Build an Arduino Self-Driving Car

# Lesson 3: Controlling Motor Speed

**Estimated time:** 60 minutes

## Introduction

In this lesson students will learn to control a motor's speed using the **`analogWrite`** command. They will learn what Pulse Width Modulation (PWM) is and what the **`analogWrite`** command does. They will use this knowledge to adjust the speed of their car's wheels and help it drive straight.

## Learning Objectives

- Learn to use the **`analogWrite`** command
- Understand the difference between pulse width modulation (PWM) and a true analog output
- Use the **`analogWrite`** command to control a car's steering

## Preparation

- Try this activity yourself before you try it with students. You can use the example ramp up/ramp down code as a demonstration.
- Students will need their working H-bridge circuits with two motors from lesson 2. These circuits may already be mounted on a car chassis. You can stand the chassis on its rear end or remove the tires from the motors to prevent it from driving around while testing the motors.

## Materials

This lesson requires all the materials from lesson 2. No additional parts are needed.

# Procedure

Walk your students through the following steps or have them watch the video and follow along.

| Step | Timestamp | Description |
|---|---|---|
| 1 | 0:00 | Demonstrate the working ramp up/ramp down circuit for students (see Appendix 3A). |
| 2 | 0:30 | Explain the concept of pulse width modulation (PWM). |
| 3 | 3:51 | Modify the circuit from lesson 2 to connect one of the H-bridge's enable pins to one of the Arduino PWM pins instead of directly to 5V (same circuit diagram as Appendix 3A). |
| 4 | 4:25 | Modify the code from lesson 2 to use the **`analogWrite`** command to control one motor's speed in the **`driveForward`** function (see Appendix 3B). |
| 5 | 7:23 | Upload the code and try different values between 0-255 with the **`analogWrite`** command. Observe how fast the motor spins for the different values. |
| 6 | 9:20 | Edit the circuit and code to control the speed of both motors independently with the **`analogWrite`** command (see Appendix 3C, but let students try this on their own first). If needed, use different values for the two motors to make the car drive straight. |
| 7 | 12:11 | Optional: modify the other functions in the program to use the **`analogWrite`** command, and/or to include additional inputs to the functions for motor speeds (see Appendix 3D). |

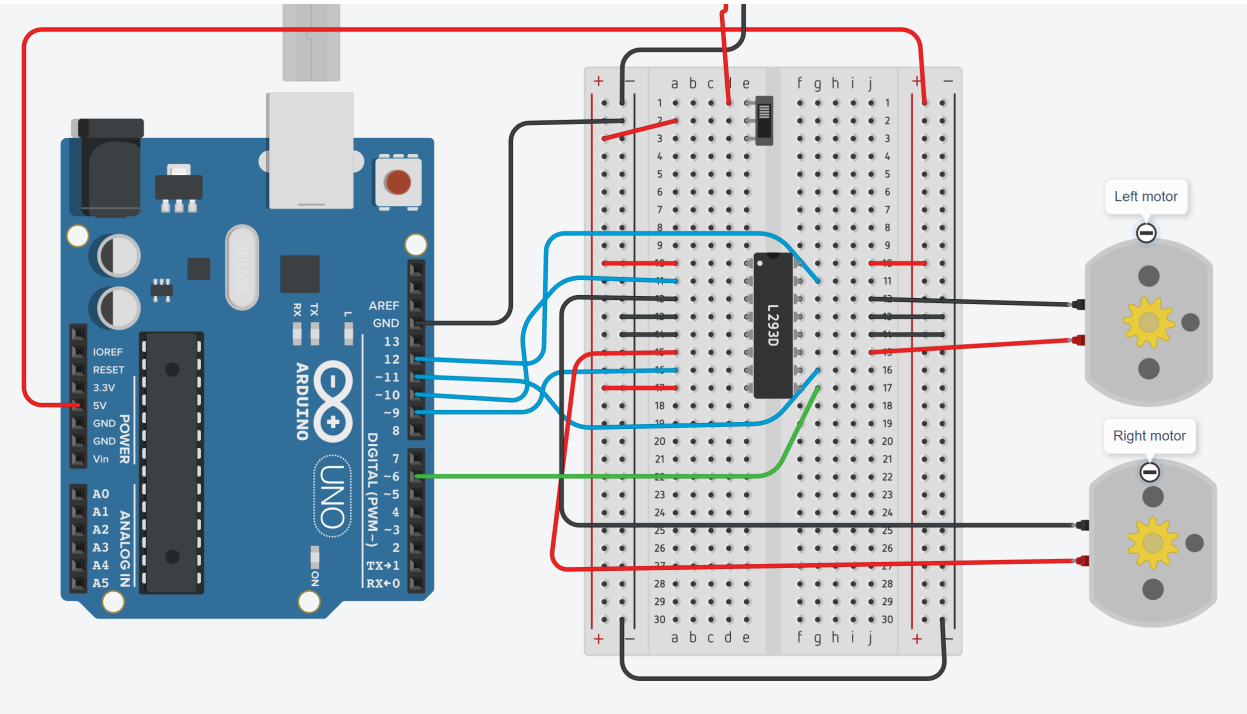
# Assessment

As you walk around the class:

- Confirm that students can get the **`analogWrite`** command working to control the speed of a single motor.
- Observe whether students can make their cars drive straight by adjusting the speeds of their left and right motors as needed.
- Optional: challenge students to program additional behaviors, such as driving forward at different speeds or going around a gradual curve.

# Appendix 3A: Motor Speed Ramp Up/Ramp Down

To control a motor's speed, you need to connect the enable pin on the H-bridge (pin 9 in the bottom right) to one of the Arduino's PWM pins (pin 6 in the diagram) instead of directly to 5V.



The following code will continually ramp up and ramp down the speed of the motor.

```
// declare motor pins
int forwardPin = 12;
int backwardPin = 11;
int speedPin = 6;     // must be a PWM pin (~)

int speed;   // speed variable will be 0-255
int delayTime = 50;  // delay time for ramp up/down in ms

void setup() {  // code that only runs once
  // set pins as outputs
  pinMode(forwardPin,OUTPUT);
  pinMode(backwardPin,OUTPUT);
  pinMode(speedPin,OUTPUT);

  // set motor direction
  // we won't change it in the loop in this program
  digitalWrite(forwardPin,HIGH);
  digitalWrite(backwardPin,LOW);
}

void loop() {  // code that repeats forever
  // ramp the motor speed up from 0-255
  for(speed = 0 ; speed<=255; speed = speed+1){
    analogWrite(speedPin,speed);
    delay(delayTime);
  }
  // ramp the motor speed down from 255-0
  for(speed = 255 ; speed>=255; speed = speed-1){
    analogWrite(speedPin,speed);
    delay(delayTime);
  }
}
```

# Appendix 3B: Single Motor Speed Control

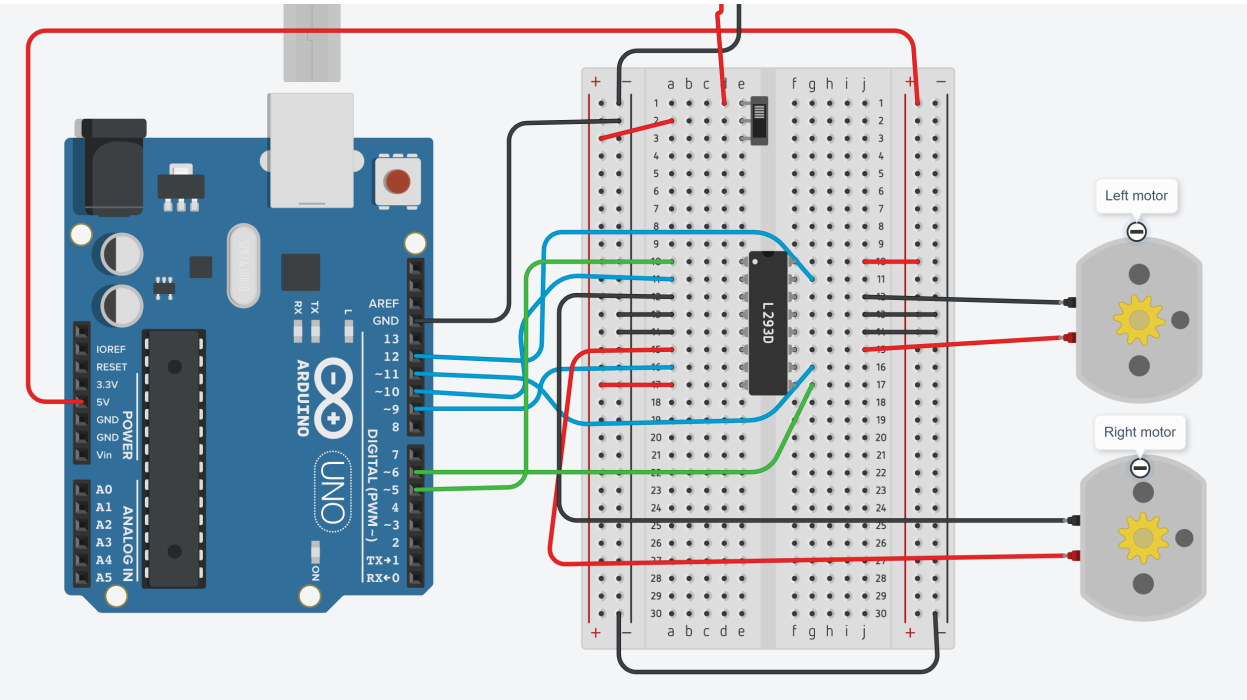
The following code uses the **`analogWrite`** command to control the speed of the left motor in the **`driveForward`** function.

```
// declare variables for motor control pins
int leftMotorForwardPin = 11;
int leftMotorBackwardPin = 12;
int rightMotorForwardPin = 9;
int rightMotorBackwardPin = 10;
int leftMotorSpeedPin = 6;

void setup(){ // code that only runs once
  // set motor control pins as outputs
  pinMode(leftMotorForwardPin,OUTPUT);
  pinMode(leftMotorBackwardPin,OUTPUT);
  pinMode(rightMotorForwardPin,OUTPUT);
  pinMode(rightMotorBackwardPin,OUTPUT);
  pinMode(leftMotorSpeedPin,OUTPUT);
  pinMode(rightMotorSpeedPin,OUTPUT);

  // drive forward forever
  driveForward(2000);
}

void loop(){  // code that loops forever
}

void driveForward(int delay_time){
  // set pins to make car drive forward then wait for delay_time ms
  analogWrite(leftMotorSpeedPin,255);
  digitalWrite(leftMotorForwardPin,HIGH);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,HIGH);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}
```

```
void driveBackward(int delay_time){
  // set pins to make car drive backward
  // then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,HIGH);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,HIGH);
  delay(delay_time);
}

void turnRight(int delay_time){
  // set pins to make car turn right
  // then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,HIGH);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,HIGH);
  delay(delay_time);
}

void turnLeft(int delay_time){
  // set pins to make car turn left then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,HIGH);
  digitalWrite(rightMotorForwardPin,HIGH);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}

void stopDriving(int delay_time){
  // set pins to make car stop then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}
```

# Appendix 3C: Two Motor Speed Control

The following circuit and code show how to control the speed of both motors independently using **`analogWrite`** commands in the **`driveForward`** function. Remember to connect the other enable pin on the H-bridge (pin 1 in the top left) to an Arduino PWM pin (pin 5 in the diagram).



```
// declare variables for motor control pins
int leftMotorForwardPin = 11;
int leftMotorBackwardPin = 12;
int rightMotorForwardPin = 9;
int rightMotorBackwardPin = 10;
int leftMotorSpeedPin = 6;
int rightMotorSpeedPin = 5;

void setup(){ // code that only runs once
  // set motor control pins as outputs
  pinMode(leftMotorForwardPin,OUTPUT);
  pinMode(leftMotorBackwardPin,OUTPUT);
  pinMode(rightMotorForwardPin,OUTPUT);
  pinMode(rightMotorBackwardPin,OUTPUT);
  pinMode(leftMotorSpeedPin,OUTPUT);
  pinMode(rightMotorSpeedPin,OUTPUT);

  // drive forward forever
  driveForward(2000);
}

void loop(){  // code that loops forever
}

void driveForward(int delay_time){
  // set pins to make car drive forward then wait for delay_time ms
  analogWrite(leftMotorSpeedPin,200);
  analogWrite(rightMotorSpeedPin,255);
  digitalWrite(leftMotorForwardPin,HIGH);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,HIGH);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}
```

```
void driveBackward(int delay_time){
  // set pins to make car drive backward
  // then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,HIGH);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,HIGH);
  delay(delay_time);
}

void turnRight(int delay_time){
  // set pins to make car turn right then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,HIGH);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,HIGH);
  delay(delay_time);
}

void turnLeft(int delay_time){
  // set pins to make car turn left then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,HIGH);
  digitalWrite(rightMotorForwardPin,HIGH);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}

void stopDriving(int delay_time){
  // set pins to make car stop then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}
```

# Appendix 3D: Function Inputs for Speed

The following code uses two additional inputs to the **`driveForward`** function for the speed of each motor. This lets you call the function multiple times with different speeds.

```
// declare variables for motor control pins
int leftMotorForwardPin = 11;
int leftMotorBackwardPin = 12;
int rightMotorForwardPin = 9;
int rightMotorBackwardPin = 10;
int leftMotorSpeedPin = 6;
int rightMotorSpeedPin = 5;

void setup(){ // code that only runs once
  // set motor control pins as outputs
  pinMode(leftMotorForwardPin,OUTPUT);
  pinMode(leftMotorBackwardPin,OUTPUT);
  pinMode(rightMotorForwardPin,OUTPUT);
  pinMode(rightMotorBackwardPin,OUTPUT);
  pinMode(leftMotorSpeedPin,OUTPUT);
  pinMode(rightMotorSpeedPin,OUTPUT);
}

void loop(){  // code that loops forever
  // drive forward forever
  driveForward(2000,255,255);
  driveForward(1000,128,128);
}

void driveForward(int delay_time, int leftSpeed, int rightSpeed){
  // set pins to make car drive forward then wait for delay_time ms
  analogWrite(leftMotorSpeedPin,leftSpeed);
  analogWrite(rightMotorSpeedPin,rightSpeed);
  digitalWrite(leftMotorForwardPin,HIGH);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,HIGH);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}
```

```
void driveBackward(int delay_time){
  // set pins to make car drive backward
  // then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,HIGH);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,HIGH);
  delay(delay_time);
}

void turnRight(int delay_time){
  // set pins to make car turn right then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,HIGH);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,HIGH);
  delay(delay_time);
}

void turnLeft(int delay_time){
  // set pins to make car turn left then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,HIGH);
  digitalWrite(rightMotorForwardPin,HIGH);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}

void stopDriving(int delay_time){
  // set pins to make car stop then wait for delay_time ms
  digitalWrite(leftMotorForwardPin,LOW);
  digitalWrite(leftMotorBackwardPin,LOW);
  digitalWrite(rightMotorForwardPin,LOW);
  digitalWrite(rightMotorBackwardPin,LOW);
  delay(delay_time);
}
```