

Buzzer_Challenge_7__Student

This is the **student version** of **Challenge 3: Morse Code Beeper**.

Follow the instructions to complete the challenge on your own.

```
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 3: Morse Code Beeper

Objective

Write a function that converts **text into Morse code** beeps and plays it using the buzzer. Implement a buzzer sequence for **SOS** (... --- ...).

Instructions

1□ Create a Morse Code Dictionary

- Define a **dictionary** that maps letters to Morse code symbols (`.` for dots and `-` for dashes).
- Example:
`morse_code = {'S': '...', 'O': '---'}`

2□ Loop Through the Letters

- Iterate through each letter in the word **"SOS"**.

- Convert the letter into **Morse code symbols** using the dictionary.

3▯ Convert Symbols to Beeps

- **Dots (.) should be short beeps.**
- **Dashes (-) should be long beeps.**
- Use `node.play_buzzer()` to play the corresponding beep.
- Add a small delay (`time.sleep(0.2)`) between each beep for clarity.

4▯ Add Pauses Between Letters

- Introduce a slightly **longer pause** between letters.
 - This helps distinguish separate characters in Morse code.
-

Available Functions & How to Use Them

** Play a Buzzer Sound**

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

- **Parameters:**
 - `frequency` : The sound frequency in Hz.
 - `on_time` : Duration the buzzer stays ON.
 - `off_time` : Duration the buzzer stays OFF.
 - `repeat` : Number of times the buzzer plays.

** Time Delay Function**

```
time.sleep(seconds)
```

- **Pauses execution** for the specified number of seconds.
 - Use this to create spacing between Morse code signals.
-

Hints

Use a **dictionary** to store Morse code mappings.

Loop through each **letter** and then each **symbol** in Morse code.

Assign **shorter beeps** for dots (.) and **longer beeps** for dashes (-).

Add **pauses** between letters to make the message readable.

** Example (Not a Solution)**

"Convert the word 'SOS' into Morse code using beeps, with short beeps for dots and long beeps for dashes. Ensure there are brief pauses between each beep and a longer pause between letters."

By following this structured approach, you will **understand how to convert text into Morse code using the buzzer** without seeing a direct solution.

In []: *# Write your solution here*