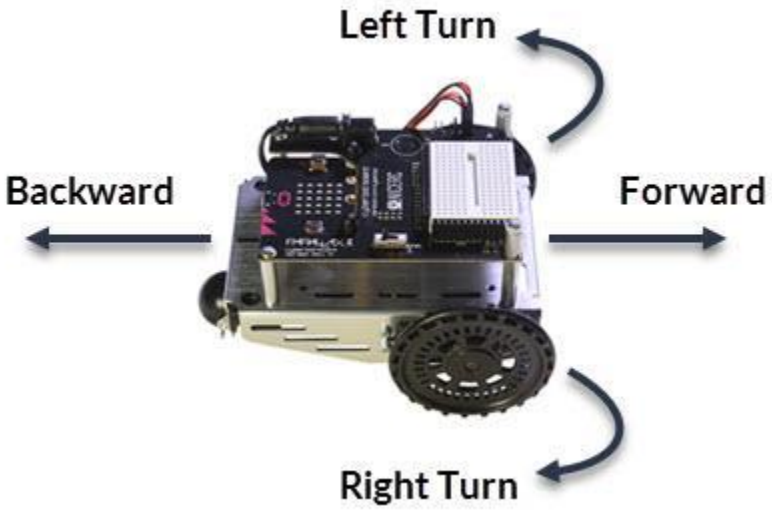
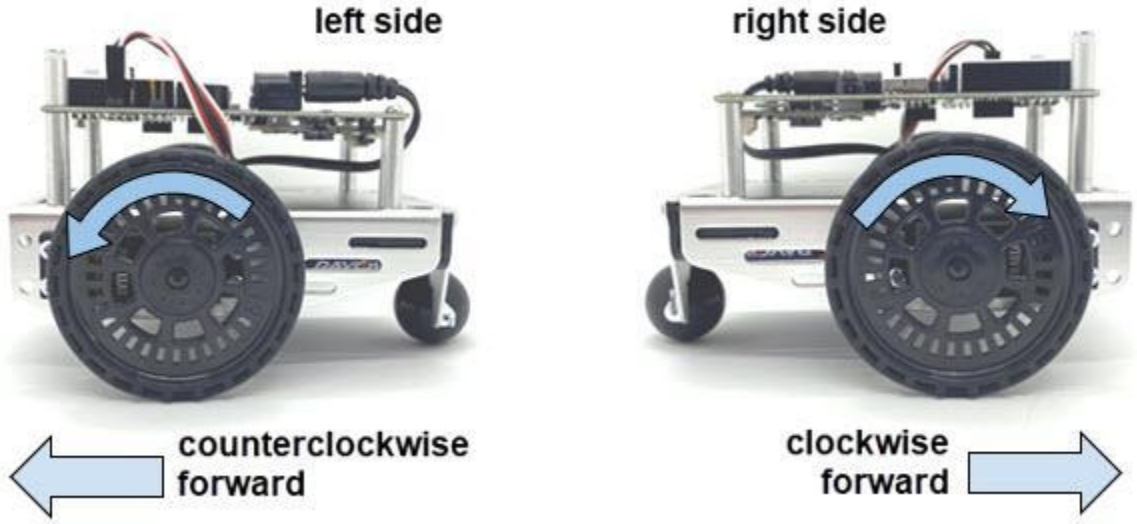
Forward, Backward, and Straightening Curves

The cyber:bot robot's forward is in the direction of its breadboard and drive wheels. Think about the ball in the back as a "tail wheel". One way that might help when thinking about forward, left, etc. is to imagine that you are sitting on the white breadboard facing forward while driving the cyber:bot.

✔ Make sure to commit the cyber:bot robot's left, right, front and back to memory before continuing.

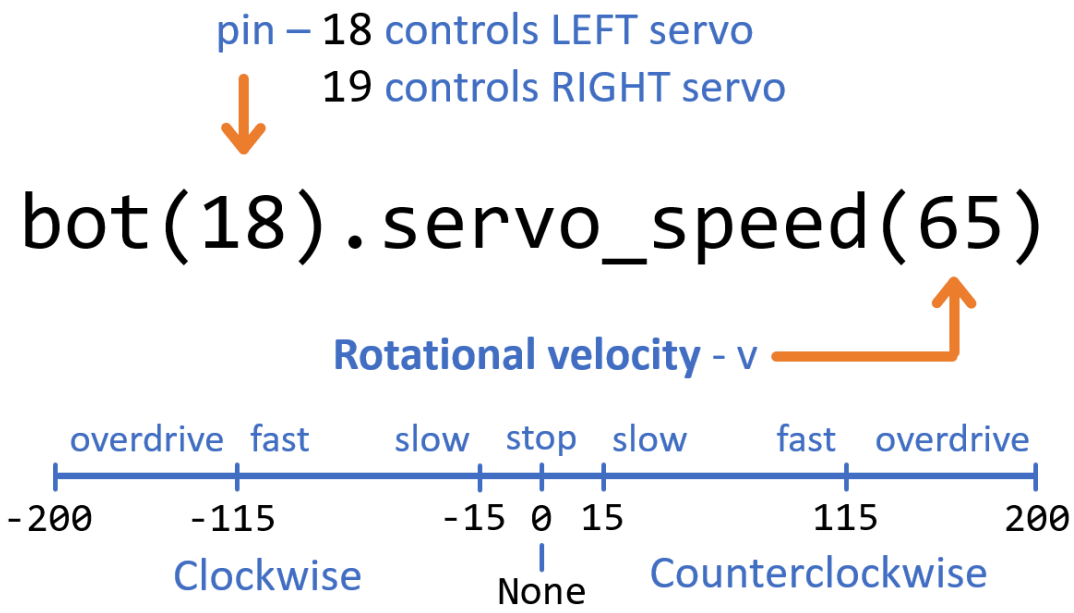


Have you ever thought about what direction a car's wheels have to turn to propel it forward? The wheels turn opposite directions on opposite sides of the car. Likewise, to make the cyber:bot roll forward, its left wheel has to turn counterclockwise, while its right wheel turns clockwise.



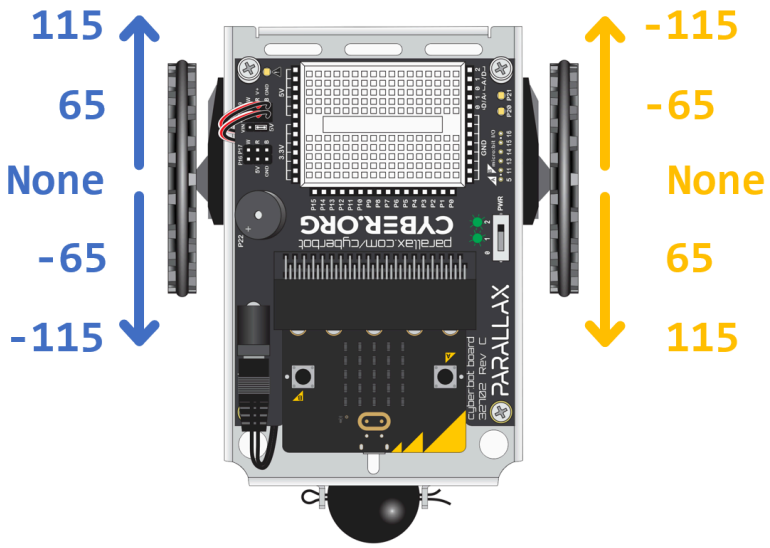
Remember that with **`bot(pin).servo_speed(velocity)`**, the velocity argument can be anywhere in the -200 to 200 range. For turning counterclockwise, the speed control range is 15

(slow) through 115 (fast). For turning clockwise, the speed control range is -15 (slow) through -115 (fast). Values in the 115 through 200 range (and -115 through -200) are considered overdrive because it's asking the servos to go faster than they can. The **stop range** or **deadband range** is -15 through + 15.



Since the wheels need to turn in the opposite direction to go forward or backward, the right wheel speed will be the opposite +/- sign of the left wheel. Use this diagram for picking wheel speeds to set the direction the cyber:bot robot goes. If you use the example of 65 for the left wheel speed and -65 for the right, the cyber:bot will roll forward at about half speed.

```
bot(18).servo_speed(65)  # Left Wheel
bot(19).servo_speed(-65) # Right wheel
```



Don't worry if the cyber:bot goes forward in a curve instead of perfectly straight ahead. You'll adjust that shortly.

Keep in mind that a script can use the **`servo_speed`** function to control the speed and direction of each servo. Then, it can use the **`sleep`** function to keep the servos running for certain amounts of time before choosing new speeds and directions.

Script: forward-three-seconds

Here's an example that will make the cyber:bot roll forward for about three seconds, and then stop.

- ✔ Make sure the cyber:bot board's power switch is set to 1 and the battery pack is plugged into the cyber:bot board.
- ✔ Go to the micro:bit Python Editor at http://python.microbit.org/v/3.
- ✔ Click Open, and find and open cyberbot-template-with-blink.hex.

  (See Add modules to your micro:bit.)
- ✔ Replace its default script with **forward-three-seconds** shown below.
- ✔ Click the Save button to save a copy of your work.

```
# forward-three-seconds

from cyberbot import *

bot(18).servo_speed(65)          # Half-speed forward
bot(19).servo_speed(-65)

sleep(3000)                      # Wait three seconds

bot(18).servo_speed(None)        # Stop
bot(19).servo_speed(None)

```

Tests

✔ Click the Send to micro:bit button to flash the script into the micro:bit.

✔ Disconnect the programming cable and put the cyber:bot on the floor.

✔ Move the switch to position 2 and press/release the reset button on the micro:bit. (Or, set PWR to 0, then back to 2.)

✔ Verify that the cyber:bot drives slowly forward for three seconds.

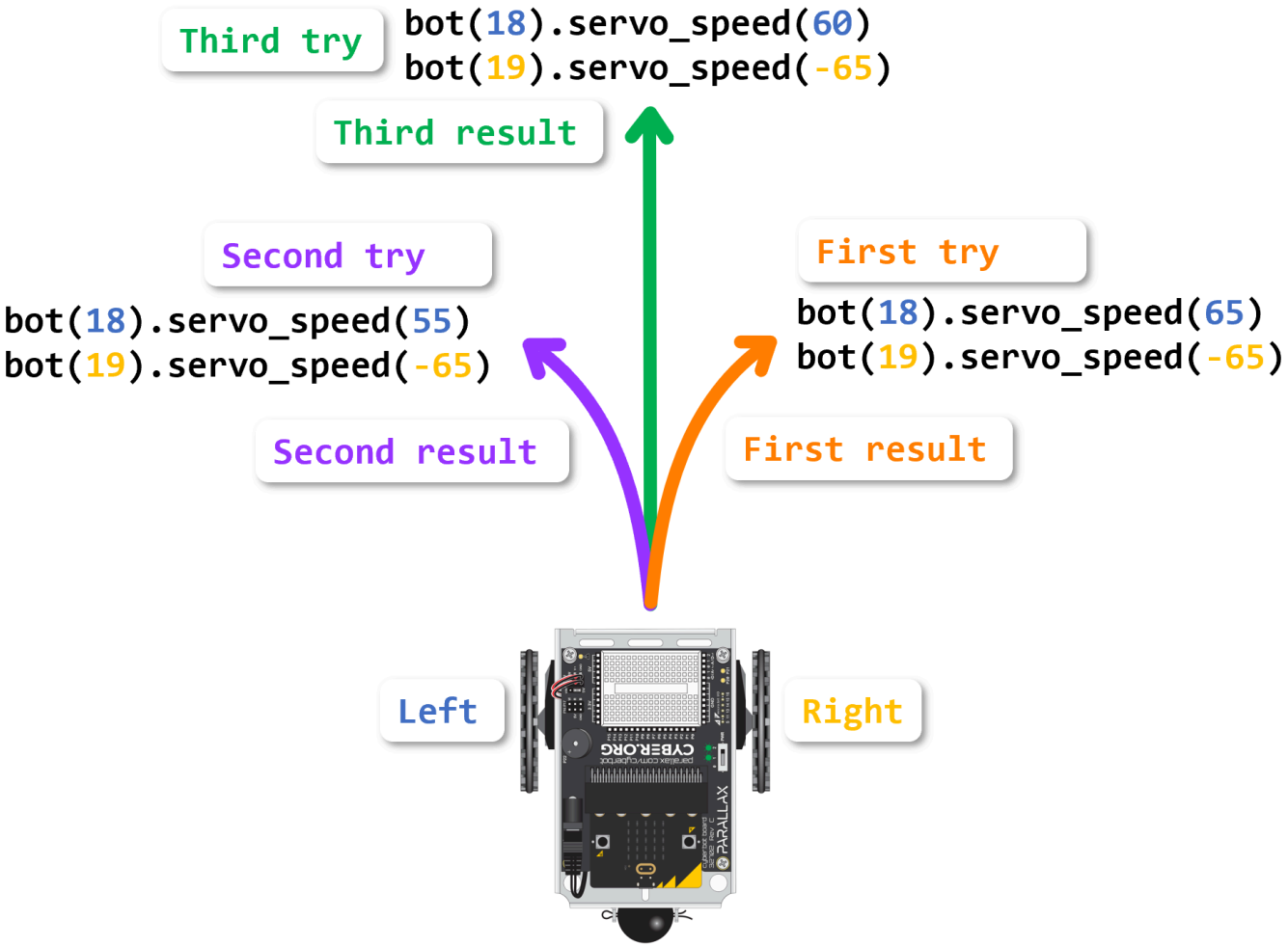
Again, if it's curving instead of going straight forward, you will correct that next.

✔ Set the PWR switch to 0 when the cyber:bot is not in use, and remember to unplug the battery pack from the cyber:bot board if you are done for the day.

Try This – Straightening the Path

Here is an example of straightening the cyber:bot robot's path if it's curving when you want it to go straight. In this example, the statements to make it go forward actually causes some curving in its path.

- On the first try, the cyber:bot curved to the right. That means the left wheel was faster than the right.
- On the second try, bot(18).servo_speed(55) slowed down the left servo too much, and it curved left instead.
- On the third try, bot(18).servo_speed(60) slowed it down just the right amount for straight-forward travel.



TIPS:

- If your cyber:bot curves to the right with identical wheel speed values, slow down the left wheel with bot(18).servo_speed(slower_speed_value) -like trying 55 instead of 65.
- If your cyber:bot instead curves to the left, slow down the right wheel with bot(19).servo_speed(slower_speed_value) -like trying -55 instead of -65.
- It will take several tries to get it right.
- You may want to repeat this for top speed too, and the difference in the wheel speed values probably won't be the same as they were at 65.

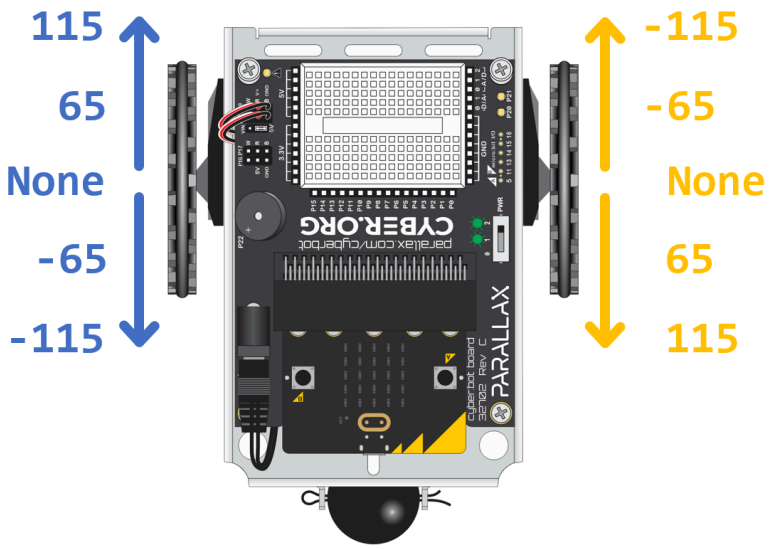
✔ If your cyber:bot robot goes in a curve instead of straight ahead with 65 and -65, modify the wheel speed arguments until it goes straight ahead. Your numbers are likely to be different from the ones in the diagram.

✔ You will have to flash the modified script (with the Send to micro:bit button) after every adjustment you make, test it and maybe try again with what you've learned from the way the curve changed.

Your Turn – Experiment with Speeds, Directions, and Run Times

Here is a repeat of the diagram for setting wheel speeds and how they affect your cyber:bot robot's wheel speeds. It's a good reference for experimenting with forward and backward speeds and turns. You can also use it as a reference for setting up turns, pivots, and curves, which we'll take a closer look at in the next activity.

```
bot(18).servo_speed(65)  # Left Wheel
bot(19).servo_speed(-65) # Right wheel
```



✔ Try servo_speed values and adjust forward paths for these speeds and list the left and right wheel speed values that straightened the forward/backward travel.

✔ Full speed forward:

bot(18).servo_speed(_____)

bot(19).servo_speed(_____)
Either bot(18).servo_speed(115) and bot(19).servo_speed(a lower negative value), or bot(18).servo_speed(a lower positive value) and bot(19).servo_speed(-115).

✔ Half speed forward:

bot(18).servo_speed(_____)
bot(19).servo_speed(_____)
Either bot(18).servo_speed(115) and bot(19).servo_speed(a lower negative value), or bot(18).servo_speed(a lower positive value) and bot(19).servo_speed(-115).

✔ Half speed backward:

bot(18).servo_speed(_____)
bot(19).servo_speed(_____)
Either bot(18).servo_speed(115) and bot(19).servo_speed(a lower negative value), or bot(18).servo_speed(a lower positive value) and bot(19).servo_speed(-115).

✔ Full speed backward:

bot(18).servo_speed(_____)
bot(19).servo_speed(_____)
Either bot(18).servo_speed(115) and bot(19).servo_speed(a lower negative value), or bot(18).servo_speed(a lower positive value) and bot(19).servo_speed(-115).

✔ For a given speed, try doubling the sleep(3000) to sleep(6000).

What happened to the distance it traveled? _____
The cyber:bot travels forward for twice the time, and also about twice as far as a result.

✔ Although the next section focuses on turns, you can probably get some interesting turns, curves, and pivots just from creative use of the numbers in the diagram.