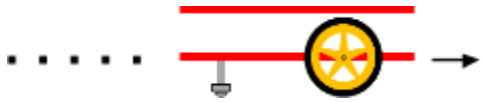
Name: _____ Date: _____

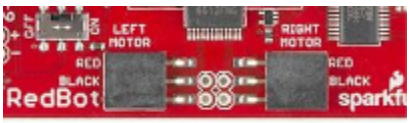
Get’n Going - Recycled STEM robots



Introduction

The RedBot is a great starting point for robotics. It is essentially an Arduino board with a motor controller. In this activity, you'll learn how to control the motion of the motors and build a frame and chassis for your very own robot out of recycled materials!

Setup Hardware



Connect the external battery pack to the RedBot Mainboard and then connect the motors. The motors have two wires (typically red and black, but not always). Connect these to the RED or BLACK pins for the LEFT MOTOR and the RIGHT MOTOR. Finally, connect the RedBot Mainboard to your computer using a standard USB cable.

Write Code

Open up a web browser and go to codebender.cc or use the regular Arduino IDE to write this code. Copy the following 14 lines of code into your window - *pay careful attention to punctuation, spelling, and capitalizations*.

```
1  // RedBot Test Code
2  #include <RedBot.h>
3
4  RedBotMotors motors;
5
6  void setup()
7  {
8    motors.drive(255);
9    delay(2000);
10   motors.brake();
11 }
12 void loop()
13 {
14 }
```

After you have the code typed out, select the **SparkFun RedBoard** as the board type and the appropriate port. Click the **Run on Arduino** button (CTRL+U) in either codebender or the Arduino IDE.

If you have everything correct, the motors should spin for 2 seconds and then stop. If you want to see it run again, simply hit the RESET button on the **RedBot Mainboard**.

Are the motors not running? Check the code for errors. Make sure you haven't missed a semi-colon, parentheses, or curly brace or check out the troubleshooting tips below:

Troubleshooting Tips

- Check code for syntax errors.
- The POWER must be set to ON.
- Make sure that the XBEE_HW_SERIAL switch is in the down position pointed toward XBEE SW SERIAL -- this is for advanced wireless control that we use later.
- Motors are, in fact, plugged into the correct ports.
- The MOTOR switch is set to RUN.

Dissecting the code

Now that you have seen your code work, let's take a look at what's going on. The first line of code reads:

```
[illegible]
```

This “includes” a reference library called RedBot.h that incorporates additional code into your program to provide access to new functions and commands. A reference guide to all of the objects and methods of the RedBot library can be found at: https://sfe.io/r110

After we've included the RedBot library, we can now use some of its *objects* and *methods* to control the RedBot. In the second line of code, we declare a [illegible] *object* called [illegible]. The [illegible] object allows us to easily control the motors' speed and direction and the general movement of the RedBot. This is done with the following line:

```
[illegible]
```

In Arduino, the [illegible] function runs just once, so any instructions or commands you put inside it will also only run once. There, you'll see two commands that we can use to control the speed of the robot:

```
[illegible]
[illegible]
```

The [illegible] command (also called a function) drives both the left and right motors in the correct direction so that the robot drives forward. The value in between the parentheses of a function is called an input parameter. For the [illegible] function, the input parameter is the [illegible], and it can be any integer value from -255 to 255.

The [illegible] command stops the motors abruptly. Notice that [illegible] does not have an input parameter. You still need to have the parentheses (), though. This is what indicates that it is a function.

Finally, you'll see a command in between the two motors commands:

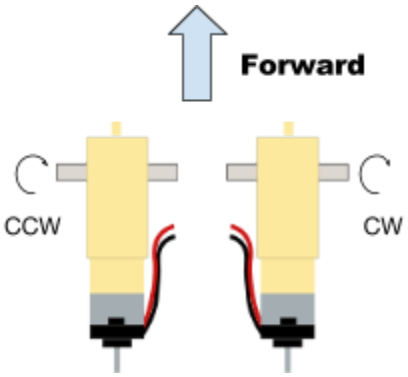
```
[illegible]
```

As implied, this command delays the program flow for 2000 time units. In this case, the time unit is a millisecond. The Arduino microcontroller runs at **16 MHz,** so if we didn't add a delay, it would drive the motors and then immediately stop them in just a few microseconds!

The [illegible] is a blocking function that allows you to control the program flow.

Play around with the code

Now that you got the motors to spin, let's check to make sure everything is spinning in the right direction. Orient the motors similar to the drawing. When you pair the motors together, the [redacted] command should spin the right motor clockwise (CW) and the left motor counter-clockwise (CCW).



When you put wheels on the motors, this allows the robot to move forward. If one of the motors is spinning in the opposite direction, simply reverse the red and black wires on that motor. We suggest labelling the motors (R) and (L) so that you remember which motor goes where as you start building your robot.

Now, play around with the code. Try changing the speed of the motors. What is the slowest speed setting that you can use before the motors *stall*? Do you notice anything peculiar? What happens when you use a negative number for the speed? Write down your observations in your notebook.

Engineering Notebook Observation Notes Sentence Starters

"The slowest speed setting for my motors is…"
"When I set this speed, we observed that…"
"We then tried changing… because we thought..."
"When we used a negative number for the speed setting..."

More Control!

Want more control? If you want to drive the right motor and the left motor separately, you can use the [redacted] or the [redacted] functions. These methods also take a single input parameter for [redacted] that can vary from -255 to +255.

[redacted]

Characterize how these numbers relate to the motion and direction of the motors.

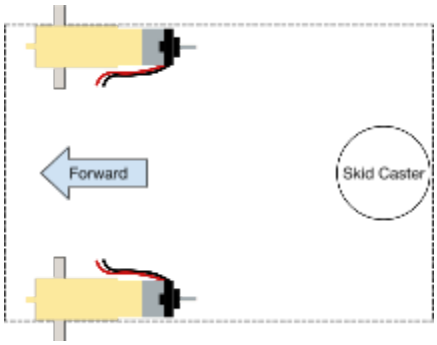
Additional Questions:

1. What does line 9 in the code example above, [redacted] do? What happens if you take it out of the code? Why does this happen?
2. The input parameter +255 in [redacted] causes the robot to move forward. Explain the direction that the right motor and the left motor spin to provide this result.
3. Write a short code snippet to make the robot:
   - move forward for 2 seconds
   - stop for 1 second
   - move backward for 1.5 seconds
   - stop for 1 second
4. What is the difference between code that you put in the setup() and code that you put in the loop() part in Arduino?

Dumpster Diving -- Build your 'Bot!

Now that you've gotten to spin the motors and you've seen how to control things in code, let's turn this into something that resembles a robot. There are a lot of robotics platforms that provide a pre-made chassis frame or building platform. Here, we encourage you to dig into your trash, clean out your recycling, and look at the potential for what these things can be. Cereal boxes? Plastic containers? Cardboard? These are all great frames for a robot!

To get you started, we provided a rough template for a chassis laid out onto a piece of cardboard. The robot works best as front-wheel drive design, and we suggest using a skid caster for the back, but this is *your* design.



Use hot-glue, tape, zip-ties or whatever you can find to secure the motors onto the chassis. Make sure you make room to mount the RedBot Mainboard and the battery pack, as well. When you're done with building your chassis, take a picture of it and/or make a sketch to include in your notebook!

Robot Dance Party

Let's add some personality! Play around with controlling the motion of your new creation using the code examples you've learned. Can you figure out how to make the robot turn or spin in place? In order to turn, we need to control the motors separately. Recall that there are two commands to control the power of each wheel independently: and

Modify the code so that the RedBot drives straight, turns 90 degrees and then stops. Now, extend this to trace out an entire square!

Find your favorite dance song and choreograph a short little routine for your robot. Now film your robot dancing using **VideoStar** (iPhone), **VideoFX** (Android), **Vine**, or whatever your favorite video making program is. Share it to us (@SparkfunEDU) on Youtube, Vimeo, or your own site.

Challenges \ Going Further

- Pick a medium speed for your robot like: Setup your robot to drive straight for 1 second and stop. Measure the distance it covers. Now, repeat this using 3 - 5 different time durations. What relationship do you notice? Write down any qualitative observations as you conduct this experiment. What is your independent variable, dependent variable, and controlled variables?
- Now, setup the robot to drive straight for 1 second and vary the speed. Write down any qualitative observations as you conduct this experiment.
- Now that you know how to get the robot to drive, let's investigate! How does the relate to "real" measurements of speed / velocity? takes a parameter from -255 to 255. Come up with a relationship (equation) between and the actual speed of the robot.
- Engineers and scientists describe the speed of a motor in terms of revolutions per second. What input parameter value do you use with to get a speed of 1 revolution per second?