

Physical Computing Self-Quiz

The following are questions you should be able to answer without reference to outside material by the middle of the semester in Introduction to Physical Computing. Try to answer all of the questions below without resorting to copying external code or diagrams. If you don't know where to start on a given question, consult with your instructor in office hours.

Give a definition for physical interaction.

Explain the difference between an analog sensor and a digital sensor. Give examples of both.

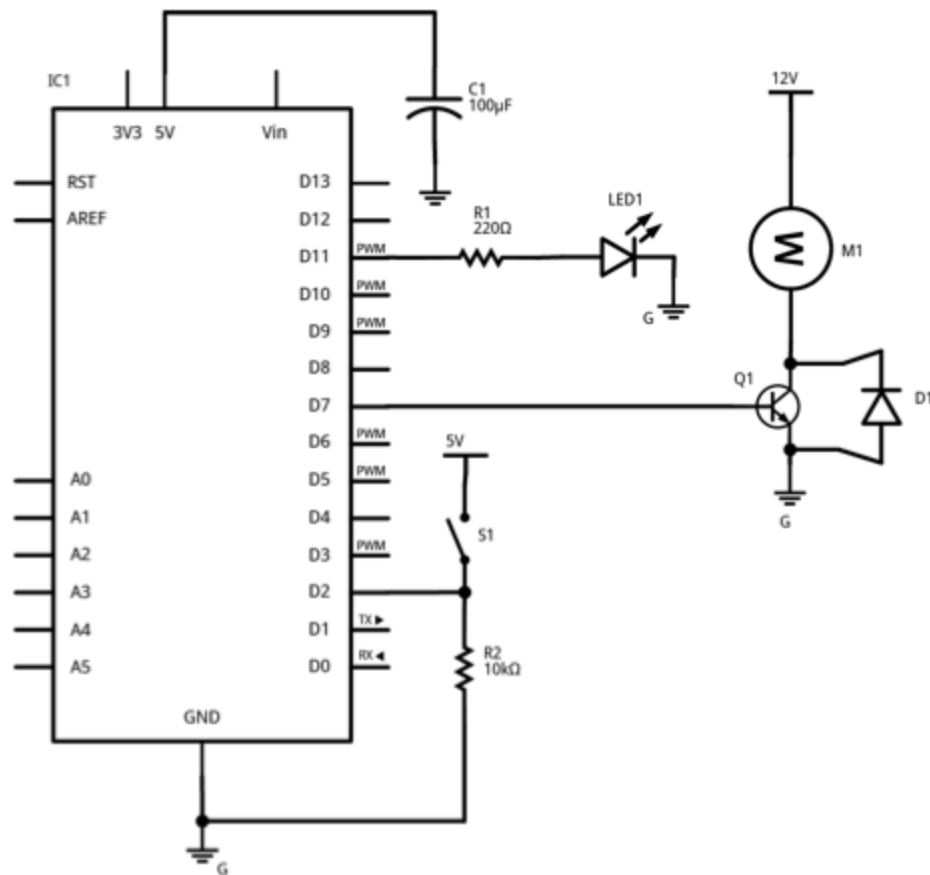
How do you determine how the range of resolution you need from a given sensor or output device (actuator)?

Explain the flow of electrical energy in a circuit.

What is a short circuit?

Describe how you might make a homemade switch.

Identify the electronic components in this diagram:



R1, R2:

C1:

IC1:

Q1:

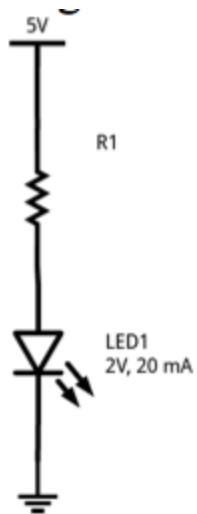
D1:

LED1:

M1:

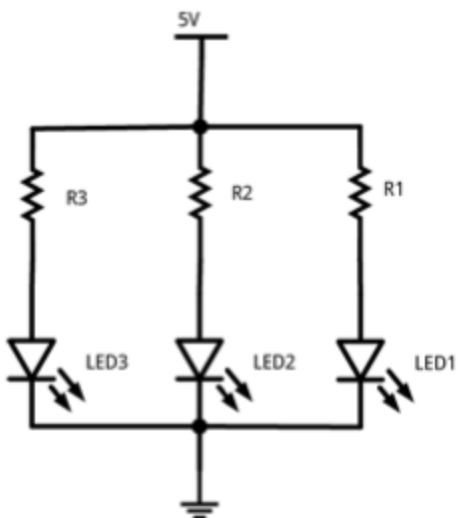
G:

Calculate the value for the resistor in the following circuit:



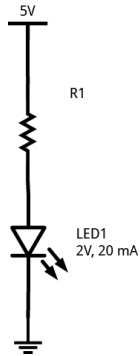
If you wanted to make the LED in the circuit above dimmer, what change would you make to the resistor?

Identify the components that are in series below, and those that are in parallel:



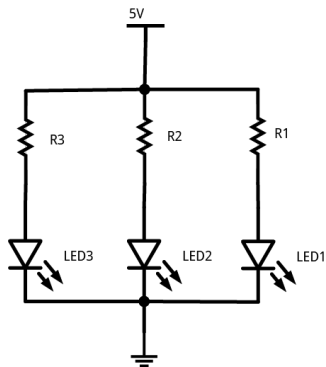


What formula would you use to calculate the value for the resistor in the following circuit? What is the value? What happens if you don't use enough resistance?



When absolutely nothing works on your breadboard, what is the most basic thing to check using a multimeter?

Identify the components that are in series below, and those that are in parallel:



Fill in the blanks:

When components are in series, the _____ through them is the same.

When components are in parallel, the _____ across them is the same.

When you put batteries in series the _____ adds up.

When you put batteries in parallel the _____ adds up.

Draw the circuit for connecting a switch to a microcontroller's digital input pin, using a pulldown resistor. What does the resistor do?

For analog input we often use variable resistors. Can you give 3 examples of variable resistors?

What is the maximum range your Arduino can give for an analog input reading? How many bits of memory does this take up? How about analog output?

Draw the circuit for connecting a potentiometer to a microcontroller's analog input pin.

Draw the circuit for connecting a variable resistor to a microcontroller's analog input pin, using a voltage divider.

Can the circuit you drew for the voltage divider output full range of analog readings? What about the potentiometer circuit?

What's the range of values that you can store in a single byte of memory?

If you need to store a larger range than that, what do you do?

Your Arduino cannot output an analog voltage. How does it fake it for dimming an LED, for example?

Write a program to read a digital input on an Arduino and print the result out to the serial monitor.

Write a program to read a digital input on an Arduino, compare the current reading to the last reading, and if the state of the input has changed, print the result out to the serial monitor.

A potentiometer attached to an Arduino's analog input returns a result of 435 when read using the `analogRead()` command. What is the voltage between the input and ground?

Write a program to read an analog input on an Arduino, map the result to a range from 0 to 255, and fade an LED.

Write a program to read an analog input on an Arduino and send the result serially to a desktop computer in a single byte.

Write a program to read three analog inputs on an Arduino and send the results serially to a desktop computer as an ASCII string, separated by commas, and terminated by a newline and carriage return.

Asynchronous serial communication is the most common way for microcontrollers to communicate with personal computers. What are some of the characteristics that must be agreed upon by both sides to allow serial communication to happen?

How many programs can control a given serial port on your computer at a time?

What does the serial monitor do?

Imagine you have an Arduino sketch sending serial data out. You open the Serial Monitor in the Arduino IDE to confirm that it's sending data. Then you open a Processing Sketch to visualize the data and you get the following error:

Error: PortInUseException.

What's the first thing you should do to change this?

You can interpret the bytes you send on a serial connection as binary values or as ASCII characters. What are the benefits of these two methods?

How does the Arduino serial monitor interpret bytes? What would you see if you sent it what it wasn't expecting?