



## Challenge #6 Multiple Solutions

2 left turns using time Scroll down for JAVA

```
to runOpMode
  Put initialization blocks here
  set frontRight . Direction to Direction REVERSE
  set backRight . Direction to Direction REVERSE
  call Challenge #6 :2 left turns waitForStart
  If call Challenge #6 :2 left turns opModelsActive
  do
    Put run blocks here
    drive forward
    set Power
    frontLeft to 1
    frontRight to 1
    call Telemetry.addData
    key to "Driving"
    text to "Forward"
    call Telemetry.update
    call Challenge #6 :2 left turns sleep
    milliseconds to 3200
    turn to right
    set Power
    frontLeft to 0
    frontRight to 1
    call Challenge #6 :2 left turns sleep
    milliseconds to 550
    drive forward
    set Power
    frontLeft to 1
    frontRight to 1
    backLeft to 1
    backRight to 1
    call Challenge #6 :2 left turns sleep
    milliseconds to 1500
    set Power
    frontLeft to 0
    frontRight to 0
    backLeft to 0
    backRight to 0
    call Challenge #6 :2 left turns sleep
    milliseconds to 100
    set Power
    frontLeft to -1
    frontRight to 1
    backLeft to 1
    backRight to -1
    call Challenge #6 :2 left turns sleep
    milliseconds to 2500
    set Power
    frontLeft to 0
    frontRight to 0
    backLeft to 0
    backRight to 0
```

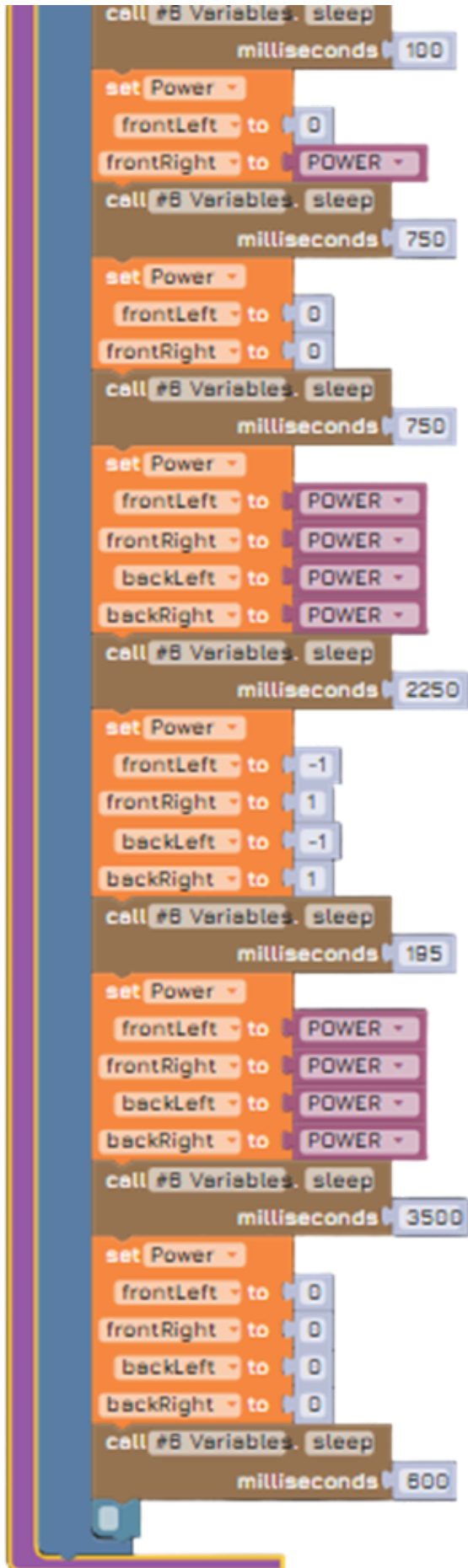


## Using Variables/Encoders

```

to runOpMode
  Put initialization blocks here
  set frontRight . Direction to Direction REVERSE
  set backRight . Direction to Direction REVERSE
  set Mode
  frontLeft to RunMode STOP_AND_RESET_ENCODER
  frontRight to RunMode STOP_AND_RESET_ENCODER
  backLeft to RunMode STOP_AND_RESET_ENCODER
  backRight to RunMode STOP_AND_RESET_ENCODER
  set Mode
  frontLeft to RunMode RUN_TO_POSITION
  frontRight to RunMode RUN_TO_POSITION
  backLeft to RunMode RUN_TO_POSITION
  backRight to RunMode RUN_TO_POSITION
  set POWER to 0.8
  set TargetPosition
  frontLeft to 3750
  frontRight to 3750
  backLeft to 3750
  backRight to 3750
  call #B Variables. waitForStart
  If call #B Variables. opModeIsActive
  do
    repeat while frontLeft . CurrentPosition < frontLeft . TargetPosition
    do
      set Power
      frontLeft to POWER
      frontRight to POWER
      backLeft to POWER
      backRight to POWER
      call Telemetry. addData
      key "S"
      number frontLeft . CurrentPosition
      call Telemetry. update
      robot will stop at encoder value 3700
      call #B Variables. sleep
      milliseconds 100
      set Mode
      frontLeft to R
      frontRight to RunMode STOP_AND_RESET_ENCODER
      backLeft to RunMode STOP_AND_RESET_ENCODER
      backRight to RunMode STOP_AND_RESET_ENCODER
      set Mode
      frontLeft to RunMode RUN_USING_ENCODER
      frontRight to RunMode RUN_USING_ENCODER
      backLeft to RunMode RUN_USING_ENCODER
      backRight to RunMode RUN_USING_ENCODER
      call #B Variables. sleep
      milliseconds 100
  
```

Sleep for the given amount of milliseconds.





```
1. package org.firstinspires.ftc.teamcode;
2.
3. import com.qualcomm.robotcore.eventloop.opmode.LinearOpMode;
4. import com.qualcomm.robotcore.eventloop.opmode.TeleOp;
5. import com.qualcomm.robotcore.hardware.DcMotor;
6. import com.qualcomm.robotcore.hardware.DcMotorSimple;
7.
8. @TeleOp(name="Lesson4", group="Tutorial")
9. public class Lesson1 extends LinearOpMode {
10.     // create DcMotor objects
11.     private DcMotor frontRight;
12.     private DcMotor frontLeft;
13.     private DcMotor backRight;
14.     private DcMotor backLeft;
15.
16.     @Override
17.     public void runOpMode() {
18.         // initialize hardware variables
19.         frontRight = hardwareMap.get(DcMotor.class, "frontRight");
20.         frontLeft = hardwareMap.get(DcMotor.class, "frontLeft");
21.         backRight = hardwareMap.get(DcMotor.class, "backRight");
22.         backLeft = hardwareMap.get(DcMotor.class, "backLeft");
23.
24.         // Set motor direction
25.         frontRight.setDirection(DcMotorSimple.Direction.REVERSE);
26.         backRight.setDirection(DcMotorSimple.Direction.REVERSE);
27.
28.
29.         waitForStart();
30.
31.         // when started
32.         if (opModeIsActive()) {
33.
34.             // drive forward
35.             frontRight.setPower(1);
36.             frontLeft.setPower(1);
37.             backRight.setPower(1);
38.             backLeft.setPower(1);
39.             telemetry.addData("Driving", "Forward");
40.             telemetry.update();
41.             sleep(1100);
42.
43.             // stop
44.             frontRight.setPower(0);
45.             frontLeft.setPower(0);
46.             backRight.setPower(0);
```



```
47.     backLeft.setPower(0);
48.     sleep(100);
49.
50.     // turn left
51.     frontRight.setPower(0.5);
52.     frontLeft.setPower(-0.5);
53.     backRight.setPower(0.5);
54.     backLeft.setPower(-0.5);
55.     sleep(250);
56.
57.     // drive forward
58.     frontRight.setPower(1);
59.     frontLeft.setPower(1);
60.     backRight.setPower(1);
61.     backLeft.setPower(1);
62.     sleep(1500);
63.
64.     // stop
65.     frontRight.setPower(0);
66.     frontLeft.setPower(0);
67.     backRight.setPower(0);
68.     backLeft.setPower(0);
69.     sleep(100);
70.
71.     // turn left
72.     frontRight.setPower(0.5);
73.     frontLeft.setPower(-0.5);
74.     backRight.setPower(0.5);
75.     backLeft.setPower(-0.5);
76.     sleep(250);
77.
78.     // drive forward
79.     frontRight.setPower(1);
80.     frontLeft.setPower(1);
81.     backRight.setPower(1);
82.     backLeft.setPower(1);
83.     sleep(2000);
84.
85.     // stop
86.     frontRight.setPower(0);
87.     frontLeft.setPower(0);
88.     backRight.setPower(0);
89.     backLeft.setPower(0);
90.     sleep(100);
91.
92.
```



93.  
94.     }  
95.    }  
96.}