

Arduino: DC Motors

Diagrams & Code

Brown County Library

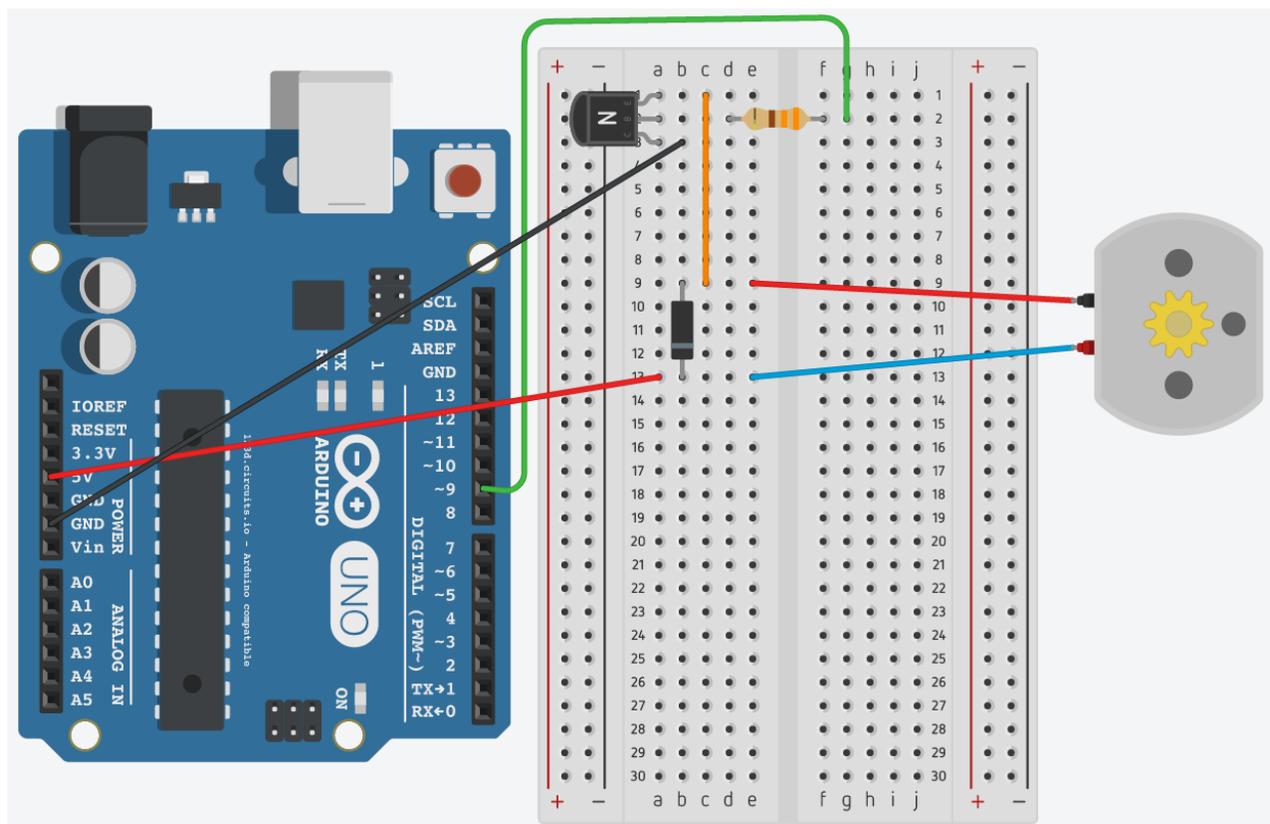
Warning: Make sure to plug the following components in correctly!

- **Transistors:** Transistors have an emitter, base and collector. With the transistors that are used by the library, when flat side of the transistor facing you the left leg is the emitter, the middle leg is the base and the right side is the collector. If the setup below isn't working, try flipping what you have connected to the left and right side of the transistor. There is an emitter, base and a collector and depending on the part it may be flip flopped. The emitter should be the one connected to ground.
- **Diodes:** Remember that diodes are polarized! Make sure the striped end is closest to the cable coming from the 5V pin (as indicated in the image below). If you put the diode the wrong way you are creating a short circuit!
- For more information on transistors and diodes, see the "Learn More" section at the end of this document.

Projects 01 & 02: Motor Turning On and Off and Changing Motor Speed

Components needed:

- Arduino Uno board
- breadboard
- 4 jumper wires
- DC toy motor
- 330 ohm resistor
- PN2222 NPN transistor
- 1N4004 diode

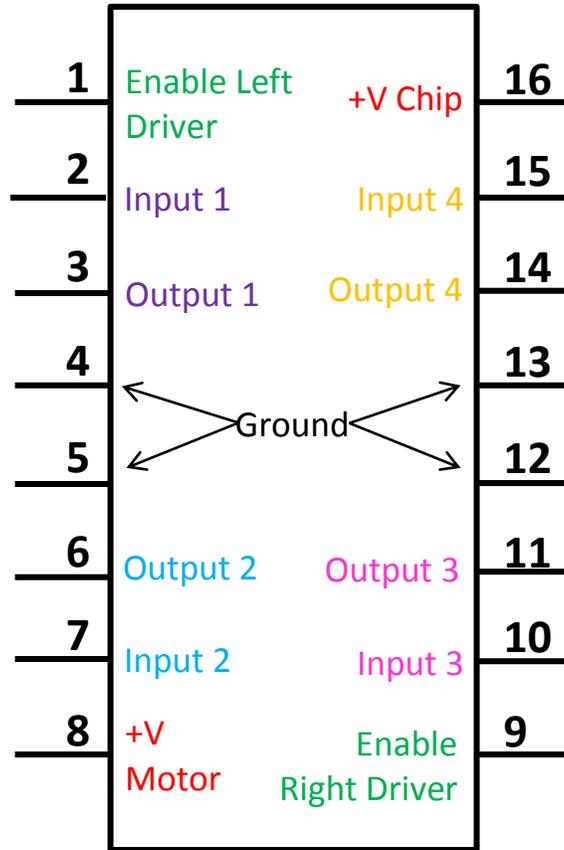


```
/*  
DC Motors 01 : Turn Motor On and Off  
Source: Code adapted from SparkFun Inventor's Kit Example Sketch 12  
(https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-12-driving-a-motor)  
*/  
  
int motorPin = 9; // motor is connected to pin 9  
  
void setup() {  
  pinMode(motorPin, OUTPUT); // motor pin is an output  
}  
  
void loop() {  
  digitalWrite(motorPin, HIGH); // turn the motor on (full speed)  
  delay(3000); // wait (delay) for this many milliseconds  
  digitalWrite(motorPin, LOW); // turn the motor off  
  delay(3000); // wait (delay) for this many milliseconds  
}
```

```
/*  
DC Motors 02 : Change Motor Speed  
Source: Code adapted from SparkFun Inventor's Kit Example Sketch 12  
(https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-12-driving-a-motor)  
*/  
  
int motorPin = 9; // motor is connected to pin 9  
  
void setup() {  
  pinMode(motorPin, OUTPUT); // motor pin is an output  
}  
  
void loop() {  
  int Speed1 = 200; // between 0 (stopped) and 255 (full speed)  
  int Speed2 = 50; // between 0 (stopped) and 255 (full speed)  
  
  analogWrite(motorPin, Speed1); // runs the motor on Speed 1  
  delay(500); // wait (delay) for this many milliseconds  
  analogWrite(motorPin, Speed2); // runs the motor on Speed 2  
  delay(1000); // wait (delay) for this many milliseconds  
}
```

Project 03: Using an H-Bridge

This is what each leg of an H-bridge means:



For more information about how H-bridges work, see:

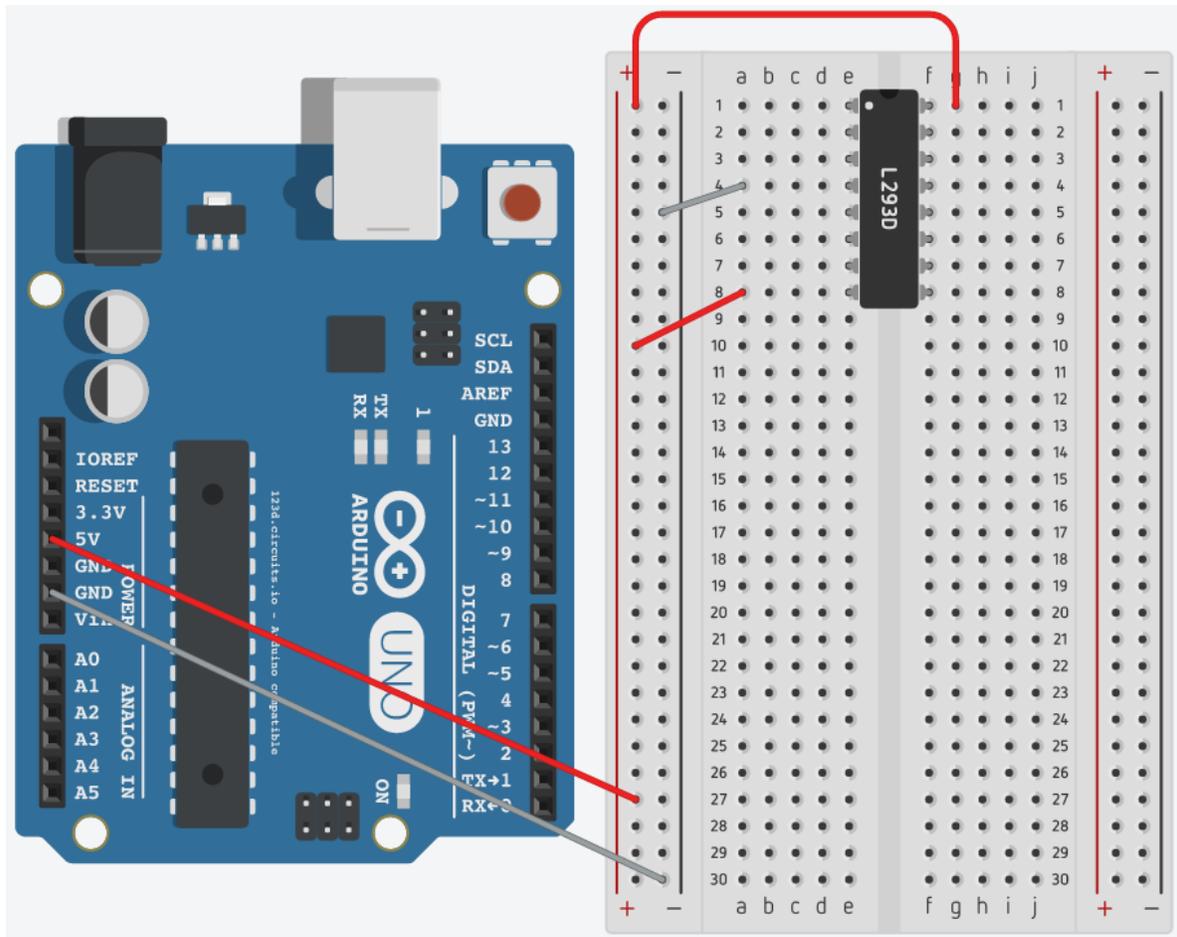
H-Bridges: The Basics. <http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>

Project 03: Using an H-Bridge

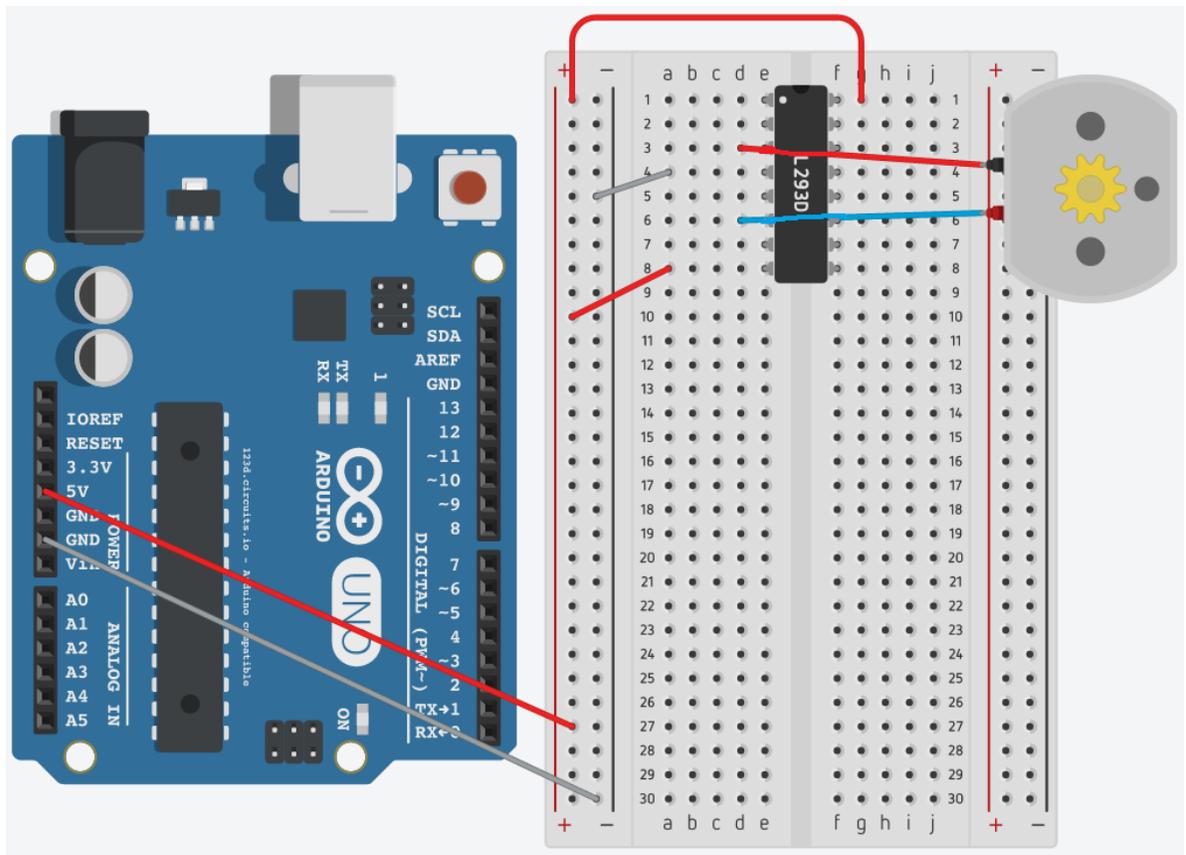
Components needed:

- Arduino Uno board
- breadboard
- 8 jumper wires
- DC toy motor
- H-bridge – L293D Motor Control Chip

Build in stages:

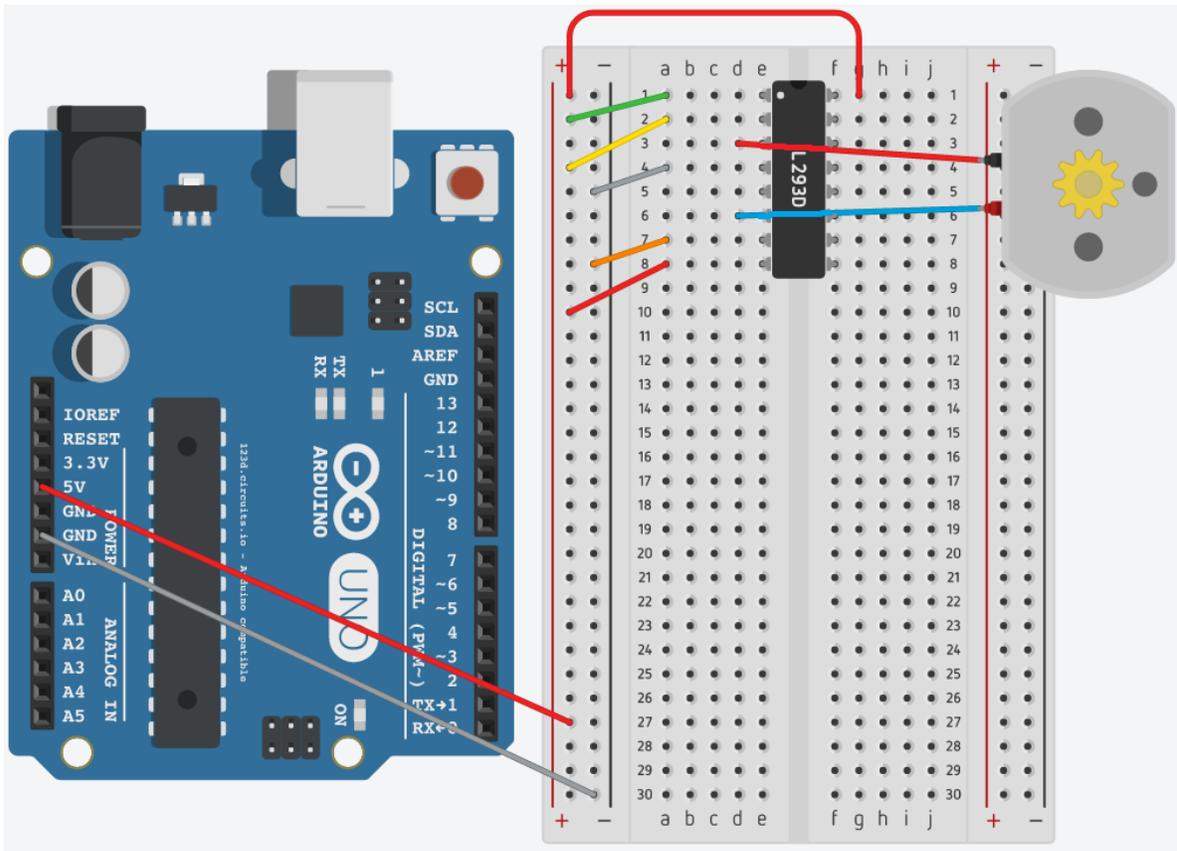


Project 3 continued:



Project 03 continued:

Note: No code is needed for this project – you will manually move jumper wires to make the DC motor start, stop and move in the direction that you want. However, make sure to plug the Arduino back into the computer to provide power once you have completed the setup below.



Now what?

- We will now control the motor manually by moving the jumper wires.
- Once the Arduino is plugged back in, the motor should be spinning – we'll call the direction that it is currently spinning "Direction A."
- Move Pin 1 (Enable) to the ground gutter. The motor will stop, as we have just disabled the left side of the driver. Reconnect to power to start the motor again.
- Now move Pin 2 (Input 1) to ground. Now Pins 2 & 7 (Inputs 1 & 2) are both connected to ground and the motor will stop.
- Move Pin 7 (Input 2) to power. As Pins 2 & 7 are connected differently again, the motor will begin but rotate in the opposite direction (Direction B).
- Connect Pin 2 (Input 1) back to power. Pins 2 & 7 (Inputs 1 & 2) are both connected to power and the motor will stop.
- Use the table to the right as a reference.

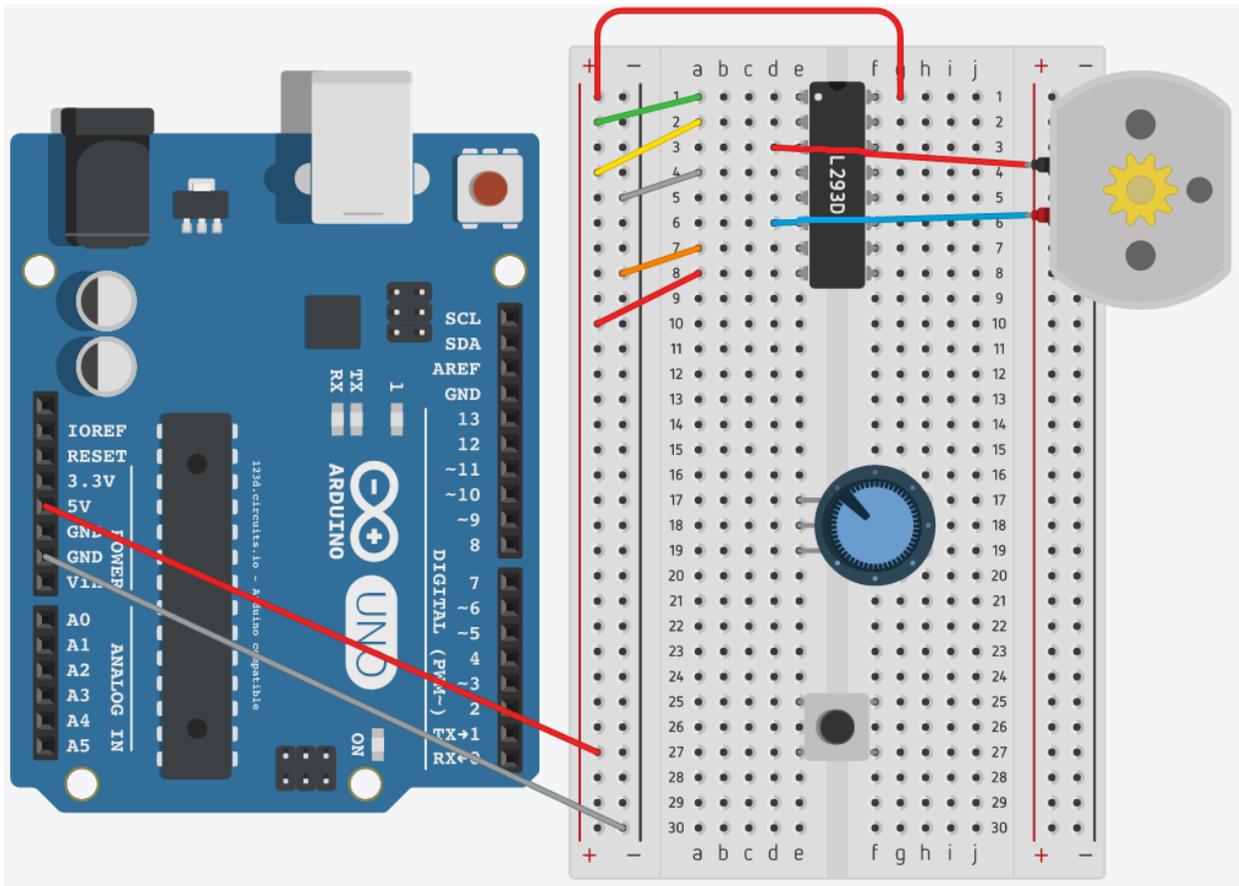
Pin 2 (Input 1)	Pin 7 (Input 2)	Motor
Power	Ground	Turns (A)
Ground	Ground	Stopped
Ground	Power	Turns (B)
Power	Power	Stopped

Project 04: Controlling a Motor

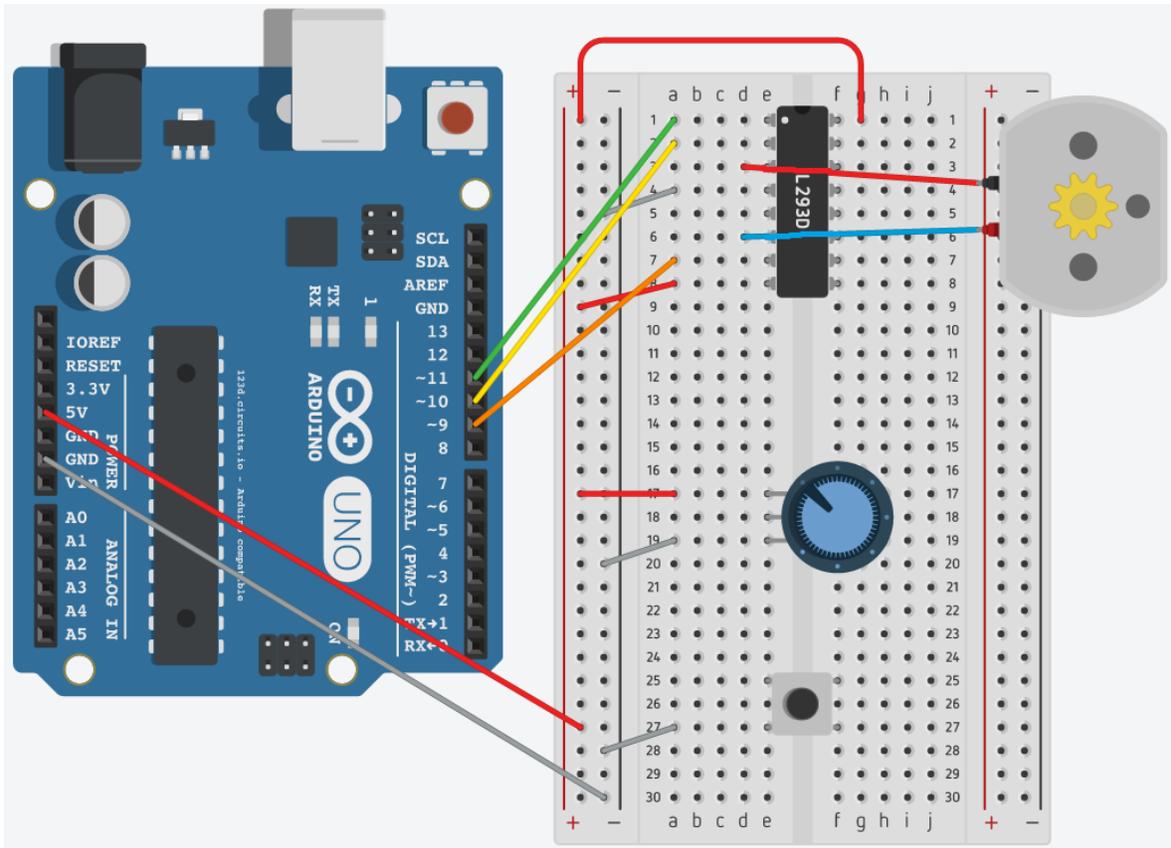
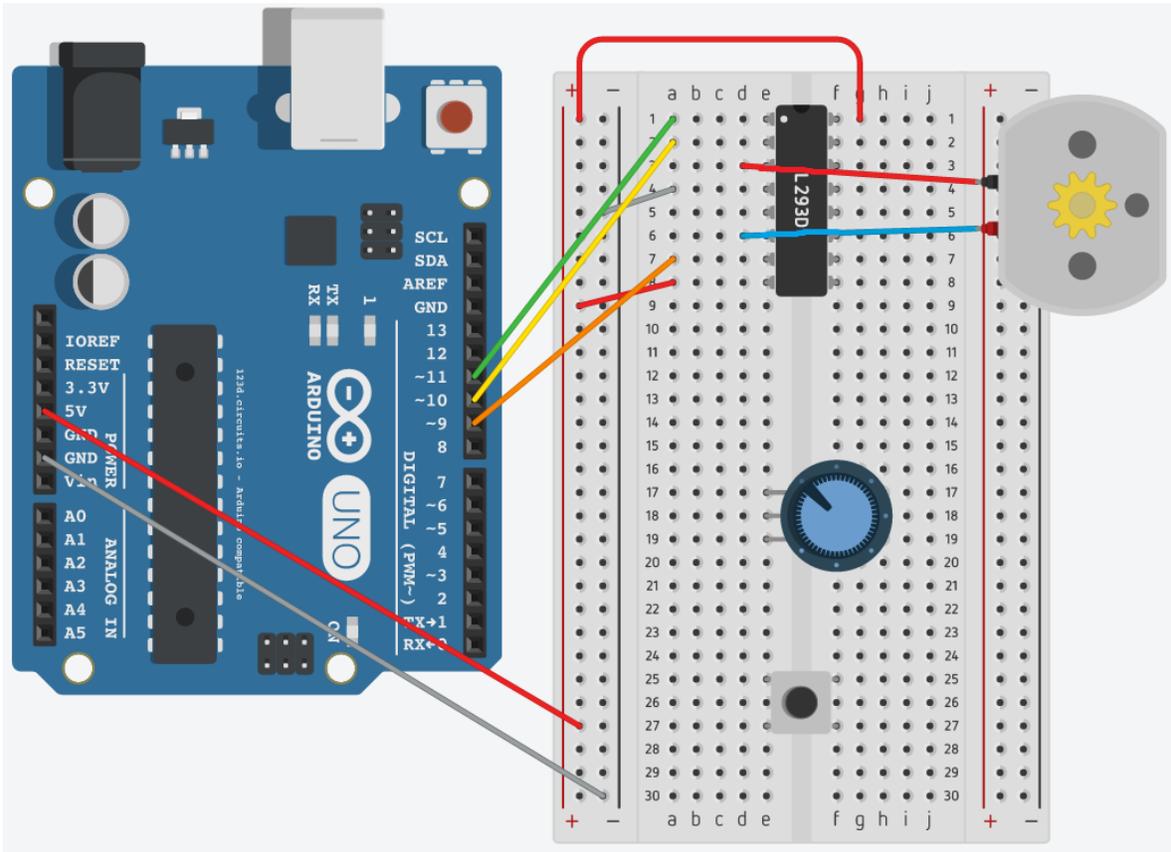
Components needed:

- Arduino Uno board
- breadboard
- 8 jumper wires
- DC toy motor
- H-bridge – L293D Motor Control Chip
- Potentiometer
- Button

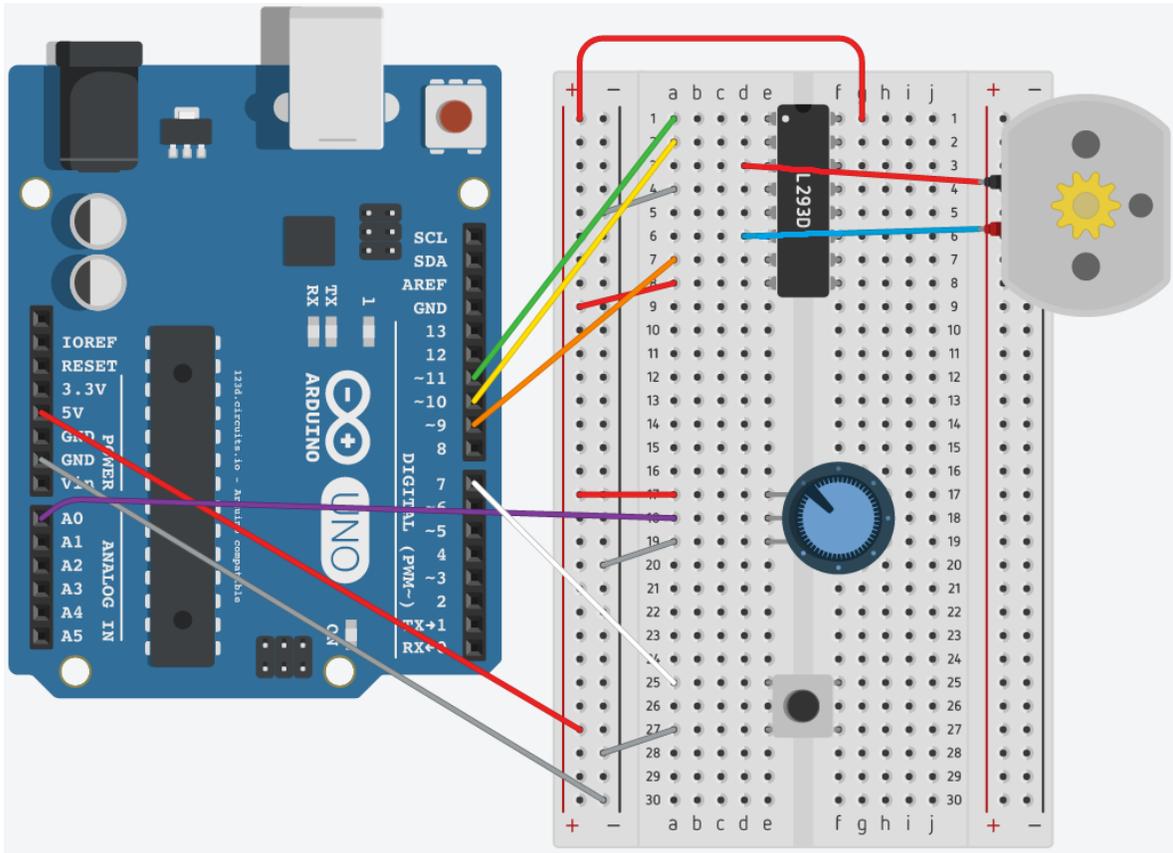
Build in stages:



Project 4 continued:



Project 4 continued:



```
/*  
DC Motors 04 : Controlling a Motor  
Source: Code adapted from Adafruit Arduino - Lesson 15. Bi-directional  
Motor(https://learn.adafruit.com/adafruit-arduino-lesson-15-dc-motor-reversing/overview)  
*/
```

```
int enablePin = 11; // enables left side of the h-bridge  
int forwardPin = 10; // pin 2 from h-bridge – tells the motor to go forward  
int reversePin = 9; // pin 7 from h-bridge – tells the motor to go backward  
int buttonPin = 7; // button  
int potPin = 0; // potentiometer
```

```
void setup()  
{  
  pinMode(forwardPin, OUTPUT); // the h bridge pins are set as outputs  
  pinMode(reversePin, OUTPUT);  
  pinMode(enablePin, OUTPUT);  
  pinMode(buttonPin, INPUT_PULLUP); // the button is an input (pullup enables the internal pull-up  
  resistor)  
}
```

```
void loop()  
{  
  int speed = analogRead(potPin) / 4; // divide the potentiometer reading by 4 to get a number between 0 & 255  
  boolean buttonPressed = ! digitalRead(buttonPin); // read the button. Since false = pressed, use not (!) to switch the value around  
  setMotor(speed, buttonPressed); // send our settings to the motor  
}
```

```
void setMotor(int speed, boolean buttonPressed)  
{  
  analogWrite(enablePin, speed); // turns motor on or off & sets speed on the enable pin  
  
  if(buttonPressed) {  
    digitalWrite(forwardPin, HIGH); // enable going forward  
    digitalWrite(reversePin, LOW); // disable going backward  
  }  
  else {  
    digitalWrite(forwardPin, LOW); // disable going forward  
    digitalWrite(reversePin, HIGH); // enable going backward  
  }  
}
```

Ideas to Build On

Set the motor to automatically accelerate and decelerate:

Use the code found on page 13 of this document and the same setup as Projects 01 and 02 above.

Control a motor using a serial monitor:

Try the serial speed function at the bottom of the code found on this page:

<https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-12-driving-a-motor>

Create a setup where a motor runs when you push a button:

<http://ninevolts.pbworks.com/w/page/10102026/dc%20motor%20pushbutton>

Add another motor to the other side of the H-bridge:

<http://hardwarefun.com/tutorials/creating-robots-using-arduino-h-bridge>

Learn More

Want to learn more about how DC motors work? Try these resources:

Adafruit Arduino Lesson 13: DC Motors. Simon Monk.

<https://learn.adafruit.com/adafruit-arduino-lesson-13-dc-motors>

Adafruit Arduino Lesson 15: DC Motor Reversing. Simon Monk.

<https://learn.adafruit.com/adafruit-arduino-lesson-15-dc-motor-reversing>

Exploring Arduino: Tools and Techniques for Engineering Wizardry. Jeremy Blum. 2013.

H-Bridges: The Basics.

<http://www.modularcircuits.com/blog/articles/h-bridge-secrets/h-bridges-the-basics/>

Sparkfun SIK Experiment Guide for Arduino V3.3 – Experiment 12: Driving a Motor.

<https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v33/experiment-12-driving-a-motor>

Sparkfun Tutorials: Motors and Selecting the Right One.

<https://learn.sparkfun.com/tutorials/motors-and-selecting-the-right-one>

Sparkfun Tutorials: Diodes.

<https://learn.sparkfun.com/tutorials/diodes>

Sparkfun Tutorials: Transistors.

<https://learn.sparkfun.com/tutorials/transistors>

```
/*  
DC Motors : Automatic Acceleration & Deceleration  
Source: Code adapted from SparkFun Inventor's Kit Example Sketch 12  
(https://learn.sparkfun.com/tutorials/sik-experiment-guide-for-arduino---v32/experiment-12-driving-a-motor) and Jeremy Blum's Exploring Arduino (http://www.exploringarduino.com/content/ch4/)  
*/
```

```
int motorPin = 9; // motor is connected to pin 9
```

```
void setup() {  
  pinMode(motorPin, OUTPUT); // motor pin is an output  
}
```

```
void loop() {  
  int speed;  
  int delayTime = 20; // milliseconds between each speed step
```

```
  // accelerate the motor
```

```
  for(speed = 0; speed <= 255; speed++) // counts from 0 to 255 (max speed) using the variable "speed"  
  {  
    analogWrite(motorPin,speed); // set the new speed  
    delay(delayTime); // delay between speed steps  
  }
```

```
  // decelerate the motor
```

```
  for(speed = 255; speed >= 0; speed--) // counts down from 0 to 255 (max speed) using the variable "speed"  
  {  
    analogWrite(motorPin,speed); // set the new speed  
    delay(delayTime); // delay between speed steps  
  }  
}
```