BotBrains\_Battle\_Round-2

Two Wheeled Self-Balancing Robots (TWSBR)

-Mokshyada Mishra

**Documentation**

The documentation file includes all the nitty-gritty details of our design process, calculations, ideation, diagrams, and more. It's got everything you need to understand and replicate our project.

#### **Breaking Down the Problem**

To tackle this project, we divided the problem into several parts:

1. **Balancing Mechanism**: The robot needs to maintain its balance at all times.
2. **Navigation**: The robot must navigate to the delivery location, avoiding obstacles.
3. **Payload Management**: The robot should be able to carry and securely hold parcels.
4. **Communication**: The robot should communicate its status and receive commands wirelessly.

#### **Development Strategy**

* **Step-by-Step Implementation**: We decided to implement the project step-by-step, starting with the balancing mechanism. Once the robot could balance reliably, we would move on to navigation and then add the communication features.
* **Testing and Iteration**: At each stage, we would test the robot thoroughly and make necessary adjustments. This iterative approach would help us catch and fix issues early.

**COMPONENTS:**

Hardware Components:

1. Microcontroller:

- Arduino Uno (or any similar microcontroller board)

2. Sensors:

- IMU Sensor: MPU6050 (Inertial Measurement Unit)

- GPS Module:A generic GPS module such as NEO-6M

3. Motors and Motor Drivers:

- DC Motors: Two DC motors with encoders (optional but useful for precise control)

- Motor Driver: L298N Motor Driver Module

4. Power Supply:

- Battery Pack: A suitable battery pack (LiPo or similar) to power the motors and the microcontroller

5. Miscellaneous:

- Wires: Jumper wires for connections

- Breadboard: For prototyping the circuit

- Connectors: Battery connectors, motor connectors

**Programming**:

* Initialization: Set up sensors and motor drivers.
* Balance Control Loop: Use a PID controller to keep the robot balanced with data from the sensors.
* Navigation Algorithm: Use LIDAR for mapping and planning the route.
* Obstacle Avoidance: Detect and avoid obstacles in real-time using LIDAR.
* Communication: Set up wireless communication for remote control and monitoring.

#### **Creativity and Uniqueness**

* **LIDAR for Advanced Navigation**: Most TWSBRs use basic sensors for navigation. By adding LIDAR, we aimed to enhance the robot's ability to navigate complex environments autonomously.
* **Modular Design**: We designed the code to be modular, making it easier to add new features or improve existing ones in the future.

### Conclusion

This ideation process helped us clarify our approach and ensured that we covered all necessary aspects of the project. By breaking down the problem and tackling it step-by-step, we aimed to create a robust and innovative Two Wheeled Self-Balancing Robot capable of efficient parcel delivery.