MASSACHVSETTS INSTITVTE OF TECHNOLOGY
Department of Electrical Engineering and Computer Science
6.081—Introduction to EECS I
Spring Semester, 2007

Thursday Lab for Week 7

- Lab on designing and using a virtual oscilloscope.
- Lab report due Tuesday April 3rd in Lecture.

The constraints view of circuits

In Tuesday's software lab, you used the constraint view of resistor networks to get the computer to analyze circuit voltages and currents. In this lab you will be designing circuits with resistors, and we hope this experience will help you develop intuition about how such circuits behave. Specifically, you will use your understanding of resistor networks to design a virtual oscilloscope, and then use that oscilloscope to investigate the behavior of a LEGO motor. This lab is new, and to allow us to experiment with some of the material, we have made some parts optional. Please also be prepared that we may make announcements if we see common difficulties which we did not predict.

Thursday Lab - Designing and using a virtual oscilloscope

In this lab you will design a resistor network that will allow you to measure voltages over a wide range. Then you will write a python program that will create a graph of a sequence of voltage samples taken at regular time intervals, effectively creating a virtual oscilloscope. Finally, you will use your virtual oscilloscope to measure the relationship between voltage and current for a LEGO motor, and use your knowledge of circuits to develop a circuit model for the motor.

For this lab you will not need the robot, but you will need a laptop and:

- National Instruments (NI) Interface Box, USB cable, and screwdriver.
- A number of resistors.
- A Lego motor with connector.
- A protoboard with built-in power supply.

We will be using the NI box in a number of labs to interface to circuits, sensors and motors. The NI box can be used to measure voltages and convert those voltages to numbers that can be accessed using python. We have provided a python program which will read numbers generated by the NI box, specifically the numbers that are related to the voltages at the NI box terminals labeled *AI0*, *AI1*, and *AI2*. Note that for the NI box, the voltages are measured with respect to the voltage at the NI box terminal labeled *GND*. Our program, *daqtry.py*, is an infinite loop which reads from and writes to the NI box and prints the results. The program is in the subdirectory *NILab* in the home directory on the lab laptops.

You will also be using a protoboard to build some small circuits. A protoboard is used for making easily modifed electrical connections between wires and circuit elements. If you look at your protoboard, you will notice many rows of five holes. These holes are electrically connected, so if you plug two wires in to two holes in the same row, the wires will be connected electrically. Resistor leads can also be plugged directly in to the protoboard holes. Also, each protoboard has several long columns of holes which are used for nodes in a circuit that have a large number of connections. These columns are often used for ground and power. **CAREFUL: the long columns are only electrically connected for HALF THE COLUMN**. If you have never used a protoboard, have one of the staff members demonstrate the board's use.

Try using the NI Box

Connect the NI box to your laptop (it has a USB connector), and try running the *daqtry.py* program to read the voltage at *AIO*. Please note that you will need to connect wires to the *AI0* and *GND* terminals to connect circuitry to the NI Box. You should also take note of the fact that the NI box has terminals which generate two reference voltages, +5 volts and +2.5 volts. We have special screwdrivers, wire, and wire strippers, just ask if you need them.

Try connecting *AI0* to *GND* and to the two reference voltages and note the readings returned by the *daqtry.py* program.

PLEASE NOTE: Sometime the interface to the USB port gets confused, and you get a pile of error messages when you run daqtry.py. If so, exit any python interpreter you are running, unplug the NI box from the USB port, wait a few seconds, and then replug in the NI box. Then try running *daqtry.py* again. If that does not fix the problem ask for help.

Checkpoint: 2:15 PM

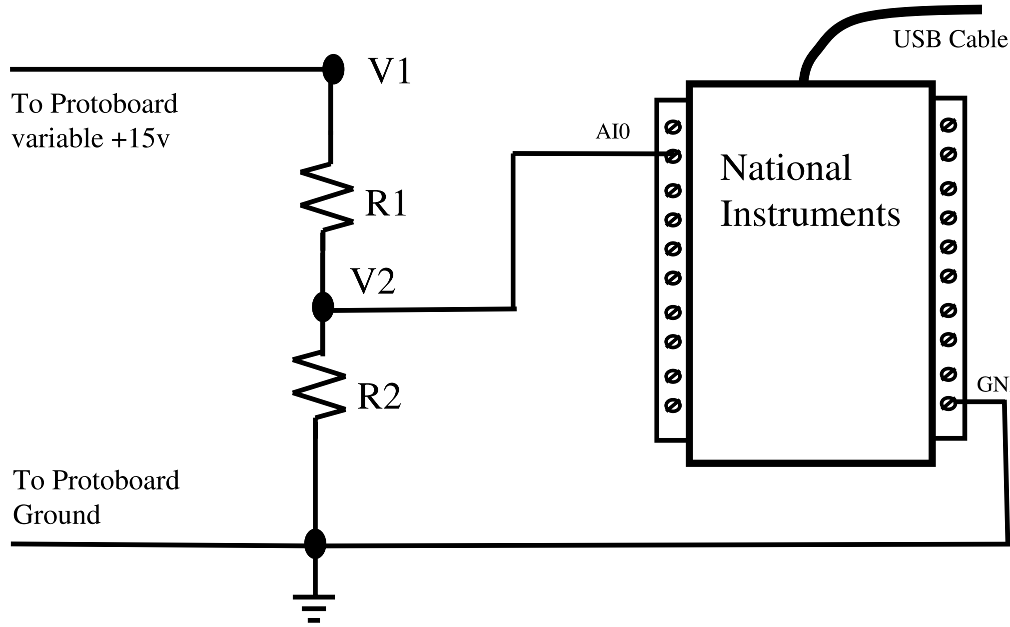
- Demonstrate that you can use python to read voltages from the NI box.

Rerange the input using a resistor network

The input voltage range for the NI box *AI0* input is from 0 volts to 5 volts, but the power supply for the protoboard can generate voltages in the range of 0 volts to +15 volts. Your job will be to design a resistor divider network that will reduce voltages in the range from zero volts to +15 volts to voltages in the range from zero volts to five volts. The reduced voltage range can then be measured using the *AI0* input, and you can use python to post-process the measured voltage and undo the effect of the resistor divider. In this way, you can use the NI box to effectively measure a larger range of voltages, making your own volt meter.

To see how to design such a circuit, consider the diagram below of the NI box, an external resistor network, and the connection points for a test voltage (supplied here by the protoboard +15 variable supply and the protoboard ground). If $R2 = 10,000\Omega = 10k\Omega$, what value should $R1$ have so that the voltage at *AI0* with respect to *GND* will be five volts when the protoboard adjustable supply voltage is 15 volts.

Once you have decided on a resistor value for $R2$, explain your solution to your LA. Then, please use the protoboard we have given you to construct your external resistor network. In order to construct your network, you may need to use series and parallel combinations of the resistors we have available to generate the resistance values you need. Ask you LA to show your our labeled box of resistors. Finally, modify *daqtry.py* so that the printed voltage matches the test voltage. You can demonstrate your *volt meter* using the protoboard's adjustable voltage supplies.



Checkpoint: 3:15 PM

- Demonstrate that you can measure voltages from 0 to +15 volts using the NI box and your resistor network.

Optional Challenge if you have time

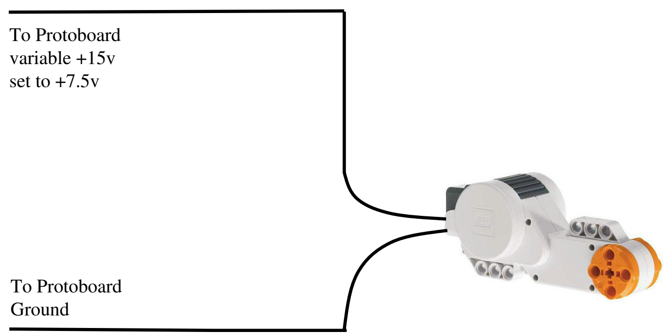
The NI box has a five volt reference voltage. Can you add a resistor to the divider network, and then adjust the values of the resistors so that you can measure test voltages from −15 volts to +15 volts?

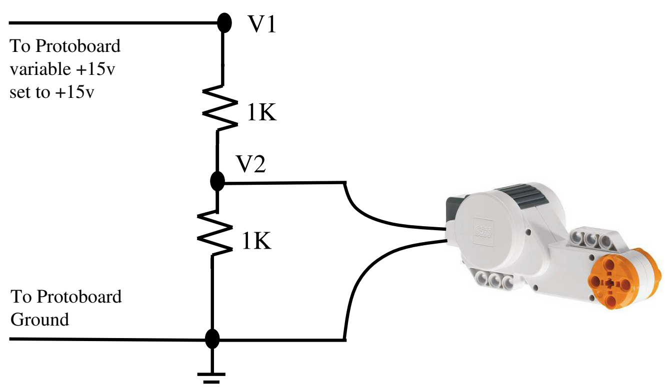
Using the Motor

We have supplied you with a motor, and you will be using the motor in the next lab to make a moving "head" for the robot. For this lab, you will be examining the voltage-current characteristics for the motor.

The speed of the motor depends on the voltage applied to the motor. Since your protoboard has an adjustable power supply, you can experiment with this relationship. As you will notice, the adjustable power supply is not calibrated, so it is quite fortunate that you just designed a volt meter that will let you measure the actual voltage of the adjustable supply. Use your volt meter to set the adjustable supply to 7.5 volts and try the running the motor using the circuit in the figure below. Note that the motor has a orange and green wire. You can interchange these two wires in the circuit, then the motor will spin in the opposite direction.

You can also generate 7.5 volts by using a voltage divider. Then you can connect the divider output to the motor, as shown in the bottom circuit in the figure below. Try this circuit and use your volt meter to understand what happened. Why did the circuit fail?





Checkpoint: 3:45 PM

- Demonstrate to your LA that you understand what happened when you tried the motor circuit using a voltage divider.

The virtual oscilloscope

For this last part, we would like you to plot the motor current versus time using python. To do this, you will have to figure out how to do three things:

- Convert motor current to a voltage you can measure with your volt meter.
- Modify *daqtry.py* so that it stores a sequence of measurements.
- Interface your measured sequence to *SOAR*'s plotting program

In order to use SoaR's plotting features, first import the needed functions by typing the following lines in the interpreter:

```
import SoaR
from SoaR.Util.GraphingWindow import GraphingWindow
```

Once you have imported the functions from SoaR, you can plot the values of a python function by opening up a GraphingWindow and then using the graphDiscrete function. For example, to plot the values of the function *yourF* from $n = 0$ to $n = 7$ on a graph whose y axis ranges from -10 to 10, one would type

```
plotWindow = GraphingWindow(500, 500, 0, 7, -10, 10, "Plot")
plotWindow.graphDiscrete(yourF)
plotWindow.helpIdle()
```

Please not that *graphDiscrete* takes a function as an argument, and in the above *graphDiscrete* command, the function *yourF* will be called for each value in the sequence $0, 1, 2, 3, 4, 5, 6, 7$, as in $yourF(0)$, $yourF(1)$, etc.

For example, given the function *lineF*

```
def lineF(n):
    return(2.0*n)
```

then *plotWindow.graphDiscrete(lineF)* will plot a line of slope 2.0.

Please determine how to use an external resistor (100Ω is a good value) and the protoboard power supply to measure the motor current. Explain your approach to your LA, and then design a program to plot the motor data by modifying *daqtry.py* and using *GraphingWindow*. Note that you may wish to change the range for the y axis in your plots to make them more informative.

Checkpoint: 4:45 PM

- Demonstrate that you can plot motor currents as a function of time.

To Hand in Tuesday April 3rd

- Write a brief description of your reranging resistor network, and explain how your determined the value of $R1$.
- Explain what happened when you connected the motor to the voltage divider using $1k\Omega$ resistors and why.
- Describe how you measured the motor current.
- Describe how interfaced the measured current to the graphing program and include a commented copy of your python program.
- Enjoy spring break!