Design of Intelligent Logistics Car based on STM32

Le Liu¹, Jiawen Xie², Wenxin Wu², Gang Du¹, *

¹ School of Information Engineering, China University of Geosciences, Beijing 100083, China
² School of Engineering and Technology, China University of Geosciences, Beijing 100083, China

*Corresponding author e-mail: dugang2021@cug.edu.cn

Abstract. In the increasingly developing logistics industry, the application of intelligent logistics cars is becoming more and more common. This paper mainly introduces a design scheme of intelligent logistics cars based on the STM32F103ZET6 single-chip microcomputer and OpenMV programmable camera. Its main process is as follows. First, through OpenMV, the car can scan a QR code to read relevant information of the task. Then, OpenMV recognizes the color of the object, and the single-chip microcomputer controls the manipulator arm to grab the object of the corresponding color. Next, the car will carry the object to the designated position along the designated route. Finally, the process is repeated several times until the delivery task is completed. The study of intelligent cars has very important practical value and theoretical significance.

Keywords: Logistics Car, STM32, OpenMV, PID, Mecanum Wheel.

1. Introduction
Since the beginning of the 21st century, the scale of China's logistics industry has rapidly grown. The level of logistics service has improved significantly. The environment and conditions for its development have been continuously improved, laying a solid foundation for further accelerating the development of China's logistics industry. Nowadays, more and more people choose online shopping, which makes the logistics and express delivery industry flourish. In the past, the logistics industry was generally low-tech. However, in recent years, with the rapid development of the Internet and other related technologies, the automation level of the logistics industry has been enhanced. Advanced Internet technology and the development of artificial intelligence have improved the technology of intelligent mobile robots. At home and abroad, many intelligent and automatic cars have been widely used. Especially in the logistics industry, many mechanical and repetitive labor of delivering goods can be completely replaced by intelligent robots. In the electronic commercial warehouse, we can see many working scenes of intelligent logistics cars. Not only can it reduce the repetitive labor of machinery, but also improve work productivity. Therefore, the research on how to design intelligent logistics cars has important practical value, and is helpful to many industries.
2. Overall Structure Design
Generally speaking, intelligent logistics cars are mainly composed of three parts: control system, detection system and power system [1].

In terms of control system, STM32F103ZET6 single-chip microcomputer is used as the control core for data processing and analysis. In terms of detection system, the car uses OpenMV to realize the function of identifying objects, and two kinds of sensors are used at the same time: ultrasonic sensors and gray sensors. In terms of power system, the car uses the JX6221 steering gear as the steering gear of the manipulator arm and the 6-DOF manipulator arm. Also, the car uses the JGB37-520 encoder deceleration motors and drive modules.

Our design scheme includes mechanical structure, circuit structure and control structure.

2.1. Mechanical Structure Design
The main mechanical structures are 4 Mecanum wheels, a 6-DOF manipulator arm, etc. The Mecanum Wheel has many advantages. It can move in all directions compared to traditional wheels, allowing the car to move in all directions. However, the disadvantage is that the wheel is heavier than normal wheels of the same size. Each wheel needs to be driven by a separate motor, which brings difficulties and inconvenience to the process of design. The function of the manipulator arm is to grasp the object. The 6-DOF manipulator arm has a large range of movement, which brings convenience for grasping objects. We have assembled the whole car shown in Fig. 1.

![Fig.1 Physical Picture of Intelligent Logistics Car](image)

2.2. Circuit Design
The circuit design includes power supply module, step-down module and drive module.

- Power supply: The car is powered by 12V Lithium batteries.
- Step-down module: We use 5V and 3.3V voltage regulator modules. Among them, the 5V one is used for ultrasonic sensors, gray sensors, OpenMV, encoders. The 3.3V one is used for power supply of single-chip microcomputer.
- Driven module: The basic principle of driving circuit is H-bridge driving principle. We use two double H-bridge motor drives to save space and improve efficiency.

2.3. Control Program Design
In order to make the car work better and play an important role, we carefully study the actual warehouse situation. Most of porters start to deliver cargoes after receiving the task and return to the starting point when finishing delivery task. Therefore, based on the actual logistics industry situation, we design a approximate program logic in Fig. 2.
3. Visual Recognition Module

3.1. OpenMV Introduction
OpenMV is a kind of programmable camera with the STM32F765VIT6 as the control core. It has many functions, such as color recognition, shape recognition, QR code-scanning recognition, face recognition, etc. In addition, it has other functions such as serial communication. We can use the MicroPython programming language to write a script to achieve corresponding functions. OpenMV has powerful functions and is convenient for users to apply in engineering, and its transmission efficiency and stability are quite high [2].

3.2. Working Process
First, the car moves to the QR code area, and OpenMV scans the QR code. It is worth noting that OpenMV will print out a string containing task information, and we need to use the code.payload() function to extract the QR code for next processing and transmission. We use the method of numbering different colors (e.g. Red is 1, blue is 2 and green is 3, etc). After the data is read by OpenMV, it will be stored so that the robotic arm can finish the task of grasping objects in order.

Then, the car moves to the object area and OpenMV starts color recognition. In the process of color recognition, we use the find_blobs function and its default entry parameter is tuple called Thresholds. Thresholds represent a tuple containing color thresholds that identifies target color. There are 6 parameters in the tuple, which are the maximums and minimums of L, A and B. L stands for light and shade, A stands for red and green, and B stands for yellow and blue. OpenMV IDE has its own threshold editor which we can use to debug threshold ranges. The threshold of the same color can vary greatly under different lights, so we should pay extra attention to the effect of light when debugging. A set of
thresholds is supposed to be debugged multiple times to achieve the best results. In the threshold editor, black represents other colors and white represents the target color. The principle of debugging is to make the target area as white as possible and the interference area as black as possible. Fig.3 is quite clear and good, so we select the threshold at this time.

![Fig.3 Threshold Editor Interface](image)

In order to make the debugging process more clear, after the target color is recognized by OpenMV, we draw a rectangular box on the image of the target object (it roughly surrounds the object), and draw a cross in the center of the rectangular box to mark the target center. Conveniently, OpenMV can print out the x and y coordinates of the object in serial terminal as shown in Fig.4, which provides convenient conditions for the movement of the manipulator arm.

![Fig.4 Serial Terminal](image)

### 4. PI Control Method

#### 4.1. PI Control

Since we use Mecanum Wheel, an omnidirectional wheel, we use PID control method to speed the motor to ensure that the car will not have a large trajectory deviation while going straight. PID is a classical control method, which means proportion, integral and differential. PID control includes position PID and incremental PID. Generally, incremental PID is applied to the motor speed control. Incremental PID is derived from the traditional PID control formula [3].

However, in the specific motor speed regulation application scenarios, we generally ignore the influence of D and only consider the influence of two parameters (P and I) on the system. PID is the classical closed-loop control. Therefore, we need to read the actual speed of the motor from the encoder, and then assign adjusted PWM value to the motor after being adjusted by the PI algorithm. In this way, the adjustment process of the two parameters is completed, and the motor speed is stabilized at the target speed.

#### 4.2. Setting method of PI parameters

We can use the serial port to print real-time speed of the motor in a certain period of time, and draw a line chart of the speed during this period of time, which is more intuitive and obvious. We can also use Matlab to carry out theoretical preliminary tuning. Based on the actual situation of specific adjustment, we finally make motor speed equal to target speed.
The value of P is the foundation of I and D and determines the speed and strength of the system response. The bigger the P is, the more severely the line oscillates. Otherwise, the smaller the P is, the more slowly the motor speed responds. In this case, the line chart shows that the motor can reach the target speed only after a long period of time.

The value of I can both eliminate static bias and increase the speed of response. I can reduce the errors generated by the whole system. In particular, it can eliminate errors such as wheel deceleration caused by external interference (e.g., The car hits an obstacle.). The smaller the I is, the more deviations we can observe from the target speed. Otherwise, too small value will cause the system to oscillate more.

To summarize, the process of PI adjustment requires much time. As the external environment changes, these two parameters may be quite different from the ones adjusted last time. Furthermore, as the program runs over time, errors will accumulate. Therefore, the speed will deviate more and more from the target speed when the program runs in the later period. We consider that adjustments in various situations are necessary to get a set of parameters, which will make the motor run at a stable speed.

5. Sensor Application

5.1. Ultrasonic Sensors

We apply four HC-SR04 sensors. First, the TRIG pin sends eight square waves of 40kHz. Then, the Echo pin detects if a signal is returned. When the Echo pin outputs a high level, we start timing until it outputs a low level. The time difference is duration that the ultrasonic wave travels through the air.

![Fig.5 Distribution of 4 Ultrasonic Sensors](image)

We fix two ultrasonic sensors on both front and back respectively. According to the differences between two sensors on the same side, we can fine-tune the position of the current car. If the left distance is greater than the right one, we need to move the left front wheel forward. If the right distance is greater than the left one, we need to move the right front wheel forward. Not only can the approach avoid the collision between the car and the obstacle, but also avoid the large deviation when the mechanical arm grabs the object, which may result in the object falling. According to (1), the approximate distance from the obstacle can be calculated.

\[
d = \frac{vt}{2}
\]  

In (1), d represents the distance from the car to the obstacle, v represents the speed of ultrasound, generally equals 340m/s, and t represents the time difference.

We design the distribution of 4 ultrasonic sensors fixed in the car. The distribution of ultrasonic sensors is shown in Fig.5.
5.2. **Gray Sensors**

The gray sensor is equipped with a light-emitting diode and a photosensitive resistor. The working process is that the diode illuminates the ground and that the photosensitive resistor senses the reflected light. According to the strength of reflected light, the optical signal is converted into electrical signal, which is transmitted to the microcontroller for processing.

The purpose of using gray sensors is to determine the position of the car when it is moving, so as to make corresponding adjustments. We use twelve gray sensors totally and each three are installed on each side to ensure that the car can drive in a straight line.

We design the distribution of 12 gray sensors fixed in the car. The distribution of gray sensors is shown in Fig.6.

![Fig. 6 Distribution of 12 Gray Sensors](image)

6. **Test and feasibility analysis**

Since the intelligent logistics car was built, we have tested the performance of the car. We generated and printed a QR code containing information about the delivery task. For example, QR code information is 213 (1 represents red, 2 represents blue and 3 represents green), which means that the car should first carry the blue object, and then the red one, finally the green one. In addition, we have built a field with a black grid on it, which includes three areas: the cargo warehouse area, the storage area and the route area.

During the process of the test, we used various task QR codes, and different shapes and colors of objects. After tests and debugging repeated many times, we found that the car can indeed carry objects from the warehouse area to the storage area in the specified order, and there is no large deviation while going along the straight line. The car can indeed complete the whole process of transportation accurately and smoothly. It can be proved that our design scheme is feasible. Our test is shown in Fig. 7.

![Fig. 7 The Picture of Grasping Objects](image)
7. Summary
Based on the background of current logistics industry and the rapid development of science and technology, this paper provides a design scheme of intelligent logistics cars. With the development of logistics industry, the phenomenon of replacing manual delivery with intelligent logistics cars will become more and more common. The reason is that logistics cars have the advantages of convenience and high efficiency. Moreover, it can take the place of human beings in dangerous or bad environments to carry cargoes in order to result in human injury. According to [4] and [5], it can be applied in logistics industry, automatic control, industrial production, hospitals and other occasions. The development prospect of intelligent logistics cars is very broad. In the near future, intelligent logistics cars will play its own role in many industries, help complete many tasks that human beings are difficult to do, and create more benefits and value for society.

Acknowledgments
Thanks for the support of the 2021 college student innovation and entrepreneurship project.

References
[1] Lin Hao, Wang Xinyu and Xu Yue, “Research and design of intelligent logistics trolley based on high performance microcontroller unit,” Henan Science and Technology, no. 5, pp. 24–28, 2020.
[2] Li Siyi and Wang Shaokun, “Intelligent line inspection robot based on OpenMV and Arduino,” Science and Technology and Innovation, no. 22, pp. 50–53, 2020.
[3] Xiao Wenjian and Li Yongke, “Design of intelligent vehicle based on incremental PID control algorithm,” Information Technology, vol. 36, no. 10, pp. 125–127, 2012.
[4] Zhao Kun, Zang Tiegang and Zheng Bowen, “Design of embedded controlling system for medical track vehicle,” Machine Building & Automation, vol. 46, no.4, pp. 192–194, 2017.
[5] Ma Zhide, Du Shilong, Zhou Yongshuai and Guo Jicheng, “Design and application analysis of multifunctional intelligent car,” Scientific and Technological Innovation, no. 31, pp. 156–157, 2020.