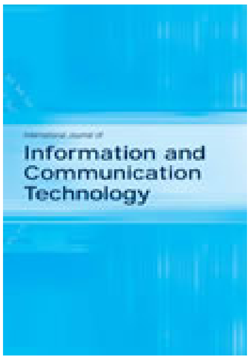



**International Journal of Information and Communication Technology**

ISSN online: 1741-8070 - ISSN print: 1466-6642
https://www.inderscience.com/ijict

---

# Design and implementation of a microcontroller-based multi-vehicle intelligent cooperation system

Xilong Qu, Xiao Tan, Siyang Yu

**DOI:** 10.1504/IJICT.2025.10071260

**Article History:**

| | |
|---|---|
| Received: | 23 October 2024 |
| Last revised: | 27 February 2025 |
| Accepted: | 27 February 2025 |
| Published online: | 11 June 2025 |

# Design and implementation of a microcontroller-based multi-vehicle intelligent cooperation system


## Xilong Qu

College of Information Science and Engineering,
Changsha Normal University,
Changsha, 410000, China
Email: quxilong@126.com

## Xiao Tan* and Siyang Yu

School of Information Technology and Management,
Hunan University of Finance and Economics,
Changsha, 410205, China
Email: tanxiao@hufe.edu.cn
Email: yusiyang@hufe.edu.cn
*Corresponding author



**Abstract:** This paper presents the design and implementation of an autonomous vehicle system for warehouse logistics, utilising a dual-vehicle coordination mechanism to enhance efficiency and adaptability in dynamic environments. The system integrates advanced hardware components, including the STM32 microcontroller, TB6612FNG motor driver, and YB-MVX01 infrared line tracking module, to achieve precise navigation and task execution. Wireless communication is facilitated through JDY-31 Bluetooth and GL24S 2.4G modules, enabling external control and inter-vehicle coordination respectively. A PID control algorithm ensures accurate line tracking, while an LCD display provides real-time system status. Experimental results demonstrate the system's effectiveness in autonomous navigation, task coordination, and adaptability to dynamic warehouse environments. The design prioritises modularity, scalability, and cost-efficiency, making it a viable solution for modern logistics automation.



**Keywords:** microcontroller; autonomous vehicle system; dynamic warehouse environments; GL24S 2.4G modules.

**Reference** to this paper should be made as follows: Qu, X., Tan, X. and Yu, S. (2025) 'Design and implementation of a microcontroller-based multi-vehicle intelligent cooperation system', *Int. J. Information and Communication Technology*, Vol. 26, No. 17, pp.41–71.



**Biographical notes:** Xilong Qu is a distinguished figure in the field of artificial intelligence and machine learning, with a keen interest in ethical AI development. His expertise encompasses the creation of algorithms that can learn from and adapt to their environment, improving decision-making processes in various sectors such as healthcare, finance, and autonomous vehicles. His work focuses on ensuring that AI systems are developed with fairness, accountability, and transparency, to benefit society as a whole.

Xiao Tan is an expert with a profound research background in chip technology, hardware security, and the internet of things. His career is focused on exploring and innovating advanced technological solutions, especially in the fields of microelectronics and information security. His research primarily involves developing more efficient methods for chip design and exploring new types of hardware security strategies, aimed at protecting data from unauthorised access and tampering.

Siyang Yu received his PhD degrees in Computer Science and Technology from Hunan University, China, in 2017. He is currently a teacher at Hunan University of Finance and Economics. His research interests include industrial internet of things, abnormal analysis, and intrusion detection.

# 1 Introduction

## *1.1 Research background and significance*

The rapid advancement of technology has significantly elevated the level of automation. Microcontrollers, as compact and efficient single-chip systems, integrate numerous functionalities – such as CPU, RAM, ROM, and various I/O ports – onto a silicon chip, playing a pivotal role in automation control across industries (Roccaforte et al., 2018; Frigerio and Matta, 2014; Nguyen et al., 2017; Cansizoglu and Elrefaie, 2017). In warehouse logistics, intelligent small cars based on microcontroller technology are increasingly vital for enhancing production efficiency and flexibility.

With the maturation and widespread adoption of microcontroller technology, its application in intelligent small cars has become increasingly prominent (Maier et al., 2017; Desai, 2013). In the current era of enhanced enterprise production technology, intelligent vehicles, and products based on them, have become central in automated logistics transportation and flexible production organisation systems (Duan, 2018).

Globally, the development of intelligent cars is intensifying. Mobile robotics, a discipline that emerged in the 1960s, has seen rapid development worldwide, with intelligent small cars becoming a significant research direction (Hamdan, 2018; Lakshmanan et al., 2020; Kamrani et al., 2023; Khan et al., 2022). Today's society, marked by widespread intelligent scenarios, welcomes the arrival of the 5G era, further diversifying the application scenarios of intelligent products. Intelligent small cars, as multifunctional intelligent mobile devices, find uses in various domains, including military, medical, logistics, and household, addressing diverse needs.

Microcontrollers, as miniature automated single-chip computers, are known for their convenience, low energy consumption, and compact size (Tsurumi and Fujii, 2018). They are widely used in everyday production and life, integrating various basic functionalities within a limited chip space. Additionally, their automated control includes components like counters, timers, serial ports, system processors, program controllers, data storage, and specific function memories (Kaewpukdee and Uthansakul, 2023). These components are interconnected to form a unified system. In microcontrollers, controllers execute various operation commands, with encoders integrating codes and transferring them to logic circuits (Bento et al., 2019). These circuits then distribute timing signals and control information to different control elements. The controller, comprising logic control systems, encoders, storage devices, and programming devices, can precisely recognise different instructions, making it an ideal main control chip for intelligent small cars (Dotoli et al., 2022).

The rapid development of computer and electronic technologies has heightened interest in intelligent product research across various fields, from industrial production to daily life. Intelligent small cars, integrating electronics, sensors, and mechanical systems, excel in automated logistics and can operate in environments inaccessible to humans, such as unmanned transportation and autonomous driving (Armstrong and Helps, 2022). However, most existing systems lack inter-vehicle communication, posing challenges in complex scenarios requiring coordinated efforts. This study addresses this gap by designing a multi-vehicle system to improve task execution in dynamic warehouse settings.

Intelligent small cars, or wheeled robots, integrate various technologies such as environmental perception, dynamic decision-making and planning, and intelligent control and execution. With ongoing technological maturity, several cutting-edge technologies have been applied to intelligent small cars.

Since the 1990s, developing intelligent cars has become a new driving force and direction in the automotive industry. Major countries in Europe, the US, and Japan have incorporated intelligent cars into their industrial systems, vigorously developing this technology. As intelligent car technology continues to evolve, it is fundamentally changing traditional car design concepts, integrating more microelectronics, intelligent algorithms, and automatic control technologies (Jang and Skibniewski, 2022).

The rapid development of computer, electronic communication, network, and intelligent control algorithms has laid a solid foundation for vehicle intelligence. From a technical perspective, intelligent car systems can be divided into three levels: intelligent perception systems, vehicle driving systems, and automatic operation systems. Intelligent perception systems use various sensors to detect the vehicle's condition, driving environment, and driver's status in real-time (Joseph and Oladayo, 2022). The vehicle driving system plans and processes information obtained from the intelligent perception system, providing driving suggestions to the driver or replacing the driver in autonomous driving to some extent. All control systems in intelligent cars are implemented through onboard computers (Hu et al., 2018; Chow et al., 2018). With the continuous development of modern control technology, many modern intelligent control techniques, such as incremental PID control, have been proposed for intelligent car speed control systems.

## *1.2 Research status at home and abroad*

### *1.2.1 International research status*

Globally, significant progress has been made in the development of new types of automobiles, with widespread applications in military, industrial, and daily life. The US pioneered automated guided vehicles as early as 1954. By the mid-1980s, Carnegie Mellon University had modified a Chevrolet car with computers, pioneering intelligent vehicle research. Italian universities also contributed, with Parma University developing autonomous safety systems for cars (Billa et al., 2020; Jozsef, 2021; Bogue, 2021; Sharma et al., 2020). In recent years, companies like Google have invested in intelligent vehicle R&D. South Korea's annual Freescale Intelligent Car Competition has seen growing popularity since the early 2000s.

### *1.2.2 Domestic research status*

In China, intelligent vehicle research is still in its infancy, initiated in the 1980s by institutions like the National University of Defense Technology and Tsinghua University. However, challenges persist in testing and safety assurance, especially on highways with complex conditions (Gueltekin et al., 2022; Ko and Lee, 2021). Recognising these challenges, in 2009, Academician Lee proposed a 'miniature intelligent vehicle testing platform'. The Miniature Car Design Competition has become a popular event among college students, boosting their practical and innovative skills (Yang et al., 2021; Schü et al., 2020).

Post-2020, the focus on individual intelligent vehicles has shifted towards multi-vehicle communication, thanks to advancements in wireless technology. Bluetooth technology now enables direct command transmission to intelligent small cars, enhancing their flexibility (Cai et al., 2014). The use of the 2.4G frequency band for inter-vehicle communication ensures stable and fast data transmission. These developments have opened new possibilities for coordinated operation among multiple intelligent vehicles (Zhang et al., 2019; Santos et al., 2022).

### *Thesis structure and main content*

This thesis elaborates on a design scheme for a microcontroller-based multi-vehicle intelligent small car system, structured into six chapters:

- Introduction: Discusses the background and significance of the multi-intelligent small car communication control system, along with an analysis of domestic and international research status on microcontrollers and intelligent small cars.
- Overall system design: presents an overview of the system's conceptual design, requirements analysis, development environment, and relevant technologies.
- Hardware design of the car system: Selects and details the hardware modules suitable for system requirements, including processor, wireless communication, motor drive, battery, and tracking modules.
- Software design of the car system: Describes the main program's design, including motor drive, display, tracking, and buzzer programs, as well as the wireless communication program and PID closed-loop control algorithm.
- System testing: Explains testing methods, processes, and results analysis to verify the system's performance and reliability.
- Conclusion: Summarises the system's design, development, and testing, and suggests potential improvements and optimisations.

## 2 System overview design

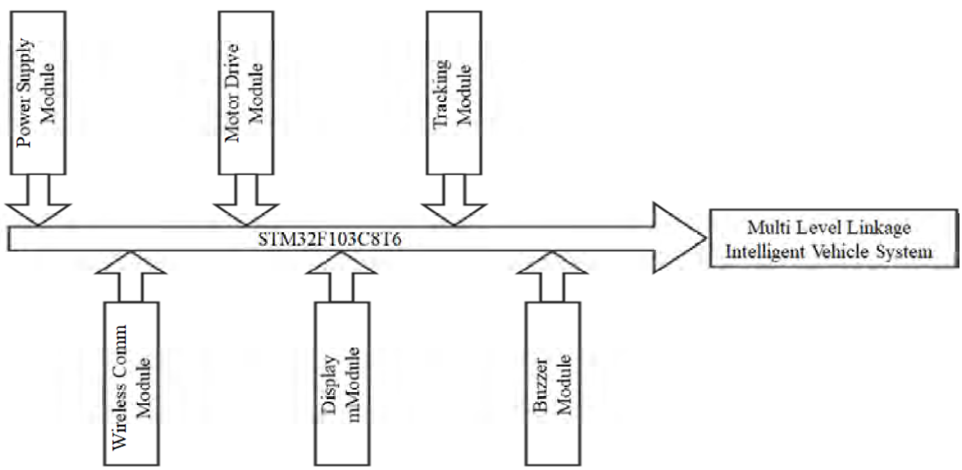
### *2.1 General design concept*

This system is primarily designed as a multi-vehicle intelligent car system based on embedded microcontroller technology and wireless communication. The system initially constructs several intelligent cars equipped with functionalities such as lighting, sound, and wireless connectivity. Two intelligent cars are preset in the system: a lead car and a following car. The lead car is capable of sending commands to the following car, which performs the instructed actions and, upon completion, sends feedback to the lead car. The lead car can also receive and execute commands sent by operators from a supervisory computer.

Another focus of the system is on automatic control, particularly the control of direct current (DC) motors by microcontrollers. A PID algorithm is employed to regulate the motor speed, ensuring stable and smooth operation of the vehicles. Additionally, the battery circuit is meticulously designed to enhance the integration of the intelligent cars.

The modules of this design are illustrated in Figure 1.

**Figure 1** Module design diagram




Beyond these core functionalities, the system integrates several safety features to ensure reliable operation. An emergency stop mechanism is implemented, allowing the STM32 microcontroller to halt the vehicle immediately if a malfunction is detected. Additionally, the YB-MVX01 infrared line tracking module serves a dual purpose as a basic obstacle detection system, enabling the vehicle to decelerate or stop when encountering unexpected objects along its path. Software safeguards, such as communication timeout protocols, further ensure that the vehicle ceases operation if wireless signals are disrupted, preventing erratic behaviour. These protective measures enhance the system's safety and dependability, making it well-suited for autonomous navigation in dynamic settings like warehouses.

### 2.2 System requirements analysis

The multi-vehicle intelligent car system is designed to meet the following requirements:

- Each intelligent car should perform line-following navigation with an error margin of less than 5 cm.
- The cars must receive and execute commands from a supervisory computer with a response time under 1 second.
- Bidirectional wireless communication between the intelligent cars should have a latency of less than 100 ms.
- Upon receiving a delivery command, the cars should perform both single and dual-vehicle delivery tasks, ensuring collision-free operation.
- The design should prioritise cost-efficiency, targeting a total hardware cost per vehicle below $100.

### 2.3 System development environment

The hardware development environment for this system utilises LCEDA, a popular Electronic Design Automation (EDA) software in China. LCEDA is primarily used for circuit schematic design, PCB layout design, and simulation analysis. It offers a comprehensive component library, robust simulation capabilities, and a convenient workflow for schematic drawing and PCB design, facilitating rapid circuit design and experimental validation for users. Additionally, LCEDA supports online collaboration and cloud storage, enhancing the efficiency of PCB circuit design.

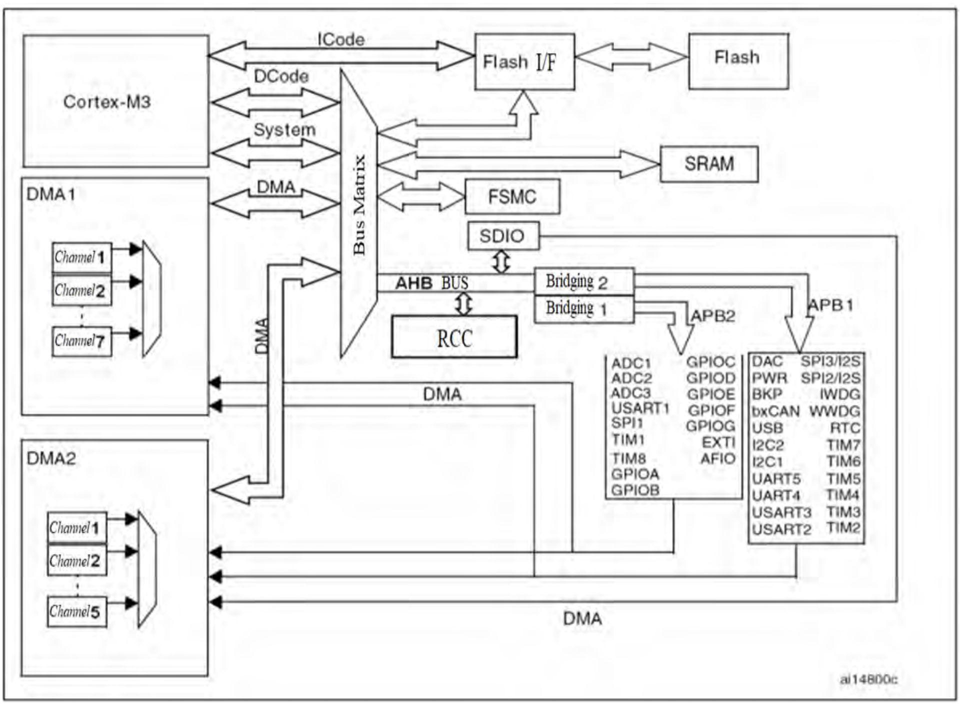
The software development for this system is mainly carried out using Keil5, a widely-used embedded development environment software, predominantly for software development and debugging in embedded systems. It integrates an editor, compiler, debugger, simulator, and microcontroller libraries, providing convenient and quick development tools along with support for libraries of numerous microcontroller models. On the Keil5 platform, there are three development approaches: library function development, register development, and bit operation. This project adopts a library function-based design method. The library function development approach follows a top-down learning methodology, tracing from the high-level application programming interface (API) to the lower levels, enabling a comprehensive understanding of the registers, the CPU’s memory distribution, boot code, and configuration of the development environment. This method simplifies development significantly, allowing the use of APIs from the library to build extensive programs and create various applications. This approach not only reduces the learning curve but also shortens the development cycle.

# 3 Hardware design of the vehicle system

## *3.1 Main control module design*

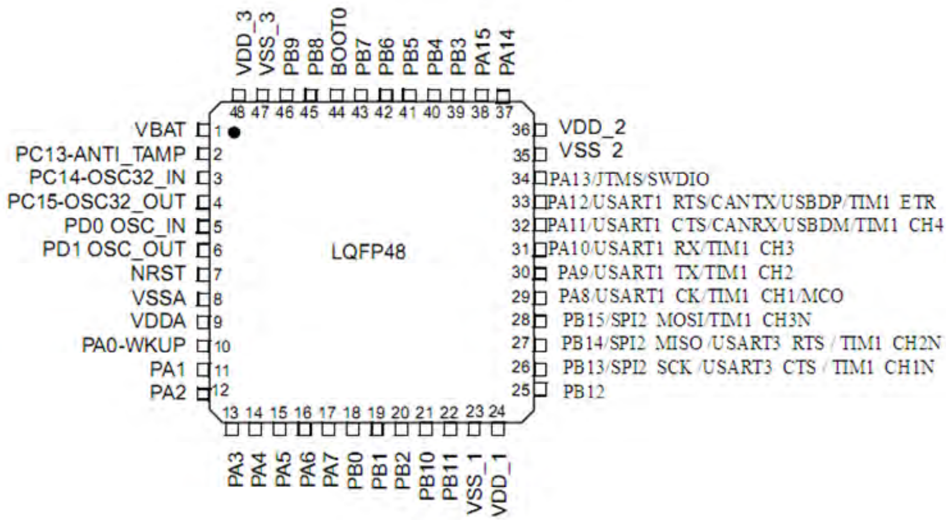
In this system, the STM32F103C8T6 is selected as the main control chip due to its balance of performance, cost, and development ease. Produced by STMicroelectronics, this 32-bit ARM Cortex-M3-based microcontroller offers a high-performance core for real-time control tasks, low power consumption, and a rich peripheral set – including multiple timers, UARTs, and PWM outputs – ideal for motor control, sensor integration, and wireless communication. Its widespread adoption ensures extensive community support, accelerating development and troubleshooting. In this design, the STM32 manages motor speed via PWM, processes infrared sensor data for line tracking, and handles bidirectional communication through JDY-31 Bluetooth and GL24S 2.4G modules, ensuring efficient and stable system operation.

**Figure 2** STM32 system structure diagram




In this system, the STM32 serves as the core controller, responsible for motor control, sensor data collection, and wireless communication. Specifically, the STM32 controls the speed and direction of motors using PWM, displays data on the OLED screen via I2C or SPI interfaces, and communicates with the JDY-31 Bluetooth module and the 2.4G wireless communication module through USART serial ports, thus facilitating vehicle control and wireless communication. Moreover, the STM32 also accomplishes the automatic line-tracking function of the vehicle, controlling the vehicle's path by processing infrared signals collected by the line-tracking module, enabling autonomous navigation within warehouses. In summary, the STM32 plays a crucial role in this system, ensuring stable and efficient operation.

**Figure 3** STM32 pin function diagram




### *3.2 Main control module circuit design*

The minimal system of a microcontroller typically consists of a crystal oscillator circuit, power supply circuit, reset circuit, and debugging circuit. As the core processor, the microcontroller operates with the crystal oscillator providing the clock signal, the power supply delivering the necessary power, and the reset circuit ensuring the microcontroller returns to its initial state. These components collectively form the basic operational framework of the microcontroller, providing the essential conditions for its operation. The specifics are as follows:

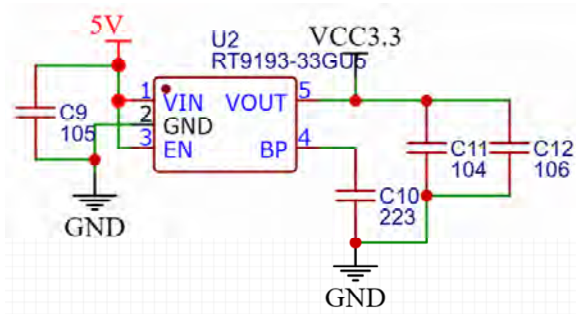
## 4 Power supply circuit

In the minimal system of a microcontroller, the power supply circuit is a crucial component. Its primary function is to provide a stable and reliable power source for the microcontroller. The design of the power supply circuit must consider factors such as the working voltage, current, and stability of the microcontroller to ensure its normal operation. In practical applications, considerations like battery life, power noise, and power consumption are also important to meet the requirements of real-world use.

In this minimal system, the power supply circuit is composed of two parts: a power supply circuit and a voltage reduction circuit. The power supply circuit utilises a standard USB interface for direct power supply, with a 5 V voltage input.

Since the maximum tolerable voltage for the STM32F103C8T6 chip is 3.3 V, it is necessary to reduce the input 5 V voltage to 3.3 V. This is achieved using an low dropout (LDO) linear voltage regulator. Filtering capacitors are added at the input and output of the linear voltage regulator to further enhance the stability of the output voltage. The schematic diagram of the voltage reduction circuit is shown in Figure 4.

**Figure 4** Voltage reduction circuit schematic diagram (see online version for colours)




## 5 Oscillator circuit

In the minimal system of a microcontroller, the oscillator circuit is imperative for providing a stable clock signal. It dictates the internal timing and operating speed of the microcontroller. Typically, this circuit comprises an oscillator, capacitors, and resistors. The oscillator's frequency is selected based on the microcontroller model and specific requirements, while the values of capacitors and resistors are calculated based on the oscillator's parameters. A stable and accurate oscillator circuit ensures normal and precise operation of the microcontroller, whereas instability in this circuit may lead to operational anomalies or errors.

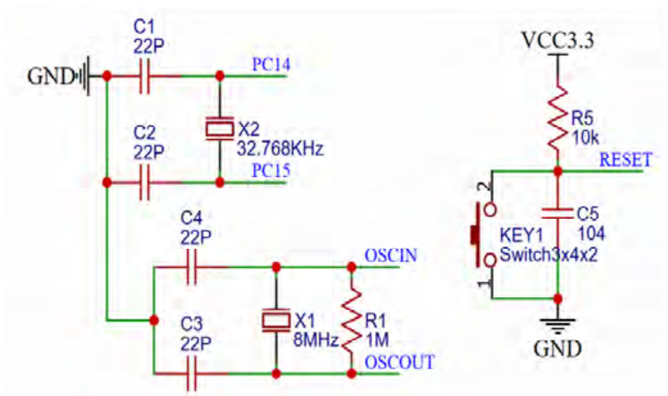
In this design, we have opted for an 8 MHz passive crystal oscillator and a 32.768 KHz passive crystal oscillator. The use of passive oscillators is preferred for their sufficient accuracy and flexibility. The 8 MHz oscillator serves as a high-speed external clock, while the 32.768 KHz oscillator functions as a low-speed external clock. This combination facilitates precise timing crucial for accurate timing circuits.

### *5.1 Reset circuit*

The reset circuit in a microcontroller's minimal system is critical for reinitialising the microcontroller during startup or under abnormal conditions, ensuring the system operates correctly. This circuit typically includes a reset circuit chip and a power supply for the reset circuit. When there's a fluctuation in the power supply voltage, the reset circuit's power supply provides a stable voltage to the reset chip, ensuring its functionality. The reset circuit chip monitors the microcontroller's power supply voltage and emits a reset signal when the voltage falls below a predefined threshold, resetting the microcontroller to its initial state. This mechanism ensures the system can start normally even under exceptional conditions.

The STM32 microcontroller supports three reset modes: power-on reset, manual reset, and software reset. In this minimal system, a manual reset approach is used. When the reset button is pressed, the RESET pin and the ground are connected, generating a low level, triggering an external reset and producing a reset pulse, thus resetting the system. The schematic diagram of this process is shown in Figure 5.

**Figure 5** Reset circuit schematic diagram (see online version for colours)




## 6 Decoupling circuit

In a microcontroller's minimal system, the decoupling circuit functions as a filtering mechanism, designed to remove high-frequency noise from the power supply signal, thereby ensuring the stable operation of the microcontroller. High-frequency noise in the power supply, originating from the resistance, inductance, and capacitance of the power lines, can disrupt the performance of the microcontroller, affecting the system's stability. The role of the decoupling circuit is to filter out these high-frequency noises, stabilising the power supply to the microcontroller and ensuring its proper functioning. Typically, the decoupling circuit consists of capacitors and resistors, where capacitors filter out the high-frequency noise, and resistors control the discharge rate of the capacitors.

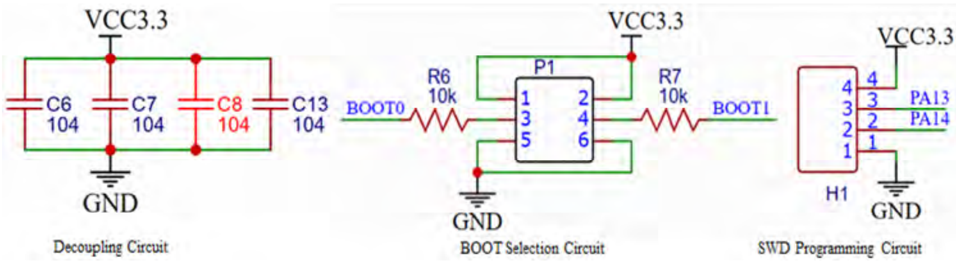
In this minimal system, four capacitors are used in parallel to form the decoupling circuit, which filters out interference and maintains stable pin voltages.

## 7 Debugging and download circuit

The STM32F103C8T6 microcontroller allows for boot mode selection by altering the BOOT connection. The modes include booting from the main flash memory, system memory, or the internal SRAM.

The serial wire debug (SWD) is a serial debugging interface that, compared to JTAG, requires only two lines: SWCLK and SWDIO. SWDIO is the serial data line for reading and writing data, while SWCLK is the serial clock line, providing the necessary clock signal. The SWD programming circuit is primarily used for programming, connecting the STM32's SWD programming interface to a downloader for program download and writing. The microcontroller's PA14 and PA13 pins serve as SWCLK and SWDIO, respectively. The schematic diagram of these circuits are shown in Figure 6.

**Figure 6** SWD programming circuit schematic diagram (see online version for colours)




## 8 Other peripheral circuits

In this minimal system, I have designed a power indicator LED, a test LED, and have externally connected all pins of the STM32 chip. The schematic diagrams for these are presented in Figure 7.

**Figure 7** Peripheral interface circuit schematic diagram (see online version for colours)

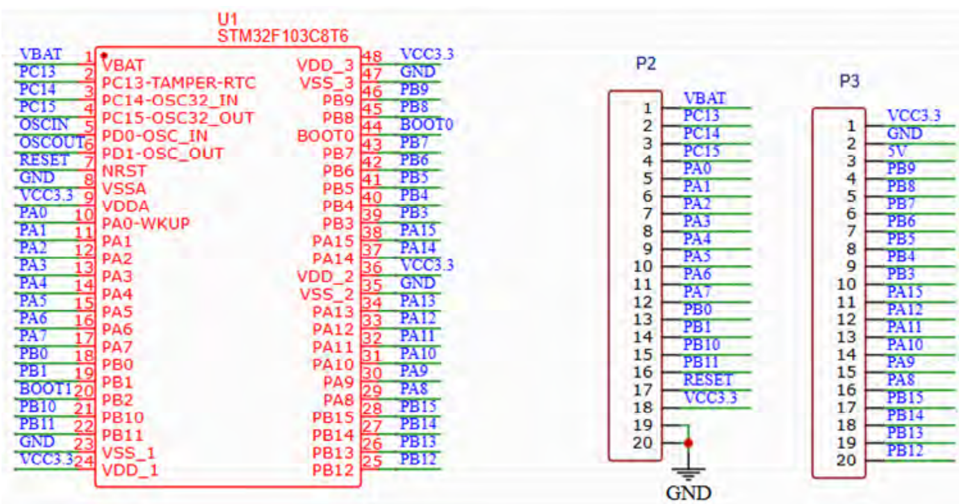
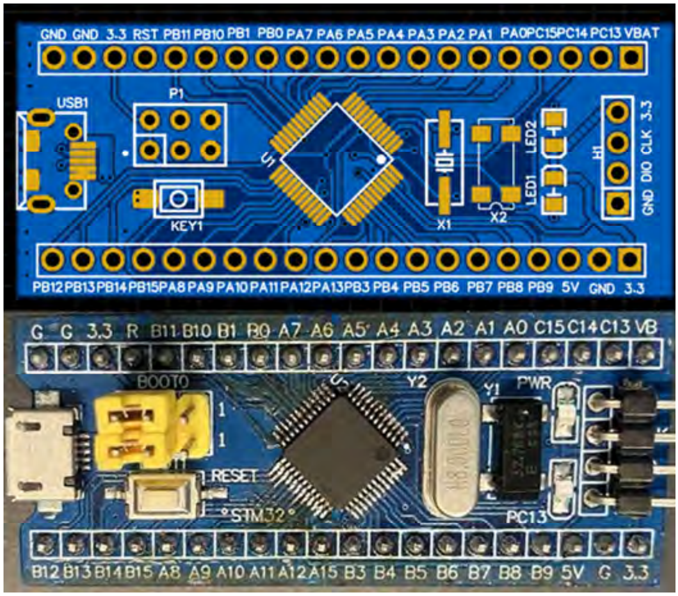



Figure 8 display the PCB design and the actual assembled minimal system of the microcontroller, respectively.

### *8.1 Wireless communication module design*

The 2.4G wireless communication module in this design utilises the GL24S chip, which employs the latest 2.4G SOC technology. Its notable features include no development requirement, a line-of-sight range exceeding 200 meters, integrated transmission and reception without the need for switching, support for serial pass-through, and provision of a communication protocol for rapid debugging. The simplicity of its debugging process and its low latency are significant advantages.

**Figure 8** Microcontroller minimal system (see online version for colours)



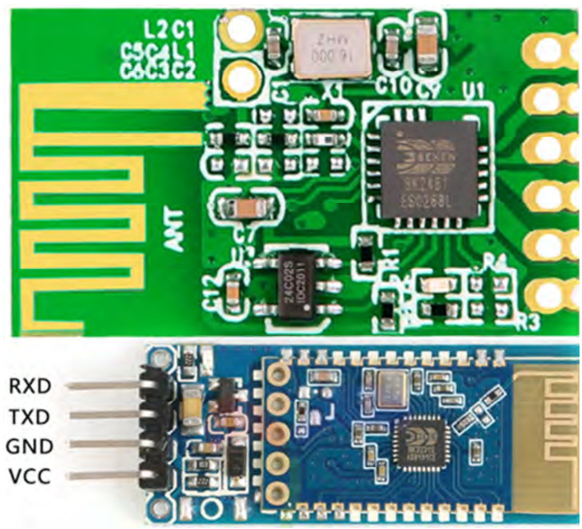

The 2.4G wireless communication module, compared to other wireless communication methods, offers broad coverage, high-speed transmission, resistance to interference, and low power consumption (the communication module is shown in Figure 15). These attributes make it highly suitable for applications in the internet of things (IoT), smart home systems, and intelligent transportation. In this design, its primary function is to facilitate inter-vehicular communication, enabling data transfer and collaborative control between small vehicles. The module's high transmission rate and stability significantly enhance communication efficiency between vehicles, thereby improving the overall system performance. Additionally, the 2.4G module supports both broadcasting and point-to-point communication, making it versatile for various scenario requirements and enhancing the flexibility of vehicle applications.

For Bluetooth wireless communication, the JDY-31 chip is employed. This module is a simple wireless communication device based on Bluetooth protocol. It is built around the BC417 single-chip Bluetooth IC and complies with the Bluetooth v2.0 standard, supporting both UART and USB interfaces.

Following the introduction of these modules, it is worth clarifying their specific roles and synergy within the system. The JDY-31 Bluetooth module is primarily utilised for receiving external commands from a supervisory computer or smartphone, such as destination warehouse numbers, enabling user-initiated control over the vehicles. Conversely, the GL24S 2.4G module is dedicated to inter-vehicle communication, facilitating the exchange of commands and feedback between the lead and following vehicles. For instance, the lead vehicle might transmit a delivery instruction to the following vehicle via the 2.4G module, which then executes the task and sends confirmation back upon completion. This dual-module architecture ensures seamless external control and efficient internal coordination, significantly enhancing the system's operational flexibility and robustness.

In this design, the Bluetooth module's role is to receive commands issued from smartphones or computers. By connecting to the vehicle's Bluetooth module, users can send destination warehouse numbers and other necessary commands. The vehicle can receive and interpret these commands to perform actions accordingly, facilitating the transport of goods between different warehouses. Compared to other communication methods, Bluetooth offers advantages like low power consumption, short-range connectivity, and ease of connection. It enables quick, simple, and flexible pairing with smartphones or computers, making it highly suitable for transmitting control commands to the vehicles in this design, the module is shown in Figure 16.

**Figure 8** GL24S 2.4g wireless communication module and JDY-31 Bluetooth module (see online version for colours)

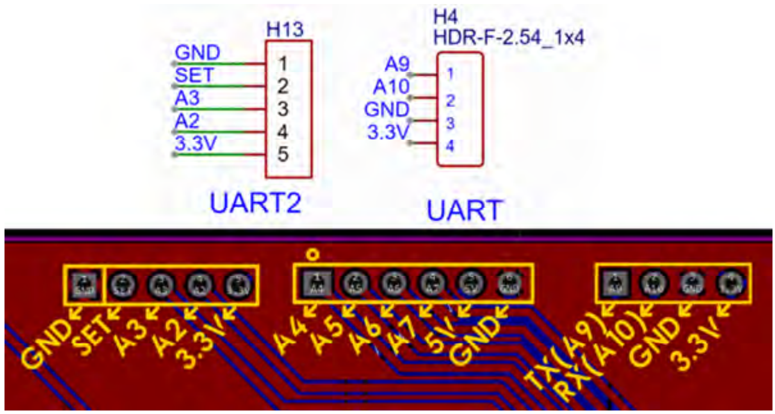



## 8.2 *Wireless communication module circuit design*

The wireless communication circuit in this design primarily consists of the GL24S 2.4G wireless module and the JDY-31 Bluetooth module (as shown in Figure 9). The GL24S module facilitates communication between vehicles, allowing for high-speed and stable data transmission. On the other hand, the JDY-31 module is used for communication with smartphones or computers, making it convenient for users to control the vehicles and issue commands. Both modules communicate with the microcontroller via the universal synchronous/asynchronous receiver-transmitter (USART) interface. The STM32F103C8T6 microcontroller features three USART peripherals, of which USART1 and USART2 are mainly employed in this system.

In this circuit design, the integration of both the GL24S and JDY-31 modules allows for a versatile communication system within the minimal system framework of the microcontroller. The GL24S module's capacity for high-speed, reliable data transfer between vehicles is crucial for effective vehicular coordination and data sharing, essential in applications like automated logistics systems or cooperative robotic tasks. Meanwhile, the JDY-31 Bluetooth module provides an accessible interface for user interaction, permitting direct control and command issuance from personal devices such as smartphones or computers. This dual-module approach not only ensures robust inter-vehicle communication but also bridges the interaction between the system and the end-user, enhancing the overall functionality and user experience of the design.

**Figure 9** Wireless communication circuit PCB layout (see online version for colours)




## 8.3 Motor drive module design

In this system, a DC motor driver from Toshiba Semiconductor is utilised, characterised by a high-current MOSFET-H bridge configuration and dual-channel output, capable of driving two motors simultaneously. The TB6612FNG, a key component in this design, is a DC motor driver based on a MOSFET H-bridge integrated circuit. Compared to traditional transistor H-bridge drivers, it offers higher efficiency.

In contrast, the L298N, with its average drive current of 600 mA per channel and a pulse peak current of 1.2 A, has only half the output load capacity. For high-power applications, the L298N requires an additional heatsink, indicating concerns regarding thermal dissipation and the external diode freewheeling circuit. Moreover, since it cannot directly drive a motor, it requires a high-power supply filter capacitor, which is a limitation in terms of minimising the system size. On the other hand, the L293D does not require an additional heatsink, and its peripheral circuitry is relatively simple. It can directly drive motors by simply connecting an external power filter capacitor, which is beneficial for reducing the system size. As for pulse width modulation (PWM) signals, the L293D's ability to handle frequencies up to 100 kHz is a significant advantage over the 5 kHz and 40 kHz frequencies supported by the aforementioned chips.

In this system, the motor driver mainly controls the movement of the vehicle. It can drive two DC motors and provide independent PWM control, enabling the vehicle to move forward, backward, turn left, and right, among other manoeuvres. This driver module is efficient, stable, and reliable, meeting the requirements for vehicular motion control in this design. Additionally, it supports various protective features like short-circuit, overheat, and under-voltage protection, ensuring the safety and reliability of the vehicle during operation. Therefore, the TB6612FNG motor drive module plays a crucial role in this design, forming the core of the vehicle's motion control, as shown in Figure 10.

**Figure 10** TB6612FNG motor drive module (see online version for colours)


This motor drive module's integration into the system underscores its capacity to handle complex motion controls and robust operational demands. The choice of the TB6612FNG, with its efficient and high-performance characteristics, ensures that the vehicles in this design can execute precise and varied movements, essential for applications requiring agile and responsive control. The module's compatibility with high-frequency PWM signals further enhances its ability to provide smooth and precise motor control, an aspect crucial for the nuanced manoeuvring of the vehicles in sophisticated systems like automated warehouses or robotic delivery services.

## 8.4 *Motor drive module circuit design*

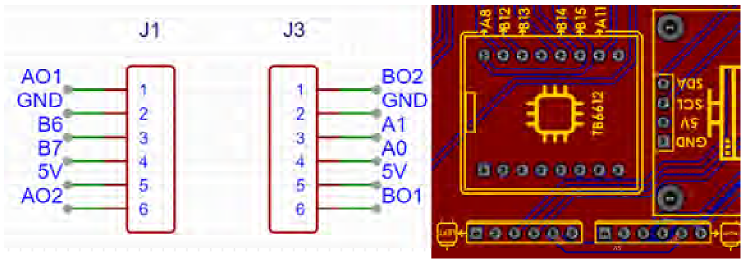
The motor drive circuit in this design is divided into two main components: the DC motor and the TB6612FNG motor drive module.

The DC motor converts electrical energy into mechanical energy. As an electromechanical actuating component, it contains a closed main magnetic circuit inside. The main magnetic flux flows through this circuit, intersecting with two electrical circuits. One circuit is used to generate magnetic flux, known as the excitation circuit, and the other, called the power circuit or armature circuit, is used for power transmission. When controlling a DC motor with a microcontroller, a drive circuit is necessary to provide sufficient driving current to the motor.

In this design, the TB6612FNG motor drive module is used. The microcontroller adjusts the motor speed by varying the duty cycle of the output PWM waves. Additionally, the DC motor in this system is equipped with an encoder, which converts the rotational angle of the motor into pulse signals. By sampling and processing these pulse signals, precise control and positional feedback of the motor can be achieved. Compared to ordinary DC motors, an encoder motor can more accurately control the rotation angle and speed of the motor, making it suitable for applications that require high precision control and positional feedback. In this design, the encoder motor is used to control the movement and positioning of the vehicle. The position signals fed back by the encoder enable precise control and positioning of the vehicle, as shown in Figure 11.

This circuit design emphasises the importance of precise motor control and feedback in the system. The integration of the TB6612FNG module and the encoder-equipped DC motor allows for sophisticated control over the vehicle's movements, crucial for applications requiring high accuracy and reliability. The use of PWM for speed control provides a flexible and efficient means of adjusting motor operation, while the encoder's feedback facilitates exact positioning and movement tracking. This combination of components ensures that the vehicle can execute complex manoeuvres with precision, essential in automated systems such as intelligent transportation or robotic handling tasks.

**Figure 11** Motor drive circuit PCB layout (see online version for colours)




## 8.5 Line tracking module design

In this system, a four-channel infrared line tracking module, model YB-MVX01, is employed. This module utilises adjustable resistors in conjunction with an integrated amplification circuit, enabling super-linear adjustment of the infrared tracking distance. It offers a broad range for fine-tuning, and the module can directly output digital signals of high and low levels, simplifying development.

Within this system, the infrared line tracking module plays a crucial role in facilitating autonomous navigation of the vehicle. By detecting the conditions of the ground beneath the vehicle, it determines the vehicle's current position and direction of travel. This information is then used to control the rotation direction and speed of the motors, enabling movements such as forward, backward, left turn, and right turn. During the process of delivering goods between warehouses, the vehicle is required to travel along predefined routes, making the function of the infrared line tracking module particularly significant. By detecting black lines on the ground, the vehicle can autonomously follow the designated routes, eliminating the need for tedious and unstable manual control, thereby enhancing the efficiency and stability of the vehicle's movement. Additionally, the infrared line tracking module can also detect the vehicle's position and direction, enabling more precise control and navigation.

The integration of the YB-MVX01 module into the system highlights its ability to streamline vehicle navigation in structured environments like warehouses. The module's capacity for super-linear distance adjustment and wide tuning range ensures flexibility and adaptability in various operational settings. By automating the route-following process, the module significantly reduces the complexity of vehicle control, allowing for more efficient and reliable operation. This capability is essential in scenarios where precise and consistent navigation is critical, such as in automated logistics and warehouse management systems, where vehicles are required to follow specific paths for material transport.

## 8.6 Display module design

In this system, we have chosen a 0.96-inch OLED display screen driven by the SSD1306 driver, with a resolution of 128 × 64, utilising the I2C interface for communication. The SSD1306 OLED display is a highly integrated screen that can be easily combined with the STM32F103C8T6 microcontroller to form a complete embedded system. Compared to the traditional LCD1602, the SSD1306 OLED display offers several advantages.

Firstly, the SSD1306 OLED display boasts a higher resolution and a wider viewing angle. Thanks to OLED technology, it provides a higher pixel density, allowing for clearer and more detailed image and text display. Additionally, the OLED screen offers a wider viewing angle, maintaining image clarity both horizontally and vertically.

Secondly, the SSD1306 OLED display consumes less power. Unlike traditional LCD screens, OLED displays do not require backlighting, which significantly reduces power consumption. This is particularly important for battery-powered embedded systems.

Lastly, the SSD1306 OLED display features a higher response speed. Due to the different working principles compared to LCD screens, OLED displays update screen content more rapidly. This makes OLED screens perform exceptionally well in applications where frequent screen refreshing is necessary.

In this system, the combination of the SSD1306 OLED display and the STM32F103C8T6 microcontroller enables various functionalities, such as displaying the operating status of the vehicle.

The integration of the SSD1306 OLED display into the system highlights its capacity to enhance user interaction and information display. Its high resolution and clarity facilitate the effective presentation of detailed information, such as status indicators, navigation data, or system alerts, essential in complex control environments. The low power consumption and fast response time of the OLED technology make it an ideal choice for mobile, battery-operated systems where efficiency and responsiveness are crucial. This combination of features makes the SSD1306 OLED display a vital component in this system, contributing to its overall functionality and user experience.

## *8.7 Power supply module design*

For the power supply in this design, two 18650 batteries are connected in series to form a battery pack capable of delivering 7.2 V. However, since the typical operating voltage for electronic components is 5 V or 3.3 V, it is necessary to step down this voltage. Generally, voltage reduction is achieved through either LDO linear regulation or DC-DC switching power supply, as shown in Figure 12.

Given the number of components in this design and the high current involved, using LDO linear regulation could lead to excessive heat in the voltage regulator and unstable output. Therefore, a DC-DC switching power supply is adopted for voltage reduction in this design.

The chosen DC-DC step-down chip is the MP1584EN. This chip is available in both dual in-line and 8-pin SOIC surface mount packages, with a 3 A output current. It accepts an input voltage range from 4.8 V to 28 V and provides an adjustable output voltage ranging from 1.23 V to 26 V. Additionally, it includes current limiting, overheat, and short-circuit protection features. Compared to the widely used LM2596, the MP1584EN is significantly smaller, which helps reduce the PCB area. The integration of the MP1584EN step-down module in this system underscores its capability to manage power efficiently for multiple components while maintaining a compact form factor, as shown in Figure 13.

Building on the power supply design, an analysis of battery life further illustrates the system’s operational capacity. The vehicle is powered by two 18650 lithium-ion batteries connected in series, providing a nominal voltage of 7.2 V and a total capacity of approximately 3,000 mAh. The system's average power consumption is estimated at 1.5 W, accounting for the combined draw of the STM32 microcontroller, TB6612FNG motor driver, sensors, and communication modules. This yields an expected battery life of approximately 14.4 hours of continuous operation (calculated as 3,000 mAh × 7.2 V ÷ 1.5 W), assuming a fully charged battery and minor power conversion losses. In practice, operational duration may vary based on task demands, such as motor usage frequency, making this a vital consideration for prolonged autonomous missions.

**Figure 12** Battery step-down module schematic diagram (see online version for colours)

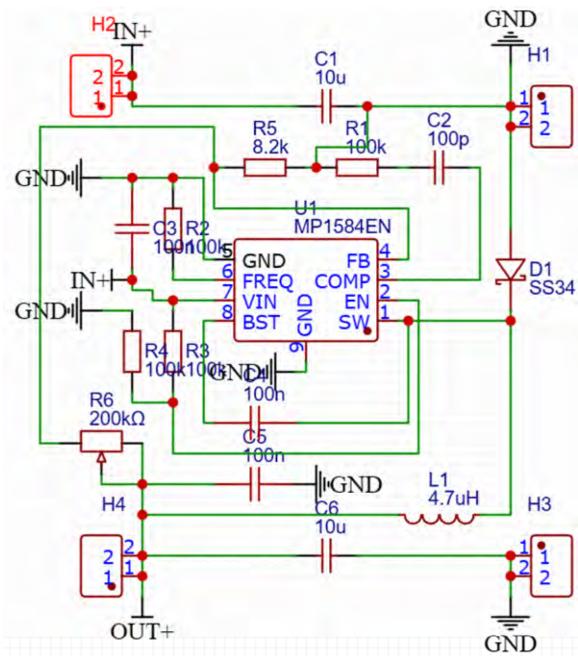
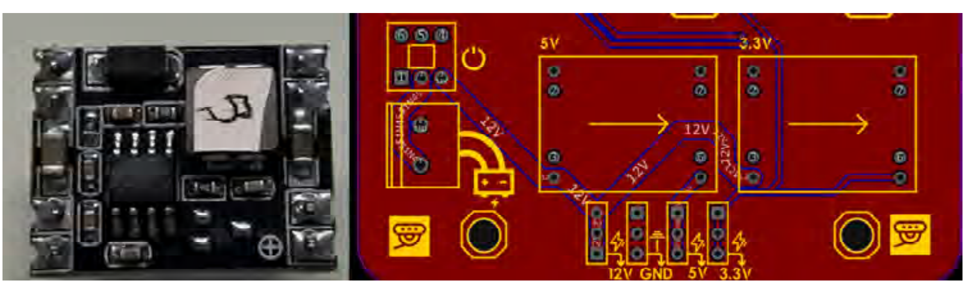
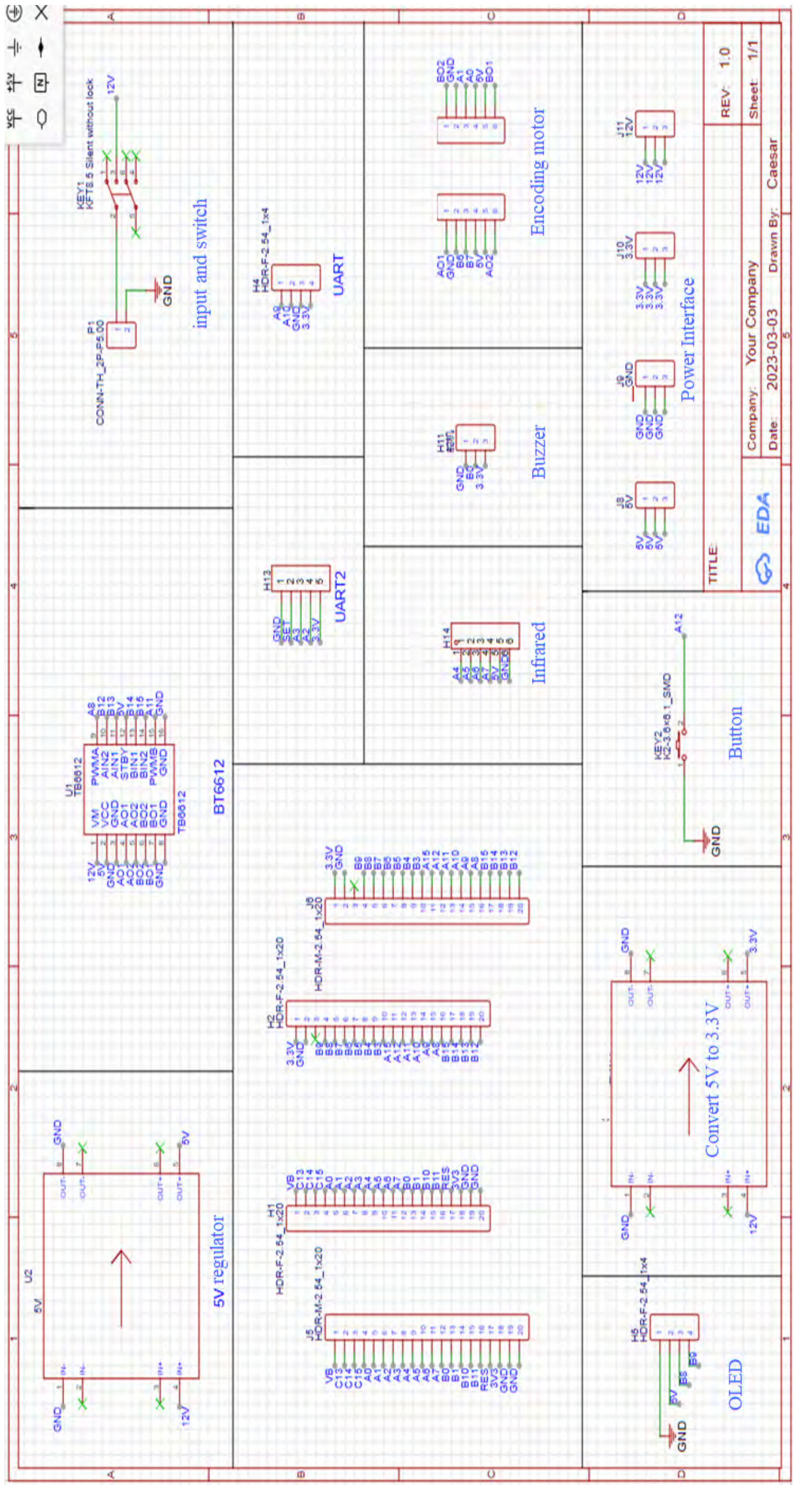



**Figure 13** PCB layout of battery step-down circuit (see online version for colours)




This choice is critical in systems where space conservation and efficient power management are essential, such as in portable or battery-operated devices. The module's protective features ensure the safety and reliability of the power supply, vital for maintaining the stability and longevity of the entire system. This thoughtful design choice contributes significantly to the overall efficacy and durability of the power supply module, making it a cornerstone of the system's architecture.

**Figure 14** Vehicle circuit schematic diagram (see online version for colours)



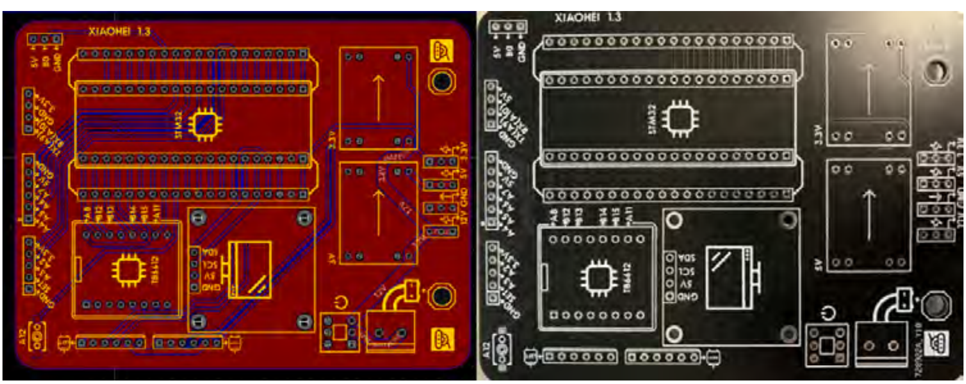
### 8.8 Overall circuit design of the vehicle

The comprehensive circuit design for this vehicle integrates all the modules discussed previously. The hardware part of this design is depicted in the schematic created in the EDA software, as shown in Figure 14.

The schematic is then translated into a printed circuit board (PCB) layout. The first step is to determine the size and shape of the board. For this design, the maximum PCB size that the vehicle can accommodate is 10 × 7.5 cm, which is the chosen dimension for this PCB. The layout process involves strategically placing the components to ensure reasonable module spacing and minimal routing distance with the least number of traces. While routing, factors such as signal interference, board thickness, and pad size need to be considered. Common routing techniques include orthogonal (right-angle) routing, straight-line routing, and 45-degree angled routing. To ensure signal stability and interference resistance, a ground plane layer is added. This ground plane needs to be connected to the pins and traces of the components. Finally, the PCB design files are produced and sent to a manufacturing facility for production.

The meticulous design of the vehicle's circuit, encompassing thoughtful component placement, routing strategy, and ground plane integration, highlights the attention to detail crucial in electronic design. The ground plane addition is particularly noteworthy as it enhances the circuit's robustness against interference, ensuring reliable operation. This comprehensive approach results in a compact, efficient, and functional PCB design tailored to the vehicle's specific requirements and constraints, thereby optimising its performance and utility. As shown in Figure 15.

**Figure 15** Vehicle circuit PCB layout and actual PCB of the vehicle circuit (see online version for colours)




## 9 Software system design of the vehicle

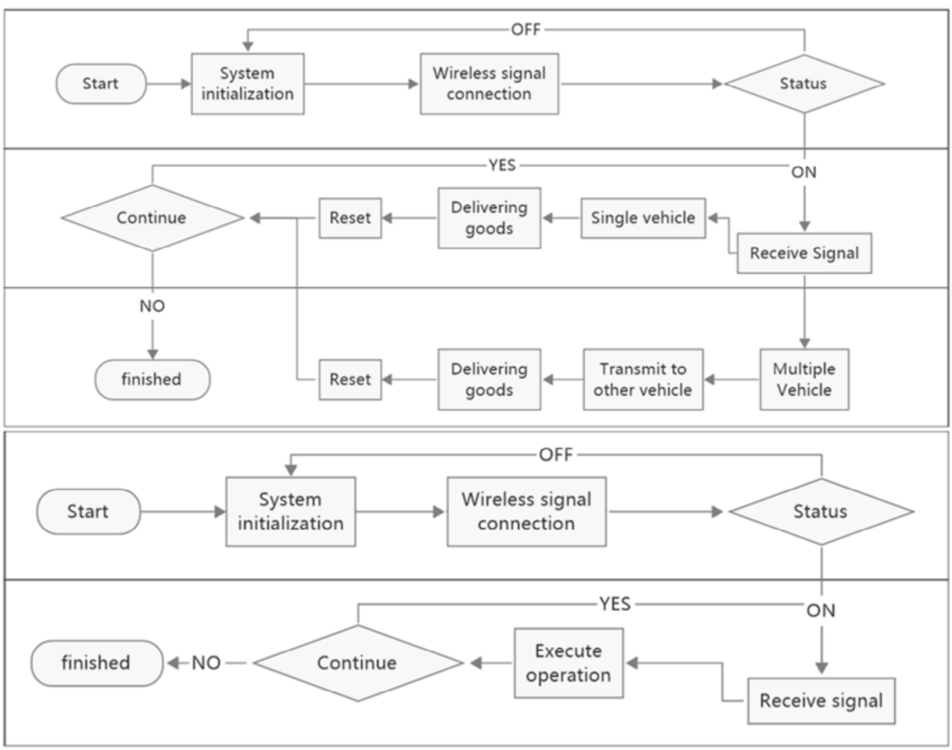
The STM32 series of microcontrollers primarily utilise two programming languages: assembly language and C language. Assembly language, being closest to the machine level, is primarily used for accessing hardware-specific features of the STM32, such as operating registers and handling interrupts. However, it has significant drawbacks like lengthy development cycles, poor code readability, and challenges in portability. Therefore, C language is chosen for this design due to its advantages.

In STM32 programming, C language is extensively used due to its readability, portability, and high degree of modularity, making it suitable for most application scenarios. Using C language for programming allows for the effective utilisation of STM32's low-level library functions, enabling rapid implementation of various functionalities. In this design, C language is used to compile programs and implement a modular program structure. This approach facilitates debugging and optimisation of the system software and also makes the program design easier for others to understand and read. The program in this design mainly consists of the main program, timer interrupt service routines, external interrupt service routines, and other modules, each performing different functions. The overall program structure is clear, making it easy to maintain and expand.

## *9.1 Main program design*

The functional flowchart of the main vehicle and its ancillary vehicles is shown in Figure 16.

**Figure 16** Vehicle program flowchart




1 Dual-vehicle delivery mode: When the main vehicle receives the dual-vehicle delivery instruction, it departs to deliver goods, followed by the subordinate vehicle. Both vehicles can deliver to different or the same warehouses. After completing the delivery, they return to their starting positions in sequence.

2 Single-vehicle delivery mode: The vehicle waits at a designated position for commands. When the main vehicle receives a single-vehicle delivery instruction, it transports the goods to the specified warehouse and then returns to the starting position.

## 9.2 *Motor drive program design*

In the STM32 series, PWM outputs are generated using timers other than the basic timers TIM6 and TIM7. The advanced timers TIM1 and TIM8 can produce up to seven PWM outputs, while the general-purpose timers TIM2 to TIM5 can generate four PWM outputs each.

**Figure 17** Motor drive core code snippet (see online version for colours)

```
void PNM_init(void)
    {
    GPIO_InitTypeDef GPIO_InitStructure;
    TIM_TimeBaseInitTypeDef TIM_TimeBaseStructure;
    TIM_OCInitTypeDef TIM_OCInitStructure;

    RCC_APB2PeriphClockCmd(RCC_APB2Periph_TIM1, ENABLE); // Enable clock for TIM1
    RCC_APB2PeriphClockCmd(RCC_APB2Periph_GPIOA, ENABLE); // Enable clock for GPIOA peripheral

    // Configure this pin to alternate function output mode, to output PWM pulse waveform of TIM1 CH1
    GPIO_InitStructure.GPIO_Pin = GPIO_Pin_8 | GPIO_Pin_11; // TIM CH1
    GPIO_InitStructure.GPIO_Mode = GPIO_Mode_AF_PP; // Alternate function push-pull output
    GPIO_InitStructure.GPIO_Speed = GPIO_Speed_50MHz;
    GPIO_Init(GPIOA, &GPIO_InitStructure);

    TIM_TimeBaseStructure.TIM_Period = 100 - 1;
    // Set the value of the period for the auto-reload register for the next update event (80KHz)
    TIM_TimeBaseStructure.TIM_Prescaler = 36 - 1;
    // Set the value of the prescaler to divide the TIMx clock frequency (no division)
    TIM_TimeBaseStructure.TIM_ClockDivision = 0;
    // Set clock division: TDTS = Tck_tim
    TIM_TimeBaseStructure.TIM_CounterMode = TIM_CounterMode_Up;
    // TIM counting up mode
    TIM_TimeBaseInit(TIM1, &TIM_TimeBaseStructure);
    // Initialize the TIMx time base unit according to the parameters specified in TIM_TimeBaseInitStruct

    TIM_OCInitStructure.TIM_OCMode = TIM_OCMode_PWM2;
    // Select timer mode: TIM Pulse Width Modulation Mode 2
    TIM_OCInitStructure.TIM_OutputState = TIM_OutputState_Enable;
    // Compare output enable
    TIM_OCInitStructure.TIM_Pulse = 0;
    // Set the pulse value to be loaded into the capture compare register
    TIM_OCInitStructure.TIM_OCPolarity = TIM_OCPolarity_High;
    // Output polarity: TIM output compare polarity high
    TIM_OC1Init(TIM1, &TIM_OCInitStructure);
    // Initialize the peripheral TIMx according to the parameters specified in TIM_OCInitStruct
    TIM_OC1PreloadConfig(TIM1, TIM_OCPreload_Enable);
    // Enable preload for CH1
    TIM_OC4Init(TIM1, &TIM_OCInitStructure);
    // Initialize the peripheral TIMx according to the parameters specified in TIM_OCInitStruct
    TIM_OC4PreloadConfig(TIM1, TIM_OCPreload_Enable);
    // Enable preload for CH4
    TIM_CtrlPWMOutputs(TIM1, ENABLE);
    // MOE Main output enable
    // TIM_ARRPreloadConfig(TIM1, ENABLE); // Enable preload register on ARR for TIMx
    TIM_Cmd(TIM1, ENABLE); // Enable TIM1
```

In this system, the timer/counter TIM2 on the STM32F103C8T6 is configured to operate in PWM output mode. The frequency of TIM2 is set to 72 MHz, and the PWM duty cycle is adjustable based on actual feedback. The TB6612FNG motor drive module is then connected to the general-purpose input/output (GPIO) pins of the STM32F103C8T6. In this design, the PWM output is connected to the PWM input pin of the TB6612FNG, and the direction control inputs are connected to the IN1 and IN2 pins of the TB6612FNG module. The control program is written such that the PWM output of the STM32F103C8T6 controls the motor speed, and changing the levels of the IN1 and IN2 pins controls the motor's direction and speed.

The core code snippet is illustrated in Figure 17.

This motor drive program design is crucial for the precise control of the vehicle's movements. By using the PWM output from the STM32F103C8T6, the system can finely adjust the motor speed, which is essential for tasks requiring accurate speed regulation and direction control. The ability to modify the duty cycle of the PWM signal allows for a wide range of speed control, from very slow to fast, providing versatility in the vehicle's operational capabilities. The integration of direction control through the IN1 and IN2 pins adds another layer of manoeuvrability, enabling complex movements such as turning and reversing. This comprehensive approach to motor control is vital for the vehicle's functionality in automated tasks, ensuring both precision and reliability in its movements.

### *9.3 Display program design*

The display program in this design drives the OLED panel through the I2C protocol to show relevant parameters on the screen. The OLED screen operates in different addressing modes: page addressing mode, horizontal addressing mode, and vertical addressing mode.

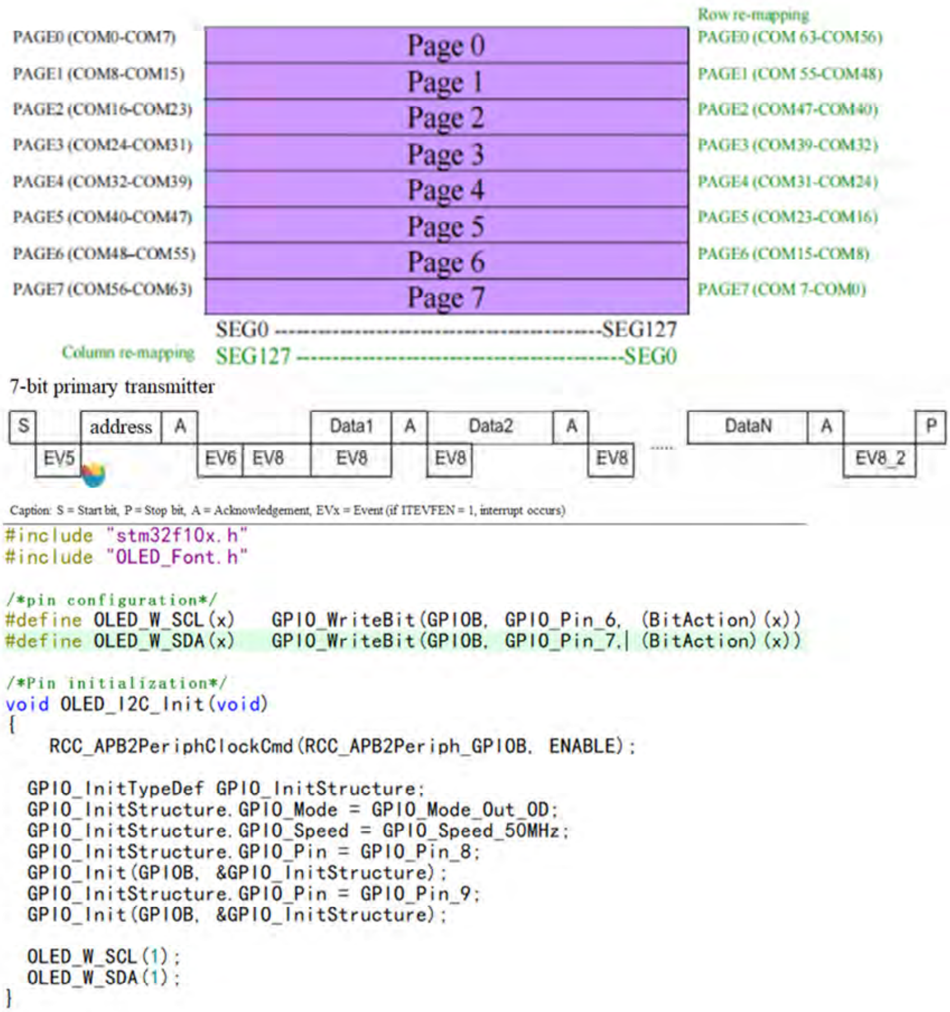
Here, we focus on the most commonly used page addressing mode. The page addressing mode divides the OLED screen into eight pages. Essentially, it splits the screen width into eight equal parts. For example, to display a dot at the beginning of the third row in the first column, you would configure it bit by bit as 0000 0100 ($0 \times 08$). The configuration steps for hardware I2C, Involve sending data and continuously receiving different acknowledgments, and the core code snippet for the OLED display, is shown in Figure 18.

The display program design is essential for providing user-friendly interaction with the system. By implementing page addressing mode on the OLED display, the program can efficiently manage the screen area, allowing for precise control over what is displayed. This feature is particularly beneficial for showing small and detailed information, such as system status, sensor readings, or navigational instructions, in a clear and organised manner.

The use of hardware I2C for communication with the OLED panel ensures reliable and high-speed data transfer, which is crucial for real-time display updates. The I2C protocol's simplicity also contributes to a more streamlined and efficient code, facilitating easier maintenance and potential upgrades.

In summary, this display program design, incorporating page addressing and I2C communication, significantly enhances the system's overall usability and functionality, making it more intuitive and accessible for users.

**Figure 18** Hardware configuration steps and core code fragments (see online version for colours)




## 9.4 *PID line tracking program design*

The design of this vehicle necessitates precise and smooth delivery of goods to designated warehouses. Traditional line tracking methods, which involve simple left and right steering adjustments based on the vehicle's heading, may lead to instability, especially at higher speeds, as the vehicle tends to oscillate around the line.

To address this, proportional-integral-derivative (PID) control is employed to stabilise and smoothen the line tracking process. PID control is a linear feedback control system that adjusts the control output based on the difference (error) between a desired setpoint ($r(t)$) and the actual output ($y(t)$). The PID controller calculates the control output $u(t)$ by summing the proportional, integral, and derivative responses to the error $e(t) = r(t) - y(t)$. The general form of the PID control algorithm in the continuous time domain is as follows:

$$u(t) = K_p e(t) + \frac{k_i}{T_i} \int e(t)dt + K_d T_d \frac{de(t)}{dt}$$

Here, $K_p$ is the proportional gain, $T_i$ is the integral time constant, and $T_d$ is the derivative time constant.

In line tracking control, the input error *e*(t) is derived from the line tracking sensor's output. The output $u(t)$ from the PID controller directly controls the vehicle's steering.

Having outlined the PID framework, it is useful to contrast it with other control strategies to justify its adoption. Compared to fuzzy logic, which demands complex rule-based tuning and greater computational resources, or Kalman filters, which excel in state estimation but require significant processing power, PID offers a simpler yet effective solution for real-time embedded systems. Its proportional, integral, and derivative terms enable rapid error correction, steady-state precision, and reduced oscillations, ensuring stable and smooth navigation along the designated path. This balance of efficiency and ease of implementation makes PID the preferred choice for this design, avoiding the overhead of more resource-intensive alternatives.

Implementing PID closed-loop control in this system involves the following steps:

- Initialisation: Setup the PWM output and direction control pins for the encoded DC motor, along with the encoder counter. Initialise the infrared line tracking sensor to read the state of the infrared signals through corresponding IO ports.
- Main program: Calculate the target speed for the encoded DC motor using the PID algorithm and compare it with the actual speed to obtain the error value.
- PWM output: Based on the error value, calculate the appropriate PWM duty cycle and output it to the motor's PWM control pin. Set the direction control pins to the appropriate state (forward or reverse) to control the motor rotation.
- Speed regulation: Continuously read the encoder counter to calculate the actual speed of the motor. Compare this with the target speed to get a new error value and adjust the motor's speed accordingly, repeating this step until a stable speed is achieved.
- Line detection: When the infrared line tracking sensor detects a black line, reduce or stop the motor to prevent the vehicle from deviating from its path. Adjust the parameters in the PID algorithm to achieve the desired control strategy.

The implementation of PID control in this vehicle's line tracking system enhances its ability to navigate precisely along the predetermined path, ensuring smooth and stable movement. This is crucial in applications like automated warehouses, where the accuracy and consistency of the vehicle's path are paramount. The PID control system's adaptability to different operational conditions and its ability to minimise oscillations around the line make it an ideal choice for this application.

## *9.5 Wireless communication program design*

In the JDY-31 Bluetooth serial communication module, there are mainly two working modes: command response mode and automatic connection mode. In these modes, the module can assume one of three roles: master, slave, and loopback. In the master role, the module automatically connects for data transfer based on pre-set configurations in automatic connection mode. In the slave role, the module executes AT commands in command response mode. Users can send various AT commands to set control parameters or issue control commands.

**Table 1** JDY-31 at command table

| *JDY-31 AT command table* | |
|---|---|
| AT+VERSION | Query firmware version |
| AT+RESET | Soft reset |
| AT+PIN | Set or query connection password |
| AT+BAUD | Set or query baud rate |
| AT+NAME | Set or query broadcast name |

In this design, the JDY-31 Bluetooth module is initially configured using AT commands. It is then connected to the host computer via Bluetooth protocol, receiving instructions sent by the host. These instructions are processed by the STM32 microcontroller via UART.

The GL24S wireless pass-through module uses 2.4G wireless communication. Like the JDY-31, it is configured using AT commands. The main and subordinate vehicles are paired, and then the main vehicle sends data to the GL24S wireless chip via UART. After processing, the chip transmits the data to the subordinate vehicle's wireless chip, which then sends it to the subordinate vehicle's microcontroller via UART to execute the corresponding commands.

The wireless communication program design is crucial for enabling efficient and reliable data exchange between the vehicle and the control system, as well as between the main and subordinate vehicles. The use of both JDY-31 and GL24S modules allows for versatile communication capabilities, catering to different communication needs in the system. The incorporation of AT commands for module configuration provides flexibility and ease of setup, making the system adaptable to various operational scenarios. This design ensures that the vehicles can receive and execute commands accurately, which is vital for the system's overall performance and reliability.

## 10 System testing

To ensure the effective implementation of the entire system, this chapter involves testing both hardware and software, followed by an overall system test. Segregated testing allows for better problem identification and clearly demonstrates the entire design process.

### *10.1 System hardware testing*

1 STM32 testing

The first step is to test whether the minimal system board functions properly, as it is the most fundamental and crucial part of the entire design. The board is connected to the computer via a programmer, and a simple program that blinks an LED is written into it. If the LED blinks as expected, it confirms that the minimal system board is working correctly.

2 PCB testing

The PCB connects various modules and ensures that they are supplied with power and can communicate with each other. After connecting the power, use a multimeter to check if the voltages in different parts are normal. Then, use the continuity test function of the multimeter to verify the connections between various points.

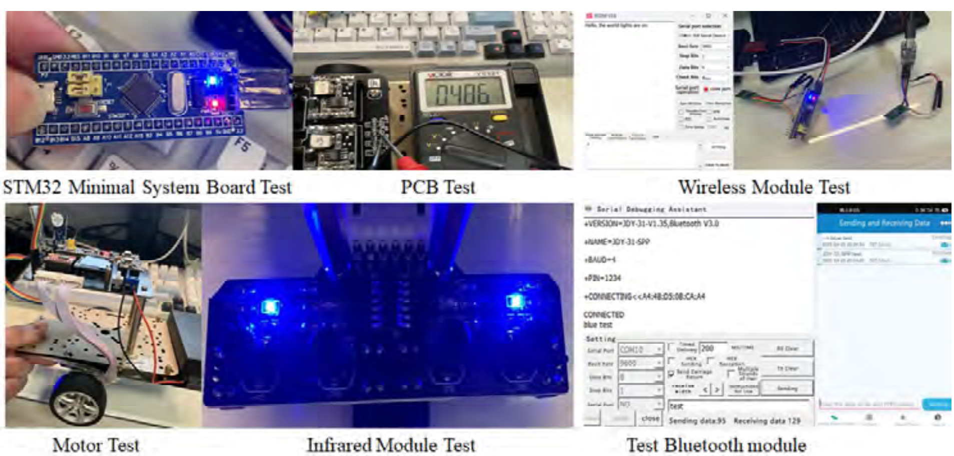
3 Motor testing

Mount the encoded motors and the motor drive module on the PCB. Write and upload a program that can drive the motors. Test the forward and reverse rotation of the motors, the controllability of motor speed, and whether the encoders return values to the STM32 as expected.

4 Infrared module testing

Mount the infrared module on the vehicle, connect it to the power supply, and connect the module's IO ports to the corresponding IO ports of the STM32. Move the module over a black line; the module's LED should correctly indicate detection of the black line, and the STM32 should receive feedback from the infrared module.

**Figure 19** System hardware testing (see online version for colours)




5 Wireless module testing

Wireless modules are crucial to the system. Test the 2.4G wireless module first: connect one module to the computer via a serial port and the other to the STM32. Upload a program to the STM32 that lights up an LED and returns the current light status when it receives a wireless signal. Check if the LED on the STM32 lights up and the serial port assistant on the computer receives the status.

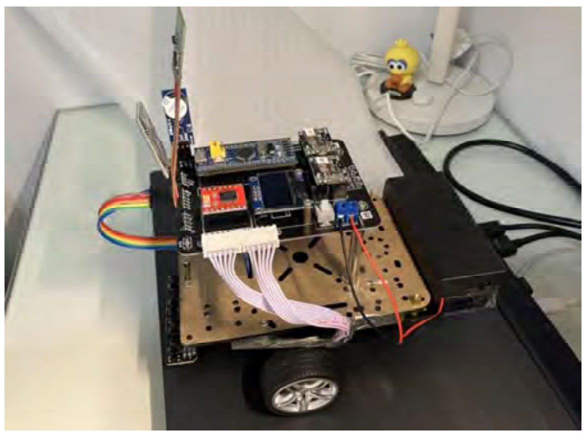
Then test the Bluetooth module. Connect it to the computer via a serial port and use a Bluetooth serial port assistant to send AT commands. Verify that the module communicates properly with the computer. Then use a smartphone to connect to the

Bluetooth module, send 'blue test' from the phone, and ensure that the computer's serial port assistant receives and displays 'blue test'. The system hardware test is shown in Figure 19.

## 10.2 Integrated system test

Assemble all modules onto the PCB and mount the PCB onto the vehicle. The final assembled vehicle is as shown. Design a simulated warehouse map on the ground as shown in Figure 20. Place two vehicles at the designated starting positions and send instructions via Bluetooth for single or dual vehicle delivery modes.

**Figure 20** Assembled vehicle (see online version for colours)



In single vehicle delivery mode, the vehicle communicates with the host machine via Bluetooth to receive the designated warehouse number. It then uses infrared line tracking technology to follow the black line on the ground to the specified warehouse. After unloading at the warehouse, confirmed by pressing a button, the vehicle returns along the same path to the starting position.

In dual vehicle delivery mode, the process is more complex. The main vehicle receives warehouse numbers and conveys them to the subordinate vehicle, so it knows where to deliver the goods. In this mode, the coordination between the main and subordinate vehicles is crucial. They must operate without conflicting with each other to avoid accidents or loss of goods.

Throughout the delivery process, the vehicles follow instructions and paths accurately. If both vehicles are delivering to the same warehouse, the subordinate vehicle waits at the nearest warehouse until the main vehicle has delivered its goods.

## 10.3 Test results and analysis

The intelligent vehicle system has been successfully tested and debugged. The motor drivers and wireless modules operate as expected. Test results of the infrared line tracking module indicate that the vehicle can accurately follow a black line and maintain stable alignment with it. The use of encoded DC motors and the PID control algorithm has shown that the vehicle's speed can be stably maintained at the set value and can respond quickly to command changes for speed adjustments.

In summary, the testing and debugging process of the intelligent vehicle involved multiple components, including the motor drive module, wireless module, and display module. Through multiple tests and necessary adjustments to the code and hardware connections, the functionality of the vehicle was confirmed, demonstrating its wireless communication capabilities.

## 11 Conclusions

The successful implementation and testing of the intelligent vehicle system underscore its potential for practical applications. The system's ability to precisely follow predetermined paths and respond to wireless commands makes it an ideal candidate for automated tasks such as warehouse management or delivery services. The integration of various modules, including motor control, wireless communication, and line tracking, highlights the system's versatility and adaptability.

Furthermore, the use of PID control for motor speed regulation proves to be effective for maintaining stable motion and rapid response to control inputs. This feature is crucial in environments where precision and reliability are paramount. The testing results also emphasise the importance of thorough system integration and debugging, ensuring that each component functions harmoniously within the overall system.

In conclusion, this project demonstrates the feasibility and effectiveness of designing and implementing an intelligent vehicle system capable of autonomous navigation and wireless control. Its successful deployment paves the way for future advancements and applications in the field of autonomous vehicles and robotics.

Looking ahead, this system's capabilities could be further expanded to meet evolving demands. Scaling the design to coordinate larger fleets of vehicles could enable sophisticated logistics operations, such as simultaneous multi-point deliveries. Incorporating advanced algorithms, such as machine learning for path optimisation or predictive maintenance, might enhance autonomy and efficiency. Furthermore, integrating advanced sensors like LiDAR or cameras could improve environmental perception, enabling navigation in unstructured environments. These potential enhancements would position the system as a versatile foundation for next-generation autonomous logistics and robotics applications, opening new avenues for research and deployment.

## Declarations

All authors declare that they have no conflicts of interest.

## Acknowledgements

This research was supported by Scientific Research Key Project of Hunan Education Department (No. 24A0791) and (No. 24B0926) and (No. 24A0786) and (21A0592). Changsha Municipal Natural Science Foundation [kq2014063]. Provincial Natural Science Foundation of Hunan (2023JJ30101) and (2025JJ70437). Ministry of Education Humanities and Social Sciences Research Youth Fund Project (22YJCZH025).

## References

Armstrong, J. and Helps, C.R. (2022) *Ac 2007-1242: Comparative Evaluation of Zigbee and Bluetooth: Embedded Wireless Network Technologies for Students and Designers*.

Bento, L.C., Parafita, R., Rakha, H.A. and Nunes, U.J. (2019) 'A study of the environmental impacts of intelligent automated vehicle control at intersections via v2v and v2i communications', *Journal of Intelligent Transportation Systems*, pp.1–19.

Billa, A., Shayea, I., Alhammadi, A., Abdullah, Q. and Roslee, M. (2020) 'An overview of indoor localization technologies: toward IoT navigation services'.

Bogue, R. (2021) 'New technologies for robotic tactile sensing and navigation', *Industrial Robot*, Vol. 48, No. 4, pp.478–483.

Cai, R., Wang, X., Wang, Y., Xiao, S. and Wang, W. (2014) 'A novel and effective user scheduling scheme for heterogeneous scenario in LTE-A system', *IEEE International Conference on Computer & Information Technology*, IEEE.

Cansizoglu, H. and Elrefaie, A.F. (2017) 'A new paradigm in high-speed and high-efficiency silicon photodiodes for communication-part II: device and VLSI integration challenges for low-dimensional structures', *IEEE Transactions on Electron Devices*, No. 99, pp.1–10.

Chow, J.C.K., Michael, P., Marco, S. and Mohannad, A.D. (2018) 'Indoor tracking, mapping, and navigation: algorithms, technologies, and applications'.

Desai, K. (2013) *Wireless Communication Technologies*, John Wiley & Sons, Ltd.

Dotoli, M., Maione, B. and Turchiano, B. (2022) *Fuzzy-Supervised PID Control: Experimental Results*.

Duan, S. (2018) *Honeycomb Structure and Key Technologies of 5G Wireless Communication Network*.

Frigerio, N. and Matta, A. (2014) 'Energy-efficient control strategies for machine tools with stochastic arrivals', *IEEE Transactions on Automation Science & Engineering*, Vol. 12, No. 1, pp.50–61.

Gueltekin, O., Cinar, E., Oezkan, K. and Yazici, A. (2022) 'Multisensory data fusion-based deep learning approach for fault diagnosis of an industrial autonomous transfer vehicle', *Expert Systems with Application*, August.

Hamdan, H. (2018) *Proceedings of the International Conference on Advanced Wireless Information, Data, and Communication Technologies (AWICT 2017)*.

Hu, C., Wang, R., Yan, F., Huang, Y., Wang, H. and Wei, C. (2018) 'Differential steering based yaw stabilization using ISMC for independently actuated electric vehicles', *IEEE Transactions on Intelligent Transportation Systems*, No. 2, pp.1–12.

Jang, W.S. and Skibniewski, M. (2022) 'Wireless sensor technologies for automated tracking and monitoring of construction materials utilizing zigbee networks'.

Joseph, E. and Oladayo, O.J. (2022) 'Design and implementation of an autonomous firefighting robot with wireless control', *International Journal of Embedded Systems & Emerging Technologies*.

Jozsef, S. (2021) 'Real-time lane line tracking algorithm to mini vehicles', *Transport and Telecommunication Journal*, Vol. 22, No. 4, pp.461–470.

Kaewpukdee, A. and Uthansakul, P. (2023) *Characteristic of Line-Of-Sight In Infrastructure-To-Vehicle Visible Light Communication using MIMO Technique*, No. 1, p.24.

Kamrani, M., Concas, S., Kourtellis, A., Rabbani, M., Kummetha, V.C. and Dokur, O. (2023) 'Drivers' reactions to connected vehicle forward collision warnings: leveraging real-world data from the thea cv pilot', *Transportation Research, Part F. Traffic Psychology and Behaviour*, January, Vol. 92, pp.108–120.

Khan, A.R., Jamlos, M.F., Osman, N., Ishak, M.I., Dzaharudin, F., Yeow, Y.K. et al. (2022) *DSRC Technology in Vehicle-To-Vehicle (V2V) and Vehicle-To-Infrastructure (V2I) IoT System for Intelligent Transportation System (ITS): A Review*.

Ko, J. and Lee, J. (2021) 'Discovering research areas from patents: a case study in autonomous vehicles industry', *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*, IEEE.

Lakshmanan, S., Richardson, P. and Yi, Y. (2020) *Line-of-Sight Optical Communication for Vehicle-to-Vehicle (V2V) and Vehicle-to-Infrastructure (V2I) Mobile Communication Networks*, us2020177276a1.

Maier, A., Sharp, A. and Vagapov, Y. (2017) 'Comparative analysis and practical implementation of the ESP32 microcontroller module for the internet of things', *7th International Conference on Internet Technologies and Applications*.

Nguyen, K-G., Tervo, O., Vu, Q-D., Juntti, M. and Tran, L-N. (2017) 'Energy-efficient transmission strategies for multiantenna downlink', *IEEE*.

Roccaforte, F., Fiorenza, P., Greco, G. et al. (2018) 'Emerging trends in wide band gap semiconductors (sic and gan) technology for power devices', *Microelectronic Engineering*.

Santos, L., Costa, T., Caldeira João, M.L.P. and Soares Vasco, N.G.J. (2022) 'Performance assessment of ESP8266 wireless mesh networks', *Information*, Vol. 13, No. 5, pp.210–211.

Schü, B., Braun, T., Otten, S. and Sax, E. (2020) 'SceML: a graphical modeling framework for scenario-based testing of autonomous vehicles', *MODELS*.

Sharma, S., Gupta, L.R. and Dubey, M.K. (2020) 'A review on emerging technologies in augmented reality navigation systems'.

Tsurumi, S. and Fujii, T. (2018) 'Reliable vehicle-to-vehicle communication using spectrum environment map', *2018 International Conference on Information Networking (ICOIN)*, IEEE Computer Society.

Yang, T., Tang, H., Bai, C., Liu, J., Hao, J., Meng, Z. et al. (2021) *Exploration in Deep Reinforcement Learning: A Comprehensive Survey*, arXiv e-prints.

Zhang, Z., Cao, S. and Wang, Y. (2019) 'A long-range 2.4g network system and scheduling scheme for aquatic environmental monitoring', *Electronics*, Vol. 8, No. 8, p.909.