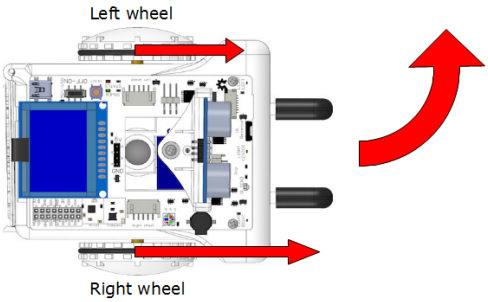
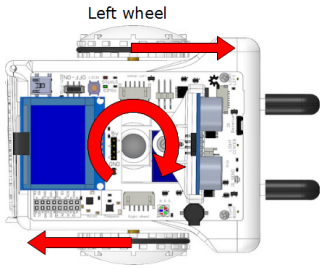
# Sparki With Arduino, Guide #2 - Motors

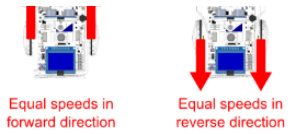
Sparki is a two wheeled robot, and its movement is based on the idea of **differential** drive. This means that Sparki's movement is controlled by sending different amounts of power to the left and the right wheels.

When the two wheels are powered by different amounts one of the wheels will cover more ground than the other. This causes Sparki to turn in one direction or the other. For example, if the right wheel is rotating faster than the left one, the robot will turn left while traveling forward

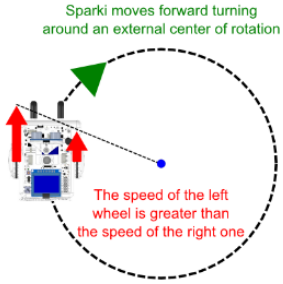


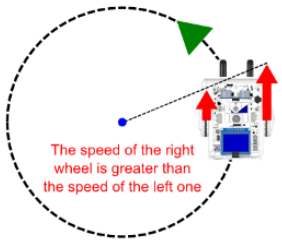



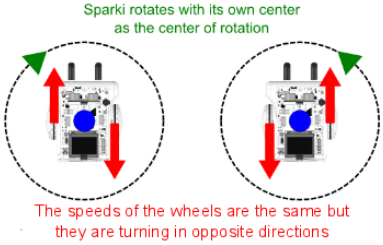

If the wheels move in opposite directions at the same speed, the robot will rotate around its center (which is also where Sparki can hold a marker to draw with, if you find some paper!)




To make Sparki travel straight forward you will have to apply the same amount of power to both wheels in the forward direction. Sparki can also moves backwards if you do the opposite and supply the same amount of power to both wheels in reverse.




There are three simple sets of instructions you can use to get Sparki moving: move forward and backward, move left or right, and stop. You'll need to place them inside the loop() function of the Sparkduino code. The code for these instructions looks like:

```
sparki.moveForward();
sparki.moveBackward();
sparki.moveLeft();
sparki.moveRight();
sparki.moveStop();
```

⚠ Before we get started, make sure that the On/Off Switch near the USB port is OFF.

⚠ Before you start the activities below make sure that you have plenty of room for the Sparki to move and be prepared to pick it up. a long fall off a table could break the Sparki.

💡 Start by choosing New from the File menu to load a new sketch with empty setup() and loop() functions.

Let's try moving Sparki forward or backward by adding some code into our new sketch.

```
#include <sparki.h> //include Sparki library
void setup(){
}

void loop(){
  sparki.moveForward();
}
```

Let's test it out to make sure it works,  go ahead and click the upload button again. It's best to change just a little bit of code at a time and check to make sure it works correctly before moving on and adding more code. To test this code first connect Sparki to your computer with the USB cable and select the board and port, and click Upload, just like in Guide #1.

Now that you've uploaded your code you'll notice nothing is happening.  That's because the motors are only powered from Sparki's batteries and not from USB, so turn the switch ON. Sparki never really stops moving, which is exactly what we told it to do. Let's try making Sparki first move forward and then backwards for the same amount.

```
#include <sparki.h> //include Sparki library
void setup(){
}

void loop()
{
  sparki.moveForward();
  sparki.moveBackward();
}
```

Upload and see what happens! Sparki isn't moving anymore? That's because Sparki's microprocessor is pretty fast. So fast that there isn't any time for the wheels to move between these two commands. So we have to add a little bit of time where Sparki's microprocessor isn't doing anything. To do this we'll use the `delay()` function. When executing a `delay()` whatever Sparki was doing before the delay keeps happening, ignoring everything else. `delay()` takes a value in milliseconds (ms), so 1000 is 1 second, but lets do 3 seconds.

```
#include <sparki.h> //include Sparki library
void setup(){
}

void loop()
{
  sparki.moveForward();
  delay(3000);
  sparki.moveBackward();
  delay(3000);
}
```

Upload your code again and see if you got Sparki moving the way you want it to move. Try changing the number inside of the parenthesis of the `delay()` function to change how long Sparki moves in either direction.

Now let's add a turn to Sparki's code. You can decide whether you want Sparki to turn left or right. In the example code Sparki will be turning right, but from here on out you can write whatever you like in your code. If you get any error messages, you can just delete the code you added or use the undo function to back up a few steps, and as always, Arch Reactor volunteers are here to help if you get stuck!

```
#include <sparki.h> //include Sparki library
void setup(){
}

void loop()
{
  sparki.moveForward();
  delay(3000);
  sparki.moveBackward();
  delay(3000);
  sparki.moveRight();
  delay(1000);
}
```

Now Sparki moves forward, then backwards and turns for one second. That's not very precise, is it? Let's make Sparki turn a certain number of degrees.

```
#include <sparki.h> //include Sparki library
void setup(){
}

void loop()
{
  sparki.moveForward();
  delay(3000);
  sparki.moveBackward();
  delay(3000);
  sparki.moveRight(90);
}
```

Notice there's no `delay()` after `moveRight(90)`, that's because when you give a move a value the code does not continue until that move is done.  Try changing the 90 and see how that affects the movement.

 From here on you won't be instructed when to upload code in any of the guides.  Anytime you see a block of example code if you feel you understand it fully you may skip ahead, or type it in and adjust it to see how it works with changes or additional code that you want to add.  Arch Reactor volunteers can help out at any point, no matter what code you put in, so feel free to experiment!

Constant moving is great, but sometimes you want to stop and wait for something. To do that we'll use the `moveStop()` function. Try changing the `delay()` function after the `moveStop()` function to make Sparki wait around for a longer or shorter amount of time.

```
#include <sparki.h> //include Sparki library
void setup(){
}

void loop()
{
 sparki.moveForward();
 delay(3000);
 sparki.moveBackward();
 delay(3000);
 sparki.moveRight(15);
 sparki.moveStop();
 delay(1000);
}
```

You may now do the optional advanced motors section on the next pages, experiment further with the code you just wrote or move on to the LCD guide.

If you’re looking for a challenge here are some patterns you can try to make Sparki recreate by writing your own code. Hint: `moveForward()` also takes a parameter, a distance in centimeters. Try `moveForward(20)`. See if you can make Sparki move in a square or a zig-zag!