What Will You Learn?

Learn to wire and write code to control a Servo and an LED at the same time.

Why Should You Learn This?

Now that you know how to use a LED and a Servo separately, you can expand on your ideas by learning how to program both together. While this is more complicated than wiring each of the components separately, it is something that is possible with a bit of time and a few extra parts.

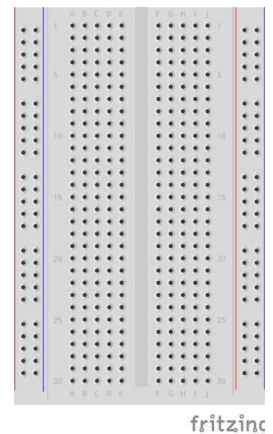# Here's What You'll Need:

Arduino Microcontroller

1 Servo

Continuous Servo is used in this activity, but you could also use a standard Servo)

Breadboard

2 Red Wires

2 Black Wires

1 White Wire

**1 resistor (100 – 300 ohms)**

1 LED

Safety First!

Never allow the red and black wires to touch together while they are connected to a powered microcontroller, as that creates a short circuit. A short circuit can potentially cause the wires and/or the microcontroller to get hot enough to burn the skin. In addition, in the event of a short circuit, the microcontroller can potentially catch on fire.

Let's Get Connected!

1. Let's focus on connecting the Servo first. Begin by connecting a red wire from Vin on the microcontroller to the red (+) wire of the Servo.
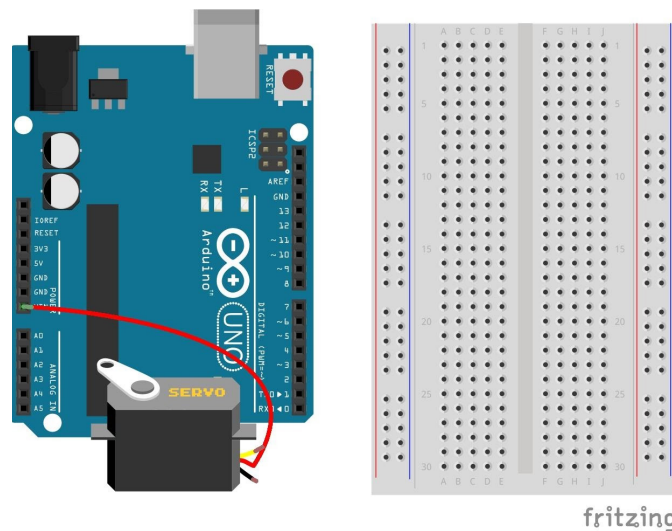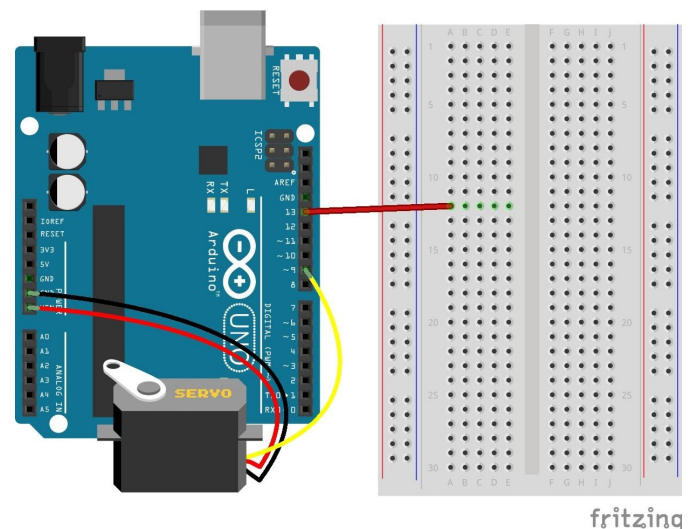
2. Next, connect a black wire from GND to the black (-) wire of the Servo.
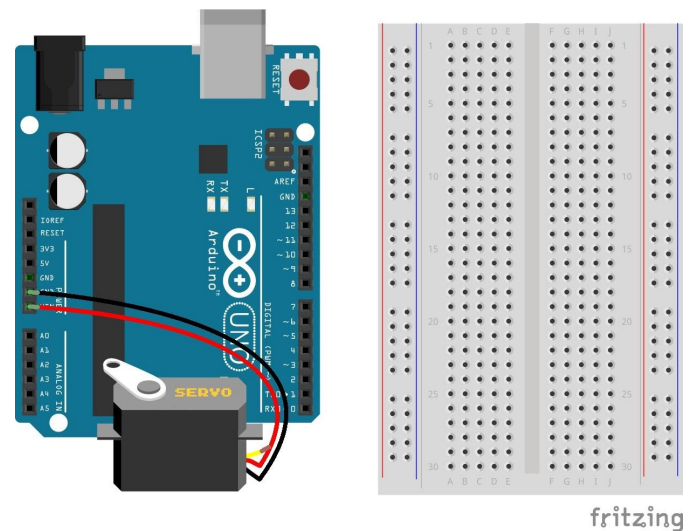
3. To finish hooking up the Servo, connect a white wire from 9 to the white (control) wire of the Servo (the wire in the diagram is yellow, but your wire may be white).
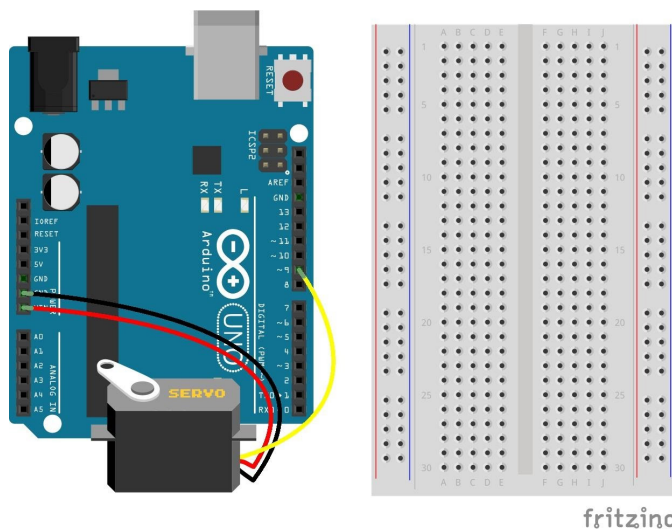
4. Now that the Servo is connected, you can focus on the LED. Connect a red wire from 13 to any row of the breadboard (row 12 is used below). You will use this same row to connect your resistor later.

5. Next, connect a black wire from GND to the negative rail, or the blue lined column.
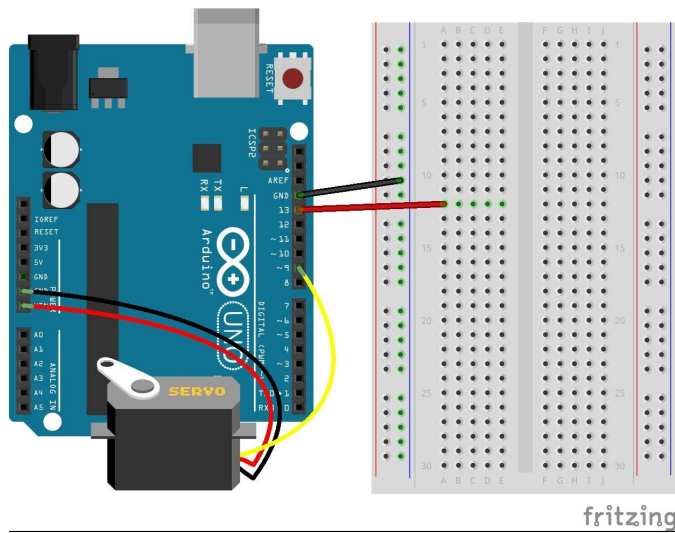
6. You will also need a resistor to make sure you don't burn out your LED. Connect the resistor from the row that has the red wire (row 12 below) to another row on the breadboard (row 8 is used below).
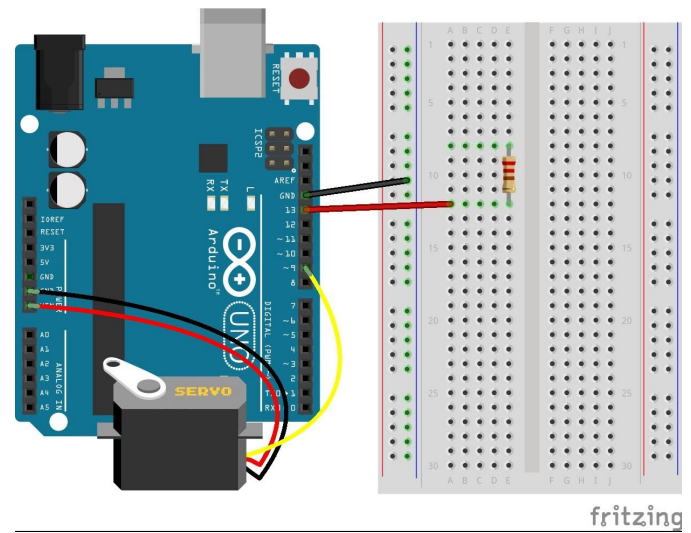

7. Lastly, connect the LED. The long leg (positive) will go in the same row as the end of the resistor that is not connected to the red wire (row 8), and the short leg (negative) will go in the negative rail.
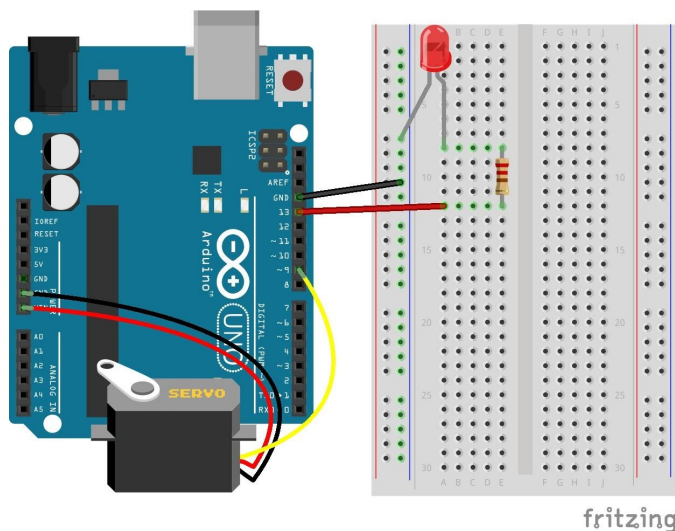
Time to Write Some Code!
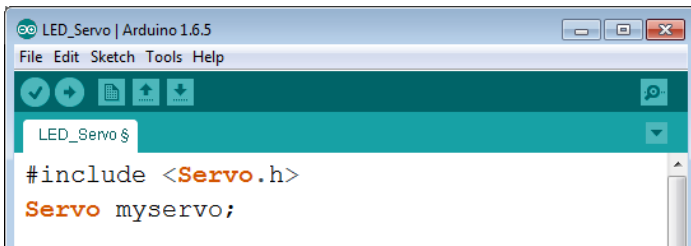
1. Start by opening the Arduino programming language and deleting the text to start with a blank document. This is a good time to make sure that you've selected the right port to connect your board (go to Tools > Port then select the port with Arduino written in brackets beside the number). It may be tempting to just copy and paste the Servo code and LED Blink code into one document, but combining the two programs directly won't work. In this activity, you'll learn how to write the code from start to finish.
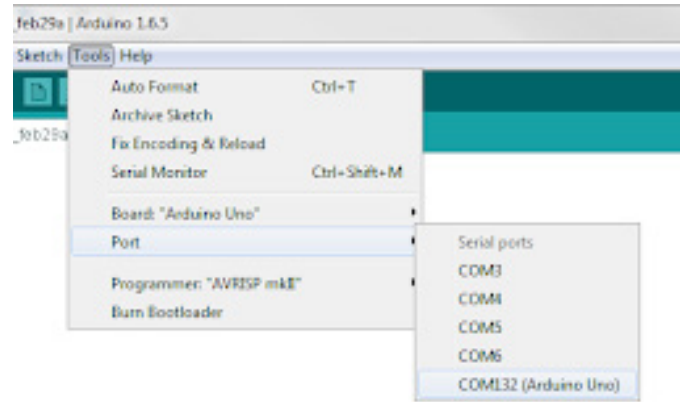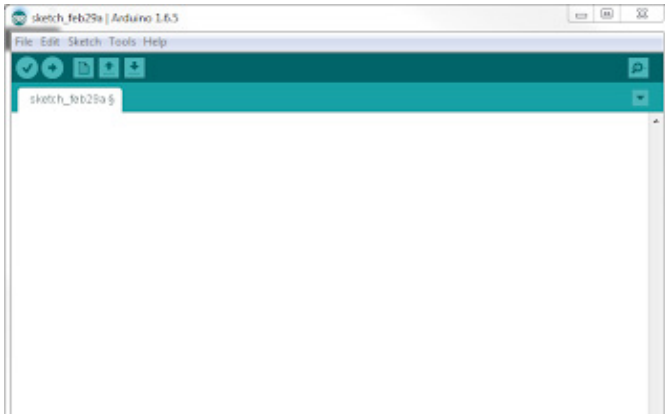
2. Whenever you use a Servo, you need to include the Servo library in your program. To do this, go to Sketch > Include Library > Servo. This will automatically insert a line that says: "`#include <Servo.h>`".
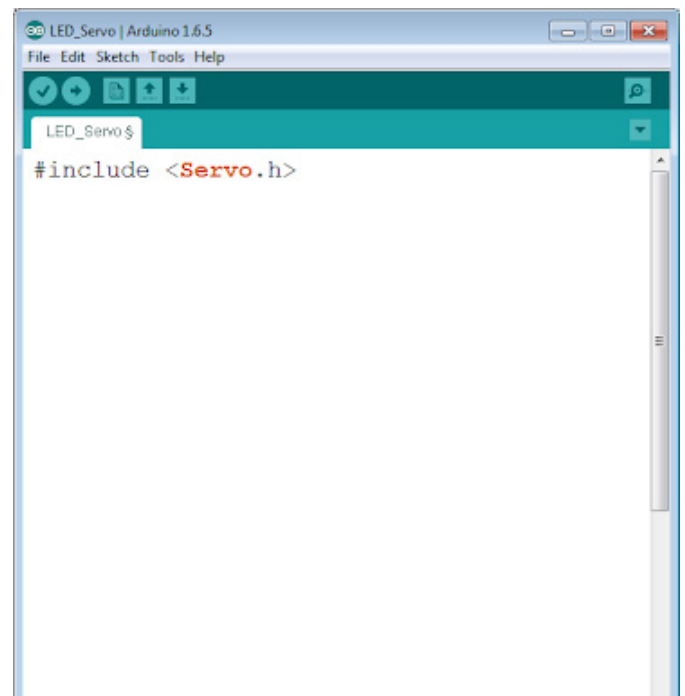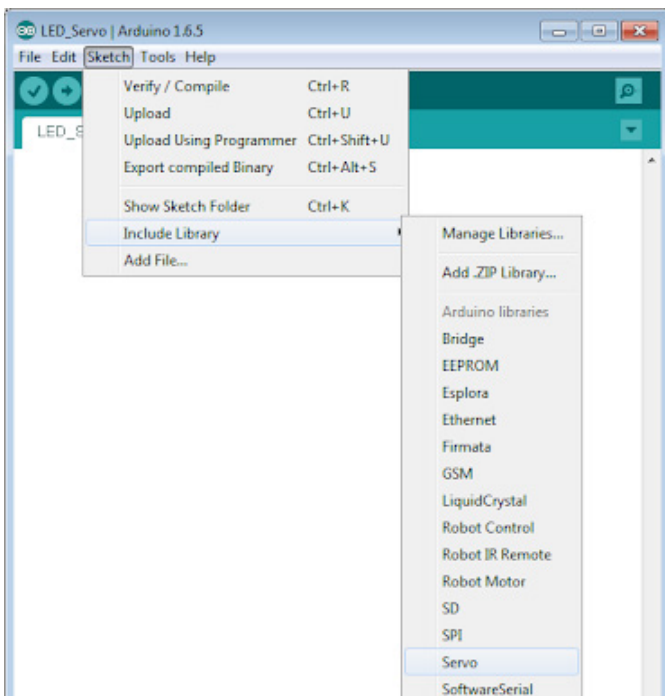
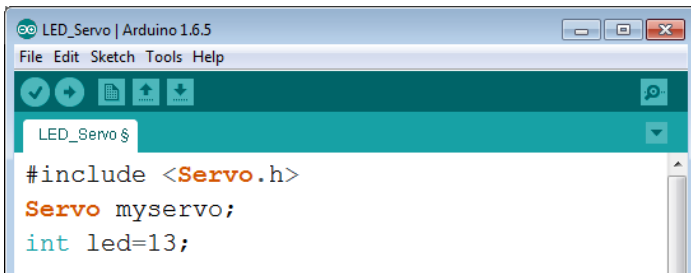3. Next, you need to name your Servo, as you did in the "Servo" activity. The example below names the Servo "myservo". You can name your Servo whatever you want, just be sure to use that name throughout your entire code.
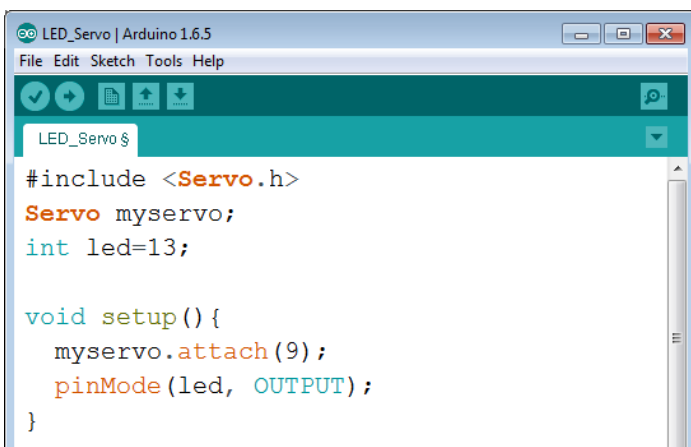
4. You are also going to make a name for your LED. To do this, you need to create a variable that tells you what pin your LED is connected to. To create a variable, use the line of code "`int led=13`". In this line of code, "led" is the name of your variable (it's good to make this name something easy to understand). 13 is the pin that your LED is connected to, and "int" is the type of variable that you are creating. "int" is short for integer, which just means that it will be a number. This is the most common type of variable that you will use.

5. Now that you have named your Servo and LED, you're ready to write the Setup section of your program. Because you are using 2 components, a Servo and an LED, you need to setup both. You'll set up the Servo and the LED the same way that you did in the "Servo" and "LED Blink" activities, but this time you will put them both in the same program.

6. Next, you can write the loop function of your program. It may be tempting to copy and paste the code from the "Servo" and "LED Blink" examples together. If you tried that you would end up with the following:

```
void loop() {
  digitalWrite(led, HIGH);      //turn the led on
  delay(1000);                  //wait 1 second
  digitalWrite(led, LOW);       //turn the led off
  delay(1000);                  //wait 1 second
  myservo.write(0);             //spin the servo to the left
  delay(2000);
  myservo.write(180);           //spin the servo to the right
  delay(2000);
}
```

Although this will work, it will not make the LED and Servo do what you want them to do – which is to have the LED blink and Servo move at the same time. The LED will turn on, stay on for a second, then turn off, but because the microcontroller runs through all the lines of code one at a time before starting over from the top, the light will stay off for 5 seconds (1000 + 2000 + 2000). The same thing happens with the Servo. Since you are using a continuous Servo, it will start out not moving, then will spin one direction for 2 seconds and then in the opposite direction for 4 seconds (2000 + 1000 + 1000). This table may help you see it more clearly:

Full Code in "void loop(0) {	Focus on LED	Focus on Servo
digitalWrite(led, HIGH); delay(1000); digitalWrite(led, LOW); delay(1000); myservo.write(0); delay(2000); myservo.write(180); delay(2000);	digitalWrite(led, HIGH); delay(1000); digitalWrite(led, LOW); delay(1000); myservo.write(0); delay(2000); myservo.write(180); delay(2000);	digitalWrite(led, HIGH); delay(1000); digitalWrite(led, LOW); delay(1000); myservo.write(0); delay(2000); myservo.write(180); delay(2000);

If you want the light to blink on and off every second and the Servo to change directions every second, you will need to do both actions (for the LED and the Servo) before each delay. It should look like the code below:

```
LED_Servo | Arduino 1.6.5
File  Edit  Sketch  Tools  Help

LED_Servo §

#include <Servo.h>
Servo myservo;
int led=13;

void setup(){
  myservo.attach(9);
  pinMode(led, OUTPUT);
}

void loop(){
  digitalWrite(led, HIGH);
  myservo.write(0);
  delay(1000);
  digitalWrite(led, LOW);
  myservo.write(180);
  delay(1000);
}
```

Congratulations! You should now have an LED blinking and a Servo spinning back and forth in unison. Feel free to change the delay or make the Servo spin slower by changing the value in `myservo.write( );` (see "Programming a Servo" for more information).

Tips for Success!

- If you're having issues with your code, review all the lines in the example code to make sure you didn't miss something or refer to the "Troubleshooting" document for more help.

- If your LED is not working, be sure that all your connections have been made properly and the LED is plugged in the right way. If you've checked each of these things, then try a different LED.

- If your Servo is not working, make sure that all your wires are in the correct pins and that the red, black and white wires are in the right order on the Servo attachment. If the wiring looks correct, you may want to try another Servo.
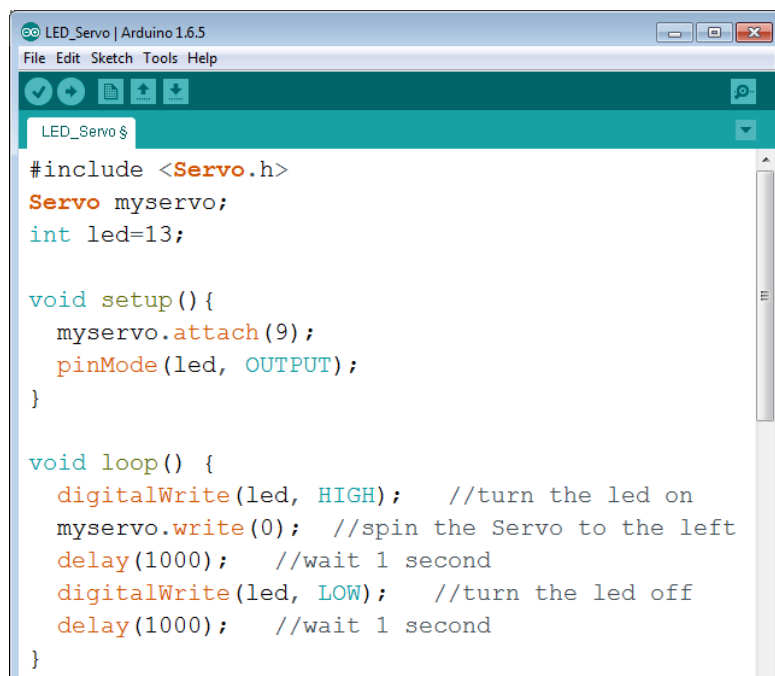
Are You Ready for Some Challenges?

Below are some other combinations of blinking and spinning for you to try to figure out on your own as challenge problems. The possible solutions are posted below, but try doing them yourself first. Remember, code can look different and have the same effect.

1. Try to make your LED continue to blink at the same rate (on for 1 second, then off for 1 second) and have your Servo spin only one way (instead of both ways). This will allow you to change the rate of the blinking without changing the motion of the Servo.

2. Try to make your LED Blink fast (with a delay of 100 milliseconds) and your Servo spin both ways for 1 second each direction.

3. Try to incorporate the LED Fade code with a Servo. You may want to start by having the Servo only spinning one way and then try to have it change directions with the rate of the fade.

Challenge Solutions:

Note: For most of these challenges, I've only posted the loop, since the variables and setup are the same. I have also posted comments, as it's good practice to write comments with your code, but you do not need to write them in your code as it doesn't change what the code does. Remember, comments are for humans.

Challenge 1 Solution:
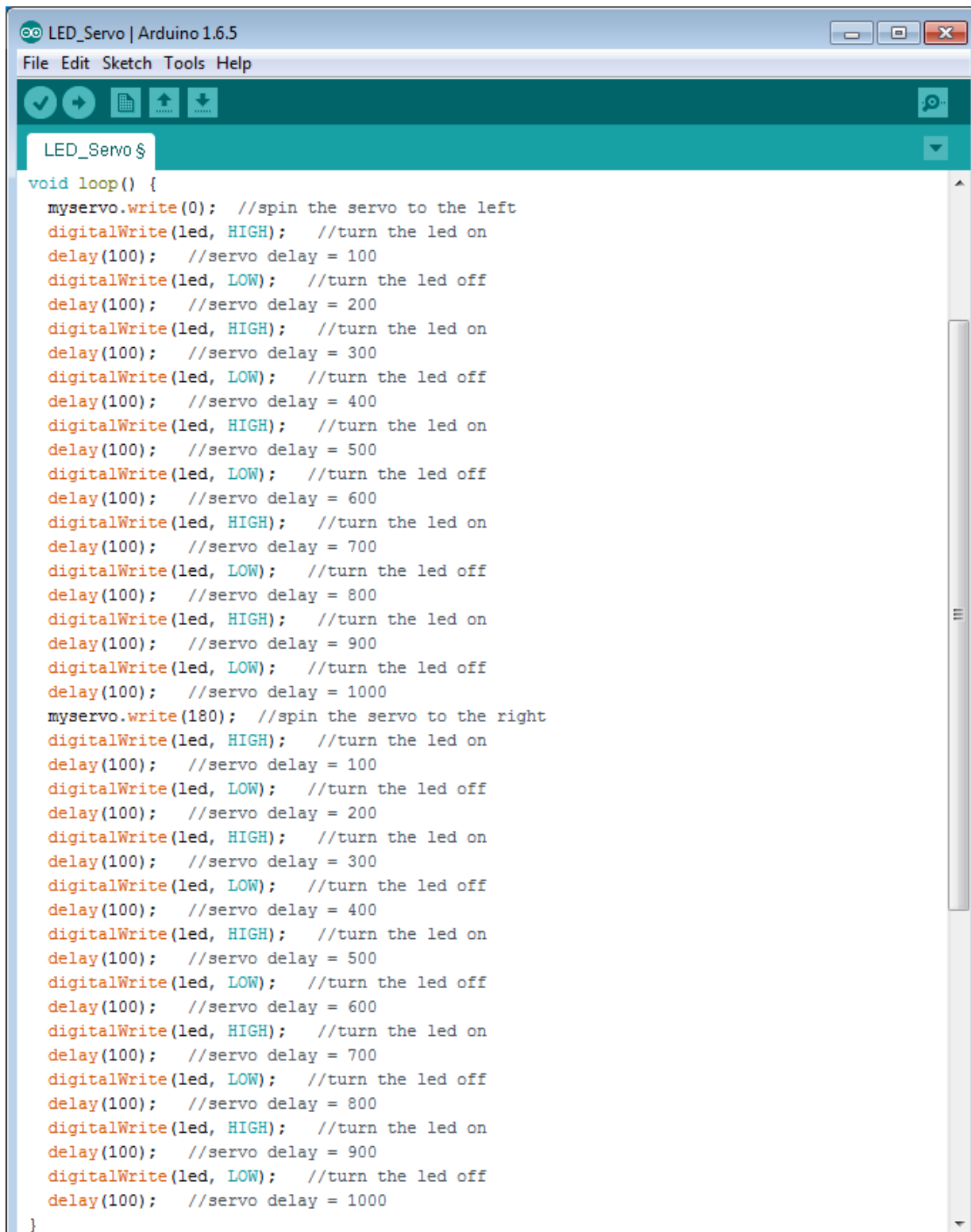

```
#include <Servo.h>
Servo myservo;
int led=13;

void setup(){
  myservo.attach(9);
  pinMode(led, OUTPUT);
}

void loop() {
  digitalWrite(led, HIGH);   //turn the led on
  myservo.write(0);  //spin the Servo to the left
  delay(1000);   //wait 1 second
  digitalWrite(led, LOW);   //turn the led off
  delay(1000);   //wait 1 second
}
```

You can always change the delay to be longer or shorter. This will change the rate at which your light blinks, but because you only told your Servo to spin one direction and never changed the direction, it will continue to spin that direction no matter the value of the delay.

Challenge 2 Solution:

We want to have a longer delay for the Servo than the LED, but we can't assign a separate delay to each part because a delay makes the entire program wait before it reads the next line. So, the loop in our program should look something like this:

```
LED_Servo | Arduino 1.6.5
File  Edit  Sketch  Tools  Help

LED_Servo §

void loop() {
  myservo.write(0);  //spin the servo to the left
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 100
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 200
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 300
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 400
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 500
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 600
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 700
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 800
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 900
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 1000
  myservo.write(180);  //spin the servo to the right
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 100
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 200
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 300
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 400
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 500
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 600
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 700
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 800
  digitalWrite(led, HIGH);    //turn the led on
  delay(100);    //servo delay = 900
  digitalWrite(led, LOW);    //turn the led off
  delay(100);    //servo delay = 1000
}
```

Challenge 3 Solution:

I started with the LED Fade example and then added in the Servo code to make the Servo spin only one way. Making the Servo change directions is a harder task, but may be easier once you have completed "Servo Sweep Challenge #2".

```
LED_Servo | Arduino 1.6.5
File Edit Sketch Tools Help

LED_Servo §

#include <Servo.h>
Servo myservo;
int led = 11;
int brightness = 0;
int fadeAmount = 5;

void setup() {
  myservo.attach(9);
  pinMode(led, OUTPUT);
}

void loop() {
  myservo.write(0);
  analogWrite(led, brightness);
  brightness = brightness + fadeAmount;
  if (brightness == 0 || brightness == 255) {
    fadeAmount = -fadeAmount ;
  }
  delay(30);
}
```