The project involves programming a Two-Wheeled Self-Balancing Robot (TWSBR) to autonomously deliver parcels to consumers' doorsteps, navigating through traffic, turns, and obstacles. This requires integrating various hardware components, developing robust software algorithms, and ensuring the system's reliability in real-world conditions.

**Ideation**

To achieve this, the robot is designed with two wheels for balance and movement, controlled by a microcontroller that processes sensor data and executes navigational algorithms. Key components include a LIDAR sensor for obstacle detection, an IMU for orientation sensing, and actuators for motor control. The navigation system employs the A\* algorithm for path planning, dynamically updating paths based on LIDAR data to avoid obstacles. Motion control is managed by a PID controller that adjusts motor speeds to maintain balance and follow the planned path. Delivery confirmation is ensured through a user interaction system, complemented by localization techniques like dead reckoning for precise positioning.

**Diagram**

A diagram of a machine

Description automatically generated

**Theory**

**Path Planning (A\*)**

The A\* (A-star) algorithm is employed for path planning, a widely used approach for finding the shortest path from a starting point to a goal. The algorithm uses a heuristic to estimate the cost of reaching the goal, combined with a search strategy to explore the paths efficiently. In this implementation, the A\* algorithm takes into account the robot's current position, the destination (goal), and updates the path based on real-time LIDAR data. The heuristic used is typically Euclidean distance, allowing the algorithm to navigate through dynamic environments by re-evaluating the path as new obstacles are detected.

**PID Controller**

The Proportional-Integral-Derivative (PID) controller is used for motion control of the TWSBR. It adjusts the motor speeds based on the difference between the desired orientation and the current orientation obtained from the IMU. The PID controller consists of three terms:

* **Proportional (P)**: Directly proportional to the current error, it steers the robot toward the goal.
* **Integral (I)**: Sum of past errors, it eliminates the residual error that remains after the proportional term correction.
* **Derivative (D)**: Rate of change of the error, it anticipates the future error and minimizes overshoot.

The PID controller continuously computes control signals to the motors, ensuring that the robot maintains balance and follows the path generated by the A\* algorithm. This dual-control approach allows the robot to navigate complex environments while maintaining stability.

**User Interaction and Localization**

User interaction is facilitated through a simple system that notifies users when the delivery is complete. Localization techniques such as dead reckoning are used to estimate the robot's position based on its previous known position and the estimated speed and direction of travel. While dead reckoning can accumulate errors over time, it provides a real-time estimate of the robot's position, which is sufficient for the delivery task.