Transactions on Computer Science and Intelligent Systems Research
ISSN: 2960-1800, eISSN: 2960-2238 | Volume 12
ESAC 2026

Design and Implementation of a Motion Control System for an Omnidirectional Mobile Platform Based on a Microcontroller Microprocessor

Zikun Yu, Xianzi Zeng, Dongrui Xu, Hao Wang, Ziyu Li *, LeleYang

School of Mechanical and Electrical Engineering, Shandong Jianzhu University, Jinan, China, 250101

* Corresponding Author Email: 15689176498@163.com

Abstract. This paper details the design and implementation of an omnidirectional mobile platform motion control system built upon a microcontroller-based microprocessor. The mechanical framework employs low-hardness, lightweight aluminum extrusions as base materials, featuring a pentagonal chassis structure integrated with omnidirectional wheels. This configuration ensures agile movement while enabling on-the-spot rotation and high-speed cornering. The execution system innovatively incorporates an electric screw-driven parallel mechanical gripper for severing fruit stems and grasping objects. Hardware design centers on the selection of the main control board (ROBOMASTER Development Board Type C, utilizing a high-performance STM32 chip) and motor selection (M3508 geared motors for the chassis, 57 stepper motors for the robotic arm). On the software side, the system employs semi-automatic control to adapt to complex environments. Key implementations include DBUS protocol parsing for receiving remote controller commands and CAN communication protocol for efficient control of multiple M3508 motors. Additionally, the software design incorporates PWM signal control and PID closed-loop speed control system design with parameter tuning. Through the seamless integration of mechanical, hardware, and software components, this solution successfully establishes a complete control system with high real-time performance and stability.

Keywords: Omnidirectional mobile platform; Microcontroller control; CAN bus.

1. Introduction

Motion control systems are pivotal for achieving comprehensive functionality and maneuverability in complex machinery. Within mobile robotics, constructing a highly integrated, stable control system is fundamental to ensuring flexible and efficient operation in challenging environments. Current mobile platforms typically employ lightweight materials for mechanical structures to balance stability with overall weight. Building upon this foundation, this research aims to design a control platform capable of omnidirectional mobility and precise grasping operations [1].

Given this context, the core question addressed in this paper is: How can we construct a highly real-time and stable omnidirectional motion control system for mobile platforms, integrating mechanical design, hardware, and software control algorithms, to achieve flexible maneuverability and precise operations in complex environments? Key technical challenges include: achieving omnidirectional mobility while maintaining structural stability; selecting control chips with ADC acquisition and PWM output capabilities to efficiently manage multi-motor motion; and effectively implementing remote controller signal parsing and multi-motor drive control (particularly CAN bus-based motor group control).

To address these issues, this paper proposes an integrated research solution: First, in mechanical design, a pentagonal chassis structure and an innovatively designed electric lead screw/parallel mechanical gripper mechanism are established. Second, in hardware design, the STM32-based ROBOMASTER C-type development board serves as the main control core, paired with M3508 and 57 stepper motors as actuators. Finally, the software design focuses on establishing a semi-automatic control architecture. This includes implementing DMA reception and decoding for the DBUS protocol, interrupt handling for CAN bus transmission and reception, and designing a PID closed-loop speed control system. These efforts collectively enable precise and stable control of the omnidirectional mobile platform [2].

2. Mechanical Design of the Motion Control System

2.1. Overview of the Mechanical System

As the skeleton of the entire machine, the mechanical system—including its drive system, transmission system, execution system, control system, and power system—determines the functional integrity of the entire machine and the feasibility of other designs [3-4].

2.2. Structure of the Mechanical System

The integrity of the mechanical system is the basic part of a robot, which directly affects the flexibility and mobility of other structures built on the platform. Common mobile platforms at home and abroad use materials with low hardness and light density such as aluminum profiles or aluminum tubes as the basic materials for constructing the mechanical structure, ensuring structural stability while considering the weight of the entire machine. The overall framework of the chassis mechanical system design is shown in Figure 1.

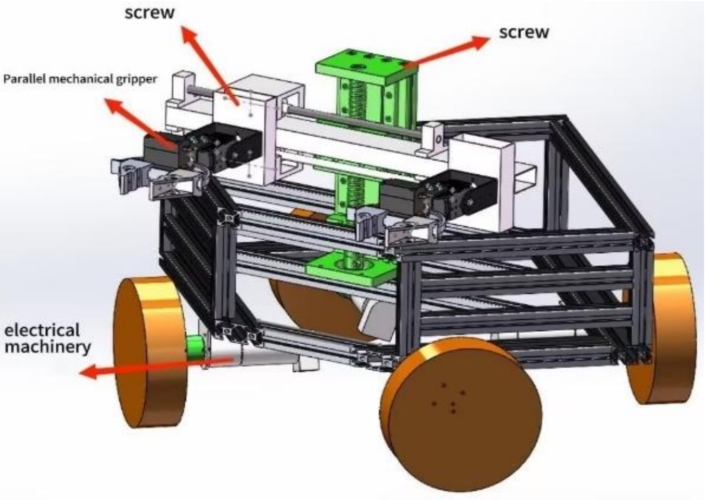

Figure 1. Overall framework of the chassis mechanical system design

The mechanical system of the mobile platform designed this time mainly features innovative designs of the execution system and control system. It includes 57 stepping motors with electric lead screws, M3508 DC brushless gear motors, omni-directional wheels, parallel mechanical claws, and several fasteners. Through the coordinated use and drive of various components, the mechanical system of the omni-directional mobile platform is completed [5].

2.3. Selection of Core Equipment

At the initial stage of design, a macro design plan is required, considering the selection of the chassis structure and manipulator structure. These two factors determine the subsequent design plan and content of the mechanical mechanism.

2.3.1. Selection of Chassis Structure

Considering the characteristics of omni-directional wheels, the chassis shape is designed as a pentagon to facilitate the flexible movement of the machine body, supporting in-place rotation or high-speed turning.

2.3.2. Selection of Manipulator Structure

The manipulator uses electric lead screws to achieve vertical changes in the spatial direction, and a horizontal lead screw is installed below to realize forward and backward movement. Parallel mechanical claws are fixed on the slider of the lead screw through aluminum tubes and screws, and blades are configured on the parallel mechanical claws to achieve the functions of cutting fruit stems and grabbing.

Since the spacing between seedlings in the competition task is fixed and known, the positions of the two parallel mechanical claws are fixed according to the competition task requirements to achieve the goal of efficiently picking two seedlings at a time.

3. Hardware Design of the Motion Control System

3.1. Overview of the Motion Control System

The core content of motion control is mainly the control of motors. In a control system, the most effective control method is to control the motion source. Therefore, when building the control system structure, the method of directly controlling the motor is generally used to achieve the purpose of controlling the controlled object.

Generally, an indispensable part of a closed-loop control system is information feedback. Common measured objects include temperature, humidity, pressure, liquid flow rate, object rotation speed, voltage value, etc. The implementation method usually uses analog circuits to convert these signals into voltage signals. The magnitude of the voltage signal reflects the magnitude of these measured objects, so the control chip of the control system needs to have AD acquisition function; the common execution mechanism in the motion control system is mainly the motor. A simple method to realize motor control is the PWM signal control mode. By changing the duty cycle, the conduction state of the motor driver is changed, thereby changing the voltage at both ends of the motor to control the speed of the motor. Therefore, the control chip of the control system needs to have the PWM function with adjustable output duty cycle. The above two points are the key to constructing the motion control system. Therefore, the mainstream methods to realize the motion control system generally adopt programmable controllers, mainly including single-chip microprocessors, FPDA/CPLD programmable logic devices, programmable DSP controllers, or special motion algorithm controllers. These methods are characterized by simple and stable circuits, ability to implement complex algorithm operations, fast signal acquisition frequency and high precision, and usually support multi-channel A/D conversion, multi-channel pulse capture function and multi-channel PWM output function. The motion control system built with programmable controllers is simple and easy to upgrade the control algorithm code, with good maintainability and high scalability [6-7].

The scheme adopted in this subject is an omni-directional mobile control system constructed based on a single-chip microprocessor.

3.2. Structure of the Omni-Directional Mobile Platform Control System

The drive and control of the omni-directional mobile platform is a key part of the robot, which can directly affect the flexibility and mobility of other structures built on the platform. Common mobile platforms at home and abroad use DC motors as the drive source, combined with appropriate reducers, which can bring reliable load-carrying capacity to the entire mobile platform [8].

This subject completes the construction of a multi-motor control system, which also requires high real-time performance and stability.

The hardware system structure of the mobile platform designed this time mainly involves the coordinated work between the chassis main controller and the chassis motor ESCs. It includes ROBOMASTER Development Board C, M3508 DC brushless gear motors, DC brushless gear motor speed controllers, DT7 remote control, DR16 receiver, and omni-directional wheels. Through the coordinated use and drive of each module, the hardware system construction of the omni-directional mobile platform is completed.

3.3. Selection of Core Equipment

At the initial stage of design, a macro design plan is required, considering the selection of the main control board and motor. These two factors determine the subsequent design plan and content of the hardware circuit.

3.3.1. Selection of Main Control Board

The main control board selected for this subject is the RoboMaster Development Board C, which adopts a high-performance STM32 main control chip, supports wide voltage input, integrates dedicated expansion interfaces, communication interfaces, and high-precision IMU sensors, and can be used with RoboMaster products or other accessories. The Development Board C has the following peripherals: user-defined LEDs, 5V interface, BOOT configuration interface, micro USB interface, SWD interface, buttons, configurable I/O interfaces, UART interface, CAN bus interface, PWM interface, DBUS interface, digital camera FPC interface, buzzer, voltage detection ADC, six-axis inertial measurement unit, and magnetometer [9-10].

3.3.2. Selection of Motors

As the execution mechanism, the performance of the motor directly affects the flexibility, speed, and load-carrying capacity of the omni-directional mobile platform. Therefore, the selection of the motor is extremely important.

The motor selected for the chassis control this time is the M3508 motor, matched with the C620 ESC. The maximum power of the M3508 gear motor set is as high as 220W, with a maximum torque of 5N·m; the maximum continuous power is 150W, with a continuous torque of 2.8N·m. The sensorless FOC control can provide stable torque regardless of the speed. It allows the robot to maintain stable power while responding quickly. The M3508 gear motor set has an industry-leading power density. While providing high power, its volume and weight are only 20% of that of similar equipment, saving a lot of space and outputting more power to make the robot operate efficiently. This motor is a chassis drive motor tailored for wheeled robots weighing 5kg to 20kg, with the characteristics of high efficiency, high reliability, and low noise, and can provide sufficient power for the omni-directional platform. Parameters of M3508 DC Brushless Gear Motor and C620 Brushless Motor Speed Controller are shown in table 1 and table 2.

Table 1. Parameters of M3508 DC Brushless Gear Motor

Parameter	Specification
Weight	365g
Outer diameter	42mm
Total length	98mm
Output shaft	D-type with threaded hole
Output shaft diameter	10mm

Table 2. Parameters of C620 Brushless Motor Speed Controller

Parameter	Specification
Rated voltage	24V
Weight	35g
Size (length × width × height, excluding wires)	49.4×25.8×11.5mm
Total length with wires	344+15mm
Signal type	CAN command, PWM
Maximum continuous current	20A

Table 3. Parameters of M3508 Gear Motor Set

Parameter	Specification
Rated voltage	24V
No-load speed	482rpm
Maximum continuous torque	3N·m
Maximum speed at 3N·m	469rpm
Operating temperature range	0-50°C

Parameters of M3508 Gear Motor Set and 57 Stepping Motor and TB6600 Driver are shown in table 3 and table 4. The motor selected for the manipulator control this time is a 57 stepping motor, matched with the TB6600 driver. The stepping motor uses imported high-quality motor steel sheets and bearings made by German technology, with small self-inductance reactance and good responsiveness. It can be directly connected to the load to run at low speed in a near-stall state without gear reduction, so it can generate a high torque-to-inertia ratio on the load shaft and eliminate system errors caused by the use of reduction gears. The internal rotor adopts German technology and is coated with imported rotor glue to prevent burrs on the surface after carbon steel oxidation and grinding, which causes increased motor noise. The TB6600 driver is powered by a DC 9-42V power supply, with 12-24V as the optimal control signal input voltage, and 3.3-24V input voltage is universal. The subdivision precision can be selected from 1-32 subdivisions, with H-bridge bipolar constant current drive and high-speed optocoupler isolation for input signals. Built-in temperature protection and overcurrent protection, automatic half-current to reduce heat generation. Motor noise optimization function, small size and space-saving.

Table 4. Parameters of 57 Stepping Motor and TB6600 Driver

Parameter	Specification
Brand	Ouli Transmission
Specification	57×56
Voltage	24VDC
Current	3.0A
Torque	1NM
Application occasions	Medical, industrial automation, printing, robots
Power supply voltage (driver)	DC 9-42V
Subdivision setting (driver)	SW1-SW3 adjustable
Current setting (driver)	SW4-SW6 adjustable

The remote control selects the DT7 remote control and the matching DR16 receiver to control the chassis movement through the DBUS protocol. The DT7 remote control is equipped with the DR16 D-Bus receiver as standard, supporting up to 16-channel control, easy installation, and higher safety of the remote control system. Specifications and Parameters of DT7 & DR16 is shown in table 5.

Table 5. Specifications and Parameters of DT7 & DR16

Component	Parameter	Specification
Common	Working frequency	2.4GHz ISM
	Communication distance (open outdoor)	1000m
DT7 Remote Control	Features	2 joysticks, 7 channels
	Working current/voltage	100mA@6V
	Battery	4 AA batteries
DR16 Receiver	Features	2.4GHz D-BUS protocol
	Receiving sensitivity (1% PER)	-97dBm
	Working current/voltage	145mA@5V
	Power supply	4-8.4V
	Size	41mm×29mm×5mm
	Weight	10g

3.4. Design of Control System Hardware Circuit

3.4.1. Voltage Regulator Power Supply Design

The power supply is the foundation to ensure the normal operation of the hardware system. In the design, detailed analysis should be carried out: the power input that the system can provide; the power output that the single board needs to generate; the current required by each power supply; the efficiency of the power circuit; the allowable fluctuation range of each power supply; the power-on sequence required by the entire power system, etc.

In the weak current control system, the common power supply voltage for sensors and various electronic components is 5V or 3.3V. The main control board we selected has multiple voltage reduction paths, which can meet the power supply needs of different electronic components.

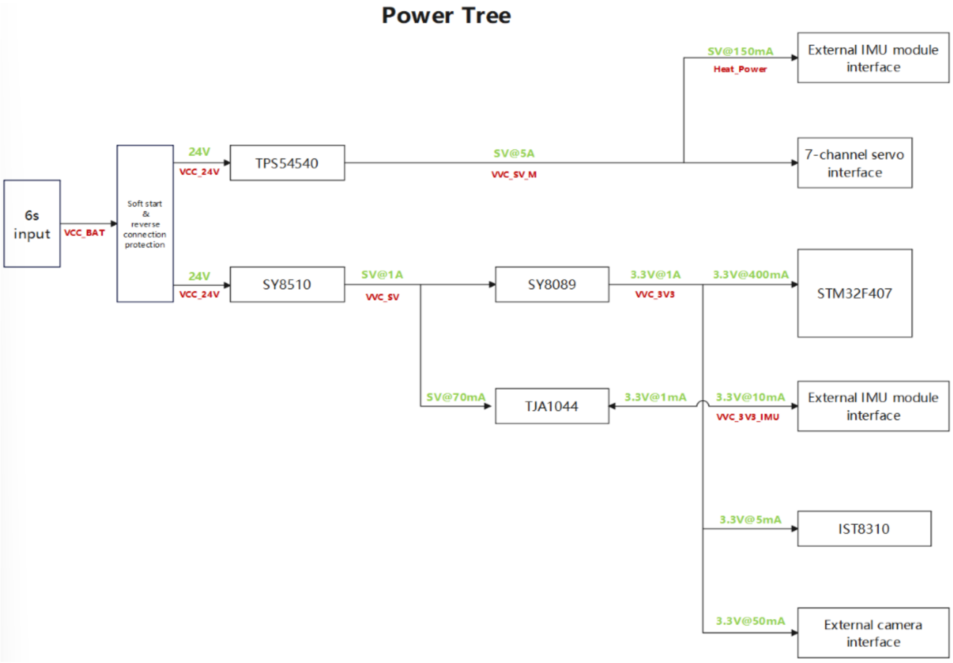

Figure 2. Power output tree diagram of the main control board

Figure 2 shows the power supply paths and output parameters of the main control board, including 24V input, 5V output, 3.3V output, and their corresponding application interfaces.

3.5. Minimum System Design of Single-Chip Microcomputer

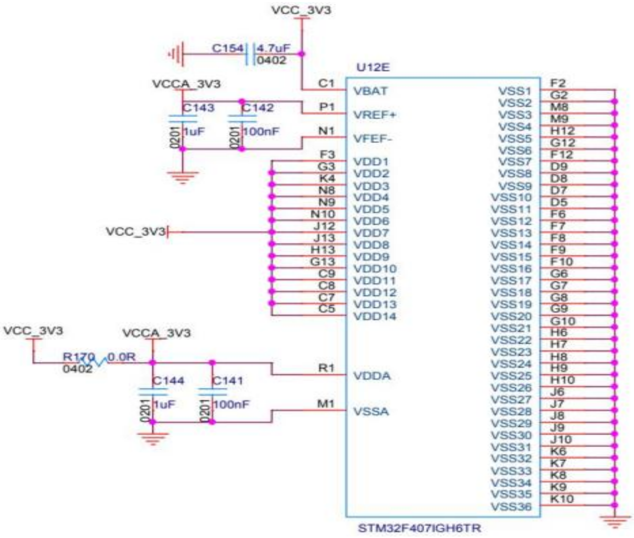

Figure 3. Schematic diagram of the minimum system of the single-chip microcomputer

The minimum system of the single-chip microcomputer is the minimum unit circuit to ensure the operation of the single-chip microcomputer chip, which includes chip power supply, reset circuit, external clock circuit, and download debugging circuit. Its schematic diagram is shown in Figure 3.

3.6. Summary

The main content of this section is to provide a detailed introduction and in-depth analysis of the hardware design of the omni-directional mobile platform. In the schematic design process, we should adopt the "take-it-yourself" principle. Now chip manufacturers generally provide reference design schematics, so we should try to use these resources and make some innovations on the basis of fully understanding the reference design. After the main chips are selected, the most critical peripheral designs include power supply, clock, and interconnection between chips.

The hardware design not only provides the circuit schematic diagram but also detailedly analyzes the impact of changes in various circuit parameters on the performance of the entire circuit.

4. Software Design of the Omni-Directional Mobile Platform

Mechanics and hardware are the visible foundation of the omni-directional platform, while software and control algorithms are the internal soul of the omni-directional platform. The three complement each other and are indispensable. They cooperate with each other to form a complete control system. Compared with hardware design, software design is more flexible, which also brings convenience to the design of the control system.

4.1. Software Design Idea and Overall Architecture

There are mainly two common control methods for mobile robots:

(1) Fully automatic mode: After the robot is powered on, it autonomously completes the corresponding actions, but generally, the motion trajectory is preset and relatively single. To adapt to a new environment, the motion trajectory needs to be reset, which is not flexible enough.

(2) Semi-automatic mode: After the robot is powered on, humans manually control the robot's movement direction, speed, and trajectory through visual feedback. This method is more flexible and can operate well in various new and complex venues.

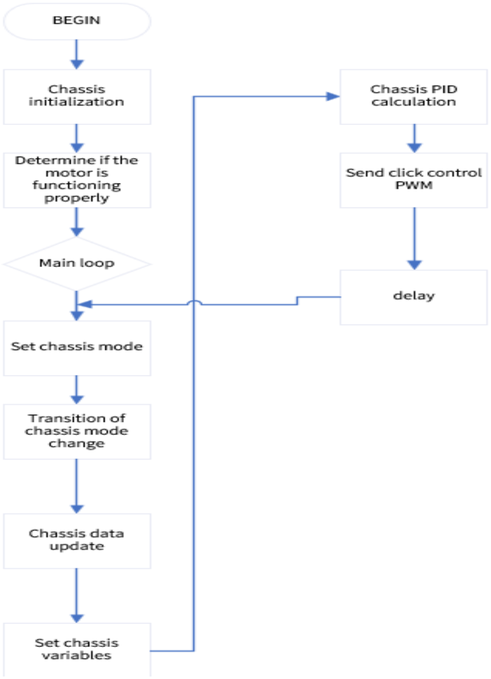

Figure 4. Software flow chart of the chassis control system

To adapt to various complex environments, the design scheme adopted this time is the semi-automatic mode. The movement direction, speed, and trajectory of the omni-directional mobile platform are issued through the remote control. This method allows the omni-directional platform to flexibly shuttle in various narrow spaces, and the control is relatively flexible and convenient.

According to the requirements of the motion control system of the omni-directional mobile chassis, the software flow chart is designed as figure 4.

4.2. Remote Control Parsing Protocol

The main functions of the remote control are as follows:

(1) Enable and disable the control of the mobile platform;

(2) Control the motion attitude of the platform, including the planning of the movement direction and trajectory, and the setting of the movement speed. The remote control selects the 2.4GHz remote control provided by DJI, with the specific model DT7 and the receiver DR16.

The output signal of the remote control receiver is standard DBUS protocol data. When the receiver establishes a connection with the transmitter, the receiver sends a frame of data (18 bytes) through DBUS every 7ms. The communication parameters of DBUS are shown in Table 6. It should be noted that the control level of the DBUS signal complies with TTL, but it is opposite to the ordinary UART signal. Therefore, a triode inversion circuit needs to be added at the MCU end so that the MCU can normally recognize the UART signal.

Table 6. DBUS Communication Parameters

Parameter	Specification
Baud rate	100k bit/s
Data length	8 bits
Parity bit	Even parity
Stop bit	1 bit
Level standard	High level = 0, low level = 1 (opposite to UART)
Frame length	18 bytes (144 bits)

The data frame structure of the DBUS protocol is shown in the following table. The meaning of each segment of data can be found according to the remote control manual, so as to splice the data and complete the decoding of the remote control. DBUS Data Frame Structure is shown in table 7 and table 8.

Table 7. DBUS Data Frame Structure (Joystick and Switch Part)

Offset (bit)	Content	Length (bit)	Sign Bit	Range	Function
0	Channel 0	11	None	Maximum 1684, Middle 1024, Minimum 364	Unsigned type, Remote control channel 0
11	Channel 1	11	None	Maximum 1684, Middle 1024, Minimum 364	Unsigned type, Remote control channel 1
22	Channel 2	11	None	Maximum 1684, Middle 1024, Minimum 364	Unsigned type, Remote control channel 2
33	Channel 3	11	None	Maximum 1684, Middle 1024, Minimum 364	Unsigned type, Remote control channel 3
44	S1	2	None	Maximum 3, Minimum 1	Remote control transmitter S1 switch position
46	S2	2	None	Maximum 3, Minimum 1	Remote control transmitter S2 switch position

Table 8. DBUS Data Frame Structure (Mouse and Key Part)

Offset (bit)	Content	Length (bit)	Sign Bit	Function
48	Mouse X-axis	16	Yes	Mouse X-axis movement data
64	Mouse Y-axis	16	Yes	Mouse Y-axis movement data
80	Mouse Z-axis	16	Yes	Mouse Z-axis movement data
96	Mouse left button	8	None	Mouse left button status
104	Mouse right button	8	None	Mouse right button status
112	Key 1	16	None	Key 1 status

First, configure the DMA for serial transmission. First, enable and configure USART1 and USART3. Among them, USART1 enables DMA transmission for serial port to send data to the PC's serial port tool, and USART3 enables DMA reception for receiving remote control data. Then, implement the process using the printf function. Use the va_start function and vsprintf function under stdarg.h together with the DMA transmission function of the serial port to implement the printf function in C language. Through the operation of the above functions, store the data content to be sent in tx_buf, store the length of the data to be sent in the len variable, and then pass the starting address of tx_buf and the data length len to the DMA transmission function to complete the current DMA data transmission.

Use the DMA reception function of USART3 to receive remote control data. Initialize the DMA reception of USART3 through the function remotecontrol_init. During initialization, enable DMA serial port reception and idle interrupt, and configure the buffer for storing data after the peripheral data arrives. Here, the double buffer function is enabled. Each frame of SBUS data is 18 bytes, and the total size of the enabled double buffer is 36 bytes, which can avoid DMA transmission out of bounds.

The flow of this program is to initialize the DMA transmission of USART1 and the DMA reception of USART3 during initialization, then use DMA to receive remote control data in the serial port reception interrupt of USART3, and use the decoding function to decode the data.

Then, call the usart_printf function implemented by the serial port in the main loop to send the decoded remote control data through the DMA transmission function of USART1.

The program flow chart is as follows:

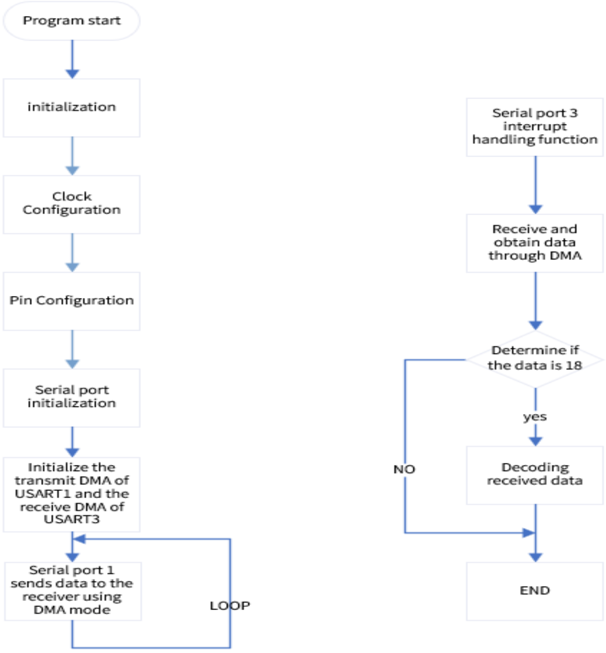

Figure 5. Program flow chart of remote control data reception and transmission

Figure 5 shows the initialization process, data reception, decoding, and transmission process of the remote control data processing program.

4.3. CAN Communication to Control M3508

CAN is the abbreviation of Controller Area Network. It was developed by the German BOACH company, which is famous for researching and producing automotive electronic products, and finally became an international standard (ISO11898). CAN is one of the most widely used field buses in the world. In North America and Western Europe, the CAN bus protocol has become the standard bus for automotive computer control systems and embedded industrial control local area networks, and there is the J1939 protocol designed specifically for large trucks and heavy machinery vehicles with CAN as the underlying protocol.

The CAN bus is composed of two lines, CAN_H and CAN_L, and various devices are mounted on the bus together. Like the I2C bus, each CAN mounted on the CAN bus has its own unique ID. Whenever a device sends a frame of data, other devices on the bus will check whether this ID is the object for which they need to receive data. If yes, they will receive this frame of data; if not, they will ignore it. The ID is stored in the arbitration field at the beginning of the data frame. CAN IDs are divided into two categories: standard IDs and extended IDs. The length of a standard ID is 11 bits. If there are too many devices and the standard IDs are not enough, extended IDs can be used, and the length of an extended ID is 29 bits.

If you want to send data to ESCs 1 to 4 to control the output current of the motor, thereby controlling the motor speed, you need to set the ID of the sent CAN data frame to 0x200 according to the content in the table, fill the 8-byte data in the data field in the order of the high 8 bits and low 8 bits of ESCs 1 to 4, set the frame format and DLC according to the content in the table, and finally send the data.

First, determine which ESC the received data comes from according to the received ID. The manual specifies that the ID of ESC 1 is 0x201, ESC 2 is 0x202, ESC 3 is 0x203, and ESC 4 is 0x204. After determining the data source, decode according to the data format in the manual, and obtain the motor's rotor mechanical angle, rotor speed, torque current, motor temperature and other data by splicing the high 8 bits and low 8 bits.

We adopted the CAN_cmd_chassis function and CAN_cmd_gimbal function to send CAN signals to the chassis motors to control motor rotation. The input of the CAN_cmd_chassis function is the expected drive current values motor1 to motor4 of motors 1 to 4. The function will split the expected values into high 8 bits and low 8 bits, put them into the 8-byte CAN data field, then add information such as ID (CAN_CHASSIS_ALL_ID 0x200), frame format, and data length to form a complete CAN data frame and send it to each ESC.

Whenever the CAN completes the reception of a frame of data, it will trigger the CAN receive interrupt processing function once. After the receive interrupt function completes the processing of some registers, it will call the CAN receive interrupt callback function. In this program, in the interrupt callback function, first determine the ID of the receiving object to see if it is the data sent by the ESC that needs to be received. After completing the judgment, decode and load the corresponding motor data into each corresponding bit of the motor information array motor_chassis.

The receiving function HAL_CAN_GetRxMessage provided by the HAL library is called during reception:

motor_chassis is an array of motor_measure_t type, which contains information such as motor rotor angle, motor rotor speed, control current, and temperature.

The decoding function actually splices the received data in the way of high 8 bits and low 8 bits to obtain various parameters of the motor.

The flow chart is as Figure 6:

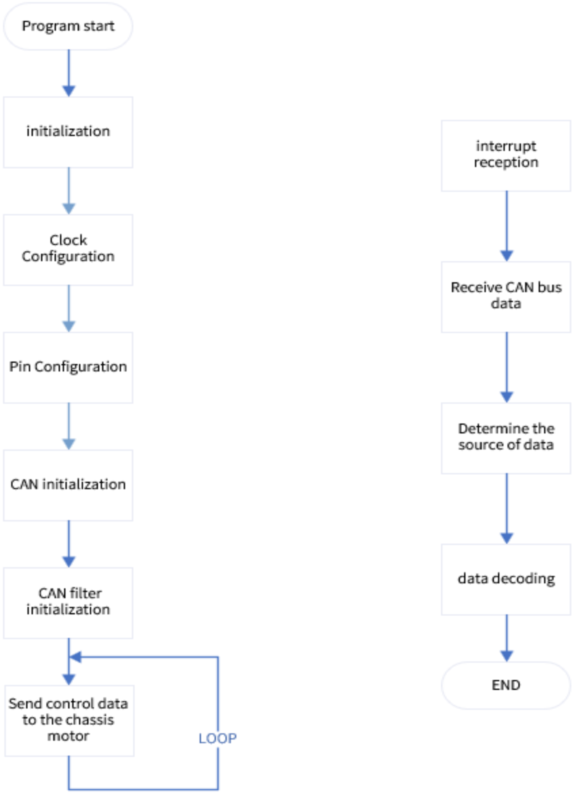

Figure 6. Program flow chart of CAN communication

4.4. PWM Control of Servo

When using __HAL_TIM_SetCompare to set the duty cycle, the corresponding inputs are shown in table 9:

Table 9. Parameters of __HAL_TIM_SetCompare Function

Parameter	Description
Parameter 1	*htim: Pointer to the timer handle, e.g., input &htim1 for Timer 1
Parameter 2	Channel: The PWM output channel of the timer, e.g., TIM_CHANNEL_1 for channel 1
Parameter 3	The value to be assigned to the comparison register; the PWM duty cycle is equal to (comparison value / timer period) × 100%

As the duty cycle of each PWM output changes in the main loop, the rotation angle of the servo also changes accordingly. Thus, the servo can realize cyclic rotation from 0 to 180 degrees.

4.5. Summary

The software design in this scheme mainly completes the following tasks:

(1) Parse the remote control channels to obtain information such as chassis enable/disable, speed given value, and chassis motion trajectory.

(2) Design CAN communication protocol packets and define the CAN communication ID addresses of the motor drivers for each wheel.

(3) Modify the duty cycle through PWM to control the servo.

(4) Design the PID closed-loop speed control system, analyze the impact of changes in K_p, K_i, and K_d on the entire control system, and perform parameter tuning to finally obtain an ideal speed controller.

5. Conclusion

The successful construction of the omnidirectional mobile platform's motion control system relies on the complementary integration of mechanical, hardware, and software components. This design ensures platform flexibility and structural stability through the use of lightweight aluminum profiles and an innovative pentagonal omnidirectional chassis. The hardware system, centered around a high-performance STM32 main control chip, drives the chassis and operates the robotic arm using M3508 and 57 stepper motors. The software design successfully implements high-speed, dual-buffer DMA reception of the remote controller's DBUS protocol and efficiently manages the drive current of the chassis' multiple motors via CAN bus. Furthermore, the application of PWM control and a closed-loop PID speed control system ensures this omnidirectional mobile platform possesses the high real-time performance, stability, and agile maneuverability required to navigate complex environments.

References

[1] Ding H, Fang X, Song J, et al. Vertical-horizontal vibration control using a novel tuned liquid column mass damper (TLCMD) system: Nonlinear coupling dynamics [J]. Structures, 2025, 82: 110463-110463.

[2] Zhang C, Su L, Zhou M, et al. Output feedback motion control with prescribed performance guarantees for hysteretic nonlinear systems and its applications [J]. Precision Engineering, 2026, 97: 547-558.

[3] Silva M T, Franchini G, Balasubramanian P, et al. Tuning and miniaturization of active vibration controllers of a fixed-wing composite aircraft drone using the experiment-on-the-loop system [J]. Aerospace Science and Technology, 2026, 168: 111004-111004.

[4] Zhao Y, Cao Y, Wu Z, et al. A theoretical and experimental study on the dynamic behavior and vibration control of a nonlinear coupling beam system under double excitation sources [J]. Mechanical Systems and Signal Processing, 2025, 241: 113459-113459.

[5] Liu L S, Zhai M, Wang R, et al. Dynamic event-based asymptotic tracking and vibration control for constrained flexible manipulator systems with intermittent faults [J]. Neurocomputing, 2025, 658: 131638-131638.

[6] Zhou J, Zuo G, Zhang S, et al. Motion control strategy for robotic arm with deep cascade feature Bayesian broad learning system [J]. Engineering Applications of Artificial Intelligence, 2025, 160: 111888-111888.

[7] Tan B, Deng Z, Liu J, et al. Energy-Conserving Nonlinear Active Control for a Bird-Inspired Quasi-Zero Stiffness Stretcher Vibration Isolation System with Asymmetric Stiffness [J]. International Journal of Structural Stability and Dynamics, 2025 (prepublish).

[8] Zhang C, He S, Mohammadzadeh A. A Novel Vibration Control System for Active Mass Drivers Based on Dynamic Fractional-order Type-2 Fuzzy Model and Adaptive Fractional Derivative [J]. Journal of Vibration Engineering & Technologies, 2025, 13 (8): 553-553.

[9] Zhou T, Zhang S, Liu S, et al. Digital technology integration in home-based exercise: a systematic review of research evolution, applications, and impact mechanisms [J]. BMC public health, 2025, 25 (1): 3528.

[10] Ma J, Hu J, Li S, et al. Stability in dynamic event-triggered control of stochastic nonlinear systems: A G-Brownian motion scenario [J]. ISA transactions, 2025.