Data and Metadata. 2026; 5:825
doi: 10.56294/dm2026826

ORIGINAL

Design and implementation of a low-cost, modular two-wheeled balancing vehicle with bluetooth gesture interface

Diseño e implementación de un vehículo equilibrado modular de dos ruedas y bajo coste con interfaz gestual Bluetooth

Peng Zhao[1], Yunkang Chen[1], Yongming Zhang[1]

[1]School of Mechanical and Electrical Engineering, Xinjiang Institute of Technology, Aksu 843100, China.

Cite as: Peng Z, Chen Y, Zhang Y. Design and implementation of a low-cost, modular two-wheeled balancing vehicle with bluetooth gesture interface. Data and Metadata. 2026; 5:826. https://doi.org/10.56294/dm2026826

Submitted: 18-10-2025 **Revised:** 07-01-2026 **Accepted:** 19-02-2026 **Published:** 20-02-2026

Editor: Dr. Adrián Alejandro Vitón Castillo

Corresponding author: Peng Zhao

ABSTRACT

Two-wheeled self-balancing vehicles have attracted considerable attention due to their compact structure and high maneuverability. However, conventional designs typically restrict the payload to the vertical axis in order to maintain mechanical centering, and most remote-control schemes rely on button-based interfaces; research on non-contact, gesture-based control remains insufficient. This study aims to develop a gesture-controlled self-balancing vehicle capable of adapting to lateral load variations. A gesture sensor is employed to recognize nine predefined gestures-forward, backward, left, right, approach target, retreat from target, clockwise rotation, counter-clockwise rotation, and wave. A cascaded PID control architecture combined with an adaptive mechanical-centering compensation algorithm dynamically adjusts the balance set-point according to the position of the counterweight. Attitude data are acquired at 100 Hz via the I^2C interface from the MPU6050's built-in DMP unit, while gesture commands are transmitted over Bluetooth at a baud rate of 9600 baudios. Experimental results demonstrate that, even with a lateral load offset of 3 cm and a mass of 150 g, the system maintains stable equilibrium, with attitude oscillations confined within ±2°, and gesture response latency below 120 ms. The nine predefined gestures successfully enable forward/backward motion, turning in various directions, in-place rotation in both directions, assessment of obstacle proximity, and stopping. The proposed method effectively expands control and load flexibility while preserving stability, providing a feasible reference for research on noncontact gesture-controlled intelligent mobile platforms.

Keywords: Gesture Control; Cascaded PID; Adaptive Mechanical Median; Wireless Transmission.

RESUMEN

Los vehículos autobalanceados de dos ruedas han atraído considerable atención debido a su estructura compacta y maniobrabilidad. Sin embargo, los diseños convencionales suelen restringir la carga útil al eje vertical para preservar la media mecánica, y la mayoría de los esquemas de control remoto dependen de interfaces basadas en botones, mientras que el control gestual no contactado sigue siendo poco explorado. Este estudio tiene como objetivo desarrollar un coche de balanceo controlado por gestos que sea capaz de acomodar variaciones laterales de carga. Se emplea una arquitectura de control PID en cascada con un algoritmo de compensación adaptativa de la media mecánica para ajustar dinámicamente el punto de equilibrio en función de la posición del contrapeso. Los datos de actitud se adquieren de la unidad DMP integrada del MPU6050 a través de I^2C a 100 Hz, y los comandos gestuales se transmiten mediante Bluetooth a 9600 baudios. Los resultados experimentales demuestran que el sistema logra un equilibrio estable con cargas laterales de hasta 150 g desplazadas 3 cm del centro, con oscilaciones de actitud dentro de ±2° y una latencia de respuesta gestual de menos de 120 ms. El vehículo ejecuta con éxito maniobras de avance, retroceso, rotación y detención basadas en cinco gestos predefinidos. El método propuesto extiende efectivamente la flexibilidad de la carga útil mientras mantiene la estabilidad, proporcionando una referencia viable para la investigación de plataformas móviles inteligentes.

Palabras clave: Control Gesto; PID en Cascada; Media Mecánica Adaptativa; Transmisión Inalámbrica.

INTRODUCTION

Balance cars are widely used in logistics distribution, industrial production line automation, smart home services, medical care, and other occasions.

The hardware of the balancing robot primarily includes a microcontroller, drivers, high-precision encoder motors, angle sensors, and an external power supply unit. Wireless transmission modules for master-slave control modes include Bluetooth, WiFi, the 2,4GHz wireless communication module NRF24L01, and infrared remote control. Operational control modes encompass line-following, obstacle avoidance, and wireless remote operation. The sensors include infrared line-following sensors, ultrasonic distance sensors, laser distance sensors, LiDAR, positioning modules, and gesture recognition sensors. For gesture-controlled self-balancing vehicles, selecting appropriate peripherals to achieve balance adjustment based on lateral load changes is crucial. The system design hinges on enabling operation according to gesture variations through non-contact gesture recognition, transmission, and control.

This paper presents a detailed, component-level guide for constructing a two-wheeled self-balancing platform. The primary objective is to document the integration of a specific set of low-cost hardware components (STM32F103, MPU6050, PAJ7602) and the development of a modular firmware architecture implementing cascaded PID control. A secondary objective is to demonstrate the functionality of a basic gesture-based remote control scheme via Bluetooth serial communication, providing a reproducible reference design for educational purposes and prototyping in embedded systems and introductory robotics.

METHOD

Mechanical Structure

The operation of the two-wheeled balancing cart is achieved by adjusting the rotational speeds of the left and right wheels to perform forward, backward, stop, turning in different directions, and rotating in different directions. Under normal conditions, if the mechanical center of gravity remains constant, balancing control can be achieved by adjusting PID parameters. If the mechanical center of gravity shifts, the cart's speed will change to maintain balance. However, if the shift is too large, the cart cannot accelerate fast enough to regain equilibrium due to motor speed limitations, causing it to tip over. (1,2) Higher precision in the motor encoder of the balancing cart allows the controller to capture more pulses per unit time, indicating greater accuracy in detecting motor speed changes. To ensure the controller acquires sufficient motor speed samples, the MPU6050's sampling rate is set to 100Hz. This captures motor pulses every 0,01 seconds, enabling the PID controller to adjust based on deviation and correct the cart's posture.If the encoder precision is insufficient, the MPU6050's sampling frequency must be reduced to obtain more pulse samples. This slows the balancing cart's posture correction cycle, potentially causing minor oscillations during operation. A high-precision motor encoder ensures more stable control of the balancing cart.

The controller reads the nine distinct posture changes from the gesture sensor and converts them into corresponding ASCII codes compatible with the balance cart's recognition protocol. For instance, ASCII codes 1-9 correspond to different gesture variations. Since the Bluetooth module's default baud rate is 9600 and the transmitted data volume is small, the transmission baud rate is set to 9600 to simplify program design.(3,4) Figure 1 shows the situation where the balance car is not weighted, and figure 2 shows the situation where the value of the side weighted mechanical device changes.

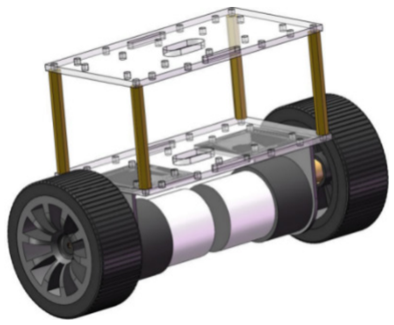


Figure 1. Balance car chassis without added weight

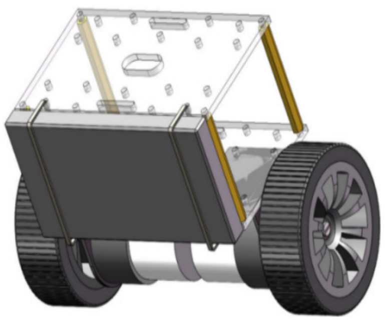


Figure 2. Balance car chassis with counterweight applied

Composition of Balance Car Control Circuit

With the main controller as the core, the entire system is generally powered by lithium batteries, while other peripheral circuits are powered by a regulated output of 5V or 3,3V. Due to the limited output current of the controller, a DC motor is driven by a driver; Motors are generally gear-reduced, and can adapt to climbing functions without involving PID speed regulation; MPU6050 is used to measure the three-axis acceleration and angular velocity data. The main controller combines PID control algorithm with data collected by MPU6050, current left and right motor speed and steering to achieve vertical and speed control of the balance car; OLED displays measurement parameters such as roll angle, pitch angle, and motor speed for the purpose of PID parameter tuning; The Bluetooth module can control its speed and steering through the host or mobile phone; Voltage monitoring enables real-time detection of lithium battery voltage. (5) The specific composition of the balance car control circuit is shown in figure 3.

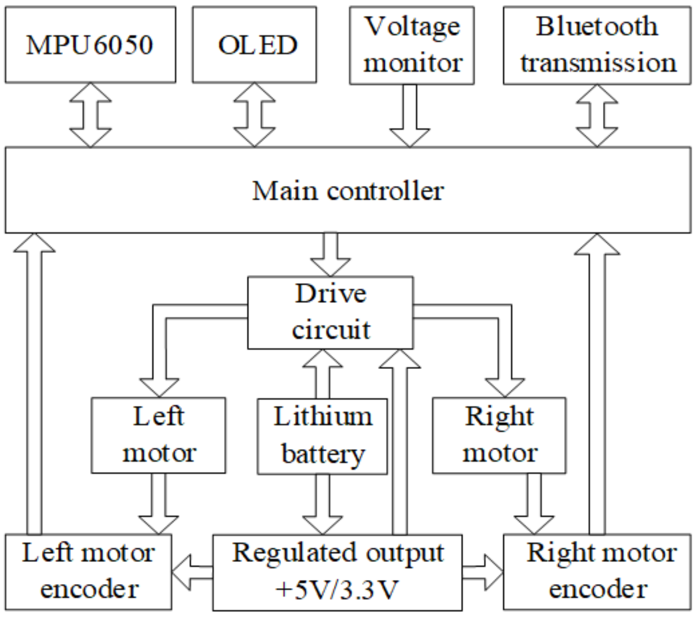


Figure 3. Structure diagram of the balance car system

Data transmission between master and slave machines

The host control system monitors the operational parameters of the balancing cart control system and enables remote control of its travel route. It employs gesture control as its operating method. The main controller receives gesture transformation data captured by the PAJ7602 gesture sensor, parses it into corresponding ASCII codes ranging from 1 to 9, and transmits this data via Bluetooth to the balancing cart controller. The balancing cart controller then adjusts the speed difference between the left and right wheels to execute the corresponding control actions. Generally, for the convenience of program porting and compatibility with wireless transmission protocols, the balanced car controller and wireless transmission module are of the same model as the main control system.(6) The specific structure of the master-slave control system is shown in figure 4.

If the master and slave Bluetooth are not connected, a wireless terminal with the same Bluetooth transmission protocol can be used to connect with the slave and control the balance car using the wireless terminal. If using a mobile app to search for slave Bluetooth, after successful connection, it can communicate with the balance car controller through the mobile app; You can also use a PC to search for Bluetooth from the slave, and after connection, communicate with the balance car controller through upper computer software such as serial debugging assistant. (7,8) The structure of the commonly used wireless terminal to control the balance car is shown in figure 5.

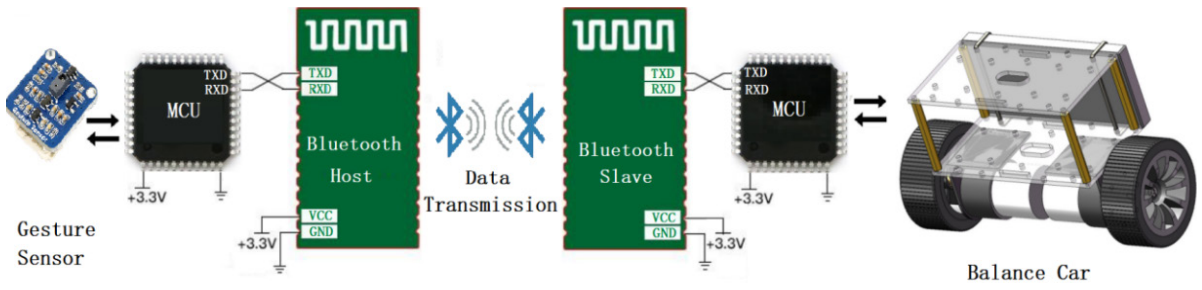


Figure 4. Gesture control with counterweight balance car master-slave data transmission

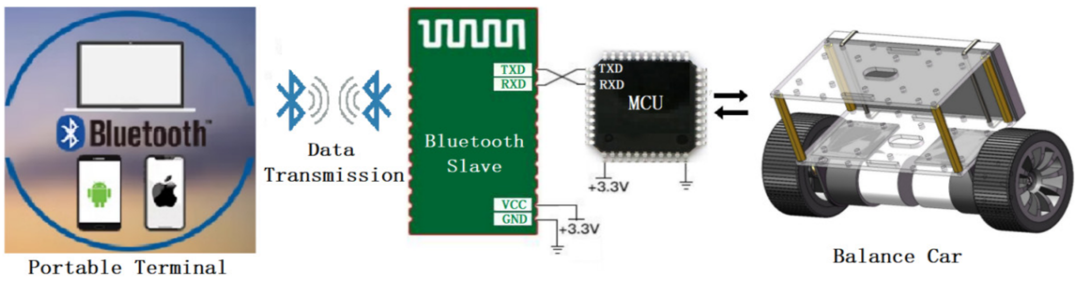


Figure 5. Wireless terminal control with counterweight balance car master-slave data transmission

Selection of master-slave hardware

Selection of Controller

Based on the structure and control requirements of intelligent vehicles, controllers equipped with encoder input and PWM output are commonly used, such as the STM32 series. For controllers with PWM output but no encoder input, such as the MSP430F5529 and STC8 series, external interrupts are used to self add or subtract the encoder output pulses based on the encoder output. The motor speed is determined based on the absolute value of the number of encoder pulses within a fixed time, and its direction is determined based on the positive or negative number of encoder pulses; This design selects the STM32F103 series controller with encoder input and PWM output as the master and slave devices. The controller is a 32-bit MCU based on ARM core and is a standard Cortex-M3 core with ARMv7 architecture. Its internal clock is multiplied by a phase-locked loop output, with a frequency of up to 72MHz, which can achieve real-time control of the balance car.(9,10)

Selection of motors and encoders

Common encoders include Hall encoders, photoelectric encoders, and giant magnetoresistance (GMR) encoders. The Hall encoder uses the Hall effect to detect changes in magnetic poles, it consistis of a Hall sensor and a magnetic encoder disk. The encoder disk is equipped with magnetic poles, and as the disk rotates, the position of the magnetic poles changes. By detecting these changes, information such as the position and speed of the rotor can be obtained; A photoelectric encoder is a sensor that uses the principle of grating diffraction to realize displacement to digital conversion. Through photoelectric conversion, the mechanical geometric displacement on the output shaft is converted into a pulse digital quantity; The GMR encoder is a position sensor based on the giant magnetoresistance effect,(11,12) which uses a series of giant magnetoresistance elements to measure the position of an object. The GMR encoder has the characteristics of high resolution, high accuracy, and fast response, and is therefore widely used in precision positioning systems and industrial automation. The number of encoder lines refers to the number of pulse signals output by the encoder per revolution, usually measured in pulses per revolution (PPR), which reflects the resolution and measurement accuracy of the encoder. The enconter should be selected appropriately based on the controller's controll preformance and the requirements of the controlled object. To achieve precise control of the balance car, this design selects a 370 reduction motor as the motor, with a reduction ratio of 1:30. Using a 330 wire photoelectric encoder, the number of pulses generated by one rotation of the motor is as follows:

$$330 \times 30 = 9900\text{PPR} \qquad (1)$$

RESULTS

Balance car control circuit design

The two wheeled balance car is powered by an external lithium-ion battery. For example, the nominal voltage of a single 18650 lithium battery is 3,7V, and the voltage can reach 4,2V when fully charged. If three lithium batteries are connected in series, the voltage can reach 12,6V. A three terminal voltage regulator is used to convert the lithium battery voltage to 5V,which supplies power to the STM32F103C8T6 core board OLED, Encoder and driver power supply; The core board contains a voltage regulator that converts+5V to 3,3V to power the angle sensor MPU6050. Excessive discharge of lithium-ion batteries can cause internal structural damage, capacity degradation, and safety hazards. Therefore, a lithium-ion battery voltage detection circuit is used to monitor the voltage. Since the internal ADC of the STM32F103C8T6 has a voltage input range of 0-3,3V, resistors R2 and R3 are connected in series for voltage division this ensures the lithium-ion battery's detection volltage does not exceed the ADC's full range voltage . Using channels 3 and 4 of timer T2 of STM32F103C8T6 to generate two PWM signals respectively to achieve speed regulation of the left and right motors; Capture the output pulses of phase A and B of the right motor encoder using channels 3 and 4 of timer T3; Capture the output pulses of phase A and B of the left motor encoder using channels 2 and 1 of timer T4; Using the I/O of the controller to simulate the transmission characteristics of the I^{2}C bus, realize the control of I^{2}C type OLED and the reading of angle sensor MPU6050 data; By configuring the MPU6050 to trigger interrupts at fixed intervals, the controller adjusts the motor speed in conjunction with the measured values of the angle sensor in the interrupt service function to correct the posture of the balance car. The Bluetooth module is connected to serial port 3 of the controller, receiving control signals sent by the host and providing feedback on the operating parameters of the balance car.(13,14) The specific hardware circuit is shown in figure 6.

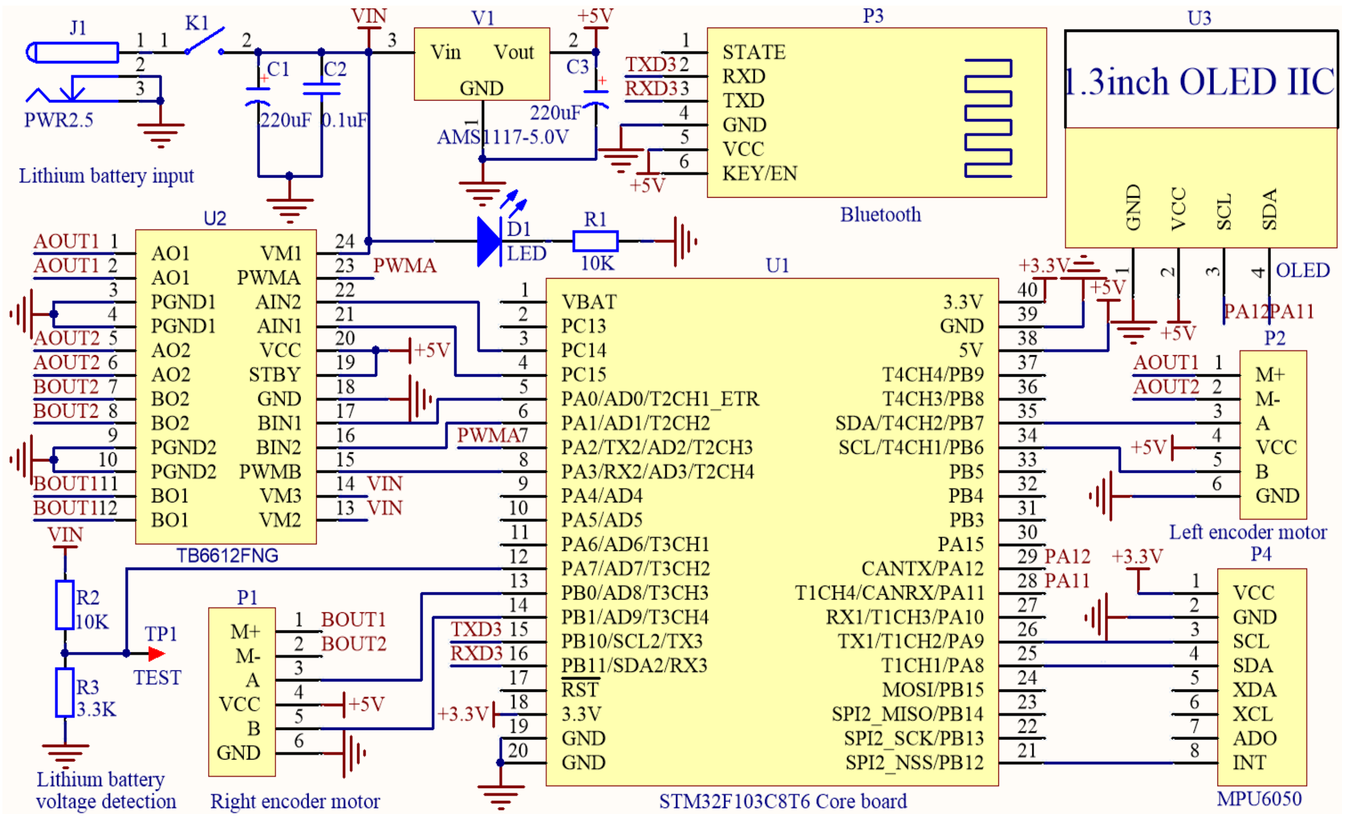


Figure 6. Balance car control circuit

Control Logic Analysis of TB6612

TB6612 adopts a MOSFET-H bridge structure and dual channel circuit output, which can drive two motors simultaneously. Compared to the heat consumption of L298N, it does not require an external heat sink, has a simple peripheral circuit, and only requires an external power filter capacitor to directly drive the motor, which is beneficial for reducing system size. The corresponding PWM signal input frequency range can reach 100KHz.(15) The control logic of the left and right motors is the same, and the corresponding control logic is shown in table 1.

Table 1. TB6612 Control Logic

Input				Output		
IN1	**IN2**	**PWM**	**STBY**	**AO1**	**AO2**	**Work mode**
H	H	H/L	H	L	L	Brake
L	H	H	H	L	H	Reverse rotation
L	H	L	H	L	L	Brake
H	L	H	H	H	L	Forward rotation
H	L	L	H	L	L	Brake
L	L	H	H	OFF		Shutdown
H/L	H/L	H/L	L	OFF		Standby

Orthogonal Encoder Control Logic Analysis

The A and B phases in the left and right motor interfaces P2 and P1 are the output terminals of an orthogonal encoder, which is a sensor used to measure rotational or linear displacement. Its working principle is based on the variation of two pulse signals with a phase difference of 90 degrees. If the outputs of encoder A and B are orthogonal signals with a phase difference of 90 degrees, the forward and reverse rotation of the motor can be determined based on whether the output pulses of A and B are ahead or behind by 90 degrees. The two forms of output pulses from orthogonal encoders A and B are shown in figure 7 and figure 8, respectively.

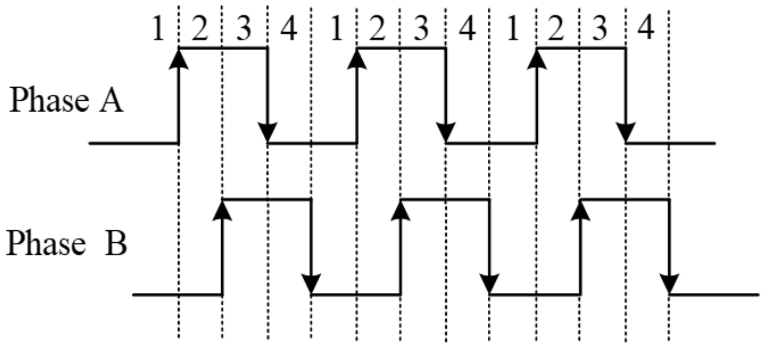


Figure 7. Output waveform of encoder when the motor runs clockwise

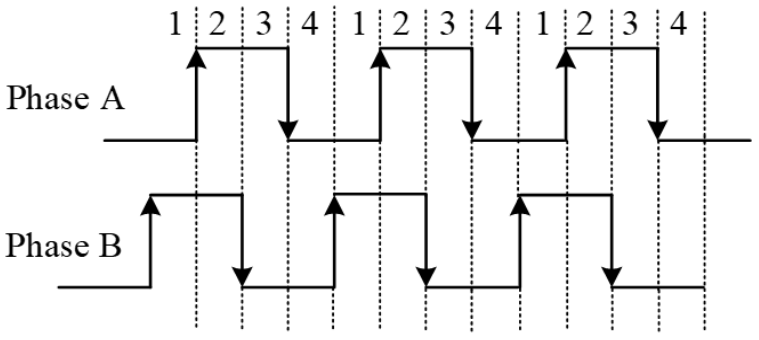


Figure 8. Output waveform of encoder when the motor runs counterclockwise

The relationship between motor direction and phases A and B is shown in table 2.

Table 2. Relationship between Motor Steering and Encoder Output

Forward rotation A phase B phase	Reverse rotation A phase B phase
L L	L H
H L	H H
H H	H L
L H	L L

From the analysis in the table above, it can be concluded that the motor speed can be determined by calculating the number of pulses per unit time for one phase of phase A orB output by the encoder;[16,17] Based on the phase difference between encoder A and phase B, the direction of the motor can be determined. When phase A leads phase B by 90°, the motor moves clockwise. When phase A lags phase B by 90°, the motor moves counterclockwise.

Host Hardware Design

The main unit of the two wheeled balance car adopts an external DC 5V power supply. In order to facilitate program porting, OLED、 The Bluetooth module and the hardware circuit of the balance car are the same; The I/O analog I²C bus of STM32F103C8T6 communicates with the PAJ7602 gesture sensor to recognize nine gesture recognition modes: up, down, left, right, front, back, clockwise rotation, counterclockwise rotation, and swing. The control signal is transmitted to the balance car control circuit through Bluetooth to achieve data exchange between the main machine and the balance car. The specific hardware circuit of the host is shown in figure 9.

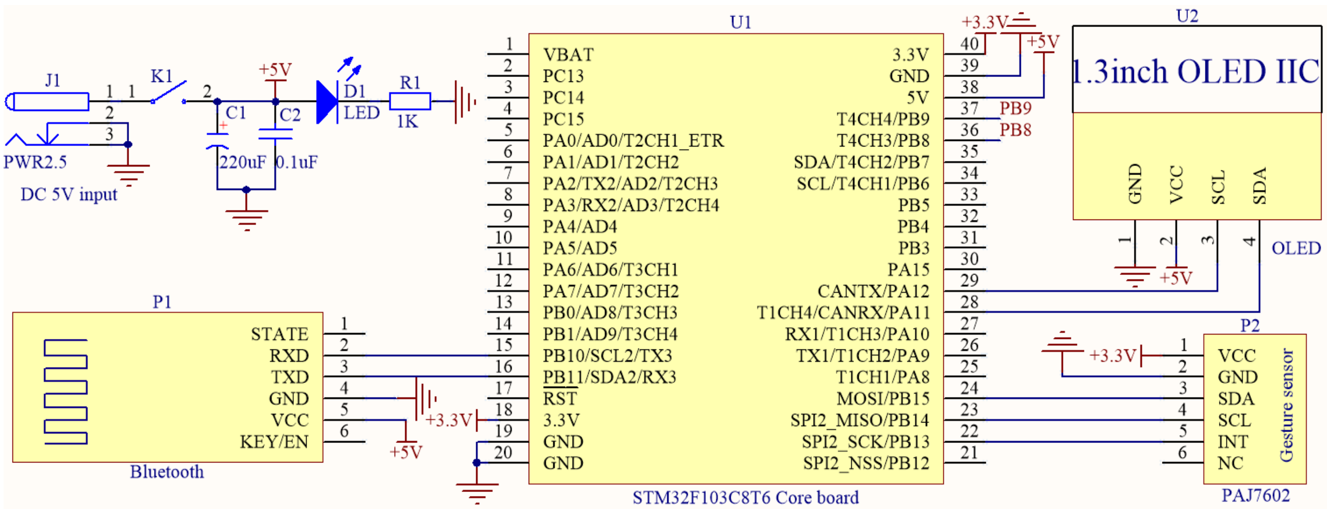


Figure 9. Host hardware circuit

Balance car software structure

Developed under the Standard Peripheral Library with STM32 as the controller, the software control system for the two wheeled balance car involves a wide range of sensors, control algorithms, display terminals, etc. In order to facilitate program porting and expansion, a modular design scheme of calling subroutines from the main program is adopted. The specific software structure is shown in figure 10.

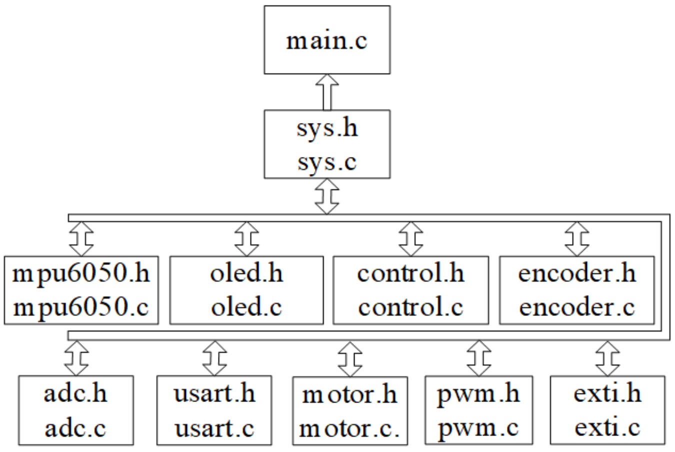
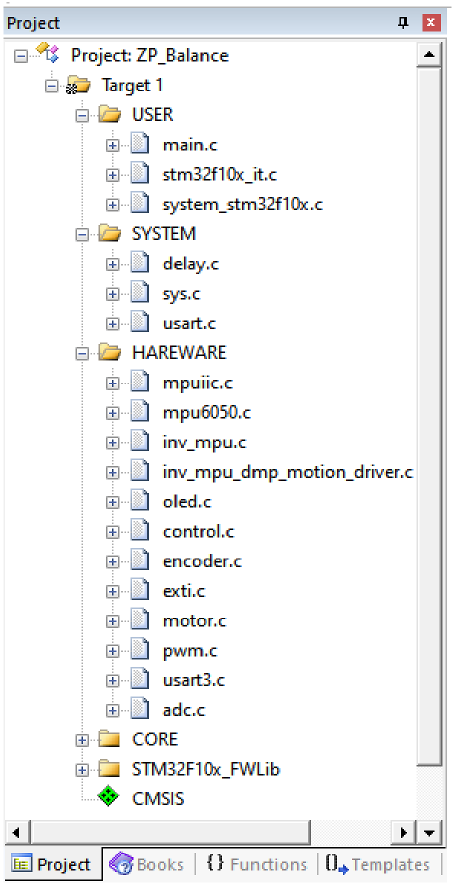


Figure 10. Software Architecture Diagram

Each subroutine file has independent header files and source code files. The header file mainly implements the declaration of variables or functions, and all header files have preprocessing instructions “# ifndef... # define... # endif” to prevent duplicate inclusion of header files. The source code file mainly implements the definition of functions, variable definitions, etc; The system header file sys. h contains all the header files used in the system, used to declare variables such as three-axis acceleration, three-axis angular velocity, left and right motor encoder values of MPU6050 used in each subfile, or to declare system initialization functions. When called by multiple subfiles, it is declared as a global variable or function. The system source code file sys. c is used to define system initialization functions, such as configuring serial port interrupts in Bluetooth transmission; The main function source code file main.c serves as the program's entry point. The main function contains the system header file, defines variables from the system header file, calls the system initialization function, and displays relevant parameters such as balance car angle, voltage monitoring value, etc. on the OLED; The source code file mpu6050. c for the attitude sensor is used to calculate the current acceleration and angular velocity. It can also be set to generate low pulses per unit time to trigger the controller's interrupt, which is used for real-time monitoring of the intelligent vehicle's operating status; The OLED source code file oled. c simulates the interaction between I^2C and OLED using a controller to display the angle and motor parameters of MPU6050; The encoder source code file encoder. c initializes the speed and direction of the left and right motors, where the direction is determined based on the polarity of the encoded data; The interrupt source code file exti.c configures how the MPU6050 triggers interrupts and collects intelligent vehicles's operating data within a unit time; The control source code file 'control. c' is the core of the entire control system, which is essentially an interrupt service program that interrupts the source code file 'exti. c'. It reads encoder data, Bluetooth transmission module data, MPU6050 data, sets the speed and direction of the balance car, implements PID control algorithms, PWM limiting, and provides protection against falls or undervoltages in the balance car; The PWM control source code file pwm. c is used to initialize the PWM signal within a fixed period; The motor control source code file motor.c is used to load the left and right motor steering settings and PWM speed control data, as well as define the anti fall function for the balance car. The A/D conversion source code file adc.c is used to monitor the voltage of lithium-ion batteries.[18] In the Keil compilation environment, the composition of the entire project is shown in figure 11.



```
Project
Project: ZP_Balance
  Target 1
    USER
      main.c
      stm32f10x_it.c
      system_stm32f10x.c
    SYSTEM
      delay.c
      sys.c
      usart.c
    HAREWARE
      mpuiic.c
      mpu6050.c
      inv_mpu.c
      inv_mpu_dmp_motion_driver.c
      oled.c
      control.c
      encoder.c
      exti.c
      motor.c
      pwm.c
      usart3.c
      adc.c
    CORE
    STM32F10x_FWLib
    CMSIS
```

Figure 11. Engineering structure of balancing car under Keil compilation software

Main Program Design for Balance Car

The main program is the entry point of the program. To achieve the balance of the balance car, the main control STM32F103C8T6 needs to calibrate the posture of the balance car at regular intervals. If the calibration frequency is too low, it may affect the stability of the balance car. If the controller processing speed is limited, it may also cause the balance car to lose control. This design utilizes the angle sensor MPU6050 to trigger an interrupt every 10ms, with the priority of the interrupt set to the highest, to ensure the normal operation of the upright car. In the interrupt service program, the control of the balance car is achieved by adjusting the speed of the left and right motors.(19) The specific flowchart is shown in figure 12.

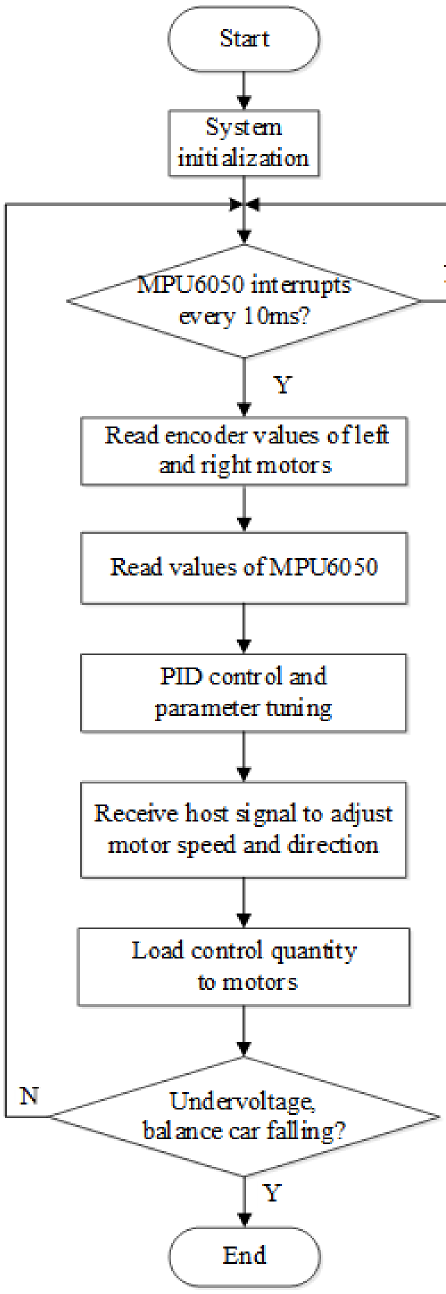

Start
System initialization
MPU6050 interrupts every 10ms?
N
OLED displays current parameters
Y
Read encoder values of left and right motors
Read values of MPU6050
PID control and parameter tuning
Receive host signal to adjust motor speed and direction
Load control quantity to motors
N
Undervoltage, balance car falling?
Y
End

Figure 12. Main program flowchart

MPU6050 attitude calculation

The MPU6050 quaternion attitude solution achieves three-dimensional attitude estimation by fusing gyroscope and accelerometer data. The quaternion differential equation based on the gyroscope angular velocity ω is:

$$\begin{bmatrix} \dot{q}_0 \\ \dot{q}_1 \\ \dot{q}2 \\ \dot{q}3 \end{bmatrix} = \frac{1}{2} \begin{bmatrix} 0 & -\omega_x & -\omega_y & -\omega_z \\ \omega_x & 0 & \omega_z & -\omega_y \\ \omega_y & -\omega_z & 0 & \omega_x \\ \omega_z & \omega_y & -\omega_x & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ q_2 \\ q_3 \end{bmatrix} \tag{2}$$

This equation is updated with quaternions using the first-order Runge Kutta method or equivalent rotation vector method. The MPU6050 has a built-in digital motion processor (DMP) that can directly read quaternions through the I^2C interface without the need for software to solve differential equations, significantly reducing the computational burden on the controller.(20,21) The controller converts quaternions into Euler angles through a program for intuitive description. Euler's angle describes the rotation of an object in three-dimensional space using three angles. Usually, these three angles are pitch, yaw, and roll. They represent rotations around three different axes. Pitch angle: Rotate around the horizontal axis; Yaw angle: rotation around the vertical axis; Roll angle: Rotate around the longitudinal axis. The pitch angle and yaw angle are related to the installation angle of MPU6050, as shown in figure 13.

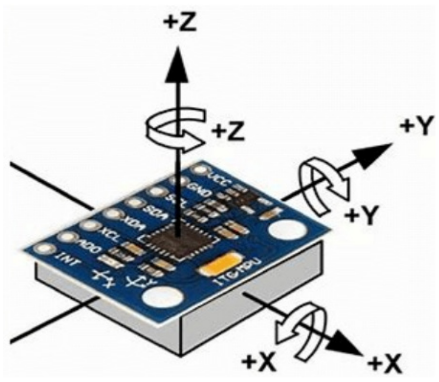


Figure 13. MPU6050 attitude sensor

The accelerometer data of MPU6050 contains Gaussian white noise and random interference, especially in dynamic environments where noise is significant. Although the gyroscope has high short-term accuracy, there is accumulated drift error over time, resulting in inaccurate long-term attitude estimation. A first-order complementary filter is used to fuse accelerometer and gyroscope data for attitude calculation, which is a lightweight algorithm that balances noise suppression and dynamic response by combining the complementary characteristics of the two types of sensors.(22,23) The specific first-order complementary filtering formula is as follows:

$$angle=\alpha^* (angle\_prev+gyro^*\Delta t)+(1-\alpha)^*accel\_angle \qquad (3)$$

Among them, α is the filtering coefficient, with a value range of 0-1. In order to balance stability and response speed, this design sets α to 0,9; Gyro * Δ t is the incremental integral angle of the gyroscope; Accel-angle is the instantaneous angle calculated by the accelerometer.

The yaw angle Yaw rotates around the Z-axis, and since there is no reference, it is impossible to determine the absolute angle of the current Yaw angle. Only the change in Yaw, which is the Z-axis angular velocity, can be obtained. The current Yaw angle can also be calculated by integrating the Z-axis angular velocity, but due to measurement accuracy issues, the calculated value may drift and become completely meaningless after a period of time. The MPU6050 module includes a biult-in temperature sensor, which monitors the chip's operating temperature in real time. By measuring temperature data, users can understand the working status of the sensor and ensure that it operates within the appropriate temperature range. The specific display effect of MPU6050 is shown in figure 14.

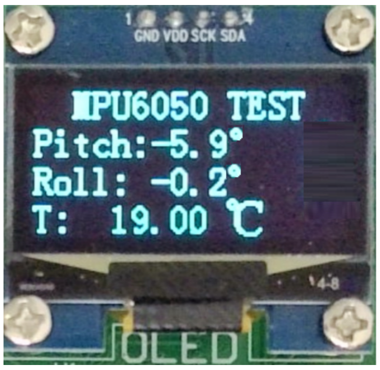


Figure 14. MPU6050 Attitude Calculation Display Interface

Analysis of Control Algorithms

The mathematical expression for PID control law is:

$$U(t) = P(e(t) + \frac{1}{I}\int_0^t e(t)dt + D\frac{de(t)}{dt}) \tag{4}$$

After discretization, the PID control expression is obtained as follows:

$$U(k) = P(e(k) + \frac{T}{I}\sum_{i=0}^{k} e(i) + D\frac{e(k)-e(k-1)}{T}) \tag{5}$$

The above equation is called a total or positional PID. If it is motion control and PWM is used to adjust the motor speed, the position based PID algorithm formula can also be expressed as:

$$P\text{wm} = K_p * e(k) + K_i * e(k) + K_d * [e(k) - e(k-1)] \tag{6}$$

For the balance car, the design concept of vertical closed-loop as the main approach, supplemented by speed closed-loop, adopts a cascade PID control scheme, and its control system is shown in figure 15.

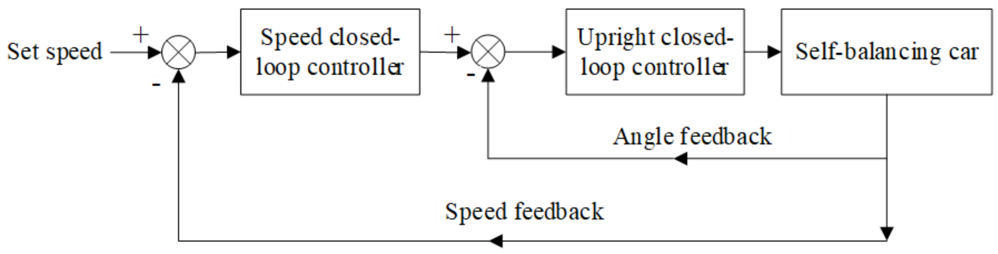


Figure 15. PID control structure diagram of balance car

The speed loop adopts PI control, and its corresponding output is:

Velocity_out =Kp2*(Encoder_real-Encoder_expect)+Ki*∫(Encoder_real- Encoder_expect) (7)

Among them, Encoder_real refers to the number of pulses captured by the left and right motor encoders within a 10ms period, without the need to divide by 2 to obtain the average value, which can reduce rounding errors; Encoder_inspect is the set target speed; Kp2 is the proportional coefficient of the speed loop; Ki integral coefficient.

The upright ring adopts PD control, and its corresponding output is:

Vertical_out=Kp1*(Real_Angle-Expect_Angle)+Kd*D(Real_Angle-Expect_Angle) (8)

Among them, Expect_Angle=Velocity_out, The output of the speed loop is used as the input of the upright loop to calculate the output of the upright loop.

For a balance car with side counterweights, due to its mechanical median not being 0, to achieve an upright position, the output of the upright ring is:

Vertical_out=Kp1*(Real_Angle-Expect_Angle-Mechanical_Med)+Kd*D(Real_Angle-Expect_Angle) (9)

Since Expect_Sngle=Velocity_out, the above equation can be simplified as:

Vertical_out=Kp1*(Real_Angle-Velocity_out-Mechanical_Med)+Kd*D(Real_Angle-Expect_Angle) (10)

Equation 10 is the output of the cascaded PID, and Vertical_out is the corresponding pulse width modulation signal (PWM) of the motor; Real_Sngle is the actual angle obtained by the attitude sensor MPU6050; Velocity_out is the output of the velocity loop; Mechanical_Sed is the mechanical median value; Expect_Sngle is the target angle, ideally 0.(24,25)

Measurement of Mechanical Median

Manually tilt the car forward to the critical point where it is about to fall, and record the angle at this time as the forward tilt critical angle; The critical point of backward tilt to the other direction is the critical angle of backward tilt. The average of the two measured angles is the mechanical median, and the specific formula is as follows:

$$\text{Mechanical median} = (\text{critical forward tilt angle}+\text{critical backward tilt angle})/2 \quad (11)$$

Implementation of Gesture Control

The host controller simulates the I²C bus transmission protocol, reads the nine gestures detected by the attitude sensor, including up, down, left, right, front, back, clockwise rotation, counterclockwise rotation, and swing, and converts the nine gestures into corresponding ASCII codes, such as' 1 'and' 2 ' ... '9' or 'A', 'B'... 'I', etc., are transmitted to the balance car controller through Bluetooth, and the balance car controller controls the steering of the balance car according to the corresponding ASCII code. Steering control is the implementation of balancing the speed difference between the left and right motors of the car; Forward and backward are the target values set for balancing the speed of the car, with positive values indicating forward and negative values indicating backward. The larger the absolute value, the faster the forward or backward movement.

System simulation

The control algorithm is the core of the balance car; MATLAB is used to simulate the avaiable control algorithms for two-wheeled balance cars, enabling parameter adjustment. Figure 16 below shows a Matlab Simulink simulation model: it constructs a three-dimensional model of a two wheeled self-balancing car, based on PID control and manual tuning.(26,27)

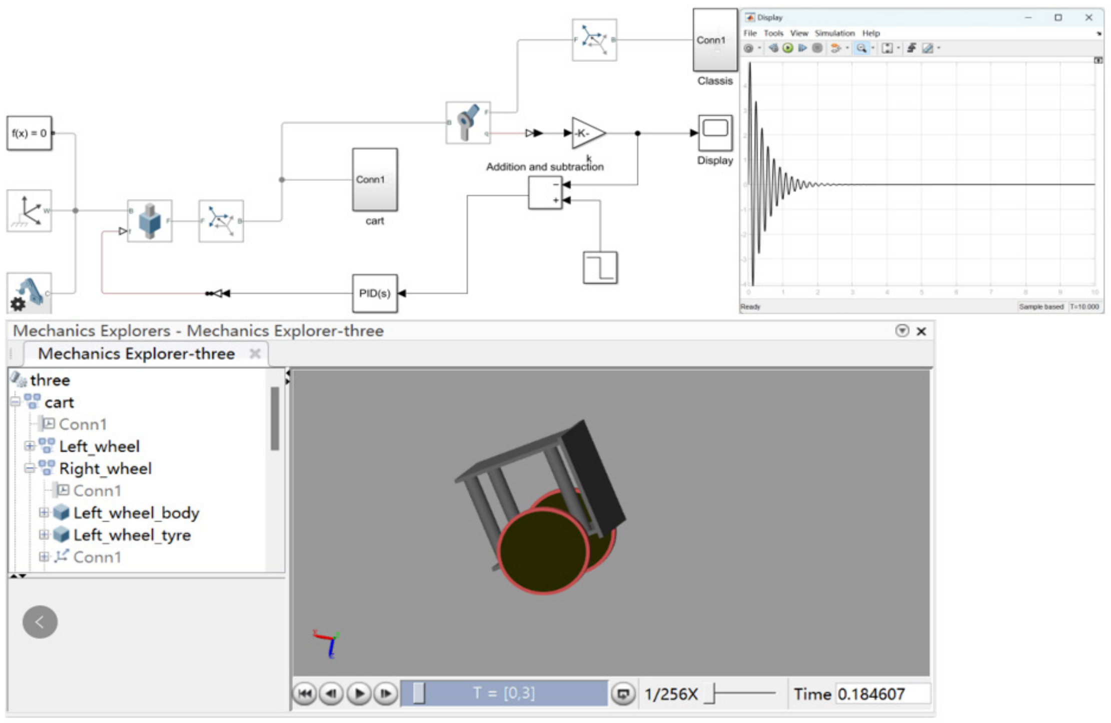


Figure 16. Simulink block diagram and control effect

System commissioning

Orthogonal Encoder Test

Using a dual channel oscilloscope to observe the output of the photoelectric encoder during motor operation, the phase relationship and direction of phases A and B are consistent. The specific waveform is shown in figure 17.

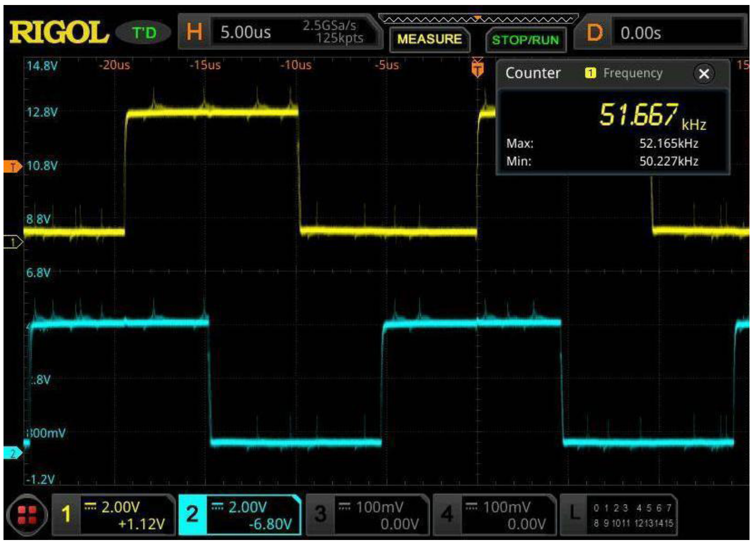


Figure 17. Output diagram of motor orthogonal encoder

Balance car PID tuning

PID tuning of vertical loop

The balance car uses position-based PID, and the upright loop adopts PD control. The positive and negative values of the upright loop's proportional coefficient Kp are determined by the following method. The PID coefficients of other links are set to 0. When the balance car falls to one side, the left and right wheels accelerate to the side where it falls, ensuring that the car has a tendency to stand upright. This indicates that the positive and negative values of the upright ring Kp are set correctly. When the positive and negative values remain constant, the absolute value of Kp continues to increase until a significant low-frequency oscillation occurs. The positive and negative judgment of the differential coefficient Kd of the upright loop is carried out by setting the PID coefficients of other links to 0. When the car rotates around the motor axis and the wheels rotate in the opposite direction without following, the polarity is incorrect. Kd should be taken in the opposite direction. With the positive and negative values unchanged, a proportional coefficient Kp that causes low-frequency oscillation is introduced and the holding value remains unchanged. The absolute value of Kd continues to increase until high-frequency oscillation occurs.(28,29) Based to engineering experience, the ideal proportional and differential coefficients for balancing the car are obtained by multiplying the oscillation-including values by 0,6.

PID tuning of speed loop

The speed loop adopts PI control, and the upright loop should be cancelled before debugging the speed loop. The positive or negative judgment of the proportional coefficient Kp of the speed loop is exactly opposite to the polarity judgment of the proportional coefficient of the three wheel or four-wheel drive intelligent vehicle. Set the target speed to 0, use external force to rotate one of the driving wheels, and the driving wheel will accelerate in the same direction as the other driving wheel until the maximum speed of the motor is reached. At this point, the polarity of the proportional coefficient Kp of the speed loop is correct. Because the upright position of the balance car is compensated by speed, when tilting to one side, the motor must accelerate towards the tilting direction to make its center of gravity perpendicular to the running plane.(30,31,32) The integration coefficient Ki can be obtained by Kp × 0,005.

Table 3. Control parameters of balance car with counterweight

Upright loop		Speed loop		Mechanical median
K_p	540,0	K_p	-0,0092	14,5°
K_d	1,2	K_i	-0,000046	

Balance car anti fall test

When the inclination angle of the balance car is too large, the speed of the motor is limited, and it is impossible to achieve system balance by increasing the speed. The balance car will tilt. When the inclination angle reaches 30°, in order to prevent the wheels from continuing to rotate after the car falls, the target speed is set to 0 to stop the balance car. The balance car completed the stop after falling according to the set requirements.

Gesture Control and Mobile APP Control Testing

Set the Bluetooth module of the gesture control circuit to master mode and the Bluetooth module of the balance car control system to slave mode, allowing for gesture control between the master and slave devices. When disconnecting the Bluetooth connection between the master and slave devices, use the mobile app to search for and connect to the Bluetooth of the slave device. Send control codes corresponding to gestures through the mobile app to control the balanced car with weights. The display effect of gesture control and mobile app control is shown in figure 18.

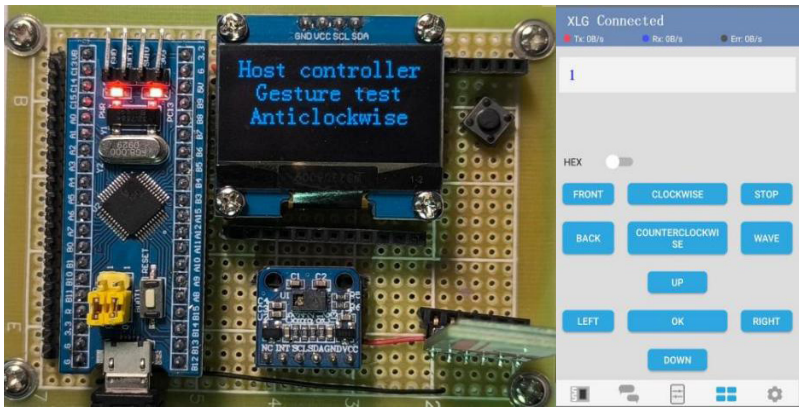


Figure18. Gesture Control and Mobile APP Control Display Effect

Balance car test with counterweight

According to the counterweight, calculate the mechanical median, set the corresponding mechanical median for the balance car, receive gesture signals transmitted by the host or mobile app, adjust the speed of the left and right motors, and achieve the normal operation of the balance car with counterweight controlled by gestures or mobile app. The specific operating status is shown in figure 19.

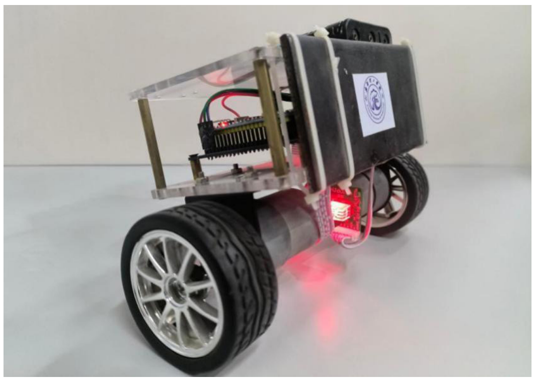


Figure 19. Operation effect of balanced trolley with counterweight

Performance Analysis

Based on multiple measurements, the corresponding gesture response latency, attitude oscillation amplitude, and Bluetooth transmission range in open areas are shown in the following table.

Table 4. Balance car performance test

Number of measurements	Average gesture response latency	Attitude oscillation amplitude	Bluetooth 5.0 transmission range
10	113ms	±1,61°	15,2m
20	108ms	±1,57°	15,3m
30	106ms	±1,54°	15,4m
40	105ms	±1,51°	15,2m

CONCLUSION

This work demonstrates the feasibility of a gesture-controlled balance car with counterweight through integrated software-hardware co-design and physical validation. The adaptive mechanical median method proves effective because it directly compensates for the center-of-gravity offset introduced by the counterweight mechanism, allowing the PID controller to operate around a dynamically calibrated equilibrium point rather than a theoretical zero. By leveraging the on-chip DMP of the MPU6050 to output quaternions via I^2C, the computational burden of attitude estimation is offloaded from the main controller, enabling deterministic real-time control within the interrupt service routine. The interrupt-driven architecture, synchronized with the MPU6050's data-ready signal, ensures consistent sampling intervals critical for stable closed-loop performance.

Compared to ROS-based balance car implementations that require substantial computational resources, complex middleware stacks, and significant development overhead, the proposed system achieves comparable functionality with a single low-cost microcontroller. Unlike LQR or model-predictive control approaches that demand accurate system identification and state-space modeling, the empirically tuned cascaded PID structure offers a pragmatic balance between performance and implementation complexity. The Bluetooth transparent transmission mode provides a cost-effective wireless link without the protocol overhead of WiFi or the power consumption of continuous RF communication, making this approach particularly suitable for educational platforms and cost-sensitive embedded applications.

Several limitations warrant acknowledgment. First, the gesture recognition vocabulary remains constrained, and the system may exhibit sensitivity to ambient lighting variations or electromagnetic interference affecting the gesture sensor. Second, the Bluetooth communication range imposes operational constraints in larger environments. Third, the determination of the mechanical median angle (θ_m) relies on empirical measurement rather than autonomous calibration, and PID parameters were manually tuned without formal optimization procedures. Finally, all validation was conducted under controlled laboratory conditions; the system's robustness to floor surface irregularities, external disturbances, or mechanical shocks remains uncharacterized.

Future research directions include: (1) implementing an automatic mechanical median calibration algorithm using iterative convergence during startup; (2) expanding the gesture command set through machine learning-based gesture classification; (3) integrating adaptive PID or fuzzy logic control to improve disturbance rejection; (4) conducting outdoor field tests to evaluate performance under real-world surface conditions; and (5) exploring low-power wireless protocols such as BLE 5.0 to extend communication range while maintaining energy efficiency.

REFERENCES

1. Tian L, Hong G, Li W. Fuzzy self-adaptive PID controller in profile bending machine with the application of feed mechanism. Hydraul Pneum Seals. 2011;31:10-13. doi:10.3969/j.issn.1008-0813.2011.01.004.

2. Altinkaya H, Khamees A, Yaghoubi E, Yaghoubi E. Control of heat exchanger liquid temperature using PID control and first order sliding mode control system. In: Proceedings of the 2024 International Conference on Electrical, Computer and Energy Technologies (ICECET); 2024 Jul 3-6; Paris, France. p. 1-6. doi:10.1109/ICECET61485.2024.10698052.

3. Eltag K, Zhang B. Design robust self-tuning FPIDF controller for AVR system. Int J Control Autom Syst. 2021;19:910-920. doi:10.1007/s12555-019-1071-8.

4. Chakraborty S, Mondal A, Das C. Fuzzy fractional order PID controller design for AVR system. In: Proceedings of the 2024 3rd International Conference on Control, Instrumentation, Energy and Communication (CIEC); 2024 May 2-4; Kolkata, India. doi:10.1109/CIEC59440.2024.10468513.

5. Zhao P, Zhang Y, Jiang LL. Research on the design scheme and control strategy of smart car with multi-sensor fusion. Ind Instrum Autom. 2024;(06):81-87. doi:10.19950/j.cnki.CN61-1121/TH.2024.06.015.

6. Yu JT, Sun YT, Zhu YM, et al. Design of indoor navigation on method based on fusion of LiDAR and vision. Transducer Microsyst Technol. 2023;42(8):69-72. doi:10.16526/j.cnki.11—4762/tp.2023.08.022.

7. Ekinci S, Hekimoğlu B, Kaya S. Tuning of PID controller for AVR system using salp swarm algorithm. In: 2018 International Conference on Artificial Intelligence and Data Processing. IEEE; 2018. p. 1-6. doi:10.1109/IDAP.2018.8620809.

8. Chu KB, Guo JJ. Tracking algorithm of intelligent vehicle movement trajectory. J Electron Meas Instrum. 2020;34(6):131-135. doi:10.16526/j.cnki.11-4762/tp.2023.08.022.

9. Zhan HB, Wang J, Fan HR. Design of autonomous tracking intelligent vehicle based on electromagnetic induction. Automation Instrumentation. 2023;38(10):16-19,23. doi:10.16526/j.cnki.11-4762/tp.2023.08.022.

10. Dogruer T, Can MS. Desgn and robustness analysis of fuzzy PID controller for automatic voltage regulator system using genetic algorithm. Trans Inst Meas Control. 2022;44:1862-1873. doi:10.1177/01423312211066758.

11. Yi YL. Design of robot remote control system based on sensor fusion technology. Comput Meas Control. 2023;31(8):142-147,182. doi:10.16526/j.cnki.11—4762/tp.2023.08.022.

12. Li L, Xiao SD, Li XK, et al. Design of intelligent vehicle positioning and navigation system based on multi-sensor fusion. Chin J Eng Des. 2019;26(2):182-189. doi:10.3785/j.issn.1006-754X.2019.02.009.

13. Wang WS, Li SJ. Design of smart car based on monocular vision sensor obstacle avoidance. Transducer Microsyst Technol. 2023;42(4):119-122. doi:10.16526/j.cnki.11-4762/tp.2023.08.022.

14. Lu YN, Lu X, Fang FC. Design and research of intelligence vehicle based on machine vision. Mod Electron Tech. 2022;45(2):177-182. doi:10.16652/j.issn.1004-373x.2022.02.034.

15. Xu N, Yan JM, Xing YH, et al. Design of self-calibration system for sensor position offset based on micro vibration. Chin J Sens Actuators. 2022;35(9):1174-1181. doi:10.3969/j.issn.1004-1699.2022.09.003.

16. Li F, Li Y, Yan C, et al. Swing speed control strategy of fuzzy PID roadheader based on PSO-BP algorithm. In: 2022 IEEE 6th Information Technology and Mechatronics Engineering Conference (ITOEC). IEEE; 2022. p. 827-833. doi:10.1109/ITOEC53115.2022.9734429.

17. Wang R, Qin JM. Fuzzy PID control method of air conditioning supply air temperature based on multi-sensing features fusion. Chin J Sens Actuators. 2023;36(6):943-948. doi:10.3969/j.issn.1004-1699.2023.03.014.

18. Shen YJ, Xing HY, Wang SZ. UAV flight control system of cascade fuzzy PID based on particle swarm optimization. Electron Meas Technol. 2022;45(1):96-103. doi:10.19651/j.cnki.emt.2107821.

19. Izci D, Ekinci S. An improved RUN optimizer based real PID plus second-order derivative controller design as a novel method to enhance transient response and robustness of an automatic voltage regulator. e-Prime Adv Electr Eng Electron Energy. 2022;2:100071. doi:10.1016/j.prime.2022.100071.

20. Briones O, Alarcón R, Rojas AJ, Sbarbaro D. Tuning generalized predictive PI controllers for process control applications. ISA Trans. 2022;119:184-195. doi:10.1016/j.isatra.2021.02.040.

21. Al Gizi AJH. A particle swarm optimization, fuzzy PID controller with generator automatic voltage regulator. Soft Comput. 2019;23:8839-8853. doi:10.1007/s00500-018-3483-4.

22. Zhou W, Hou J. A new adaptive robust unscented Kalman filter for improving the accuracy of target tracking. IEEE Access. 2019;7:77476-77489. doi:10.1109/ACCESS.2019.2921794.

23. Bitriá R, Palacín J. Optimal PID control of a brushed DC motor with an embedded low-cost magnetic quadrature encoder for improved step overshoot responses in a mobile robot application. Sensors. 2022;22:7817. doi:10.3390/s22207817.

24. Gozde H, Taplamacioglu MC. Comparative performance analysis of artificial bee colony algorithm for automatic voltage regulator (AVR) system. J Frankl Inst. 2011;348(8):1927-1946. doi:10.1016/j.jfranklin.2011.05.012.

25. Qin H, Wang L, Wang S, Ruan W, Jiang F. A fuzzy adaptive PID coordination control strategy based on particle swarm optimization for auxiliary power unit. Energies. 2024;17:5311. doi:10.3390/en17215311.

26. Control Tutorials for MATLAB and Simulink. https://ctms.engin.umich.edu/CTMS

27. Wati T, Subiyanto S. Simulation model of speed control DC motor using fractional order PID controller. J Phys Conf Ser. 2020;1444:012022. doi:10.1088/1742-6596/1444/1/012022.

28. Wu T-Y, Jiang Y-Z, Su Y-Z, Yeh W-C. Using simplified swarm optimization on multiloop fuzzy PID controller tuning design for flow and temperature control system. Appl Sci. 2020;10:8472. doi:10.3390/app10238472.

29. Liu Z, Dong H, Ma X. Temperature control strategy for hydrogen fuel cell based on IPSO-Fuzzy-PID. Electronics. 2024;13:4949. doi:10.3390/electronics13244949.

30. Suid MH, Ahmad MA. Optimal tuning of sigmoid PID controller using nonlinear sine cosine algorithm for the automatic voltage regulator system. ISA Trans. 2022;128:265-286. doi:10.1016/j.isatra.2021.11.037.

31. Chatterjee S, Mukherjee V. PID controller for automatic voltage regulator using teaching-learning based optimization technique. Electr Power Energy Syst. 2016;77:418-429. doi:10.1016/j.ijepes.2015.11.010.

32. Mosaad AM, Attia MA, Abdelaziz AY. Whale optimization algorithm to tune PID and PIDA controllers on AVR system. Ain Shams Eng J. 2019;10:755-767. doi:10.1016/j.asej.2019.07.004.

FINANCING

"Tianchi Talent" Introduction Plan Funding Project.

CONFLICT OF INTEREST

The authors declare that there is no conflict of interest.

AUTHOR CONTRIBUTIONS

Conceptualization: Peng Zhao and Yunkang Chen.
Methodology: Peng Zhao.
Software: Yunkang Chen.
Validation: Peng Zhao, Yunkang Chen and Yongming Zhang.
Formal analysis: Peng Zhao.
Investigation: Yongming Zhang
Resources: Peng Zhao.
Data curation: Peng Zhao.
Writing—original draft preparation: Peng Zhao.
Writing—review and editing: Yunkang Chen.
Visualization: Yongming Zhang.
Supervision: Yunkang Chen.
Project administration: Peng Zhao.
Funding acquisition: Peng Zhao.