

Programming the EV3 with Swift Playgrounds

Lesson 3 – Curved Move

by Ahmad Sahar

Hi, It's Shah again. For those who don't know, I'm a professional trainer specialising in MacOS and iOS software and hardware, and I also conduct classes on iOS App Development and LEGO® Mindstorms.

In the last lesson, we built a robot and made it move forward and back. In this lesson, we're going to make our robot turn.

What you need:

An iPad with Swift Playgrounds installed.

LEGO® Mindstorms EV3 Education (set no. 45544) or Home (set no. 31313).

Before you begin

For this lesson you need to build either the Educator robot from the 45544 set or the DrivingBas3 from the 31313 set.



45544



31313

The instructions for the Educator robot can be found here:

http://robotsquare.com/wp-content/uploads/2013/10/45544_educator.pdf

The instructions for DrivingBas3 can be found here:

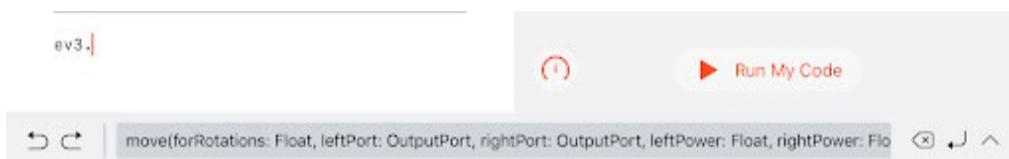
<http://ev3lessons.com/RobotDesigns/instructions/DrivingBas3.pdf>

Make sure you have fresh batteries in your robot, and that your iPad is fully charged and paired to your robot.

Curved Move

You can use the template you used in the last lesson, or you can download a fresh copy. Remove all the code from the template, and connect to the EV3 brick.

Tap ev3 from the suggestions list. It appears on the page. Tap the dot in the suggestions list. Scroll through all the suggestions in the suggestions list until you see this one:



move(forRotations: Float, leftPort: OutputPort, rightPort: OutputPort, leftPower: Float, rightPower: Float, brakeAtEnd: Bool)

Tap it to insert onto the page.

This is a Swift function which is similar to the EV3-G Move Tank block.

It has a number of parameters:

forRotations: Number of rotations to execute

leftPort: The port the left motor is connected to. Can be .a, .b, .c or .d

rightPort: The port the right motor is connected to. Can be .a, .b, .c or .d

leftPower: The power level specified for leftPort.

rightPower: The power level specified for rightPort.

brakeAtEnd: Brake or coast at the end of the move.

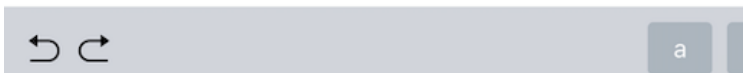
In general, to change the value for the parameters, tap a parameter, and you can either use the on-screen keyboard to enter the value, or choose from the suggestions list.

We'll start by changing the forRotations parameter. Tap the parameter, which is represented by the word Float. When the picker appears, choose 3. This sets the number of rotations to be executed to 3.

```
ev3.move(forRotations: 3, leftPort:
OutputPort, rightPort: OutputPort,
leftPower: Float, rightPower: Float,
brakeAtEnd: Bool)
```

Tap the leftPort parameter, OutputPort and type a dot.

```
ev3.move(forRotations: 3, leftPort: .|,
rightPort: OutputPort, leftPower: Float,
rightPower: Float, brakeAtEnd: Bool)
```



After you have done so, the motor ports a, b, c and d appear in the suggestions list. Choose b. This sets leftPort to Port B on the programmable brick.

```
ev3.move(forRotations: 3, leftPort: .b|,
rightPort: OutputPort, leftPower: Float,
rightPower: Float, brakeAtEnd: Bool)
```

Tap the rightPort parameter, OutputPort and type a dot. Choose c from the suggestions list. This sets rightPort to Port C on the brick.

To make the robot turn we vary the power sent to the motors. We will set leftPower = 50 and rightPower = -50, which makes the left side move forward and the right side move backward, making the robot turn clockwise on the spot.

Tap the leftPower parameter, Float. Enter 50. This sets the power level of the left motor to 50.

Tap the rightPower parameter, Float. Enter -50. This sets the power level of the right motor to -50.

Finally, tap the brakeAtEnd parameter, Bool. Tap true from the suggestions list.

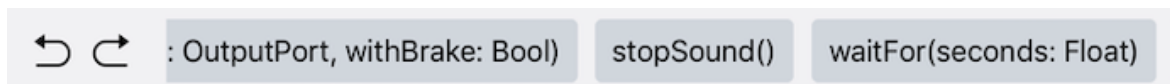
The command at this point should look like this:

```
ev3.move(forRotations: 3, leftPort: .b,
rightPort: .c, leftPower: 50, rightPower:
-50, brakeAtEnd: true)
```

```
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)
```

Next we're going to make the robot pause using a wait function.

Tap on the space below the code you've entered and choose `ev3` from the suggestions list. Tap the dot, and scroll through the list of suggestions until you find `waitFor(seconds: Float)`. Tap it.



This is a Swift function which is similar to the EV3-G Wait block. It has one parameter, `seconds`, which is the number of seconds to wait.

Tap the `seconds` parameter, `Float`. Type `1` to make the program pause for 1 second.

The function should look like this:

```
ev3.move(forRotations: 3, leftPort: .b,  
        rightPort: .c, leftPower: 50, rightPower:  
        -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)
```

```
ev3.waitFor(seconds: 1)
```

Select all the code you've typed in so far, and copy it. Tap below the existing code and paste. The result should be as follows:

```
ev3.move(forRotations: 3, leftPort: .b,  
        rightPort: .c, leftPower: 50, rightPower:  
        -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b,  
        rightPort: .c, leftPower: 50, rightPower:  
        -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)
```

```
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)
```

For the second `move` command, set `rightPower` to `0`. This makes the left side go forward, pivoting clockwise on the stationary wheel connected to the right motor.

```
ev3.move(forRotations: 3, leftPort: .b,  
        rightPort: .c, leftPower: 50, rightPower:  
        0, brakeAtEnd: true)
```

```
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: 0, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)
```

Tap below the code you've typed in so far and paste again. The result should be as follows:

```
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: 0, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)
```

```
ev3.move(forRotations: 3, leftPort: .b,  
        rightPort: .c, leftPower: 50, rightPower:  
        25, brakeAtEnd: true)
```

For the third move command, set rightPower to 25. This makes the left side go forward faster than the right side, making the robot turn gradually to the right.

Remove the last line of code, and you should get this:

```
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: -50, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: 0, brakeAtEnd: true)  
ev3.waitFor(seconds: 1)  
ev3.move(forRotations: 3, leftPort: .b, rightPort: .c, leftPower: 50, rightPower: 25, brakeAtEnd: true)
```

Run the program. The robot should turn clockwise on the spot, pause for one second, pivot clockwise, pause for one second and finally make a gradual right turn.

To make left turns, set rightPower to 50, and leftPower to -50, 0 and 25 for each move command.

Can you figure out how to make the robot turn when moving backwards?

Great job! We've come to the end of the lesson. In the next lesson, we'll build and program a robot arm.

If you wish to know more about me and what I do, feel free to visit my company website, <http://tomafuwi.tumblr.com>, like my Facebook page at <http://facebook.com/tomafuwi>, follow me on Twitter at https://twitter.com/shah_apple and subscribe to my YouTube channel at <https://www.youtube.com/c/CoderShah>.

All the best and take care.

#

