

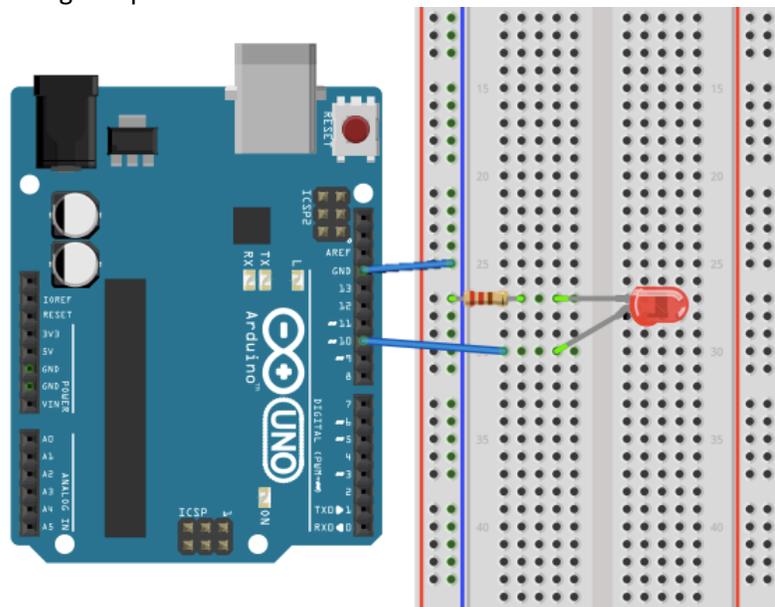
## Activity #3: Making an LED Fade Worksheet

### 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?

**This function is NOT used with analog pins – it is used with digital pins. It allows a digital pin to output a simulated voltage between 0V and 5V using pulse-width modulation (PWM). This can only be used on certain digital pins on the Arduino board (they have a ~ sign next to them). This function takes two arguments:**

```
analogWrite(arg1, arg2);
```

**The first argument in the function specifies the number of the digital pin that you want to control. The second argument specifies the simulated output voltage. A value between 0 (always off, 0V) and 255 (always on, 5V) must be specified.**

### 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.

**In the first line of code:**

```
analogWrite(led, brightness);
```

**the Arduino is told to output a (simulated) voltage to pin 10 at a level corresponding to “brightness” (where “brightness” is between 0 and 255).**

**In the second line of code:**

```
brightness = brightness + fadeAmount;
```

**the value of the variable “brightness” is increased by a certain amount (called “fadeAmount,” which in this case is equal to 5). That way, in the next iteration of the program, the output voltage will get just a little bit higher.**

**In the third block of code:**

```
if (brightness == 0 || brightness == 255) {  
    fadeAmount = -fadeAmount;  
}
```

**the Arduino examines the value of “brightness.” If “brightness” ever reaches the boundary values of `analogWrite` (0 or 255), then the sign of the `fadeAmount` value is inverted (positive/negative), so on the next iteration, “brightness” continues to stay between the boundary values.**

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.

**There are a few ways to do this. One way is to modify the values of the delay function. Increasing the delay time slows the fade process, while decreasing the delay time speeds up the fade process. Or, instead of changing the delay times, modify the “fadeAmount” variable. If `fadeAmount` is made larger, the output of the digital pin will fluctuate between**

Name: \_\_\_\_\_ Date: \_\_\_\_\_ Class: \_\_\_\_\_

**0V and 5V more quickly. If fadeAmount is made smaller, the output of the digital pin will fluctuate between 0V and 5V more slowly.**

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.

**For the hardware setup, see Activity 2 Answer Key, Alternative #2.  
Below is the modified code with the added lines in bold.**

```
/*  
Fade with 2 LEDs  
  
This example shows how to fade LEDs on pins 6 and 10 using the analogWrite() function. This example code is in  
the public domain.  
*/  
  
int led = 10;      // the pin that the LED is attached to  
int led2 = 6;      // the pin that the second LED is attached to  
int brightness = 0; // how bright the LED is  
int fadeAmount = 5; // how many points to fade the LED by  
  
// the setup routine runs once when you press reset:  
void setup() {  
    // declare the LED pins to be an output:  
    pinMode(led, OUTPUT);  
    pinMode(led2, OUTPUT);  
}  
  
// the loop routine runs over and over again forever:  
void loop() {  
    // set the brightness of the LED pins:  
    analogWrite(led, brightness);  
    analogWrite(led2, 255-brightness);  
  
    // change the brightness for next time through the loop:  
    brightness = brightness + fadeAmount;  
  
    // reverse the direction of the fading at the ends of the fade:  
    if (brightness == 0 || brightness == 255) {  
        fadeAmount = -fadeAmount ;  
    }  
  
    // wait for 30 milliseconds to see the dimming effect  
    delay(30);  
}
```