SELF BALANCING BOT

By- SACHIN KUMAR

❖ SELF BALANCING BOT OR ROBOT IS A TYPE OF ROBOTIC AND AUTOMATION SYSTEM .

- **KEY POINTS:-** **It is designed to balance itself on two wheels. Its primary goal is to maintain its balance and stability while moving around, without toppling over. This is achieved through the use of a "closed-loop feedback control" system, which involves real-time data from motion sensors to control the motors and quickly compensate for any tilting motion.**

The self-balancing bot works similarly to an upside-down pendulum, where it regularly adjusts its movement to maintain its center of gravity above the pivot point. This is done by driving its wheels in the direction it is falling , similar to how we balance a stick on our finger.

• WORKING PRINCIPLE OR MECHANISM

Self-balancing bots use a control method called PID (Proportional-Integral-Derivative), which is a close feedback control mechanism that adjusts the motor speed based on the error between the desired and actual angle of inclination. The PID controller calculates the error and adjusts(has three constant called Kp(proportionality constant),Ki(integral constant) and Kd(derivative constant)) the motor speed accordingly to maintain balance.

As mentioned above it uses real time data responded using MPU6050(gyroscope and accelerometer), as if it tilted force fully in forward direction then the bot will respond itself as to stand vertically hence move forward same as for backward.

• KEY COMPONENT

HARDWARE:

1. **Microcontroller:** **Arduino or other compatible boards (e.g., ESP32, Raspberry Pi) to process sensor data and control motors.**

2. **Sensors:** Accelerometer-Gyroscope module (MPU6050) to measure pitch, roll, and yaw.
3. **Motors:** Two DC motors with gearboxes to provide the necessary torque and speed.
4. **Motor Driver:** L298N or similar motor driver to control motor speed and direction.
5. **Power source:** Li-ion battery or other power sources to power the bot.
6. **Chassis:** A balanced chassis to hold the components together.
7. **Wheels:** Two wheels with a suitable diameter and material to provide stability and traction.

SOFTWARE:

PID controlling, motor control, balancing algorithm all are done by the code using ARDUINO IDE software.

- CODE: The link for the code is provided below
  https://drive.google.com/file/d/1TpSpnSuvepVydDFo5NAV8Rz2QjAa1Tf0/view?usp=drive_link

This code need some of the libraries that is mentioned below for proper functioning of the MPU6050
https://drive.google.com/file/d/114cLwQutvjO9R6zlOVaoiTeOI12R-TLe/view?usp=drive_link

In addition to it there is tuning of PID which will be different for different design of the bot that can be set using trial and error method.

- ## APPLICATION AND USES

- **Hotel and restaurant service robots for guest assistance and deliver.**
- **Self-balancing scooters for short-distance commuting.**
- **Hoverboards for personal mobility and recreation.**
- **Spacecraft stabilization and navigation(principle).**

- ## FUTURE ASPECTS AND STEPS

We can enhance it to be automated as well as controlled from specific device.

Some more sensors (e.g. camera, gps) can be used to navigate or have a good communication with robot even in harsh condition.