PRIME LESSONS

By the Makers of EV3Lessons

TURNING WITH THE GYRO

BY SANJAY AND ARVIND SESHAN

LESSON OBJECTIVES

- Learn how to turn using the built-in gyro sensor
- Learn how to use the wait_until() function with sensors
- Note: Although images in this lessons may show a SPIKE Prime, the code blocks are the same for Robot Inventor

METHODS YOU NEED IN THIS LESSON

- Motion Sensor methods – Used to read and reset the values of the gyro sensor

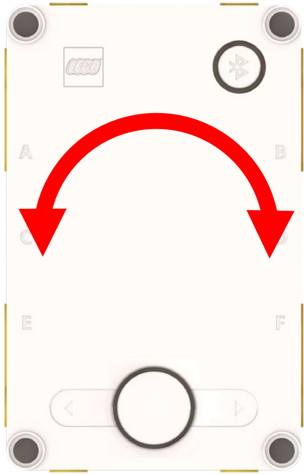
```
hub.motion_sensor.get_yaw_angle()

hub.motion_sensor.reset_yaw_angle()
```

- Operator Functions – Tests a relation between two values and outputs a Boolean (true/false) result

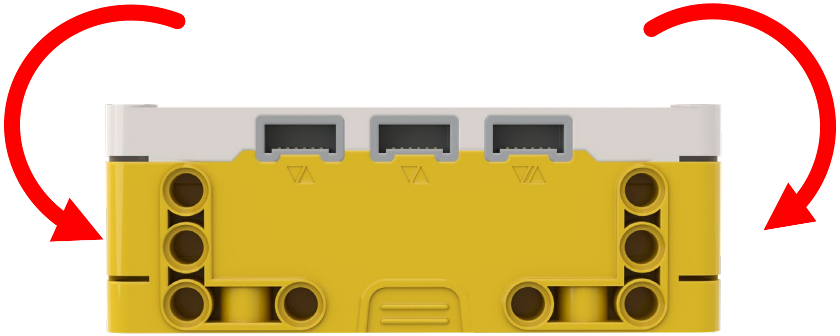
```
greater_than_or_equal_to(a, b)
```

ROBOT ORIENTATION: YAW, PITCH AND ROLL

Yaw is turning the Hub to right or left



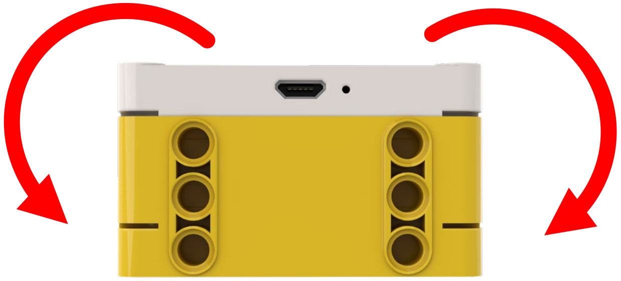
Pitch is turning the Hub up and down

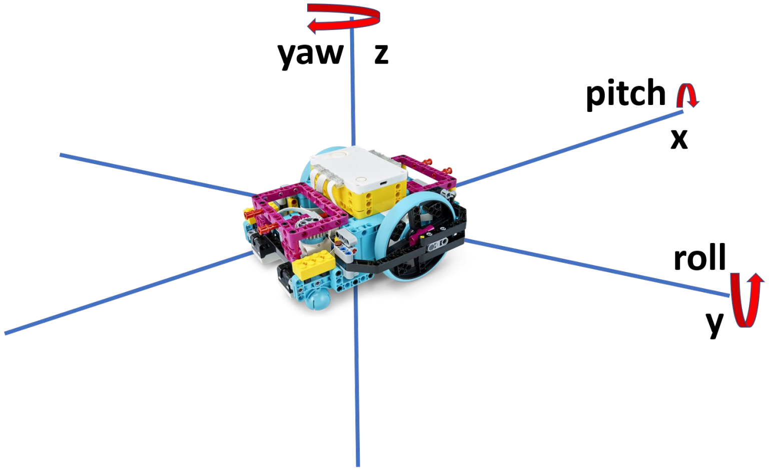


Just like x, y and z coordinates are used to describe a robot's position, yaw, pitch and roll are terms used to describe a robot's orientation. Yaw is rotation around the z-axis. Pitch is rotation around y-axis. Roll is rotation around the x-axis.

The built-in Gyro Sensor can measure the robot's orientation

Roll is turning the Hub to side-to-side





USING THE GYRO SENSOR TO TURN

- The gyro sensor can be programmed to measure the hub's yaw, pitch and roll
- These values can be used to sense if the robot has turned around x, y, or z axes
- In this lesson, we will focus on yaw which can be used to determine if a robot has turned left or right
- For pitch and roll, the robot uses gravity to determine what is a zero reading. Flat on the ground is 0 pitch and 0 roll.
- For yaw, the robot doesn't have a compass to tell it what is north or south. Therefore, you need to tell the robot what it should consider zero. This is done with the reset_yaw_angle() method.
    - Note that clockwise is positive in yaw measurement

```
hub.motion_sensor.get_yaw_angle()
```

```
hub.motion_sensor.reset_yaw_angle()
```

WAITING FOR THE GYRO TO REACH AN ANGLE

- There are two options to measure if the robot has reached the desired angle
- Option I: LEGO-specific API
  - Use the operator functions. Then, use the Wait Functions – Makes the program pause execution for some time.

```
from spike.operator import greater_than_or_equal_to, equal_to, [etc.]
from spike.control import wait_until
wait_until(get_value_function, operator_function=<function equal_to>, target_value=True)
```

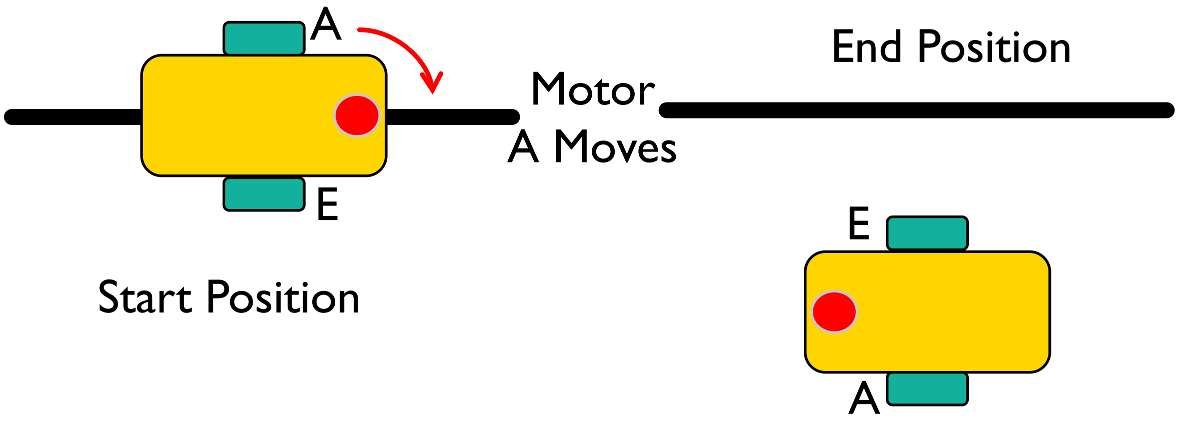
  - This option is easier to use
- Option II: General Python API
  - Use while loops

```
start moving....
while (hub.motion_sensor.get_yaw_angle() < ANGLE):
    <code>
stop moving....
```

  - Easier for running code while waiting. You could also use a user defined operator_function in wait_until() – but a while loop makes the code clearer.
  - If you do not want to run code, you can place `pass` in place of `<code>` to skip the iteration of the loop

THERE ARE TWO TYPES OF TURNS YOU CAN DO

180 Degree Pivot Turn



Motor A Moves

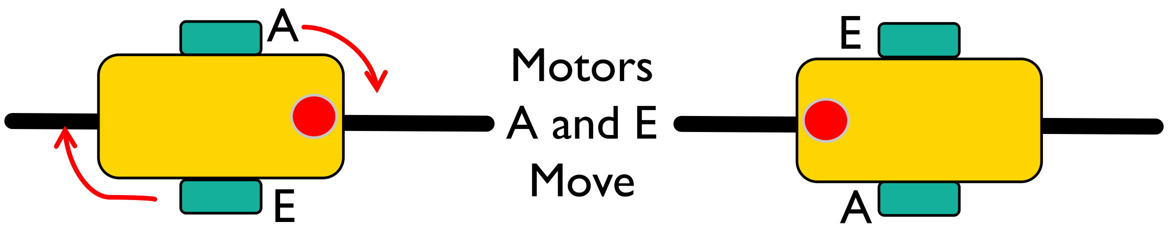
Start Position

End Position

180 Degree Spin Turn

Start Position

End Position



Motors A and E Move

Notice where the robot ends in both pictures after a 180 degree turn.

In the Spin Turn, the robot moves a lot less and that makes Spin Turns are great for tight positions. Spin turns tend to be a bit faster but also a little less accurate.

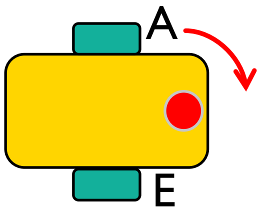
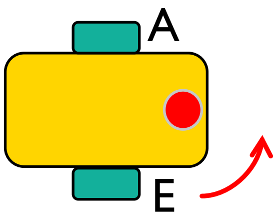
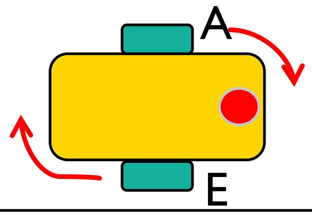
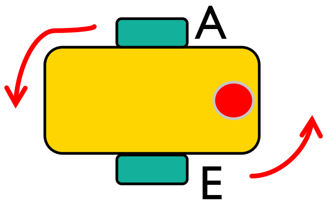
So when you need to make turns, you should decide which turn is best for you!

(Last edit: 01/17/2021)

HOW TO MAKE PIVOT AND SPIN TURNS

Change % Speed values here

```
move_tank(1, unit='rotations', 0, 50)
```

Move Tank Values			
Speed, 0	0, Speed	Speed, -Speed	-Speed, Speed
			
Pivot Turn Right	Pivot Turn Left	Spin Turn Right	Spin Turn Left

CHALLENGE 1

- Write a program that turns 90 degrees to the right using a pivot turn
- Basic Steps:
  - Import operator function

    ```
    from spike.operator import greater_than_or_equal_to
    ```

  - Configure Robot Movement
  - Make your robot start slowly turning right by just turning on the left wheel motor

    ```
    motor_pair = MotorPair('A', 'E')
    motor_pair.set_stop_action('brake')
    motor_pair.start_tank(20, 0)
    ```

    - Use low speeds here to improve keep the turn accurate
  - reset the gyro sensor angle to 0

    ```
    hub.motion_sensor.reset_yaw_angle()
    ```

  - Wait until the gyro yaw angle has reached the degrees you want

    ```
    wait_until(hub.motion_sensor.get_yaw_angle, greater_than_or_equal_to, 90)
    ```

    Or

    ```
    while (hub.motion_sensor.get_yaw_angle() < 90): pass
    ```

  - Stop moving

CHALLENGE I SOLUTION

```
1 from spike.operator import greater_than_or_equal_to
2 motor_pair = MotorPair('A', 'E')
3 motor_pair.set_stop_action('brake')
4 motor_pair.start_tank(20, 0)
5 hub.motion_sensor.reset_yaw_angle()
6 wait_until(hub.motion_sensor.get_yaw_angle, greater_than_or_equal_to, 90)
7 motor_pair.stop()
```

Line 6 can be replaced with:

```
while (hub.motion_sensor.get_yaw_angle() < 90): pass
```

TURNING RIGHT VS. TURNING LEFT

- To change the direction of the turn, you have to:
  1. Change which wheel should turn (or the direction of the power)
  2. The final angle should be -90 degrees instead of 90 degrees
  3. The wait until gyro angle comparison must change also
     1. wait_until function: The comparison should be "`less_than_or_equal_to`" instead of "`greater_than_or_equal_to`" since the angle is decreasing instead of increasing
     2. while loops: The comparison should be `while (...angle() < 90): pass` when the angle is increasing (right turn generally) and `while (...angle() > -90): pass` when the angle is decreasing (left turn generally)

Right Turn

```
motor_pair.start_tank(20, 0)
hub.motion_sensor.reset_yaw_angle()
wait_until(hub.motion_sensor.get_yaw_angle, greater_than_or_equal_to, 90)
motor_pair.stop()
```

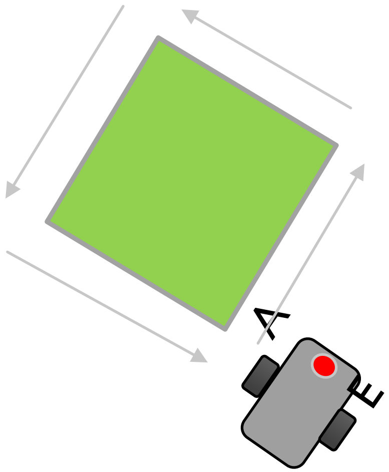
Left Turn

```
motor_pair.start_tank(0, 20)
hub.motion_sensor.reset_yaw_angle()
wait_until(hub.motion_sensor.get_yaw_angle, less_than_or_equal_to, -90)
motor_pair.stop()
```

TURNING CHALLENGES

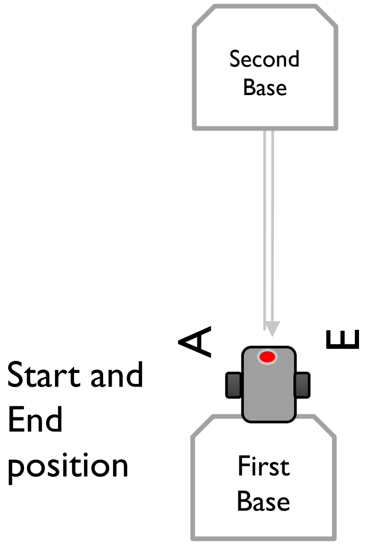
Challenge 1

- Your robot is a baseball player who has to run to all the bases and go back to home plate.
- Can you program your robot to move forward and then turn left?
- Use a square box or tape



Challenge 2

- Your robot baseball player must run to second base, turn around and come back to first.
- Go straight. Turn 180 degrees and return to the same spot.



CHALLENGE SOLUTIONS

Challenge 1

You probably used a combination of the move() method to go straight and do **pivot turns** to go around the box.

Challenge 2

You probably used a spin turn because it is better for tighter turns and gets you closer to the starting point!

CREDITS

- This lesson was created by Sanjay and Arvind Seshan for Prime Lessons
- More lessons are available at www.primelessons.org