PHYS605

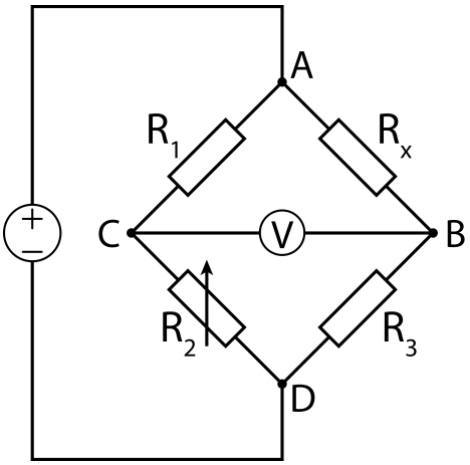
HOMEWORK 2

DUE WEDNESDAY FEB 14, 2024.

For each problem the number of points are indicated in the square brackets [ ].

RESISTORS

1) [30] Consider the circuit in the diagram to the right. This setup of resistors (sorry, the picture is using the European symbol for resistors, a rectangular box) is called a Wheatstone Bridge. The power supply (circle with + and -) has a voltage Vin. The resistor R2 is a variable resistor.
   a) Calculate the reading of the voltmeter in terms of R1, R2, R3, Rx and Vin. (Hint: the schematic can be re-drawn to look like two voltage dividers.)
   b) Consider R1=R3=1kΩ, R2 is variable, and Rx is a light sensitive resistor, called a photoresistor, which without any light has a value Rx(0)= 1.2 kΩ. What value do you need to set R2 so that the volt reading is zero when there is no light?
   c) When light falls on the photoresistor its resistance decreases. Consider that some light falls on the photoresistor and the value is decreased by about 10% to 1.1 kΩ. What is now the reading on the voltmeter if you keep R2 the same as in b?



CIRCUIT SIMULATION I-V CURVES

Computational work. Submit the following problems as a Jupyter notebook with the plots (should have 2 plots), and your data files, and the calculations. Make the notebook organized, using comment cells, and use inline comments in your code where needed.

In this part of the homework you will use an online simulator to simulate the circuit of your Lab1 part 2 experiment. You can use an online simulator, PartQuest https://explore.partquest.com, for this[1]. The program is free to use, but you will need to create a login to save your modifications. I found that the simulation does not work properly on the Safari browser, but works great with Chrome, so you may need to get Chrome to use this.

[1] An equivalent simulation with MultiSim can also be used, however Multisim is not as accurate as ~~PartSim~~ PartQuest. See here: https://www.multisim.com/content/ct5uhRENUbLkvX5TEauY8e/forward-biassed-diode

The first simulation to try is a current-voltage test of a diode. To make this easier, I have already setup an example circuit that is ready to go: https://explore.partquest.com/node/278959. (or

https://explore.partquest.com/groups/mauriks-workspace/designs/diode)

Open the document (click) and then click on the yellow "Click to Open Design Analyzer". Look at the circuit, and make sure you understand it. There are two measurement devices, a voltmeter "u1" and a current meter "u2". On the bottom right of the screen is a button, "Edit in PartQuest Explore", which you need to click on for the circuit to become interactive. Log in (or create a new account) so you can save your circuit, top right yellow button.
For this case, the example circuit is already all set up for you to test a 1N4148 small signal diode. To run the simulation, click on the green triangle at the top of the screen. (To the left of the green triangle is a configuration button for the run, which is already configured correctly.) In this case, you will do a DC sweep from -10 to 10V. Click the green triangle to run the simulation. After it is finished a new entry is added to the tree on the graphing window. If you click on this, you will see a "V" for the volts. Double clicking on it gives you the volts versus time for "U1". The same for "I", the current in "U2". You can now save these curves by right-clicking (control-clicking) on the legend symbol and choosing "Download". The resulting CSV files can now be imported into Python using Pandas, or you could combine the two files first using Excel. Then use this data to make a proper I-V curve. I suggest you use Python to do this, but any method will be accepted.

Now go back to the simulation and remove the diode that is there. Instead put in a Zener diode. On the left in the search box search for "zener" and then add this component in the correct place and with the correct orientation in your circuit. Make sure the "wires" are connected. Run the simulation again, and save the files. Make sure you rename these files so you know which they are.

Next, run the simulation again, but this time with a more accurate model for the zener diodes. Since this is a rather tricky operation with this software, I have created the model for you already.

The 1N5223B 3.3V diode is found here:

https://explore.partquest.com/groups/mauriks-workspace/designs/1n5223-33v-zener-diode

The BZT52C2V7 diode is found here:

https://explore.partquest.com/groups/mauriks-workspace/designs/bzt52c2v7-zener-diode

Tasks:

1. [10] Create the I-V curve for the (1N4148) diode and the zener diode, on the same graph. Explain and comment on the curves.
2. Create the I-V curve for the two zener diode models on the same graph.
    a. [10] 3.3V zener diode: "1N5223B 3.3V"
    b. [10] 3.3V zener diode: "BZX55C2V7"
    c. [10] Explain and comment on the curves.
3. [20] Extra bonus: Play around with this simulation. Change parts, try things, and report anything that you did, especially if you thought it was interesting and can explain it.

PROGRAMMING

- If you are not familiar with programming in Python and using Python (Jupyter) Notebooks, then please contact me right away!
- You should already have an environment setup for creating Python code and making notebooks. There is some help available on the course website on how to get started if you need a refresher.
- Note that it is acceptable to not get all the way to an answer on these tasks, but do document the headway you made *and* ***submit your notebook***, with ***documentation***.
- It is fully expected with this assignment that you use all available resources to accomplish these tasks, so feel free to search for solutions. Note that you do yourself a favor to understand the solutions you find. So see how far you get with one of the following assignments.

I. A MORSE CODE TRANSLATOR

Points: [30] - This task works better as a Python program than a notebook. Make sure it runs without giving errors before you submit. Submit this as a morse_program.py

Here you will write some code that can take any character string and then return a list that gives instructions for how to send this string in Morse code. You could then use this code to blink out the Morse code with an LED on the Raspberry Pi. Comment all your code! Provide references for any sources you used.

1. Step one. Write a Python program that prints any string you give as an argument to the terminal.
    a. Get a string into your program is a command line argument. Fully parsing a command line can be a bit tricky, so there is a library called "argparse" to help you with it. Simply getting a string is not so bad, see:
        i. http://www.pythonforbeginners.com/system/python-sys-argv
        ii. https://www.tutorialspoint.com/python3/python_command_line_arguments.htm
    b. You will need to figure out how to run your program from the command line!
2. Next, you need to translate this string. The first step is to break the string into the individual characters (including space). Python makes that easy: `mystring[i]` will get you the letter in position `i`, starting at `i=0` (zero). The next step is to translate each character to Morse Code. The best way to do that in Python is to create a dictionary for the translation. Lookup "Python dictionary" and find a tutorial you like. The *very* official rules for translating text to Morse code if found in the International Morse Code document[2].
    a. Create the dictionary so that morse_code['A'] returns the string ".- " and morse_code['B'] returns "-… " etc. For space, return two spaces, i.e. more time.
    b. You need to now "read" your string one character at a time, and then use the dictionary to print the morse code out as dots and dashes. Don't add a return after each more code character!

[2] https://www.itu.int/dms_pubrec/itu-r/rec/m/R-REC-M.1677-1-200910-I!!PDF-E.pdf

3. If you get here, great. Next you want to *change* the step 2b to give you a long string of all the dots, dashes and spaces, instead of printing them. You now want to write another function that reads this string and for each dot it would want to blink an LED a short time, for each dash a long time, and between characters it would leave the LED off for the same length as a dash. Since you don't have the LED setup at home, you can write the code and use print statements to check that it does the correct thing.

I. USING THE RESISTOR CLASS.

This part of your homework requires you to start your own **Jupyter notebook**, which you will hand in for the assignment.

In the last homework you did a complicated resistor problem. Here you will check your work with a Python program in a Jupyter notebook. To make this easier, there is a Python package available on the course GitHub site at: https://github.com/mholtrop/Phys605 and navigate to Homework and then HW2. You will find the file Electronics.py which contains the package and an example notebook that makes use of the package in a very simple manner.

You load the class just as you would load something from a Python package:

```
from Electronics import Resistor  # load the Resistor Class.
R = Resistor(1000,1./8.)        # Define a resistor of 1kOhm 1/8W
```

Look in the Electronics.py file on how to use this class. For series you would add resistors using the + operator (add). For parallel, you would use the * operator (multiply).

TASKS:

1) [20] Use the resistor class to check your solutions in HW1 problem 3. Also check the posted solutions from HW1 for problem 4. Submit the results in your notebook.
2) [Bonus 20] It would be nice to have the resistor class also give the new tolerance of the equivalent resistor it computes.
   a) Add a new property to the resistor class: "_tolerance", that defaults to 1%.
   b) Add the tolerance to the print output. (modify __str__()).
   c) Add calculations for the tolerance to the serial (__add__()) and parallel (__mult__()) functions.
   d) Test your new feature for a 1kΩ and 2kΩ resistor in series, and separately, in parallel.