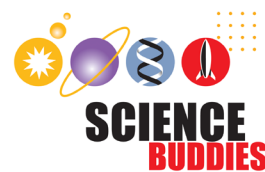


Build an Arduino Self-Driving Car

Lesson 2: Steering with Two Motors



Estimated time: 60 minutes

Introduction

In this lesson students will learn to independently control two motors using an H-bridge. When attached to a chassis, the motors can be used to make a car steer forward, backward, left, and right. Students will write functions to make their car steer and follow a pre-programmed path.

Learning Objectives

- Control two motors with an H-bridge
- Write functions to control a car's steering
- Write a program to make the car follow a pre-programmed path

Preparation

- Try this activity yourself before trying it with students. You can keep an assembled car to demonstrate it for students.
- Each group can start with their circuit from lesson 1.
- Each group will need an assembled BlueBot chassis. Once you have assembled the chassis, you can re-use them and do not need to take them apart again. If this is your first time doing the lesson, follow the instructions in the "How to Assemble Your BlueBot Chassis" video. **Note:** estimated time to complete this lesson does not include chassis assembly.
- You may wish to print out some of the appendices as a reference for the students.

Materials

In addition to their assembled circuit from lesson 1, each student or group of students will need:

- A second motor
- 9V battery
- 9V battery snap connector
- Slide switch
- Assembled BlueBot chassis
- Double-sided foam tape

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 circuit for students. Note: this could be either the motors connected to the breadboard or the entire circuit assembled and mounted on the chassis so the car drives around.
2	0:52	Starting with the assembled circuit from lesson 1, connect a second motor to the H-bridge. See if students can figure out how to do this on their own before you show them the breadboard diagram (Appendix 2A).
3	4:51	Review how you can use two motors to make the car steer forward, backward, left, and right.
4	5:46	Declare variables for all four motor control pins and set them as outputs (let students try this on their own).
5	7:27	Write a function to make both motors spin "forward" (let students try this on their own). Test it to see which way the motors spin.
6	10:00	Write additional functions to make the car drive backward, turn left, turn right, and stop (let students try this on their own first, but see Appendix 2B for complete example code). Try calling the functions for different amounts of time and observe the motor behavior.

7	11:30	Add a power switch to the circuit. This will allow you to turn the motors off without disconnecting the battery (see Appendix 2C). Turn the power switch off for now (the down position, assuming you followed the diagram in Appendix 2C) so the motors do not spin.
8	12:42	Mount the breadboard, Arduino, and 4xAA battery holder to the chassis using foam tape. Plug the 9V battery into the Arduino using the snap connector. Once you have uploaded code to the Arduino, you can disconnect the USB cable, and the Arduino will be powered by the 9V battery.
9	13:40	Test your driveForward function. Upload code that only calls the driveForward function and not any of your other functions. Turn the power switch on, put the car down, and watch which way it moves. If the car spins in place, then one wheel is spinning backwards. If the car drives backwards, then both wheels are spinning backwards. You can reverse a wheel's direction by switching the motor's red and black wires or by switching the respective forward and backward pins for the motor in your code. If needed, reverse the direction for one or both motors, then test your driveForward function again. Note: even when both motors are spinning forward, the car might not drive perfectly straight. This will be addressed in the next lesson.
10	15:32	Test your remaining functions to make sure the car moves as expected.
11	15:53	Try to program your car to follow a specific path. For example, make it drive in a square by repeatedly driving forward and then turning.

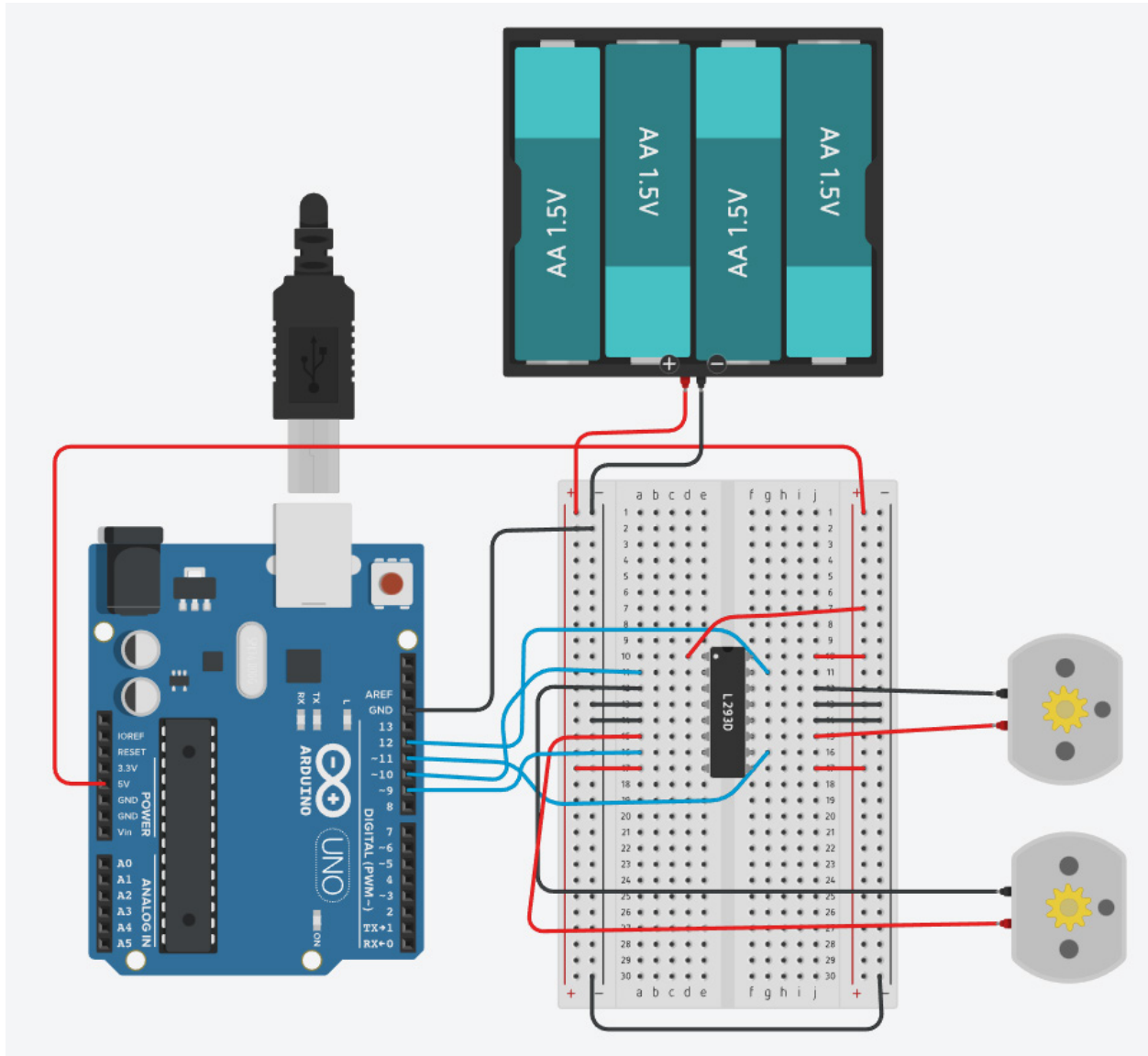
Assessment

As you walk around the class:

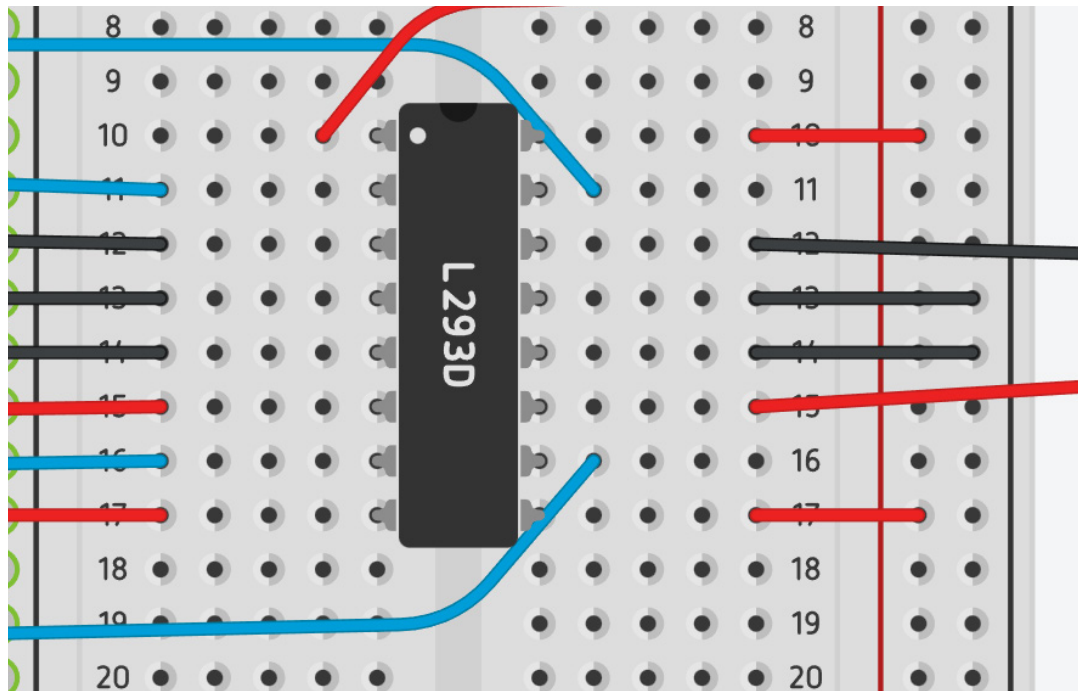
- Confirm that students are able to connect and control the second motor.
- Observe whether students can properly use functions to control both motors.
- Make sure students can mount all of their circuit components on the chassis.
- Confirm that students can edit their circuit or code if needed to make their motors spin in the correct directions for driving forward/backward, etc.
- Confirm that students can at least attempt to make their car follow a basic pre-programmed path. This may be difficult if the cars are not driving straight, but successful students may wish to try more complex paths.

Appendix 2A: Breadboard Diagram and Code for Single-Direction Spinning

The following diagram shows the entire circuit including the Arduino, battery pack, and motors.



This diagram shows a close-up of the H-bridge on the breadboard:



The H-bridge pins are connected as follows. Pins are numbered *counterclockwise starting from the top left*.

H-bridge pin	Connection
1	+5V bus (from Arduino)
2	Arduino pin 10
3	Right motor wire
4	Ground bus
5	Ground bus
6	Other right motor wire
7	Arduino pin 9
8	+6V bus (from 4xAA battery pack)
9	+5V bus (from Arduino)
10	Arduino pin 11
11	Left motor wire
12	Ground bus
13	Ground bus
14	Other left motor wire
15	Arduino pin 12
16	+5V bus (from Arduino)

Appendix 2B: Code for Steering with Two Motors

The following code contains functions to make the car drive forward, drive backward, turn left, turn right, and stop.

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

void setup() {
  // put your setup code here, to run once:
  // set motor control pins as outputs
  pinMode(leftMotorForwardPin,OUTPUT);
  pinMode(leftMotorBackwardPin,OUTPUT);
  pinMode(rightMotorForwardPin,OUTPUT);
  pinMode(rightMotorBackwardPin,OUTPUT);
}

void loop() {
  // put your main code here, to run repeatedly:
  driveForward(1000); // drive forward for 1 second
  stopDriving(1000);  // stop for 1 second
}

void driveForward(int delay_time){
  // set pins to make car drive forward then wait for delay_time ms
  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 2C: Power Switch Breadboard Diagram

This diagram shows how to add a power switch to the circuit. This lets you turn power from the 4xAA battery pack on and off without disconnecting the wires. Place the pins of the switch in three different rows of the breadboard. Connect the battery pack's positive wire to one of the outer pins. Connect a jumper wire from the middle pin to the +6V power bus (do not connect it to the +5V bus that is connected to the Arduino's 5V pin).

