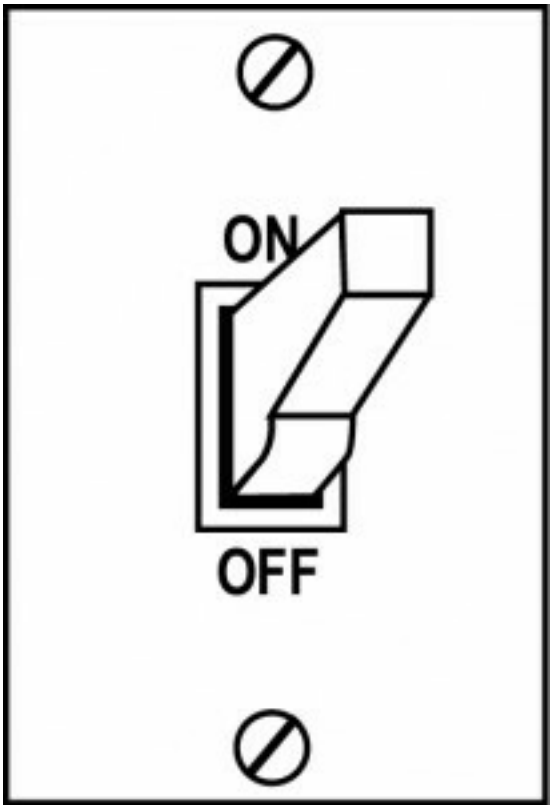
Digital vs Analog

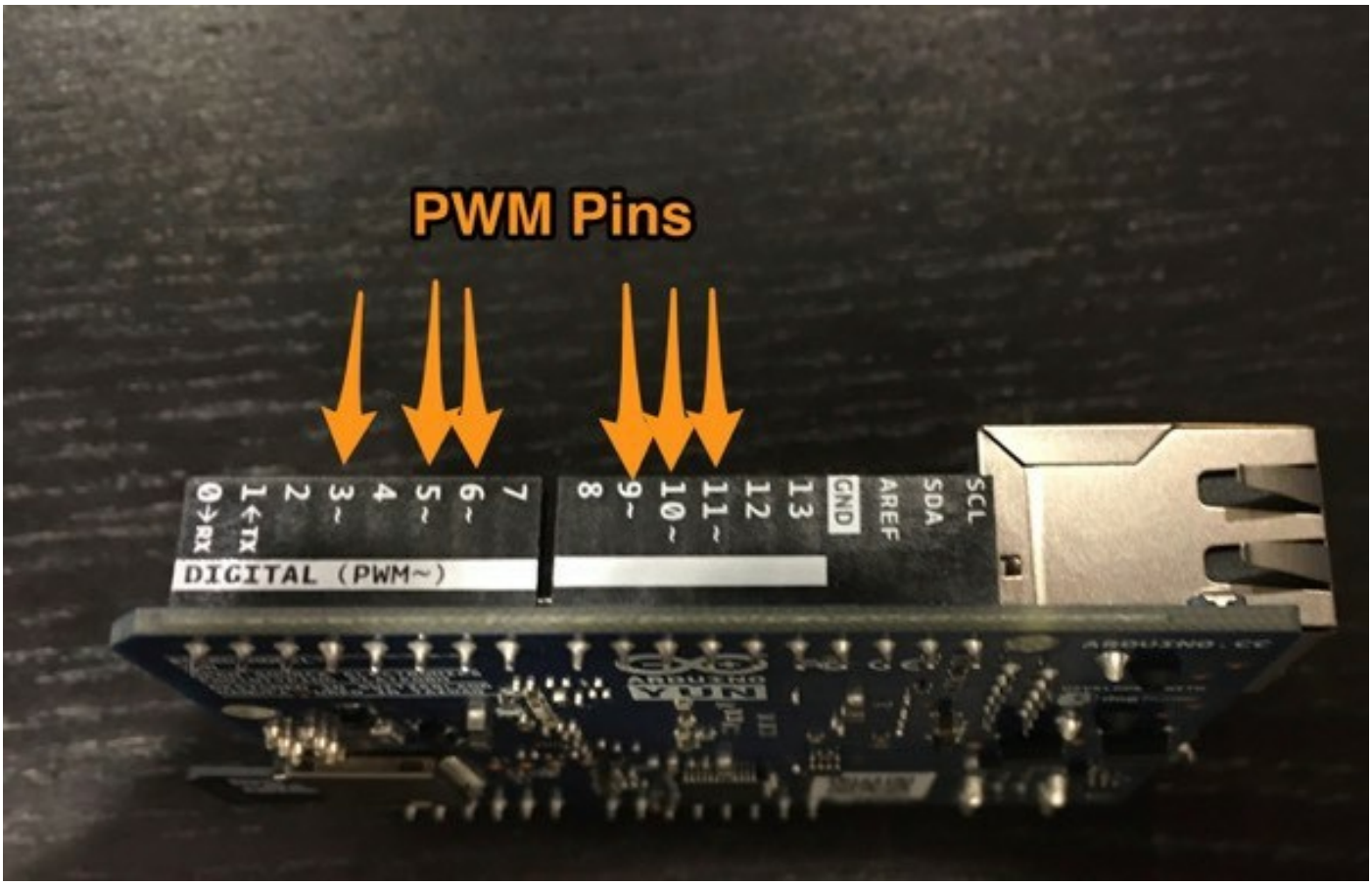
In the *Blinking LED Experiment* you used *digital* output, but that's not the only type of output: you can also perform analog output. The difference between the two is best shown using a light switch. With *digital output*, we have two options on or off, or as the method **digitalWrite()** refers to them, HIGH and LOW.



But what if we want a switch that could control the brightness of our light and have a range of "on-ness" rather than being so finite. In that instance we are going to want to use *Analog* output.

Analog Out and the Arduino

The Arduino platform is primarily a digital platform so the voltage coming out of the pins is always the same which makes analog a little more difficult to do. The Arduino uses a trick where it cycles the power on and off very quickly on a pin in order to generate a voltage that varies in strength. This is called Pulse Width Modulation or *PWM* for short. Only certain pins on the Arduino support *PWM* so if you need Analog Out, you will want a pin with a squiggly line near it:



Breadboard Diagrams

To help with your experimenting process, we have created and provided you with breadboard diagrams to help you setup any of your circuits.

![AnalogOut - Breadboard[(http://d3nnidcq81r9m6.cloudfront.net/wp-content/uploads/2016/04/06225909/AnalogOut-Breadboard.jpg)

Now that you have the basic understanding of *Analog Output*, let's try a project: the Fading LED.

Project: Fading LED

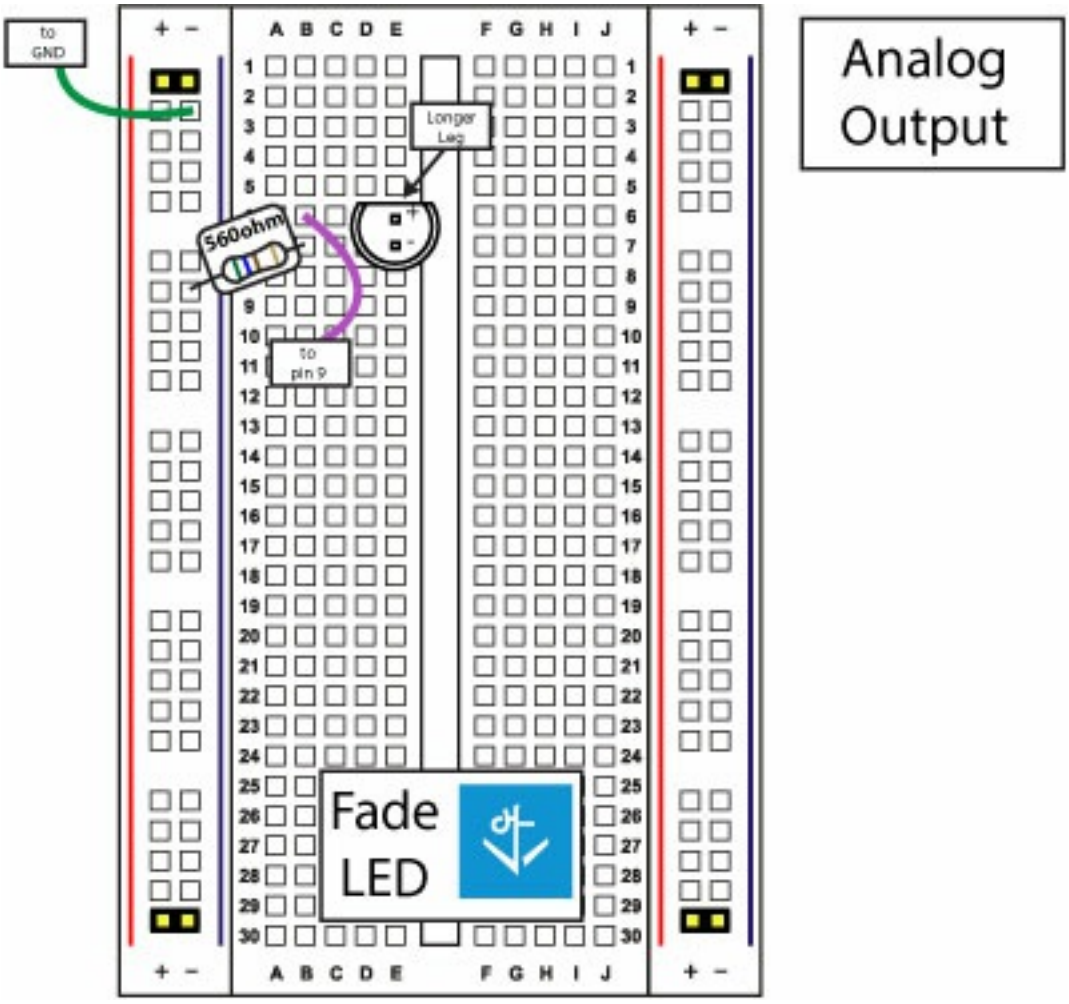
For this experiment, you're going to setup a circuit and create a new sketch that controls the brightness of an LED.

Supplies:

- Uno board and breadboard
- Jumper wires
- LED of your choice
- 220 or 560 Ohm resistor

Breadboard Diagram

Setup your breadboard as shown below:



Resistor?

You might be wondering why we have a resistor between our LED and the GND column when we didn't in the past. This is the proper way to

setup this circuit.

When we were doing the simple Blinking LED, I didn't include the resistor because we were turning the LED off at pretty fast rate and I didn't want to throw too many components at once. This resistor should be in place to help absorb any left over current coming from the LED and reduce it to 0 to protect the Arduino.

The Sketch:

Create a new sketch with the following code and upload it to your Arduino. You should see your LED light up in 4 different brightnesses: off, 25% on, 50% on, 75% on, and 100% on.

```
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);
}
```

analogWrite(pin,value)

The above sketch introduced a new method called *analogWrite()*. This method takes two parameters: the pin number and a value between 0 and 255. In the highlighted lines above, I've introduced a new method called *analogWrite()*.

This method takes two parameters: the pin number and a value between 0 and 255. This value controls how "on" the pin is. A value of 0 turns the pin **off** and a value of 255 means completely **on**. Any values in between 0 and 255 result in different levels of brightness. This is used to create a fading effect.

Bonus Challenge: Reverse Fade!

Here is a challenge to help you understand some of the concepts we have learned so far. To see the answer to the challenge, you can expand the code snippet in the next section.

Right now the code sketch results in the LED fading from off to max value. Your challenge is to make the LED **fade from max back to off**.

Attempt this first on your own and then look at the solution when you're ready.

Challenge Solution

```
void setup() {
  //No need for setup this time
}

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);
  analogWrite(9,189); //Turn the LED on pin 75% off
  delay(500); //Wait 500 milliseconds
  analogWrite(9,126); //Turn the LED on pin 9 50% on
  delay(500); //Wait 500 milliseconds
  analogWrite(9,63); //Turn the LED on pin 9 25% on
  delay(500);
  //If we left the next two lines on, then the LED would be
  //because the loop would start back over and it would star
  //analogWrite(9,0);
  //delay(500);
}
```