Using the Buzzer on the RRC Lite Board - Interactive Notebook

This notebook will guide you through controlling a **buzzer** using the **RRC Lite Board** in **ROS2**.

Learning Objectives

- Understand the difference between **active** and **passive** buzzers.
- Learn how to use ROS2 commands to control a buzzer.
- Experiment with different frequencies and sound patterns.
- Reference functions from **OmniWheelControlNode** for buzzer operations.

In [ ]:

```
import rclpy
from omni_robot_controller import OmniWheelControlNode  # Import control nod

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

1. Definitions

Fill in the definitions for the following terms:

- **Buzzer:**
- **Active Buzzer:**
- **Passive Buzzer:**
- **Frequency:**
- **ROS2:**

2. How Does a Buzzer Work?

Explain how a buzzer converts **electrical signals** into **sound waves**.

3. Why Use Buzzers in Robotics?

List three reasons why buzzers are useful in robotics: 1. 2. 3.

4. ROS2 Buzzer Commands

Match the commands to their functions:

- `node.play_buzzer(freq, on_time, off_time, repeat)` :
- `node.stop_buzzer()` :

5. Coding Tasks

Task 1: Play a Sound

Goal: Make the buzzer play a **1000Hz sound for 3 cycles**.

In [ ]:

```
# Play a 1000Hz sound for 3 cycles using ROS2 command
# Your code here:
# node.play_buzzer(?, ?, ?, ?)
```

Task 2: Stop the Buzzer

Goal: Write the Python command to stop the buzzer.

In [ ]:

```
# Stop the buzzer using ROS2 command
# Your code here:
# node.stop_buzzer()
```

Task 3: Try Different Frequencies

Goal: Play different frequencies in sequence (100Hz, 200Hz, 300Hz, 400Hz).

In [ ]:

```
# Play different frequencies sequentially using ROS2 command
# Your code here:
# node.play_buzzer(?, ?, ?, ?)
# node.play_buzzer(?, ?, ?, ?)
# node.play_buzzer(?, ?, ?, ?)
# node.play_buzzer(?, ?, ?, ?)
```

6. Student Challenge

Choose one of the challenges and implement it in your code.

Challenge 1: Robot Alerts

- Play a **short beep** when the robot **starts moving**.
- Play a **longer beep** when the robot **stops moving**.

In [ ]:

```
# Your code here:
# if robot_moving:
#     node.play_buzzer(?, ?, ?, ?)
# else:
#     node.play_buzzer(?, ?, ?, ?)
```

Challenge 2: Custom Sound Patterns

- Play a **low tone (500Hz)** when moving forward.
- Play a **high tone (2000Hz)** when rotating.

In [ ]:

```
# Your code here:
# if moving_forward:
#     node.play_buzzer(?, ?, ?, ?)
# elif rotating:
#     node.play_buzzer(?, ?, ?, ?)
```

7. Debugging & Troubleshooting

Fill in the missing solutions:

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

8. Reflection

- What was the most challenging part of this activity? How did you solve it?
- What are some ways you could use the buzzer in future robotics projects?

In [ ]:

```
node.destroy_node()
rclpy.shutdown()
```