

Buzzer_Challenge_5__Solution

This notebook contains **Challenge 5: Melodic Robot Tune**.

```
In [ ]: import rclpy
        from omni_robot_controller import OmniWheelControlNode # Import control node

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

i ROS2 Node Initialization

This cell includes the correct libraries and initializes the `OmniWheelControlNode` for use with the challenges. **Make sure ROS2 is installed and sourced in your environment.**

Challenge 5: Melodic Robot Tune

Objective

Program the buzzer to **play a simple melody** using a sequence of frequencies.

Instructions

1□ Create a Melody

- Define a **list of frequencies** that represent musical notes.
- Use different values to create a **simple melody**.

2□ Play Each Note

- Use a **loop** to play each frequency in the melody.
- Each frequency should play for **0.3 seconds**.

3□ Add Pauses Between Notes

- Insert **0.1 seconds of silence** between each note to create rhythm.
-

Available Functions & How to Use Them

** Define a Melody**

Store the sequence of frequencies in a list:

```
melody = [500, 700, 900, 700, 500]
```

** Play a Note**

```
node.play_buzzer(frequency, on_time, off_time, repeat)
```

Parameters:

- `frequency` : The note frequency in Hz.
- `on_time` : Duration the note plays (`0.3` seconds).
- `off_time` : Delay before the next note (`0.1` seconds).
- `repeat` : Number of times the note plays.

** Add a Pause Between Notes**

```
time.sleep(0.1)
```

Creates a short silence between each note.

Hints

Store the melody frequencies in a list.

Use a `for` loop to play each note.

Set `on_time = 0.3s` and `off_time = 0.1s` to maintain rhythm.

Experiment with different frequencies to create unique tunes.

** Example (Not a Solution)**

"The robot should play a sequence of five tones, each lasting 0.3 seconds with a 0.1-second break in between."

```
In [ ]: # Your code here:
melody = [500, 700, 900, 700, 500]
for freq in melody:
    node.play_buzzer(freq, 1.0, 1.0, 2)
    time.sleep(0.1)
```

3. Debugging & Troubleshooting

Fill in the missing solutions:

Issue	Solution
Buzzer not playing?	
No sound?	
Syntax errors?	

4. Reflection

- What was the most **challenging** part of these activities?
- How can you **improve** your buzzer functions for **real-world applications**?
- What **new ideas** do you have for buzzer-based interactions in robotics?

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