Learning To Program the Robot

This notebook will guide you through controlling an **omni-wheel robot** step by step. You'll see an **example**, then try to complete a similar challenge on your own!

Learning Objectives

- Understand how to move the robot using Python commands.
- Experiment with different speeds, directions, and durations.
- Observe and adjust movement patterns to improve control.

In [ ]:
```
# This part allows VScode to communicate with the robot
# This imports the required libraries and then creates a node for our robot
import sys
import os
import time

# Add parent directory to the Python path
sys.path.insert(0, os.path.abspath('..'))

import rclpy
from controllers.omni_robot_controller import OmniWheelControlNode  # Import


# Initialize ROS2 node
rclpy.init()
node = OmniWheelControlNode()
```

Example 1: Moving Forward

To move forward you can use the command below

node.move_Forward(speed in m/s, duration in seconds)

In [ ]:
```
# Example: This moves the bot forward at 0.5 m/s for 2 seconds.
# The bot should cover a distance of 1 meter.
# Lay out a meter stick and run this code to check that the bot behaves as e

node.move_forward(0.5,2)
```

Challenge 1: Moving Forward

In [ ]:
```
# Challenge - a student wants a robot to move forward for a distance of 1.5
# They whant this done over a 3 second time span.
```

Example 2: Multi-Motion

Sometimes you are going to have to use multiple motions that require starting and stopping motion. You can make the robot wait using the command:

time.sleep(insert time to wait)

In [ ]:
```
# Example: This moves the bot forward at 0.5 m/s for 2 seconds.
# The bot then pauses for 5 seconds.
# The bot then moves forward at 1 m/s for 1 second.
# Run this code to confirm this movement occurs


node.move_forward(0.5,2.0)
time.sleep(5)
node.move_forward(1.0,1.0)
```

In [ ]:
```
# Challenge
# 1) Move the robot forward at 0.7 m/s for 3 seconds.
# 2) The bot should then pause for 3 seconds.
# 3) The bot should then cover a distance of 2 meters in 5 seconds.
# Write the code below
```

Example 3: Moving Backward

To move backwards you can use the command below

node.move_Backward(speed in m/s, duration in seconds)

In [ ]:
```
# Example: This moves the bot forward at 0.5 m/s for 2 seconds.
# The bot should cover a distance of 1 meter.
# Lay out a meter stick and run this code to check that the bot behaves as e

node.move_backward(0.5,2)
```

Challenge 4: Moving Backward

In [ ]:
```
# Challenge - a student wants a robot to move backward for a distance of 1 m
# They whant this done over a 2 second time span.
```

Challenge 5: Multi-Motion Movement

In [ ]:
```
# Challenge
# 1) Move the robot forward at 1 m/s for 1 seconds.
# 2) The bot should then pause for 5 seconds.
# 3) The bot should then move backward at 1 m/s for 1 seconds.
# Write the code below
```

Shutting Down the Node

Once you're done, **shutdown the node** properly.

In [ ]:
```
node.destroy_node()
rclpy.shutdown()
```