

IUEE Project 3, part A

Fade using a Button

Fading

Demonstrates the use of the `analogWrite()` function in fading an LED off and on. AnalogWrite uses pulse width modulation (PWM), turning a digital pin on and off very quickly, to create a fading effect.

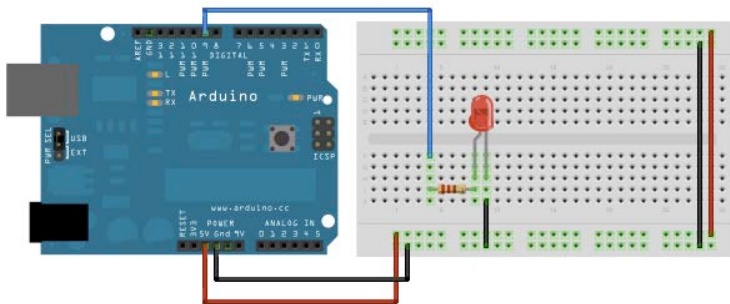
Hardware Required

Arduino board
Breadboard
a LED
a 220 ohm resistor

Circuit

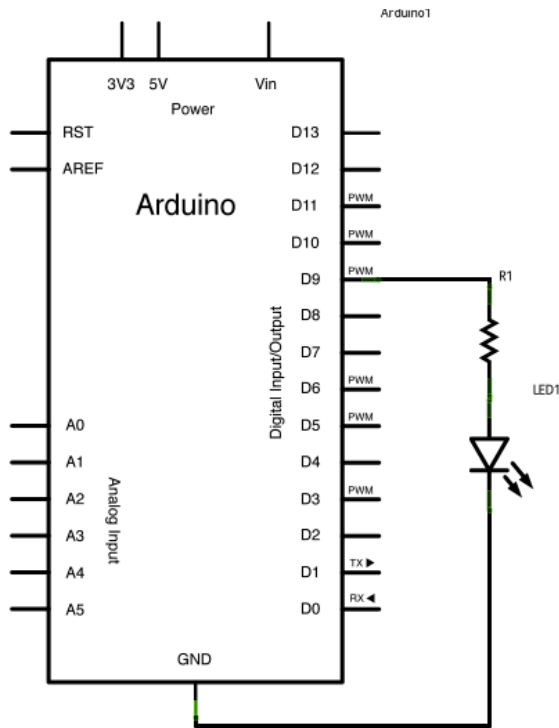
Connect the **anode** (the longer, positive leg) of your LED to digital output pin 9 on your Arduino through a 220-ohm resistor. Connect the **cathode** (the shorter, negative leg) directly to ground.

click the image to enlarge



Schematic

click the image to enlarge



Code

After declaring pin 9 to be your `ledPin`, there is nothing to do in the `setup()` function of your code.

The `analogWrite()` function that you will be using in the main loop of your code requires two arguments: One telling the function which pin to write to, and one indicating what PWM value to write.

In order to fade your LED off and on, gradually increase your PWM value from 0 (all the way off) to 255 (all the way on), and then back to 0 once again to complete the cycle. In the sketch below, the PWM value is set using a variable called `brightness`. Each time through the loop, it increases by the value of the variable `fadeAmount`.

If `brightness` is at either extreme of its value (either 0 or 255), then `fadeAmount` is changed to its negative. In other words, if `fadeAmount` is 5, then it is set to -5. If it's 55, then it's set to 5. The next time through the loop, this change causes `brightness` to change direction as well.

`analogWrite()` can change the PWM value very fast, so the delay at the end of the sketch controls the speed of the fade. Try changing the value of the delay and see how it changes the program.

```
/*
```

```
Fade
```

```
This example shows how to fade an LED on pin 9  
using the analogWrite() function.
```

Intel Ultimate Engineering Experience (IUEE)

This example code is in the public domain.

```
*/

int led = 9;           // the pin that the 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 pin 9 to be an output:
    pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
    // set the brightness of pin 9:
    analogWrite(led, brightness);

    // change the brightness for next time through the loop:
    brightness = brightness + fadeAmount;

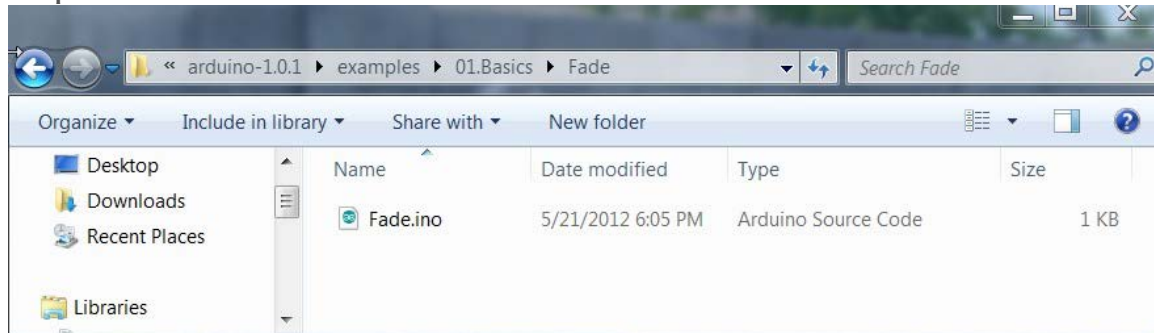
    // reverse the direction of the fading at the ends of the
fade:
    if (brightness == 0 || brightness == 255) {
        fadeAmount = -fadeAmount ;
    }
    // wait for 30 milliseconds to see the dimming effect
    delay(30);
}
```

IUEE Project 3, part B

Fade using a Button

You will be writing your own code, draw your own schematic and construct your own circuit. You will use the button (SPST) and LED from the previous projects to construct a third project.

Note: Use the Fade example program in the examples/01.Basics/Fade directory as a template.



Requirements

When the button is “off”, the LED is “on”.

When the button is “pressed”, the LED light starts “fading”.

This is similar to what happens when someone presses a doorbell.

Hardware

Arduino Board

LED

momentary button or switch

270 ohm resistor

breadboard

hook-up wire

Circuit

Construct your own circuit, using the previous sketch circuits as a guide.

Schematic:

Draw your own schematic, using the previous sketch circuits as a guide.

Code

Write (or modify another sketch) your own code, commenting the sketch to describe the changes or operations you have to add.

IUEE Project 3, part C

myProject

If you have time, try coming up with another circuit and sketch using the basic items in your kit. You could use any additional parts. Remember to write a description, a set of requirements, design the circuit, draw a schematic and write the code.