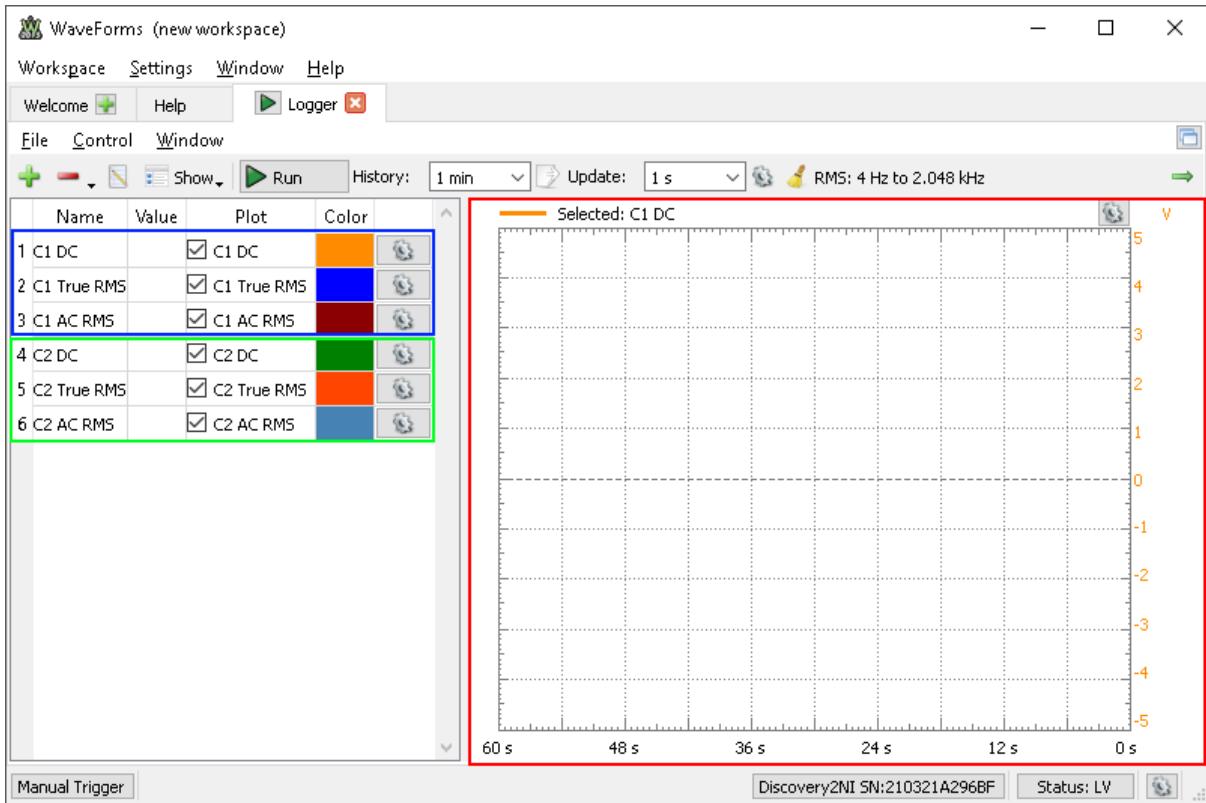


Figure C.6: Logger Window

The AD2 uses the Logger tool to measure DC and AC voltages, which can be accessed through the tab seen in Figure C.2. Clicking on this tab brings up the Logger window, seen in Figure C.6. You will also notice that a tab appears at the top of the screen, where you may start and stop the tool.

While this tool performs the same functions as the previously mentioned Voltmeter tool, the Logger tool allows you to observe the evolution of certain values over time. In instances where you need only measure specific values, you may use the Voltmeter tool. If you need to be able to explain how a certain value changes over time, the Logger tool may help to provide more insight.

Table 29: Logger Window Description

Color	Label	Description
Red	Main Window	Space where you will see the measured value over time. You can change the color of the screen by right clicking on it and changing the color.
Blue	Channel 1 Values	Displays the DC, True RMS, and AC RMS values associated with Channel 1.
Green	Channel 2 Values	Displays the DC, True RMS, and AC RMS values associated with Channel 2.

C.4.3 Oscilloscope

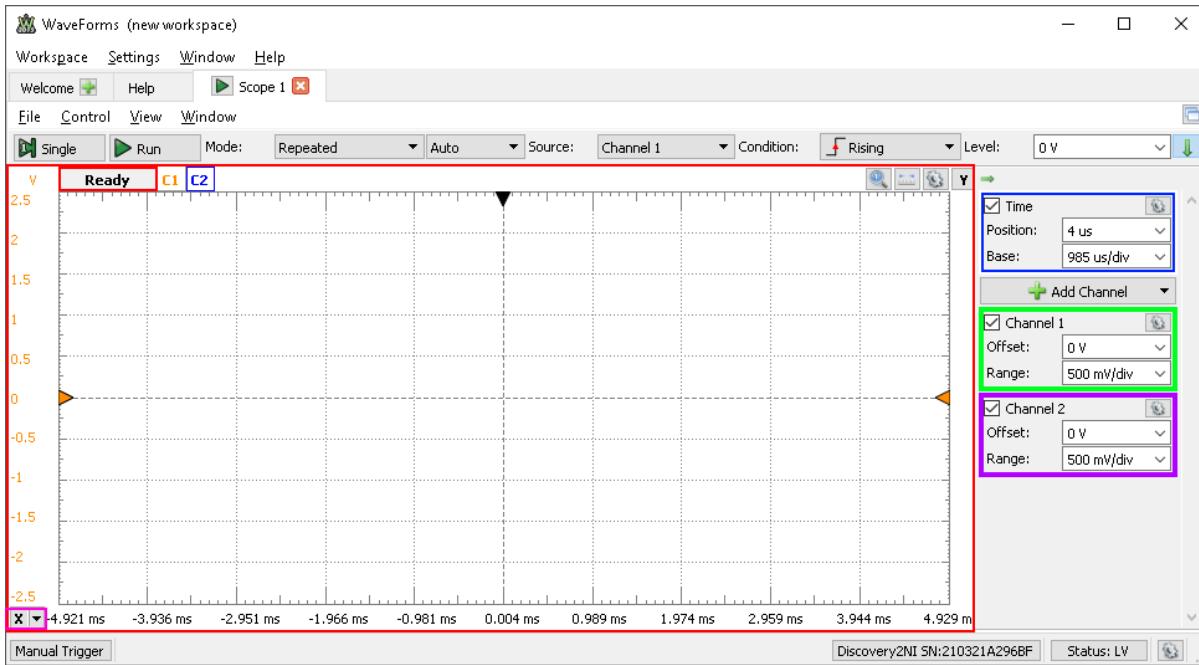


Figure C.7: Oscilloscope Window

The AD2 has a built in Oscilloscope tool that is used to measure AC values. It is accessed through the Scope tab in Figure C.2. This brings up a window as seen in Figure C.7.

Table 30: Oscilloscope Window Description

Color	Label	Description
Red	Main Window	Space where you will observe all signals. You can change the color of the screen by right clicking on it and changing the color.
Blue	Time Scale	Allows you to change scale of the x-axis and move the screen along the signal. Increasing the scale compresses the plot, while decreasing the scale expands the signal.
Green	Channel 1 Scale	Allows you to change the scale of the y-axis and move the screen along the signal for Channel 1. Increasing the scale decreases the height of the plot, while decreasing the scale increases the height of the plot.
Purple	Channel 2 Scale	Allows you to change the scale of the y-axis and move the screen along the signal for Channel 2. Increasing the scale decreases the height of the plot, while decreasing the scale increases the height of the plot.
Pink	Cursors	Clicking this button will create a cursor that can be dragged along the signal. Clicking the button multiple times will create multiple cursors.

C.5 Example Circuit Measurements

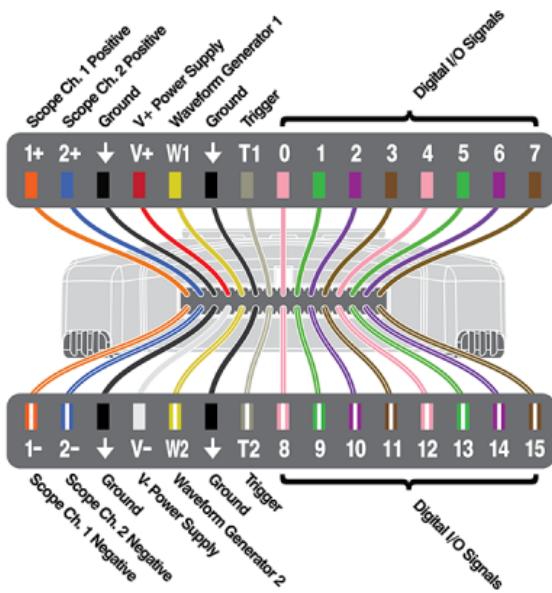


Figure C.8: Jumper Wire Labels

C.5.1 Example DC Circuit

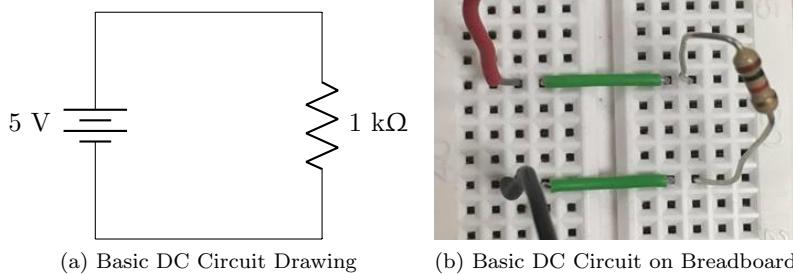


Figure C.9: Basic DC Circuit

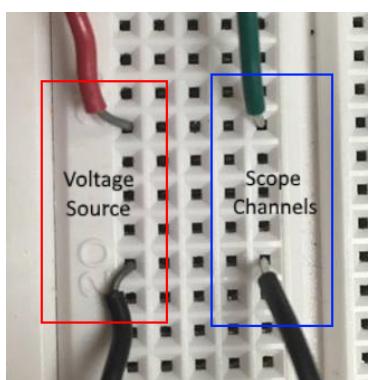


Figure C.10: Voltage Source Measurement

In order to illustrate the use of the AD2, we will use an example and show how measurements are made in lab. First and foremost is that there are a lot of jumper wires, but you only need a select few. They are labeled in Figure C.8. It should be noted that you do not need to use any of the Digital I/O Signals. Instead focus on the left half of the wire diagram.

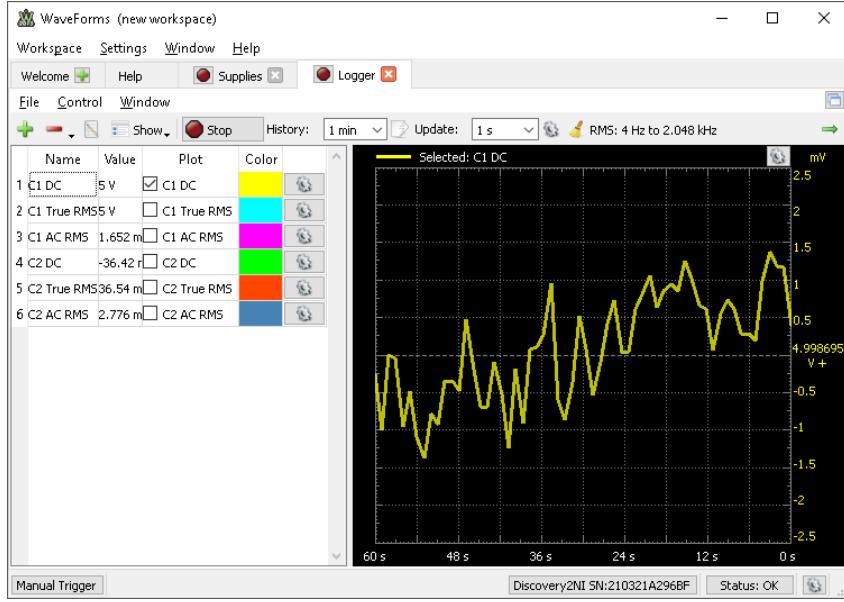


Figure C.11: Waveforms 2015 Logger Window for Voltage Source

At first glance, the waveform in Figure C.11 looks very erratic and does not seem stable. However, it is important to look at the scale on the right side of the window. From there, we see that the waveform only oscillates on the scale to 2 mV. Thus the resulting output of the DC power supply is extremely close to 5 V, but not exactly 5 V. The other measurement that can be made in this basic example is measuring the voltage across the resistor. This is done using the same logger tool as previously used, and an example circuit with the scopes is seen in Figure C.12. This results in the window seen in Figure C.13.

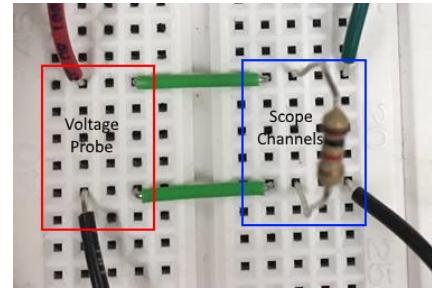


Figure C.12: Resistor Voltage Measurement

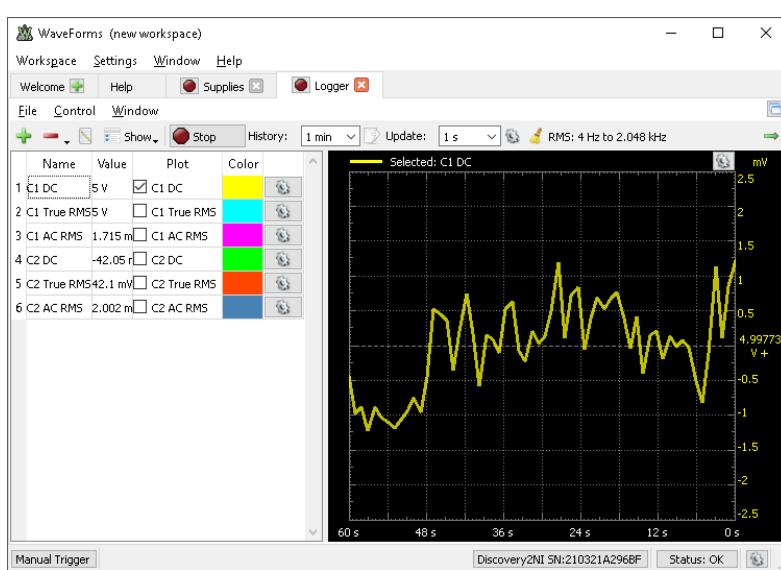


Figure C.13: Waveforms 2015 Logger Window for DC Circuit

C.6 Digital Multimeter

As powerful as the AD2 is, there are some measurements that it is not equipped to make that you will need for this lab. Instead you will need to make these measurements using a digital multimeter (DMM). These devices are not as small as the AD2, but they are still very practical. The DMM requires the use of prongs, seen in Figure C.14. The metal tips of these prongs are placed on the terminals of whatever component or wire you are measuring. The DMM itself is seen in Figure C.15, and has a myriad of functions.



Figure C.14: Digital Multimeter Prongs

Table 31: Descriptions of DMM Options



Figure C.15: Digital Multimeter

Color	Label	Description
Red	Ohms	Sets the DMM to measure resistance.
Blue	Diodes/Capacitors	Sets the DMM to measure diode voltage drop or capacitance.
Green	Current	Sets the DMM to measure current.
Purple	Switcher	Switches between white and yellow options on the dial.
Pink	Current Socket	The red prong must be plugged in here if you wish to measure current.
Orange	Ground Socket	The black prong must always be plugged in here.
Cyan	Non-Current Socket	The red prong must be plugged in here if you wish to measure anything besides current.

C.6.1 Measuring Resistance

In order to measure the resistance of a resistor, simply tap the metal leads of the DMM prongs to either end of a resistor that is **isolated** from any power sources or other components. An example of this can be seen in Figure C.16. Measuring resistance is important due to the imperfect nature of the components you will be using in lab. As an example, measuring the resistance of the resistor in Figure C.16 results in a resistance of $.944\text{ k}\Omega$, despite it being labeled as a $1\text{ k}\Omega$ resistor.



Figure C.16: Measuring Resistance Example

C.6.2 Measuring Capacitance

Likewise with a capacitor, you must measure the capacitance because despite labels, actual measured values of capacitance can vary greatly. The same principle for resistors applies to capacitors; the metal ends of the prongs touch the terminals of the capacitor, like in Figure C.17. From this result, we see that the capacitance is $90.87 \mu\text{F}$, which is different from the labeled value of $100 \mu\text{F}$.

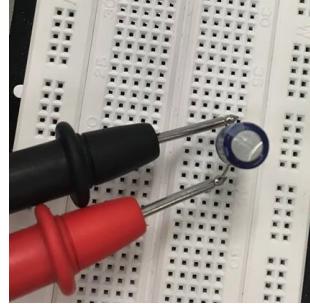


Figure C.17: Measuring Capacitance Example

C.6.3 Measuring Inductance

Unfortunately, the DMM cannot measure inductance. As a result, a secondary device, known as a fluke meter, must be used. The fluke meter can be seen in Figure C.18. While seemingly complex, you only need to use it for one purpose, which can be achieved easily. After the machine is turned on, simply tap the two terminals of the inductor to the black and metal prongs on the bottom right of the fluke meter. The display will then show the inductance.



Figure C.18: Fluke Meter

C.6.4 Measuring Current

In order to measure current, you must apply the same principle as when you are measuring current using MultiSim. Using the basic DC circuit example in Figure A, we replace one of the connecting wires with two separate wires, as seen in Figure A. Current must flow through the white wire into the black wire. Thus we complete the circuit using the prongs of the DMM. It should be noted that the circuit is incomplete without using the digital multimeter to measure current; without a connection between the white and black wires, no current can actually flow through the circuit. The resulting accurate current can be seen in Figure C.19b.

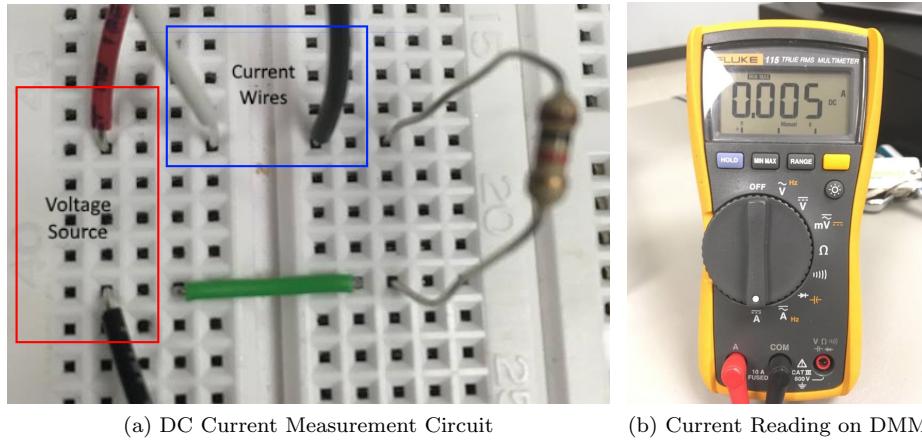


Figure C.19: DC Current Measurement with DMM

C.7 Presentation of Results

Much like in MultiSim, how you present your results is almost as important as what the results are themselves. Clear results make for easier interpretation, and therefore, make it significantly easier for your graders to provide constructive feedback.

C.7.1 Presenting Numerical Results

Numerical results are simple to present. Simply write down or type up the numbers that you read off the screen. While you may screenshot results, in lab it may be more space efficient for your reports to type up the results instead. However, be sure to be very clear about what units you use.

C.7.2 Presenting Graphs

Like MultiSim, Waveforms defaults the background color of the graphs to black. While this improves visibility on the screen, when printing the plots, a black background is both a waste of ink and makes it incredibly difficult to discern specific differences between waveforms in some situations. As a result of this, all plots that are submitted should have a white background.

Figure C.20 shows the default window of Waveforms when measuring a signal using the oscilloscope. When you right click anywhere on the waveform window, a small window will pop up. The most important thing to note is the existence of a "Color" option. By default, this option is set to dark, and you may switch it to "Light." This results in the plot seen in Figure C.21. From here, you may screenshot or snip this waveform, and printing will result in a much easier to read plot.

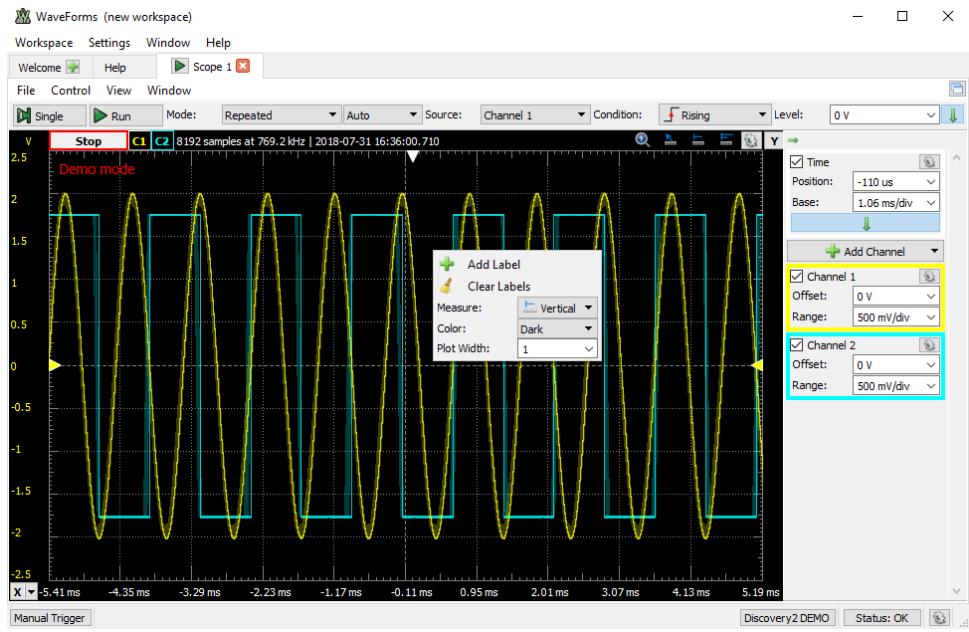


Figure C.20: Oscilloscope with Dark Background

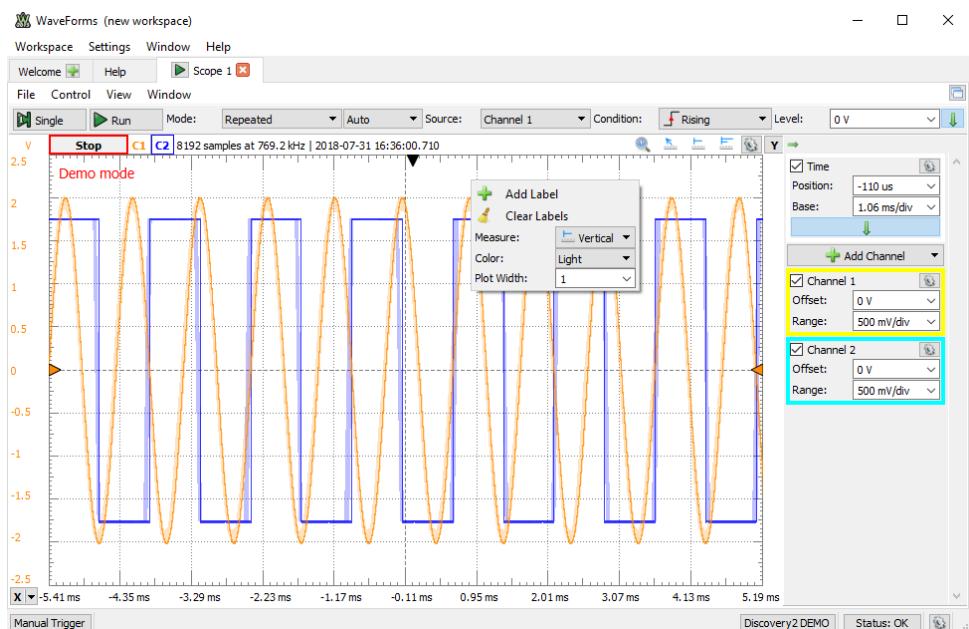


Figure C.21: Oscilloscope with Light Background

D Appendix D: Introduction to Microcontrollers

D.1 Computers, Processors, and Microcontrollers

A computer is a system (device, machine, etc.) which processes information. Thus, a computer could be an abacus, a computer, or a human. A simplified block diagram of a computer is shown in Figure D.1.

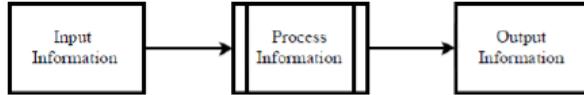


Figure D.1: Simple Block Diagram of a Computer

Modern computers are mainly digital, electronic devices and require many input and output devices (peripherals). An input peripheral might be a keyboard, a touchpad/mouse, a touchscreen, or a mouse. Output peripherals might be monitors or speakers. Any information this computer processes must be in a format which is compatible with that computer. This compatible information is data, typically in the form of digital (binary) and electronic signals.

The actual data processing is mainly carried out by a single Integrated Circuit (IC), or "Microprocessor," or by a set of microprocessors working together. Note that a microprocessor needs many subsystems working together (different types of memory, different levels of software, and input/output peripherals) to be part of a functional computing system.

A microcontroller is a microprocessor with only basic peripherals and memory. See Figure D.2a for a simple block diagram of a microcontroller. It is much more limited in capability, but has the advantage of being compact in size, with a low power requirement compared to typical, modern computers. A microcontroller, also known as Micro-Controller Unit (MCU), may be embedded into other devices, such as household appliances, automobiles, or robots, to make them more functional. See Figure D.2b for some simple diagrams of MCU's being embedded in appliances.

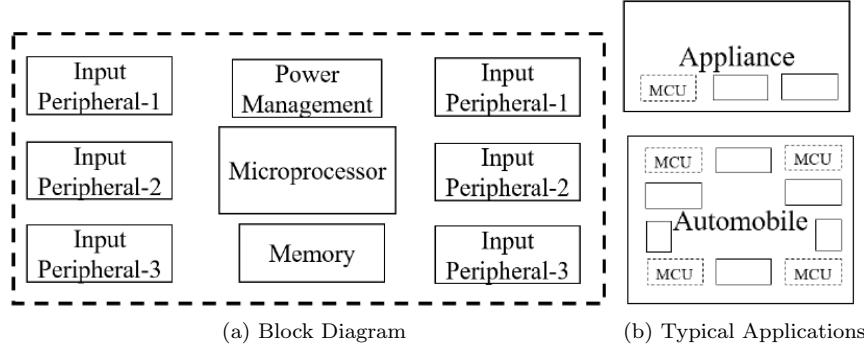


Figure D.2: Microcontroller Diagrams

Since MCU's have very limited hardware compared to typical computers, they also have much more limited software capabilities. This is an advantage for MCU's, since embedded MCU's do not need a lot of space for keyboards or monitors, and do not need advanced operating systems or software. MCU's are designed to run as standalone programmable computing systems. On the other hand, the limited hardware and software can be disadvantageous, as MCU's are not user-friendly and capable as compared to typical computers.

A typical microcontroller does not have, or even need, its own operating system, and does not have a "Run" command. A typical microcontroller simply executes whatever program it has in its program space (of non-volatile memory) which is typically of the structure shown in Figure D.3. The "Global Initialization" block contains the declaration and initialization of global variables, and includes any special hardware libraries needed (for particular sensors or actuators). The "Setup Function" block initializes any of the hardware resources needed. Note that the "Global Initialization" and "Setup Function"

block both execute only once, generally when the microcontroller is powered on or reset. The "Loop Function" block contains the "meat" of the microcontroller's program, repeating indefinitely until the microcontroller is powered down or reset.

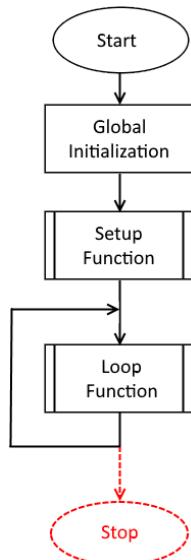


Figure D.3: Generic MCU Program Flowchart

D.2 MSP432 LaunchPad - Hardware Overview

This lab uses a specific MCU platform, known as the MSP432 LaunchPad, which is manufactured by Texas Instruments. A photograph of this device is shown in Figure D.4. You may visit http://energia.nu/pin-maps/guide_msp432p401r/ or <http://www.ti.com/tool/MSP-EXP432P401R> for more information.

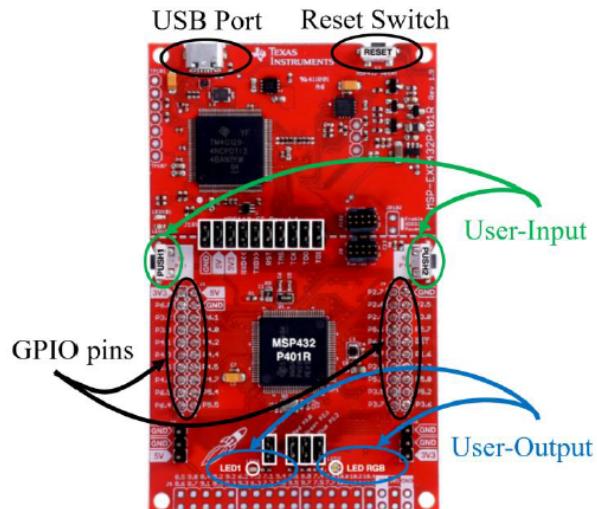


Figure D.4: MSP432 LaunchPad Hardware Layout

When looking at Figure D.4, there are a few things to consider:

- Orientation and Handling
 - Orient the LaunchPad as shown in Figure D.4. That is, ensure that the Texas Instrument Logo is at the top of the board, and the majority of the white labels were right side up.
 - Keep the LaunchPad on a level surface, away from conductive materials and fluids.
 - When holding the LaunchPad, hold it by the edges.
- USB Port
 - Power and communication is provided by a USB cable connecting a host computer USB port to this micro-USB port on the LaunchPad.
 - This micro-USB port is delicate, so handle it carefully.
 - If you need to temporarily remove power from the LaunchPad, disconnect the other end of the USB cable.
- RESET
 - User reset is provided by a pushbutton switch actuated from the top side of the LaunchPad.
 - Use this switch if you need to "reboot" the LaunchPad.
- **User-Output** is provided by means of LED1 (a single red LED) and LED2 (a three-color, RGB LED). These devices are useful indicators of a program while it is running.
- **User-Input** is provided by means of PUSH1 (a pushbutton switch, actuated from the Left side of the LaunchPad) and PUSH2 (a pushbutton switch, actuated from the Rights ide of the LaunchPad). These devices are to prompt a program to change execution while it is running.
- The **GPIO (General-Purpose Input/Output) pins** of a microcontroller provide the user access to connecting other devices to it. These are discussed in Section D.2.1.

D.2.1 MSP432 LaunchPad Pin Map

A pin map is a table of MCU pins, their physical locations, their hardware names (also known as schematic labels), their software names (also known as software addresses), and their available functions. The microcontroller's pins provide the means for users to connect physical devices to the microcontroller. Thus, a pin map is truly the "map" for connecting the MCU hardware and software to physical applications. A pin map for the MSP432 LaunchPad can be seen in Figure D.5, but is also available online at http://energia.nu/pin-maps/guide_msp432p401r/.

Note that the physical locations of the MSP432 LaunchPad pins are listed in white-on-black text, while the hardware names are color coded depending on the pin's programmable functions. As an example, pin P6_0, located at J1,2, is shaded green because it is only capable of using the "Digital Read" and "Digital Write" functions, whereas P3_2, located right below at J1,3, is shaded purple because it is capable of "Analog Write" along with "Digital Read" and "Digital Write" functions.

pins with specific hardware names are listed in white-on-gray text, in addition to being color-coded by pin label. For example, P1_1 is also listed as PUSH1 because it is connected to the Left pushbutton switch as previously described. While a pin map may seem overwhelming at first, it is essential to connecting a microcontroller to various applications using physical devices and as you use the device more, you will be more familiar with it.

D.3 Energia - Software Overview

This lab uses Energia to program the MSP432 LaunchPad. Energia is one of many forms of Embedded-C, which is a C-language platform with additional libraries that enables the microcontroller to interface directly with physical devices such as sensors, actuators, or other microcontrollers. "Arduino" is another popular version of Embedded-C which runs many Arduino MCU's.

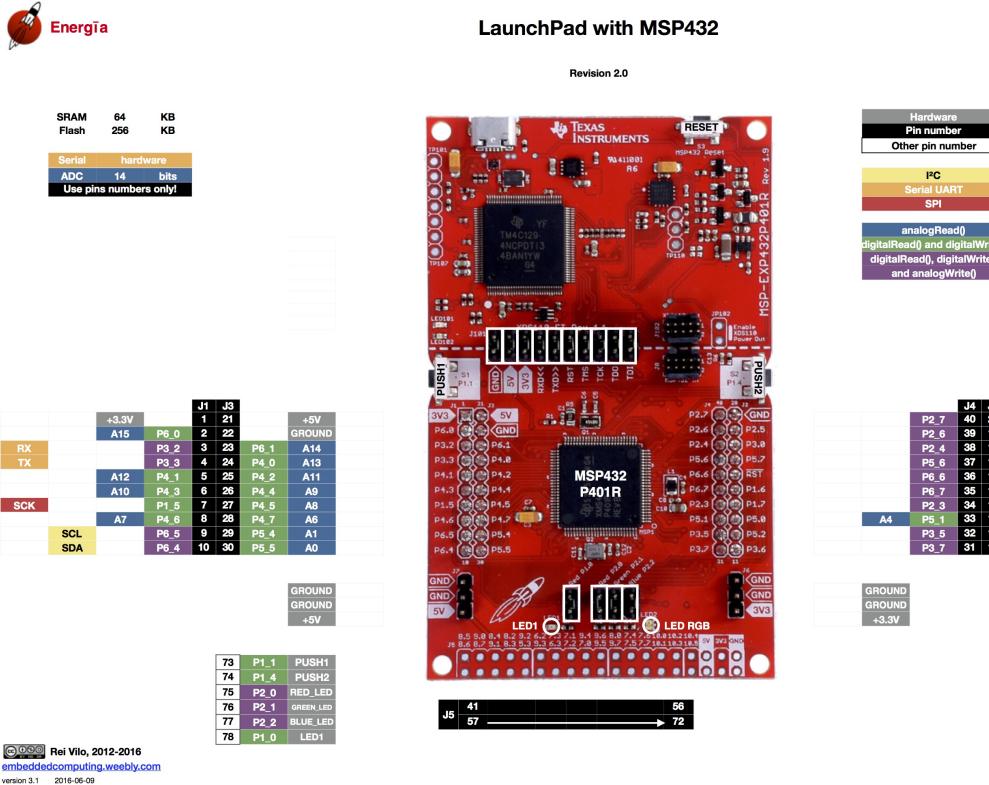


Figure D.5: MSP432 LaunchPad Pin Map

The interface of Energia's programming environment is seen in Figure D.6. Much more information, including video tutorials and documentation is available both online at <http://energia.nu/>, and in the installation folder, under the "references" sub-folder.

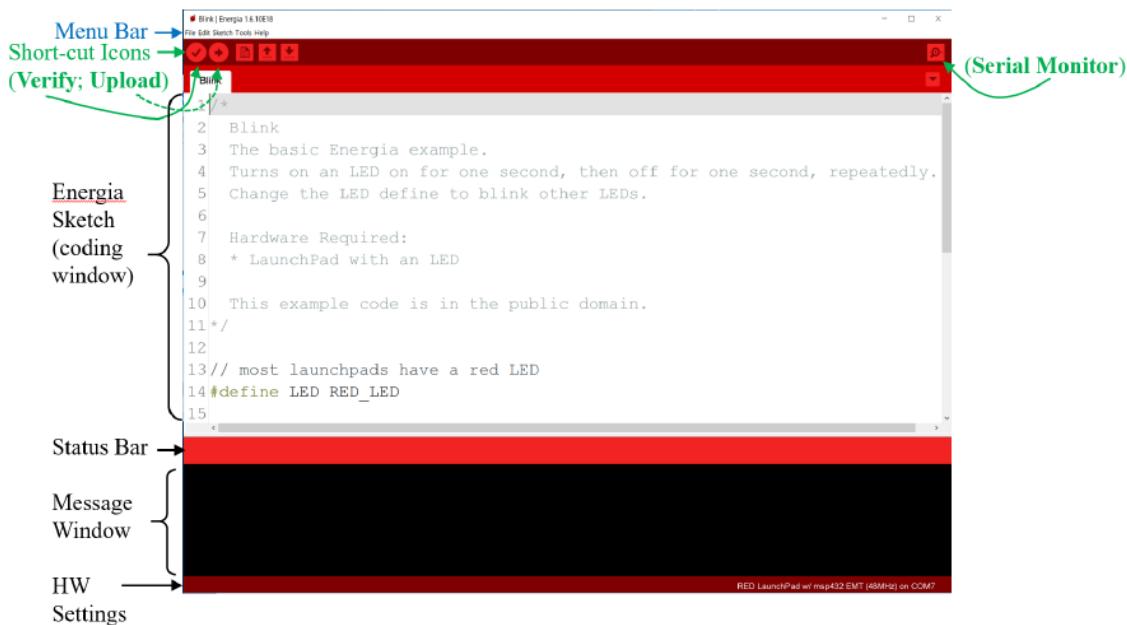


Figure D.6: Energia Interface Window

When considering Figure D.6:

- **Menu Bar**
 - Most Energia functions may be accessed through this interface.
 - The Help menu provides a quick connection to a Reference of all Energia commands.
 - The Tools menu provides a way to check and change the COM port.
- **Short-cut Icons**
 - The **Verify** and **Upload** icons are extremely important to compiling and uploading code from Energia's host programming environment to the LaunchPad.
 - * There is no "Run" command or icon. Once a sketch is uploaded to the LaunchPad, the LaunchPad executes this code automatically.
 - * There is no "Stop" command or icon. The LaunchPad executes whatever program it has in its program memory until it is reset or if power is disconnected.
 - The **Serial Monitor** provides a means for the user to see data output from the program while the program is running. This is very helpful for verifying the program's execution, especially when the program does not execute its instructions as expected.
- **Energia Sketch (coding window)**
 - This is where a user inputs or modifies Energia commands.
 - Note that Energia saves the sketch in a folder of the same name as the sketch each time a sketch is compiled.
- **Status Bar:** This is where Energia's operations are reported.
- **Message Window:** This is where Warnings and Error Messages are reported.
- **HW Settings:** This is a summary of the Energia/LaunchPad setup.

D.3.1 Running an Energia Sketch

Before compiling any code, you must ensure that Energia is set up properly with correct settings.

- On the Energia main window, verify the following settings:
 - Tools → Boards: "RED LaunchPad w/ msp432 EMT (48MHz)" should appear.
 - Tools → Port: should appear to be grayed-out when the LaunchPad is not connected.
- Connect the LaunchPad to the PC.
 - Use the USB(male) to microUSB(male) cable that is provided.
 - When connecting the LaunchPad, be sure to handle the ports carefully.
 - When successfully connected, a green LED on the left side of the LaunchPad should be on.
- Verify the COM port:
 - Tools → Port: "COMx" should now appear, now that the LaunchPad is connected.
 - "COMx" will be different for different combinations of PC and LaunchPad (COM4, COM5, etc.)
 - Generally, two COM ports should appear available as serial ports.
 - * However, only one serial port is used by Energia at any given time.
 - * Select the highest-numbered COM port that is available.
 - * If you can't see data in the Serial monitor, try switching COM ports.

After confirming all your settings are correct, open an Energia program (they are called sketches in the software). There are many examples available directly through Energia or various forums online. For this course, it is usually best to start with a program which has already been developed that is similar to your requirements, and modify it to meet your particular needs. From there, you must both verify and upload the sketch that you want to run. The buttons are seen in Figure D.7.



Figure D.7: Verify and Upload Buttons

Verifying the code will display many compiler messages in the status window, which is seen in Figure D.8.

- While these messages will be in red text, there is no cause for concern. This is normal, even for working code.
- Watch the status bar in the top right of Figure D.8. When the verification is done, you should see a "Done Compiling" message on the status bar. This indicates that the program was compiled without errors.



Figure D.8: Status Window

After verifying that the code compiles successfully, you will upload the program to the LaunchPad.

- Again, you will see many compiler messages in red displayed in the status window. Like before, this is normal, but you may note that there are some errors when reading the messages.
- When you again see the "Done Uploading" status, you should see the "Success" message pop up in the status window, like in Figure D.9. This indicates that the program is running successfully, and should be seen on your LaunchPad.



Figure D.9: Complete Status