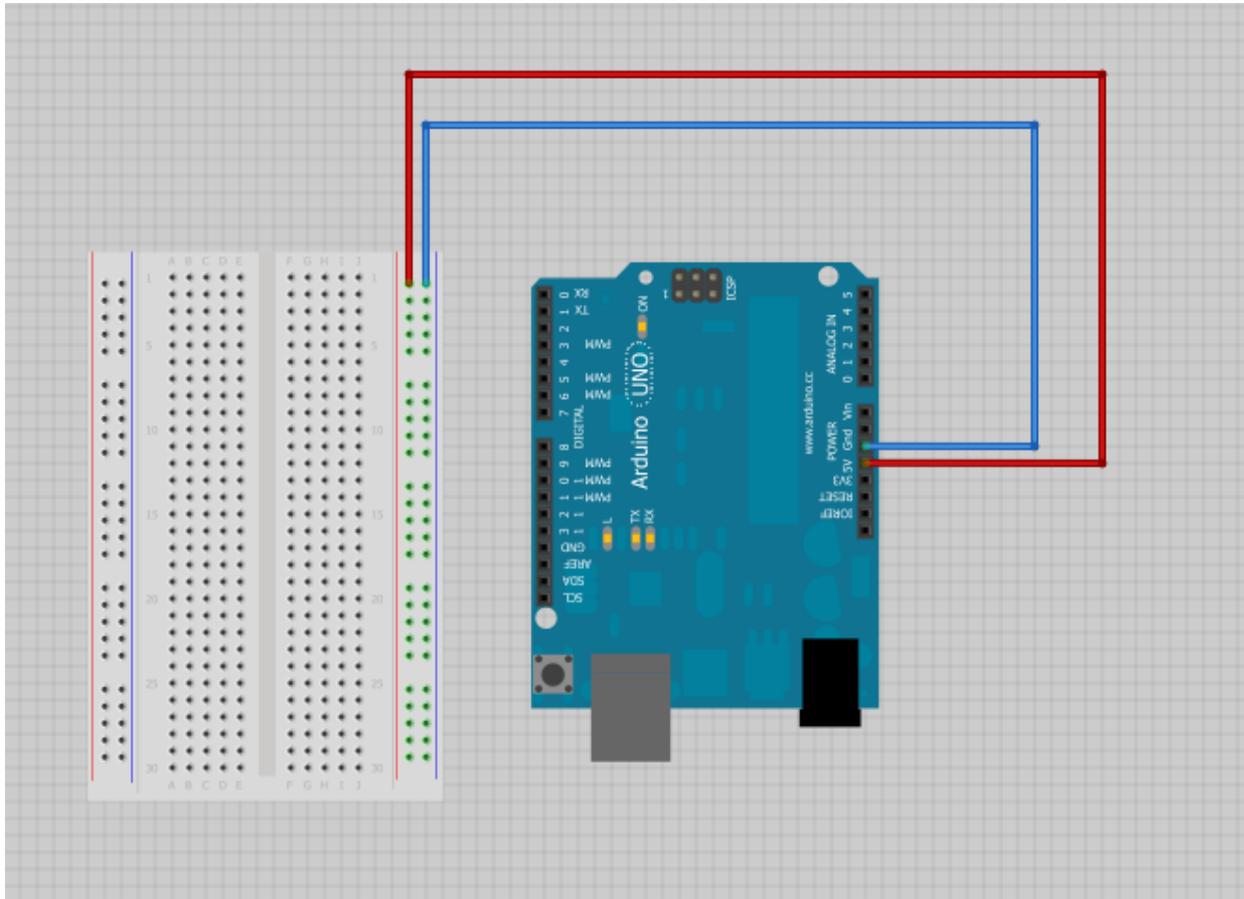
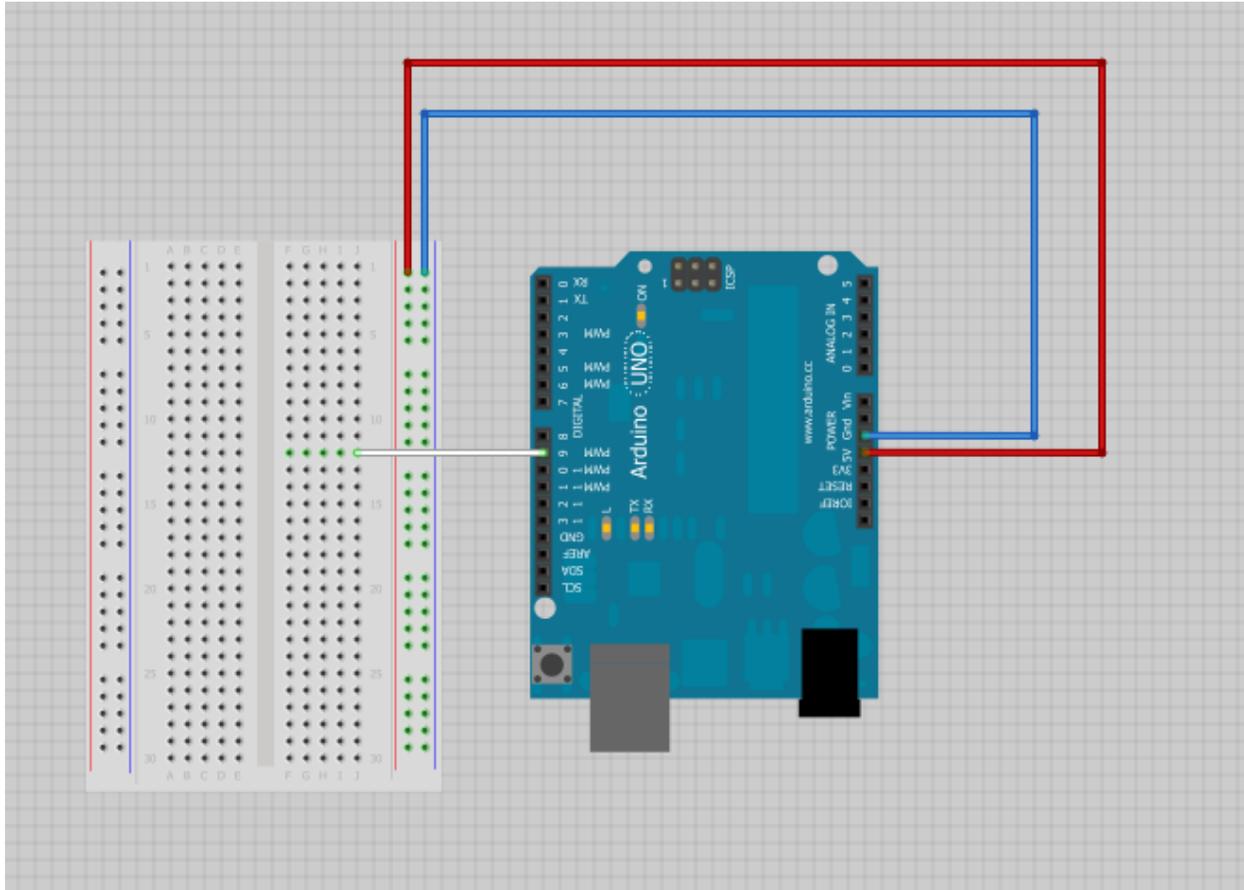


Process:

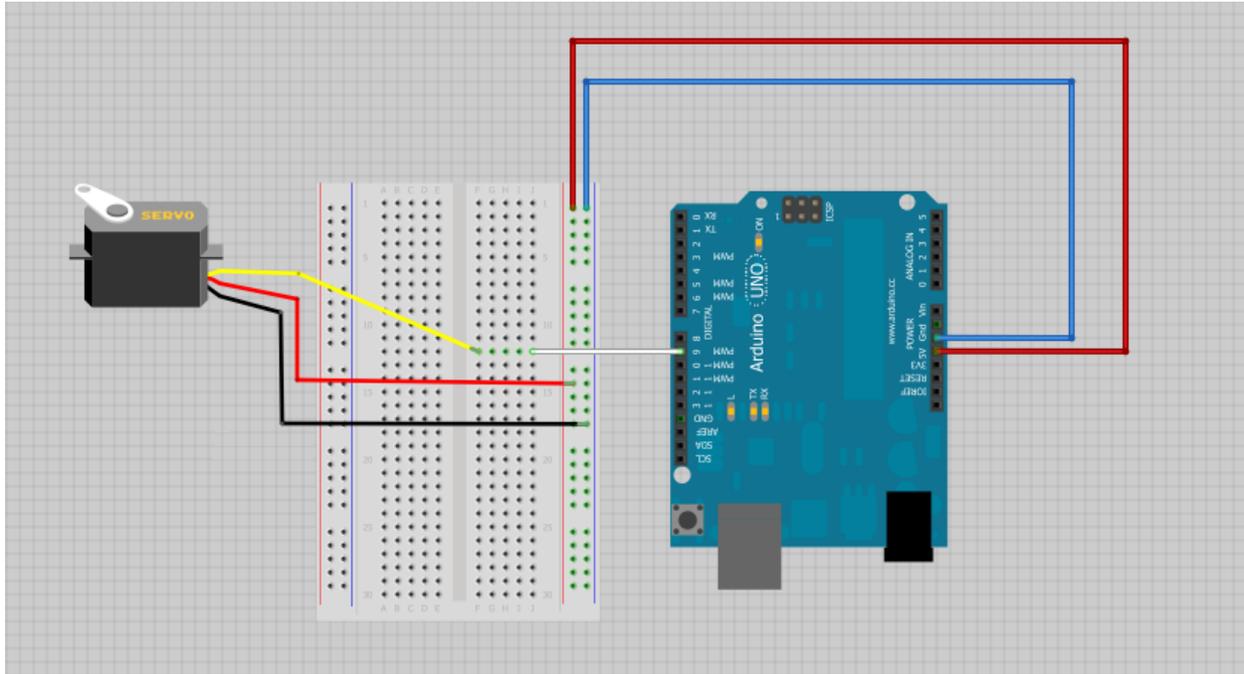
1. Remove the LED lights and Piezo Speakers from Projects 1,2, and 3. Make sure these are stored safely in the kit.
2. Remove the 330 Ohm resistors and store them in the required package.
3. Configure the Breadboard and Arduino to the following form:



4. We will now set up for Servo 1. Wire Pin 9 on Arduino to J12



5. We can now connect the Servo.
 - a. Servo Black to Ground (Blue Rail)
 - b. Servo Red to Current (Red Rail)
 - c. Servo White to F12 (Signal)



Sketch Code:

We will now program the Servos to move in the following manner:

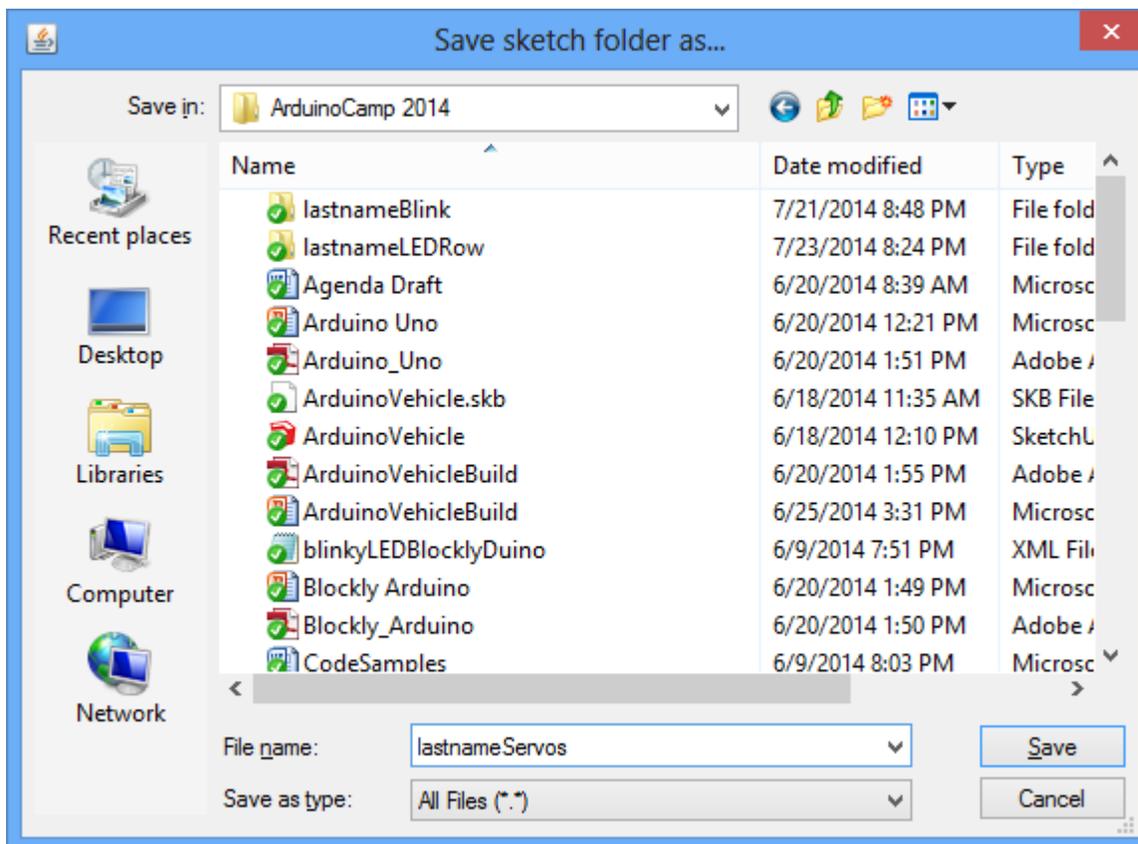
-forward for 2 seconds.

Process:

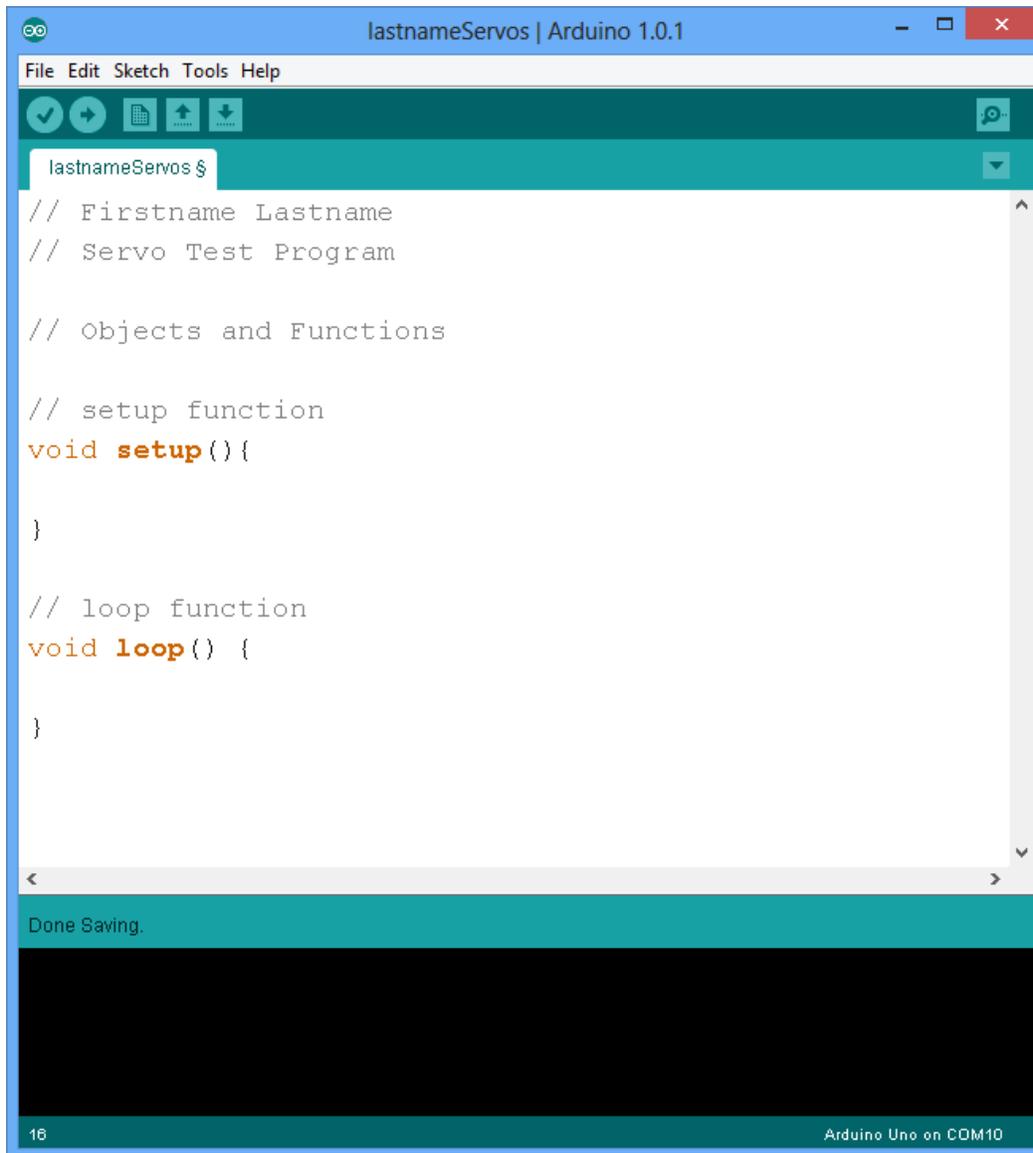
1. Start your Arduino Sketch Program by clicking on the Sketch Icon.



2. Save the code by selecting "File-Save" and naming the program "lastnameServos".



3. We want to set up the code by writing comments and defining the two required functions (void setup() and void loop()).



The screenshot shows the Arduino IDE interface. The window title is "lastnameServos | Arduino 1.0.1". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". The toolbar contains icons for saving, running, and uploading. The sketch editor shows the following code:

```
lastnameServos $
// Firstname Lastname
// Servo Test Program

// Objects and Functions

// setup function
void setup(){

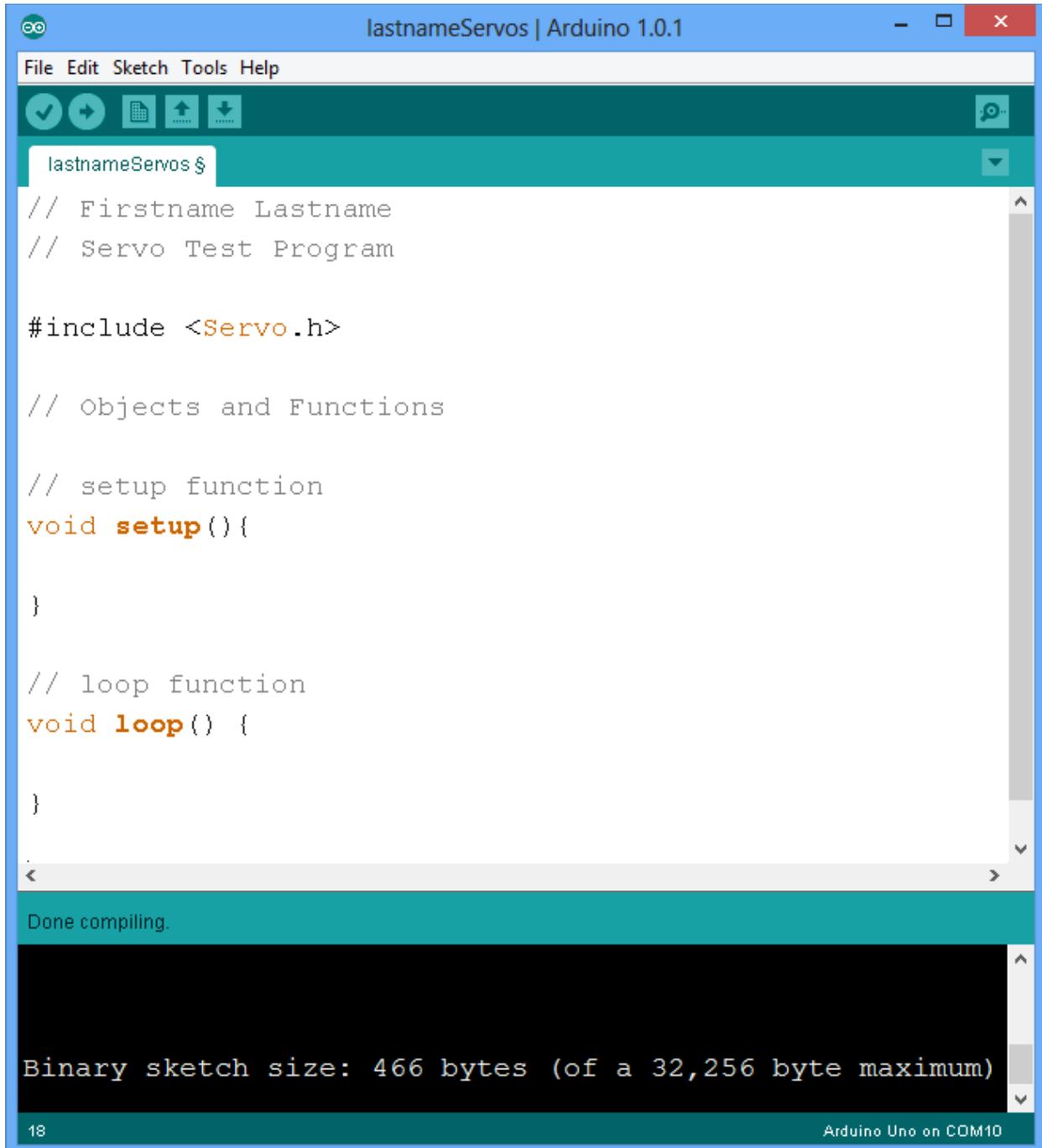
}

// loop function
void loop() {

}
```

Below the editor, a status bar displays "Done Saving." and "Arduino Uno on COM10". The page number "16" is visible in the bottom left corner.

4. Now we need to include an #include statement to bring in the library for the Servo functions.



```
lastnameServos | Arduino 1.0.1
File Edit Sketch Tools Help
lastnameServos$
// Firstname Lastname
// Servo Test Program

#include <Servo.h>

// Objects and Functions

// setup function
void setup(){

}

// loop function
void loop() {

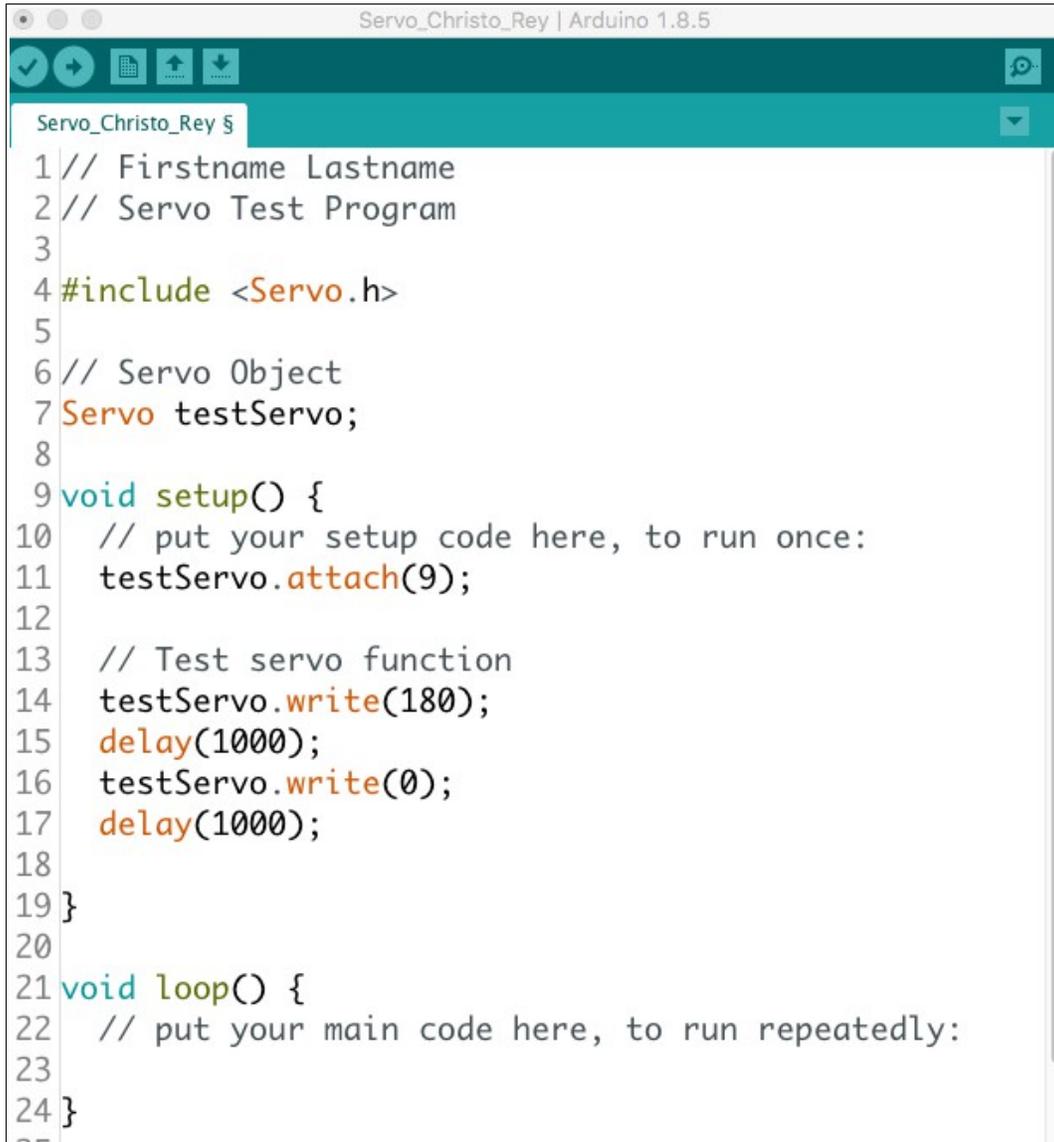
}

Done compiling.

Binary sketch size: 466 bytes (of a 32,256 byte maximum)

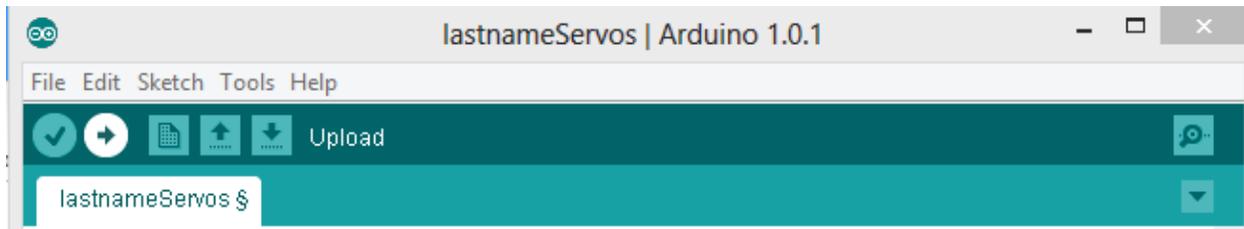
18 Arduino Uno on COM10
```

5. We will now declare the Servo object, map servo to pin 9, and test the servo.



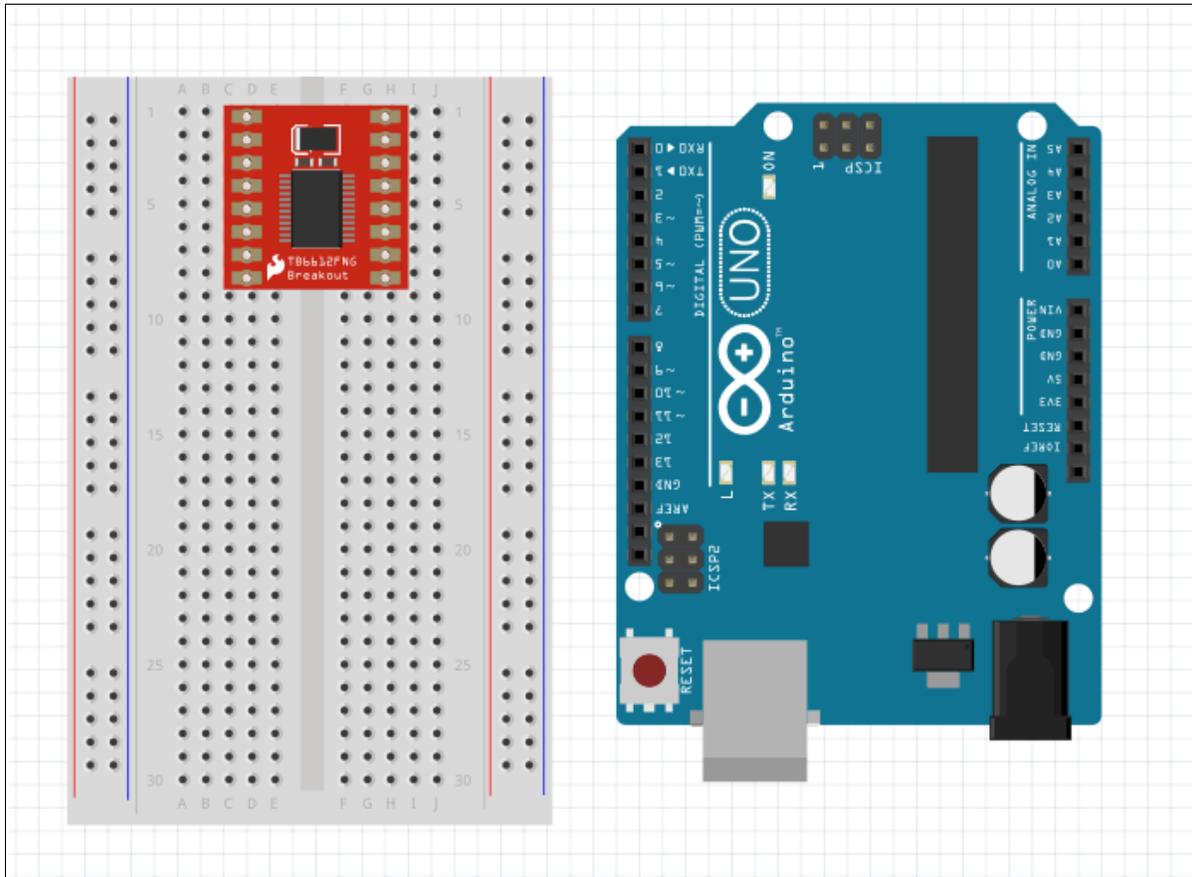
```
Servo_Christo_Rey | Arduino 1.8.5
Servo_Christo_Rey §
1 // Firstname Lastname
2 // Servo Test Program
3
4 #include <Servo.h>
5
6 // Servo Object
7 Servo testServo;
8
9 void setup() {
10 // put your setup code here, to run once:
11 testServo.attach(9);
12
13 // Test servo function
14 testServo.write(180);
15 delay(1000);
16 testServo.write(0);
17 delay(1000);
18
19 }
20
21 void loop() {
22 // put your main code here, to run repeatedly:
23
24 }
```

6. Save the program and plug in your Arduino and Servo setup. Click the upload button to upload the code to the Arduino and run the program. The servo should turn in one direction and then another.

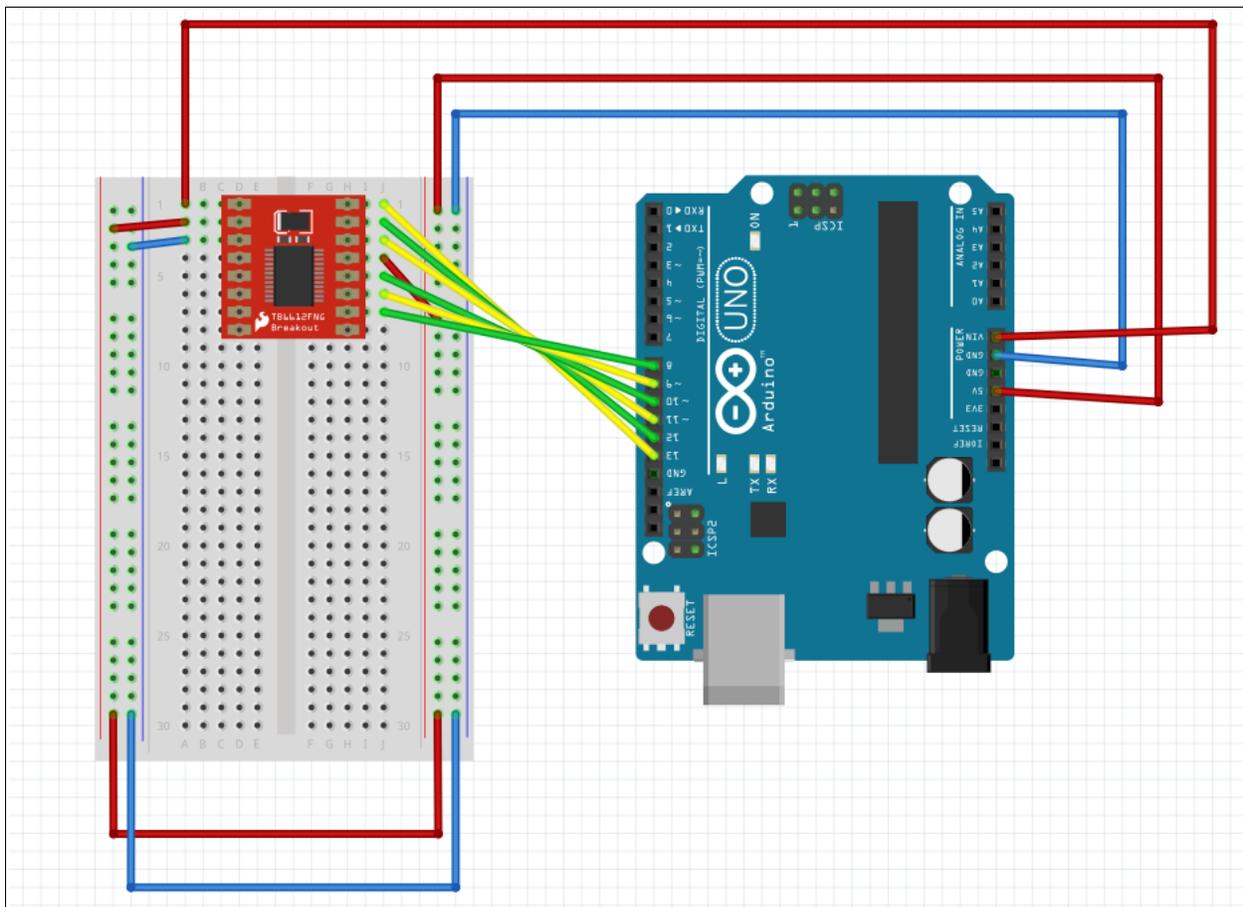
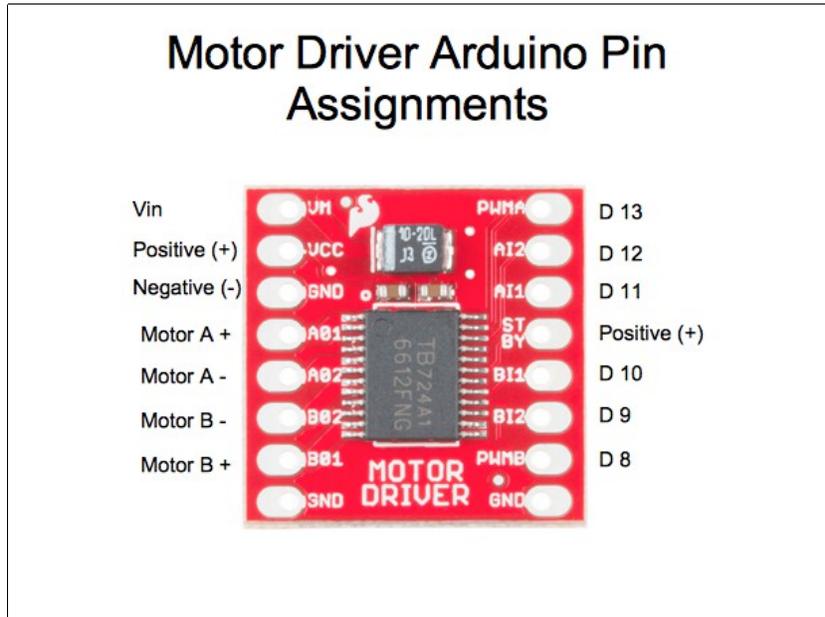


7. Now we want to add the two motors and the motor driver. The motor driver takes three inputs per motor; PWM for power, and two pins for direction.

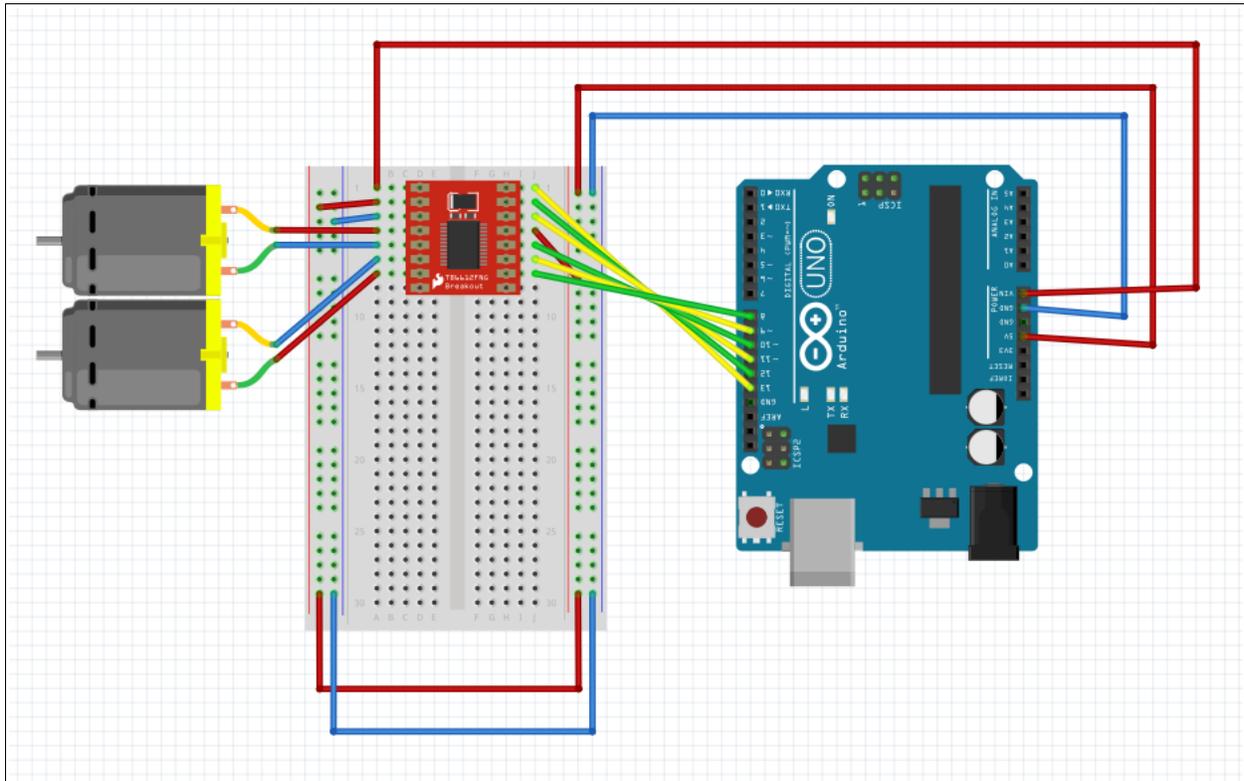
8. Plug the Motor Driver into the top of the breadboard as shown:



9. Wire the Arduino and Motor Driver as shown. The detailed pin assignments are also shown



10. Wire the two motors as shown below:



11. Start a new Arduino sketch and save it as "DC_Motor_Functions".

12. Write the following to set up the variables:

```
DC_Motor_Functions
1 // DC Motor Base Code for SIK Kit 4.0
2 // Based on SIK Example Code Circuit 5A
3 // Mr. Michaud: www.nebomusic.net
4
5 // PIN Variables
6 int AIN1 = 11;
7 int AIN2 = 12;
8 int PWMA = 13;
9
10 int BIN1 = 10;
11 int BIN2 = 9;
12 int PWMB = 8;
13
14 // Motor Speeds
15 int leftPower = 0;
16 int rightPower = 0;
17
```

13. Code the setup function to initialize the output pins:

```
17
18 void setup() {
19   // put your setup code here, to run once:
20   // Setup Motor Pins
21   pinMode(AIN1, OUTPUT);
22   pinMode(AIN2, OUTPUT);
23   pinMode(PWMA, OUTPUT);
24
25   pinMode(BIN1, OUTPUT);
26   pinMode(BIN2, OUTPUT);
27   pinMode(PWMB, OUTPUT);
28
29 }
30
```

14. Write the "spinMotorA()" function:

```
60
61 // Functions to Run Motors
62
63 void spinMotorA(int motorSpeed) {
64     if (motorSpeed > 0)
65     {
66         digitalWrite(AIN1, HIGH);
67         digitalWrite(AIN2, LOW);
68     }
69     else if (motorSpeed < 0)
70     {
71         digitalWrite(AIN1, LOW);
72         digitalWrite(AIN2, HIGH);
73     }
74     else
75     {
76         digitalWrite(AIN1, LOW);
77         digitalWrite(AIN2, LOW);
78     }
79     analogWrite(PWMA, abs(motorSpeed));
80 }
81
```

15. Write the "spinMotorB" function.

```
81
82 void spinMotorB(int motorSpeed) {
83     if (motorSpeed > 0)
84     {
85         digitalWrite(BIN1, HIGH);
86         digitalWrite(BIN2, LOW);
87     }
88     else if (motorSpeed < 0)
89     {
90         digitalWrite(BIN1, LOW);
91         digitalWrite(BIN2, HIGH);
92     }
93     else
94     {
95         digitalWrite(BIN1, LOW);
96         digitalWrite(BIN2, LOW);
97     }
98     analogWrite(PWMB, abs(motorSpeed));
99 }
```

16. Go back to the setup() function and write code to test the motors:

```
28
29 // Test the Motors
30 spinMotorA(150);
31 delay(500);
32 spinMotorA(0);
33 delay(500);
34
35 spinMotorA(-150);
36 delay(500);
37 spinMotorA(0);
38 delay(500);
39
40 spinMotorB(150);
41 delay(500);
42 spinMotorB(0);
43 delay(500);
44
45 spinMotorB(-150);
46 delay(500);
47 spinMotorB(0);
48 delay(500);
49
```

17. Save the code and upload to the Arduino. The motors should spin forward and backward.

Other Ideas:

-Add the other Servo to the system. The wiring should look something like this:

- Write a procedure to control both motors at the same time.

Vocabulary:

Servo: An electric motor and encoder that work together to use a pulse width modulated signal to control direction and speed of motor.

External Battery Pack: Provides additional power for motors or other devices in electronic device. Wired in separately from Arduino power.