



Self-Balancing Robot ANTER

Full Documentation & Technical Breakdown

Meet the Team Behind ANTER



Ahmed Fahmy

Project Lead & Hardware Integration



Mohamed Mohsen

Firmware & Algorithm Development



Malak Alsayed

Sensor & Calibration Expert



Marwa Abdelmoneim

Testing & Performance Analysis

Together, we engineered the future of personal mobility.

Project overview

ANTER is a sophisticated self-balancing robot designed for precision and stability. This project focuses on integrating advanced sensor technology with robust control algorithms to achieve autonomous upright motion.

- Demonstrate principles of dynamic stability
- Implement real-time sensor data processing
- Develop effective PID control for balance
- Provide a platform for advanced robotics research



Goals of the System

Achieve Stable Self-Balancing

Maintain vertical equilibrium under varying conditions and external disturbances.

Responsive Control

Ensure quick and accurate response to changes in tilt angle and user input.

Modular Hardware Design

Utilize easily replaceable and upgradeable components for flexibility.

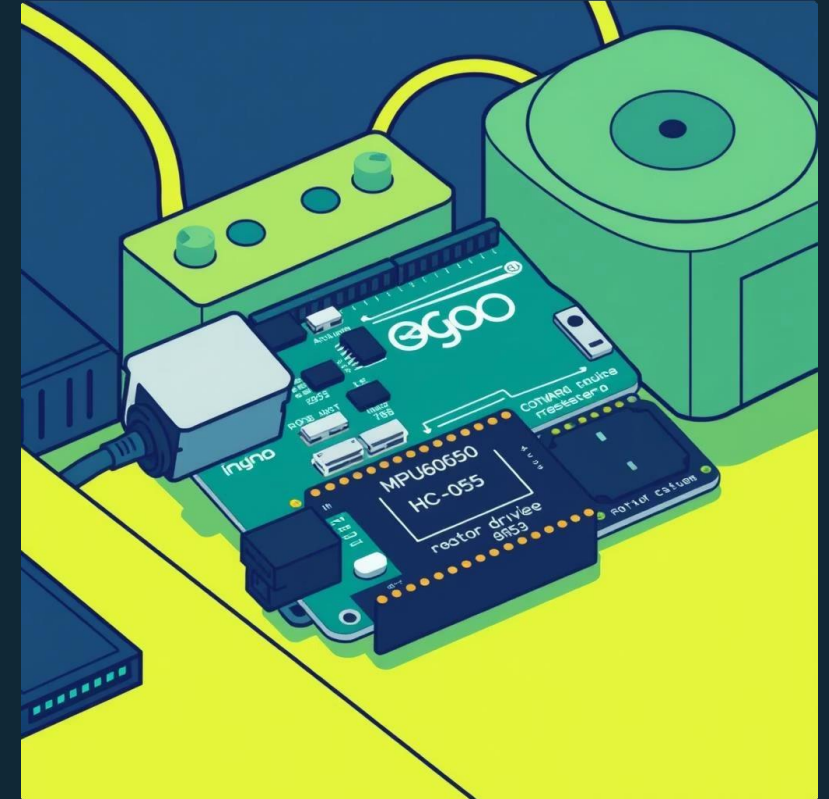
Intuitive Bluetooth Control

Enable seamless command and monitoring via a mobile application.

Our primary objective was to engineer a robot that not only balances but does so with grace and precision, responding intelligently to its environment.

Key Hardware Components

Arduino Nano	Microcontroller	Brain of the robot
MPU6050	Accelerometer/Gyroscope	Measures tilt angle
DC Motors (x2)	High-torque 12V	Propulsion & Balance
Motor Driver (L298N)	H-bridge module	Controls motor speed/direction
HC-05 Bluetooth	Wireless module	Remote control
LiPo Battery	11.1V, 2200mAh	Power supply



Each component was selected for its reliability and performance in embedded systems.

Essential Libraries Used



Wire.h

Facilitates I2C communication, crucial for interacting with the MPU6050 sensor.



MPU6050.h

Provides an interface for reading accelerometer and gyroscope data from the MPU6050.



SoftwareSerial.h

Enables serial communication on digital pins, used for the HC-05 Bluetooth module.



PID_v1.h

A robust PID control implementation, optimized for real-time balancing applications.

These libraries are the backbone of ANTAR's functionality, ensuring seamless operation and control.

Wiring & Pin Connections

Precise wiring is fundamental for the robot's stability and proper functioning. Below is a simplified overview of critical connections.



Power Distribution

LiPo battery to motor driver and Arduino via voltage regulator.



MPU6050 to Arduino

I2C (SDA, SCL) pins connected for sensor data.



Motor Driver to Arduino

PWM pins for speed control (ENA, ENB) and digital pins for direction (IN1-IN4).



HC-05 to Arduino

SoftwareSerial (RX, TX) pins for Bluetooth communication.

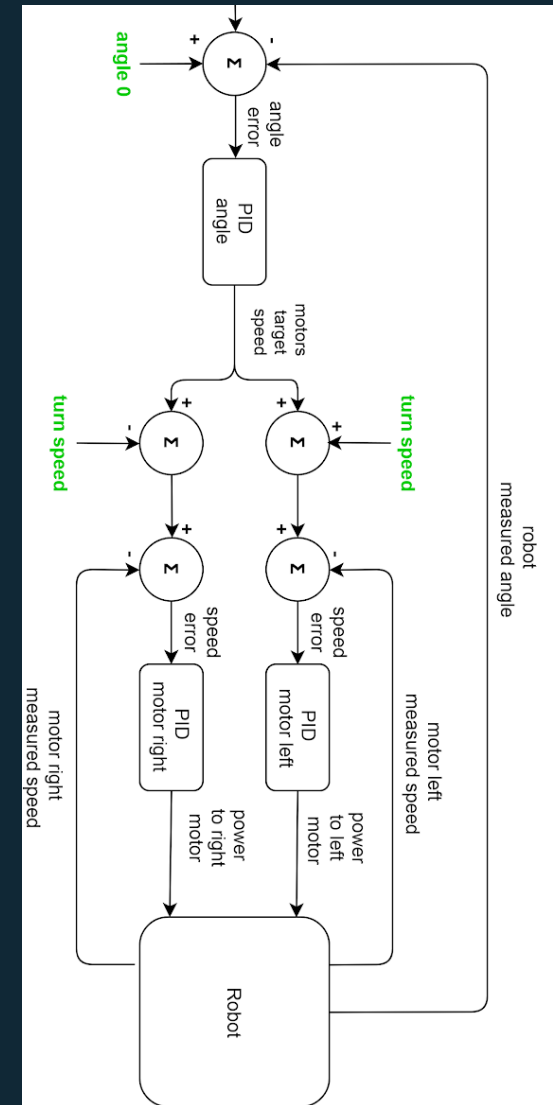
Careful attention to these connections prevents electrical noise and ensures reliable data transfer.

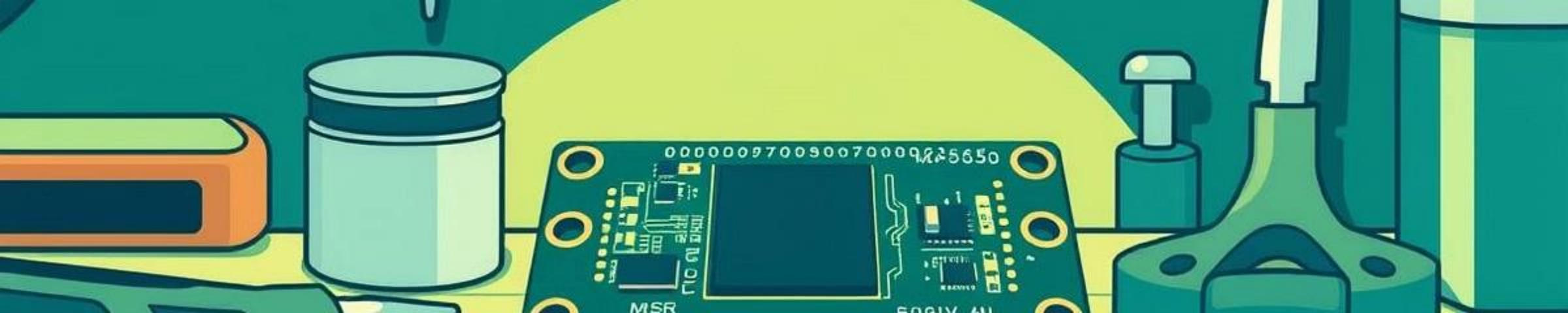
PID Control Explained

The Proportional-Integral-Derivative (PID) controller is the core algorithm for ANTAR's self-balancing capability. It continuously calculates an error value as the difference between a desired setpoint (0 degrees tilt) and a measured process variable (current tilt angle).

- **Proportional (P):** Responds to the current error. A larger K_p means a stronger, faster correction.
- **Integral (I):** Accumulates past errors, helping to eliminate steady-state offset. K_i addresses persistent errors.
- **Derivative (D):** Predicts future error based on the rate of change. K_d dampens oscillations and provides predictive control.

📌 Optimizing K_p , K_i , K_d values is critical for stable and responsive balancing.





MPU6050 Calibration for Accuracy

The MPU6050, our primary tilt sensor, requires careful calibration to provide accurate angle readings. Inaccuracies can lead to erratic balancing behavior.

Code Architecture & Algorithms



Initialization

Setup MPU6050, Bluetooth, motor pins, and PID constants. Includes a brief delay for sensor warm-up.



Sensor Data Acquisition

Read raw accelerometer and gyroscope values from MPU6050. Apply calibration offsets.



Angle Calculation (Complementary Filter)

Combine accelerometer (long-term stability) and gyroscope (short-term accuracy) data to get a precise pitch angle. $\text{Angle} = 0.98 * (\text{Angle} + \text{GyroY} * dt) + 0.02 * \text{AccAngle}$.



PID Computation

Calculate PID output based on the current angle error relative to the desired setpoint (0 degrees). Incorporate Kp, Ki, Kd values.



Motor Control

Translate PID output into PWM signals for motor speed and direction. Implement saturation to prevent motor burnout.



Bluetooth Command Processing

Receive and parse commands (e.g., move forward/backward, adjust PID) from the HC-05 module.

This structured approach ensures robust and efficient balancing, allowing for real-time adjustments and control.

Robot Design & Structure

The ANTER robot features a robust and modular physical design, meticulously engineered for stability and performance. Its compact form factor ensures agility while providing ample space for all essential components.

- **Chassis:** Constructed from lightweight aluminum alloy and high-strength ABS plastic for durability and reduced mass.
- **Wheels:** Custom-designed 65mm diameter rubber wheels provide optimal traction and support, driven by powerful DC motors.
- **Center of Gravity:** Strategic placement of the battery and motors at the base ensures a low center of gravity, critical for maintaining dynamic stability.

