

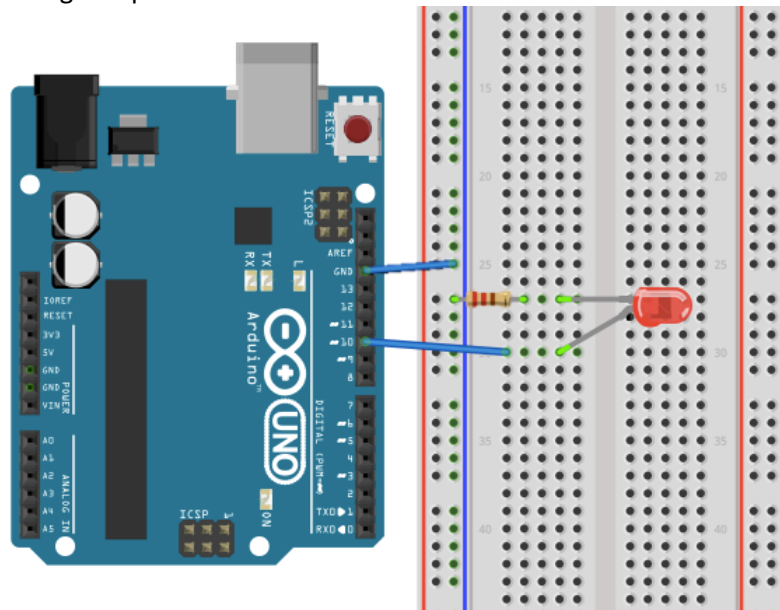
Activity #3: Making an LED Fade Worksheet

Overview

In this activity, we will program the Arduino such that it causes an LED to “fade” in brightness, rather than just blink on and off.

Construct the Hardware

1. Gather the following materials:
 - One Arduino & USB Cable
 - One breadboard & two jumper wires
 - One LED & one 220Ω Resistor
2. Construct the following setup:



- A. Connect 1 jumper wire from GND (“ground”) to one of the (-) columns.
- B. Place 1 leg of the resistor in the same column as the GND wire. Place the other leg of the resistor in a new row.
- C. Place the short leg of the LED to the same row as the resistor leg. Place the long leg in a new row.
- D. Connect 1 jumper wire from the previous row to digital pin 10.
- E. Check that your setup matches the setup in the diagram above or the one at the front of the room.

Create the Program

3. Type up the associated program for Activity #3 (see the next page).
4. Save the file as **LED_Fade**.
5. To check that the Arduino can compile your code, “Verify” it. You can do this by either pressing **Command + R** or by clicking on the Verify button (it looks like a check mark).

Name: _____ Date: _____ Class: _____

```
/*  
Activity 3: LED_Fade  
Causes an external LED to fade (get brighter and then turn off) repeatedly.  
*/
```

```
int led = 10;  
  
void setup() {  
  
    pinMode(led, OUTPUT);  

```

Troubleshoot Errors

6. Take time to trouble-shoot any errors that arise. Start by reading the first error message and checking that line of code. Compare what you have written to the original program. Common errors:
 - If you get the message “_____ was not declared in this scope,” it is possible you misspelled one of the variables or functions.
 - If you get the message “expected ‘;’ before ____” —that means you forgot to add a semi-colon at the end of the previous line.

Run the Program

7. Using the USB cable, connect your Arduino to your computer. Expect to see some lights flash on the Arduino board. Then the Arduino starts to execute the last program that was uploaded to it.
8. From the “Tools” menu in the Arduino IDE, make sure the appropriate serial port is selected.
9. Now upload the “LED_Fade” sketch.

Name: _____ Date: _____ Class: _____

10. After a few seconds, expect the LED to start to fade in brightness! If it is not fading, check your hardware first, and then your code.

Reflection Question

1. What do you think the function **analogWrite()** does? Research it on the Internet and incorporate what you learn into your response. What does the second input in its parentheses represent? What numbers must it be between?

Alternatives

1. The Arduino IDE comes with a different program for getting the LED to fade. Go to “File” > “Examples” > “Basics” > click on “Fade.” Change the LED pin number from 9 to 10. Then upload and execute the program. Notice how this program achieves a much smoother fade, and the LED gets brighter and dimmer. Explain what is happening in the **loop()** function that allows these two things to happen.
2. For either program (the one in this packet or the example from the Arduino IDE), figure out how to change the fade-rate of the LED. For example, try to get the LED to fade twice as fast, twice as slow or with a completely new pattern. Describe what you had to change in the program to do this.
3. Figure out how to add a second LED and get it to fade opposite the first LED (so that as the first one is getting brighter, the second one is getting darker). Sketch the new setup below and describe the changes you made to the program to accomplish this.