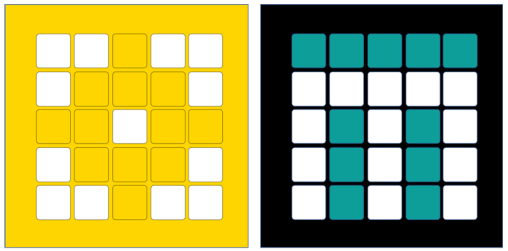
PRIME LESSONS

By the Makers of EV3Lessons



CONFIGURING ROBOT MOVEMENT

BY SANJAY AND ARVIND SESHAN

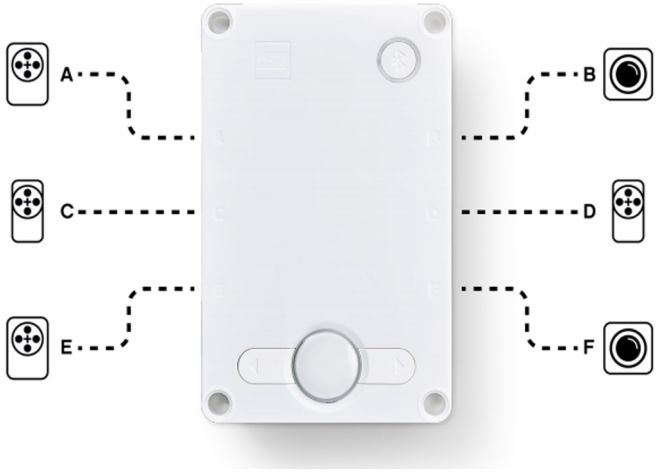
This lesson uses SPIKE 3 software

LESSON OBJECTIVES

Learn to configure robot movement on a SPIKE Prime robot

Learn how to add you first lines to the programming canvas

Note: Although images in this lessons may show a SPIKE Prime, the general concepts are the same for Robot Inventor.



WHY CONFIGURE YOUR CODE?

Every robot is different

Before you can program to move or turn, you need to first set how you have configured your robot:

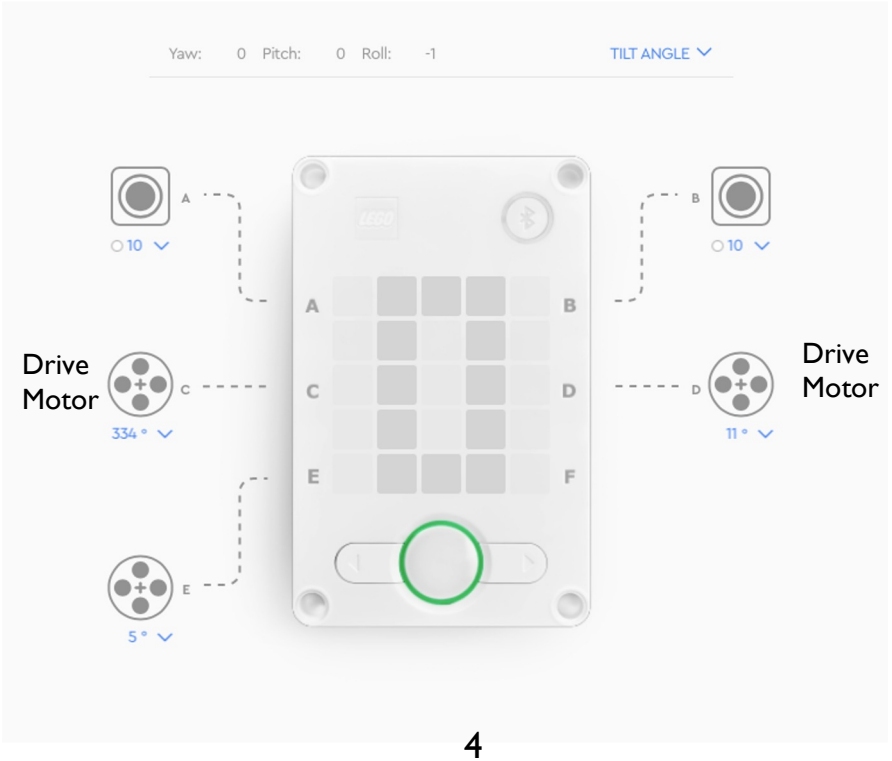
- What ports are the drive motors connected to?
- What type of wheels are you using?
- How fast do you want to move?
- Do you want to stop immediately at the end of a move?

This information needs to be in every program you write

NOTE: SP3 does not allow for as much pre-configuration as SP2 did; many functions need to have these values passed in each time. Using constants can help.

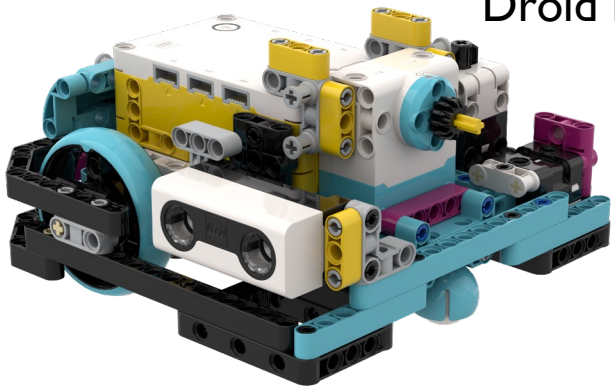
WHAT IS CONNECTED TO EACH PORT?

Drive Base 1 with one extra color sensor on port A (not shown in image)

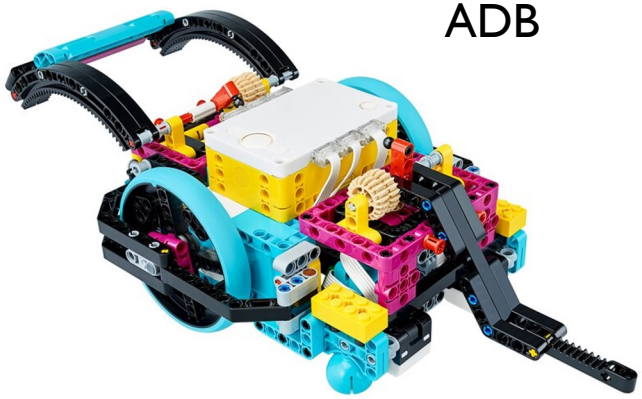


WHAT IS CONNECTED TO EACH PORT?

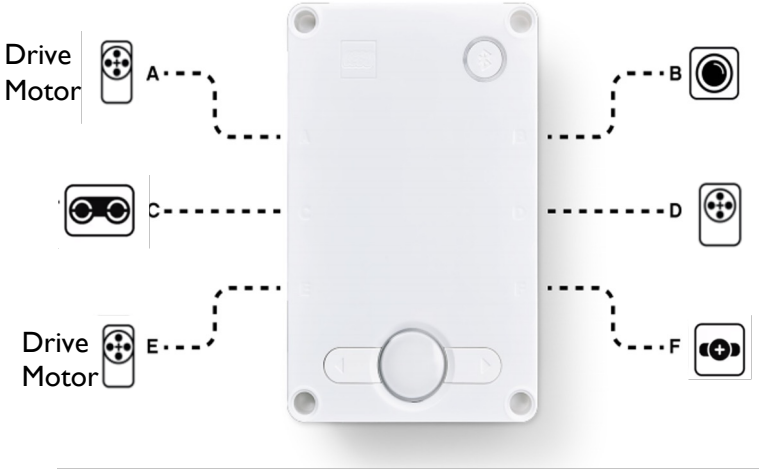
Droid Bot IV



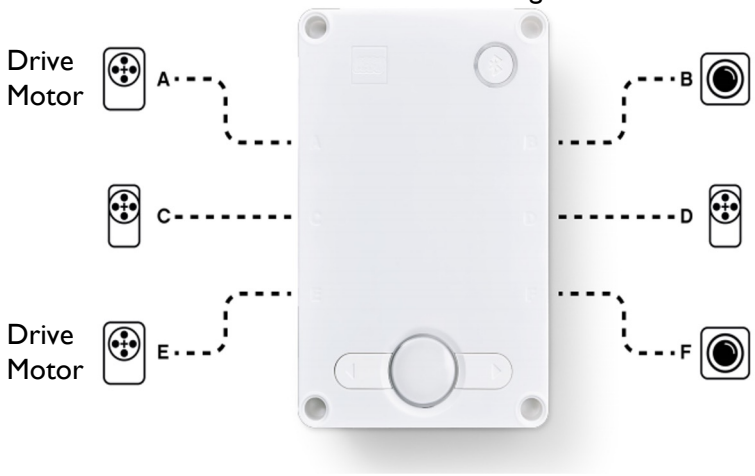
ADB



Droid Bot IV Configuration



ADB Default settings

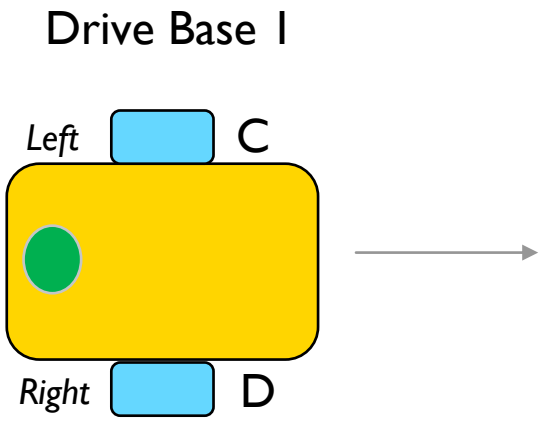


CONFIGURING MOVEMENT BLOCKS

Before using movement methods, you must configure the robot first. The below constructor (a special function that returns an object of the requested type) creates a MotorPair that can be used to move the robot.

```
motor_pair.pair(pair, left_motor_port,
right_motor_port)
```

Determines which motors are connected to the left & right wheels (change the settings for your robot). Whenever functions have 2 inputs for wheels – the first one is for the left wheel and second is for right.



Unlike in Spike 2, you cannot initialize motor speed, stop action or motor rotation by themselves. You can pass speed (now known as velocity) as a parameter when moving the pair, and the stop action when stopping the pair.

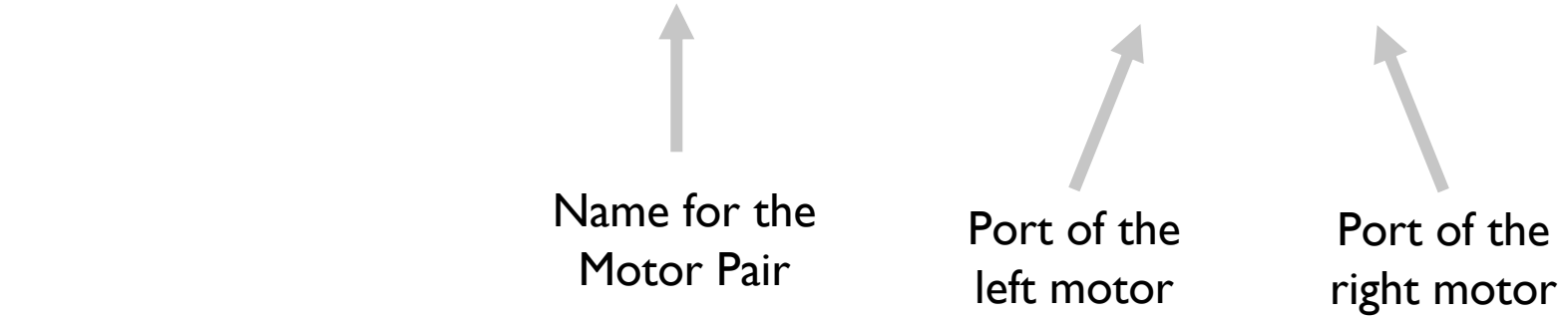
motor_pair APIs can be found in the Motor Pair tab in the Knowledge Base.

MOTOR PAIR INITIALIZATION

To use MotorPair, both motors in the pair must be initialized.

For Drive Base 1:

```
motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
```



WHEEL SIZE AND MOVEMENT CONFIGURATION

The move_for_degrees method for a motor_pair rotates the drive motors by a certain number of degrees

```
motor_pair.move_for_degrees(pair, degrees, steering=0, velocity=360)
```

But motor rotation typically needs to be calculated based on what distance you want your robot to move.

You will need to calculate the `degrees` value as it depends on what wheel you use. The next two slides explain different ways to calculate this value.

Note that you can use inches instead of centimeters if you prefer

HOW MANY CM DOES THE ROBOT MOVE IN 1 ROTATION? (METHOD 1)

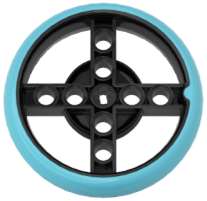
1. Look up the wheel size in mm printed on your tire and divide by 10 to convert to cm (because 1cm=10mm)
2. Multiply the answer in step 1 by **π (3.14) to compute circumference**
3. Assuming you have a direct drive (no gears), rotating the motors by 360 degrees will move the robot forward by the circumference of the wheel. If you use gears, the gear ratio will be a multiplier.

Example calculation using the standard small SPIKE Prime wheels (used in Drive Base 1):

1. Small SPIKE Prime Wheels = 5.6cm in diameter
2. 5.6cm × **π = ~17.5cm per rotation**

It is helpful to write a function that returns the number of degrees to rotate the motors, given a distance. Spike 3, unlike Spike 2, does not have a built-in function to set this:

```
WHEEL_CIRCUMFERENCE = 17.5 # cm, this is a constant for your robot
def degreesForDistance(distance_cm): # input must be in the same unit as WHEEL_CIRCUMFERENCE
    return int((distance_cm/WHEEL_CIRCUMFERENCE) * 360) # Add multiplier for gear ratio if needed
```



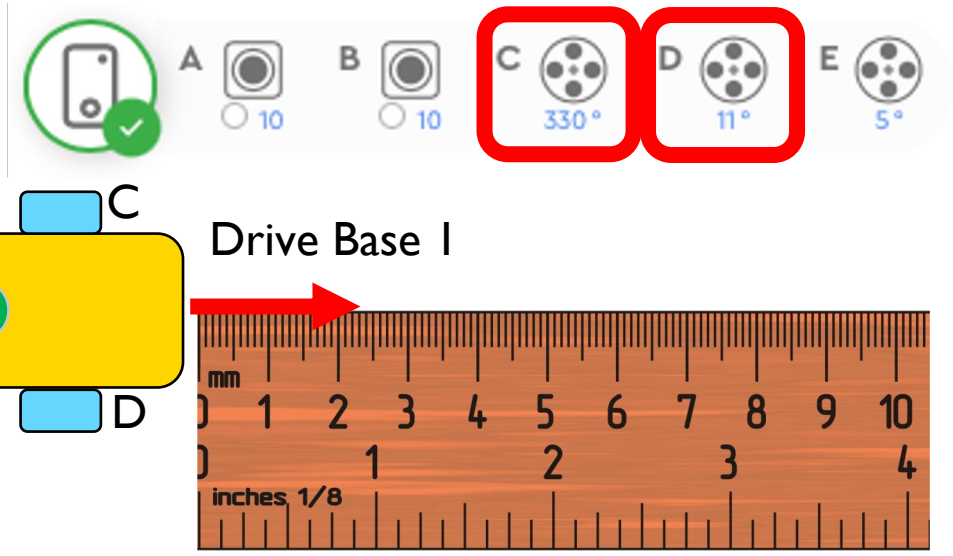
Helpful chart with common LEGO wheels and their diameters.

http://wheels.sariel.pl/

HOW MANY CM DOES THE ROBOT MOVE IN 1 ROTATION? (METHOD 2)

Use the Dashboard to view sensor data to find the Motor Degrees value

1. Put your ruler next to your wheel/robot at 0 centimeters (whatever part of the robot you use to align with 0, you should use to use to measure distance in step 2)
2. Roll your robot forward until the motor encoder reading (in the SPIKE software) reaches 1 rotation, or 360 degrees. Once you learn to program movement, you can program the robot to move 1 rotation forward.
3. Read the number of CM the robot moved along the ruler
4. Use the values to configure your robot's movement



STOP MODES: BRAKE VS. HOLD VS. COAST

`'BRAKE'` – after move, bring motors to a hard stop. This is the default

`'HOLD'` – after move, bring motor to a hard stop and use motor power to counter any further movement until the motor is used again. You will not be able to move the motor by hand.

`'COAST'` – after move, allow motors to move due to momentum

`'SMART_BRAKE'` – to make the motor brake and continue to brake after stop and compensate for inaccuracies in the next command. New in SP3.

`'SMART_COAST'` – to make the motor brake until stop and then coast and compensate for inaccuracies in the next command. New in SP3.

In general, we will use `'HOLD'` or `'BRAKE'` in most of our programs.

```
motor_pair.stop(motor_pair.PAIR_1, motor.BRAKE)
```

MOTOR VELOCITY

If no specific speed velocity is given as an input to the move method, the method will use the default velocity. According to the knowledge base the default velocity (speed) is 360.

It is not, as in Blocks programming, expressed as a percentage of maximum. The min and max velocities are defined by the motor specification.

Small motor (Spike Essential): -660 to 660
Medium motor: -1110 to 1110
Large motor: -1050 to 1050

For example, the code below will move at 360 velocity because no other velocity is specified in the method.

```
motor_pair.move_for_degrees(motor_pair.PAIR_1, 0, 1000)
```

PUTTING IT TOGETHER

For Drive Base 1 and Droid Bot 4, smaller wheels are used. One rotation only moves the robot by 17.5cm.

For ADB, the larger wheels are used. One rotation moves the robot by 27.6cm. A lower velocity can be used for additional control.

Drive Base 1

```
motor_pair.pair(motor_pair.PAIR_1, port.C, port.D)
motor_pair.move_for_degrees(motor_pair.PAIR_1,
degreesForDistance(20), 0, velocity = 400, stop = motor.COAST)
```

ADB

```
motor_pair.pair(motor_pair.PAIR_1, port.A, port.E)
motor_pair.move_for_degrees(motor_pair.PAIR_1,
degreesForDistance(20), 0, velocity = 200, stop = motor.HOLD)
```

CREDITS

This lesson was created by Sanjay and Arvind Seshan for Prime Lessons

Additional contributions by FLL Share & Learn community members

More lessons are available at www.primelessons.org