Fading LED

This project builds on the basic concept that you practiced in the *analog output* guide. Remember, Arduino doesn't have an actual analog output, so to get the dimming effect you have to use *Pulse Width Modulation (PWM)* to send short cycles of power to the pin.

You used this to create a basic fading LED. This project will challenge you to take it even further!

Base Activity

You'll be writing code that alters the behavior of the LED. Remember, with *digital output* you made LEDs that were either ON or OFF. *Analog output* allowed you to add additional functionality by creating LEDs that could fade in and out.

This project will challenge you to create LEDs that fade at different levels and at different times!

Supplies

- Uno board and breadboard
- Jumper wires
- At least 2 LEDs of your choice
- 220 or 560 Ohm resistor *for each* LED

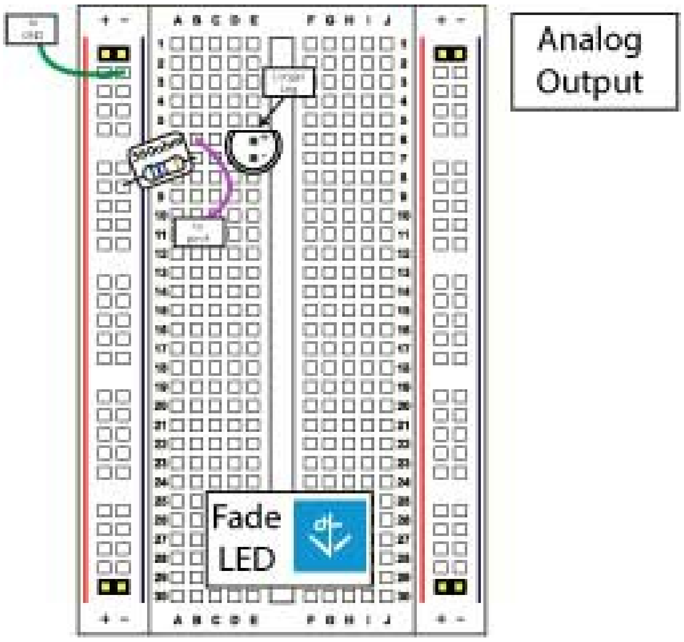
Steps

This project follows the same setup as the *breadboard LED* project. The difference will be the code that you write in the sketch.

Here's a reminder about how to setup an LED circuit.

Circuit

To help with your experimenting process, we have created and provided you with breadboard diagrams to help you setup any of your circuits. Refer to this diagram as you work through the following steps.



1. Connect the GND pin on the Arduino to the Ground (-) rail on the breadboard.
2. Connect the LEDs to the breadboard. Place them far enough apart so that you have room to work. As a tip, try to keep the LED placement consistent. A good habit is to put the long leg (+) as the top leg.
3. Connect each + leg of each LED to a pin on the Arduino. Remember, these must be the pins with a ~ !
    - For example, if you're using pins ~9 and ~10, connect the + leg of one LED to ~9, and the other to ~10.
4. Connect a resistor from the - rail of the breadboard to the short (-) leg of each LED. Each LED needs it's own resistor!

Sketch

This is the basic code template for this project. Remember, you'll need to include a variable for each LED that you're using!

In the example, "int ledPin = 9;" is used to set ONE of the LEDs. If you add another, you'll need to create *another* variable!

```

void setup() {

  int ledPin = 9;
}

void loop() {
  analogWrite(9,0); //Turn the LED on pin 9 off
  delay(500); //Wait 500 milliseconds
  analogWrite(9,63); //Turn the LED on pin 9 25% on
  delay(500); //Wait 500 milliseconds
  analogWrite(9,126); //Turn the LED on pin 9 50% on
  delay(500);
  analogWrite(9,189); //Turn the LED on pin 9 75% on
  delay(500);
  analogWrite(9,255); //Turn the LED on pin 9 100% on
  delay(500);
}
```

Fading LED: Experimenter Challenges

Now that you've faded one LED, the very first challenge is to make the LED back down! Right now, you've made a sketch that has the LED fade in a single direction. Next, make the LED fade in and out, so that it looks as if it is "breathing."

After you've done that, the next challenge is to fade more than one LED at different rates and at different intensities. This can be difficult to manage, so do your best!

Bonus: Instead of writing in the intensities as a definite number (such as 0, 63, 255, etc.) write the code so that they're fractions *of the total intensity*. Think about what this means and how you'd write the code.