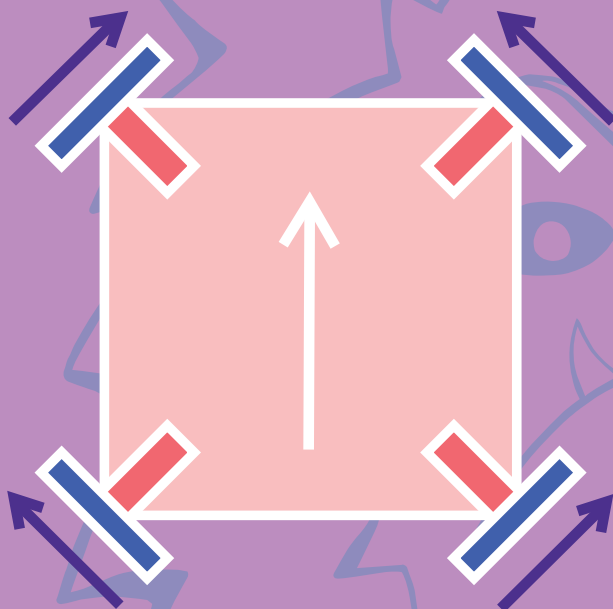


# A Guide to Omnidirectional Chassis

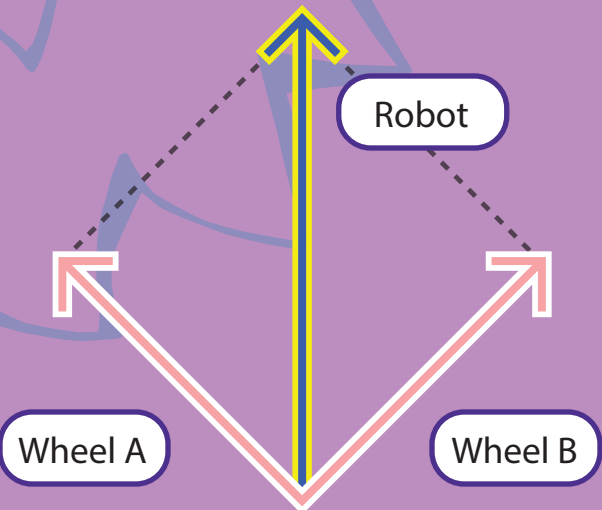
Omniwheels are wheels with rollers which allow for motion in the direction perpendicular to that of the wheel's rotation. This allows for the construction of robots which can move in any direction, rather than just forwards and backwards. Robots constructed in this way have both advantages, such as easier maneuverability and turning, and disadvantages, such as decreased forward speed and pushing power.



In order to create a holonomic chassis, mount the motors on the corners of the robot at 45 degree angles. You can mount one or two wheels to a single motor—two omniwheels are mounted together in the image above. Omniwheels have spaces between the individual rollers, so only having one wheel makes it uneven. When constructing the robot, an important consideration is balancing the weight evenly; an uneven weight distribution will make the robot harder to move and steer predictably.



An omnidirectional robot's movement can be understood mathematically using simple vector addition; each wheel's speed direction can be modeled as a vector. In a working omnidirectional robot, these vectors add up to a single vector which determines the velocity of the robot.



# Coding an Omnidirectional Robot

To code a very simple omnidirectional robot (if you don't yet know how to code a basic OpMode, go to the link below), start by creating instance variables for each of your motors in the OpMode. Also, create constants containing the clockwise and counterclockwise motor powers.

```
private static final double CLOCKWISE = 1;
private static final double COUNTER_CLOCKWISE = -1;

private DcMotor frontRight = null;
private DcMotor frontLeft = null;
private DcMotor backRight = null;
private DcMotor backLeft = null;
```

Then, in your init code, get each of your motors from the hardware map. Make sure the names you use (e.g. "frontRight") are the same as those set in the config.

```
frontRight = hardwareMap.get(DcMotor.class, "frontRight");
frontLeft = hardwareMap.get(DcMotor.class, "frontLeft");
backRight = hardwareMap.get(DcMotor.class, "backRight");
backLeft = hardwareMap.get(DcMotor.class, "backLeft");
```

In your loop method, create boolean variables and set them to the values from the D-pad.

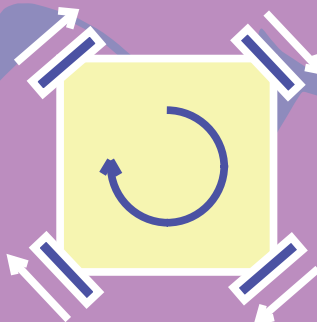
```
boolean up = gamepad1.dpad_up;
boolean down = gamepad1.dpad_down;
boolean left = gamepad1.dpad_left;
boolean right = gamepad1.dpad_right;
```

FTC has a tutorial on how to program a basic opmode at [https://github.com/ftctechnh/ftc\\_app/wiki/Creating-and-Running-an-Op-Mode-\(Android-Studio\)](https://github.com/ftctechnh/ftc_app/wiki/Creating-and-Running-an-Op-Mode-(Android-Studio)).

**Finally, add code to set the the power of the motors to the loop method. This code allows the robot to move in all four directions, depending upon which D-pad button is pressed.**

```
if (up && !down && !left && !right) {  
    frontLeft.setPower(COUNTER_CLOCKWISE);  
    backLeft.setPower(COUNTER_CLOCKWISE);  
    frontRight.setPower(CLOCKWISE);  
    backRight.setPower(CLOCKWISE);  
} else if (!up && down && !left && !right) {  
    frontLeft.setPower(CLOCKWISE);  
    backLeft.setPower(CLOCKWISE);  
    frontRight.setPower(COUNTER_CLOCKWISE);  
    backRight.setPower(COUNTER_CLOCKWISE);  
} else if (!up && !down && left && !right) {  
    frontLeft.setPower(CLOCKWISE);  
    frontRight.setPower(CLOCKWISE);  
    backLeft.setPower(COUNTER_CLOCKWISE);  
    backRight.setPower(COUNTER_CLOCKWISE);  
} else if (!up && !down && !left && right) {  
    frontLeft.setPower(COUNTER_CLOCKWISE);  
    frontRight.setPower(COUNTER_CLOCKWISE);  
    backLeft.setPower(CLOCKWISE);  
    backRight.setPower(CLOCKWISE);  
} else {  
    frontLeft.setPower(0);  
    frontRight.setPower(0);  
    backLeft.setPower(0);  
    backRight.setPower(0);  
}
```

**This code is enough to make a robot which can move in four directions. However, there are many directions to go from here; while this robot can move, it can only do so with fixed velocities, and it cannot turn. As such, the your next step might be to add turning capability. Here's a hint: you can make an omnidirectional robot rotate by moving all four motors in the same direction, as shown in the diagram below.**



The omni wheels we use are TETRIX MAX Omni Wheels. Andymark sells wheels that have metal bodies and rubber rollers, however in our experience the Tetrix omni's nylon rollers hold up better than Andymark's rubber rollers. Also, TETRIX's omni wheels are thinner than their Andymark equivalent.

