

# Intelligence Rover Design and Demonstration

Team Design Project and Skills Final Report

Team-04: Friday Garage





# 电子科技大学 格拉斯哥学院

## Glasgow College, UESTC

Team Design Project and Skills — 2021-22

Semester II

Team Number		Team 04	
Team Name		Friday Garage	
No	UESTC ID	GUID	Full Name
1	2019190602008	2510874	Zeyu Chang
2	2019190602034	2510900	Ziheng Zhang
3	2019190602018	2510884	Yingchao Jian
4	2019190504027	2510739	Ke Wang
5	2018190601034	2429202	Shuyu Wang
6	2018190606009	2429366	Yian Zhuang
7	2019190602014	2510880	Yang Hu
8	2019190608012	2511096	Yunhao Gu
9	2019190502032	2510666	Jincong Zhang
10	2019190501030	2510625	Chenjingzhi Wen

# Abstract

*Robots can accomplish various tasks in modern society and have a wide range of applications. However, since the practical application of robots in the engineering field requires the integration of multi-domain knowledge, it is quite a challenging task. This report aims to provide a thorough solution for a team design project. According to the given tasks, a rover is built to complete the tasks, including path tracing, obstacle avoidance, shape matching, color recognition, wireless communication, and other functions. Our solution mainly consists of a machine vision module, a microcontroller, drive circuits, and some peripherals. The machine vision module is used for path planning in the color and shape recognition. The ultrasonic sensor is acted as a supplement for increasing the accuracy of the route. The mechanical arm and communication module connected to the Mbed microcontroller can effectively complete the required tasks. To ensure the movement ability of the rover in various environments, the team uses a motor to drive the caterpillar track. With a suitable project planning methodology, the project and system integration complete smoothly. The final rover, after many tests, showed a good performance that can well complete all the tasks in the given project. After repeated research and discussion, some deficiencies and limitations have been put forward, which may become the future development direction of this solution.*

# Acknowledgments

First, the project is the joint effort of our whole team, so we would like to acknowledge the contribution of each member in the team. All members of the team did their best and achieved good results.

We would like to express my deep and sincere gratitude to our instructors, Dr. Abdullah Al-Khalidi, Dr. Wasim Ahmad, and Dr. Oluwakayode Onireti, for allowing us to do the project and providing invaluable guidance throughout this project. Meanwhile, the work and help from the teaching assistants also greatly benefit our project.

We are also grateful to our parents and school. Thank them for sparing no effort to support us in this extremely difficult time of the epidemic.

Finally, our thanks go to all the people who have supported us to complete the research work directly or indirectly.

# Contents

Abstract.....	i
Acknowledgments .....	i
Contents .....	ii
List of Figures .....	iv
List of Tables.....	vi
1 Introduction .....	7
2 Overall system design .....	9
2.1. Task Breakdown.....	9
2.2 System Structure .....	9
2.3 Detailed design .....	10
3 Subsystem Design .....	12
3.1 Auto and Motors Modules .....	12
3.1.1 Requirements of Performance.....	12
3.1.2 Ideas of Design .....	14
3.1.3 Motor Choice .....	31
3.1.4 H bridge Design .....	31
3.1.5 Electronic Compass .....	38
3.1.6 PID Algorithm.....	39
3.1.7 3D Printing .....	40
3.2 Vision Modules .....	41
3.2.1 The OpenMV Cam.....	41
3.2.2 Path Tracing System.....	42
3.2.3 Shape Matching .....	51

## List of Figures

3.2.4 Color Recognition Module.....	56
3.2.5 Communication Protocol .....	61
<b>3.3 Function Modules .....</b>	<b>62</b>
3.3.1 Demo Board NUCLEO-L432KC.....	62
3.3.2 Sonar-based Distance Sensor HC-SR04 .....	63
3.3.3 Wireless Communication.....	65
3.3.4 Tennis Ball Release Device .....	68
<b>4 System Integration.....</b>	<b>72</b>
<b>5 Results &amp; Discussion.....</b>	<b>74</b>
<b>6 Conclusion .....</b>	<b>81</b>
<b>References.....</b>	<b>82</b>
<b>Appendix .....</b>	<b>85</b>
<b>Appendix B.....</b>	<b>86</b>
A.1 Road Patrolling.....	86
A.2 Shape Matching .....	87
A.3 Patio 1 Function Group .....	88
A.4 Patio 2 Function Group .....	90

## **List of Figures**

- 1.1 Gantt chart of the project schedule
- 2.1 Overall system structure
- 2.2 Flow chart of patio 1
- 2.3 Flow chart of patio 2
- 3.1 The schematic diagram of Patio 1
- 3.2 The schematic diagram of Patio 2
- 3.3 The shape and dimensions of the Bridge
- 3.4 The shape and dimensions of the Arch
- 3.5 Plastic granules
- 3.6 Aluminum alloys
- 3.7 The design of a regular wheeled rover
- 3.8 The design of a simple track rover
- 3.9 The design of a complex track rover
- 3.10 The schematic diagram of the Ackermann structure
- 3.11 Simplified model of the differential principle of motion
- 3.12 The chassis chosen
- 3.13 The diagram and dimension of the metal mount
- 3.14 The diagram and dimension of the plastic mount
- 3.15 The picture of the plastic mount
- 3.16 The diagram and dimension of the L-shaped holder
- 3.17 The diagram and dimension of the Long-U-shaped holder
- 3.18 The diagram and dimension of the multi-function holder
- 3.19 The diagram and dimension of the new chassis
- 3.20 Studs of different length
- 3.21 The broadside of the rover
- 3.22 The front of the rover
- 3.23 The top of the rover
- 3.24 Schematic diagram of H-bridge
- 3.25 Schematic diagram of NMOS
- 3.26 H-bridge consists of NMOS
- 3.27 Connection of H-bridge driver circuit
- 3.28 Overall design of driver circuit and its PCB allocation
- 3.29 A typical PWM signal
- 3.30 PWM signals with different duty cycles
- 3.31 DuPont thread
- 3.32 Connection between HMC5883L and OpenMV
- 3.33 Calibration of electronic compass
- 3.34 Diagram of PID algorithm
- 3.35 Model of the Terrace Constructed by Computer Software
- 3.36 The pin configuration of the OpenMV Cam
- 3.37 Image Processing Flowchart

## List of Figures

- 3.38 Example of 3\*3 Gaussian Function Template
- 3.39 Threshold Segmentation System Design
- 3.40 Before Processing and After Processing
- 3.41 Sobel Operator
- 3.42 Wight Identification
- 3.43 Performance in Different Wet Conditions
- 3.44 Performance in Different Light Conditions
- 3.45 PID Algorithm for Image Processing
- 3.46 The LCD Screen Pixel Pattern Supermacro
- 3.47 Round pupil detected by Hough transform
- 3.48 A transformation of a line from Cartesian coordinate to  $\theta - r$  coordinate
- 3.49 A transformation of two points from Cartesian coordinate to  $\theta-r$  coordinate
- 3.50 points in ROI produce a circle with radius r
- 3.51 QR code
- 3.52 Red blocks
- 3.53 Lab color space [H2]
- 3.54 Color Threshold of red cardboard
- 3.55 the pixel coordinate system of OpenMV
- 3.56 The pins of NUCLEO-L432KC
- 3.57 HC-SR04
- 3.58 The working principle of HC-SR04
- 3.59 The signal timing diagram of HC-SR04
- 3.60 HC-12 wireless transceiver
- 3.61 HC-12 schematic diagram
- 3.62 Half-duplex physical communication line and HC-12 wireless communication line
- 3.63 The general form of UART communication
- 3.64 DS1302
- 3.65 structure of arm part
- 3.66 Structure of claw part
- 3.67 External appearance of micro servo motor
- 3.68 Example of PWM control signal
- 4.1 System integration with all involved modules
- 5.1 The finalized design of the rover's mechanics
- 5.2 Calibration of electronic compass
- 5.3 Performance in Different Wet Conditions
- 5.4 Performance in Different Light Conditions
- 5.5 The installation of the mechanical jaw

## **List of Tables**

- Table 1.1 Division chart  
Table 3.1 Comparison between plastic and aluminum  
Table 3.2 Comparison between wheel and track  
Table 3.3 parameters of brushed motor MG540  
Table 3.4 Control logic of H-bridge  
Table 3.5 Parameters of NMOS NTMFS4833N  
Table 3.6 Performance of the Algorithm in Different Wetness Conditions  
Table 3.7 Performance of the Algorithm in Different Brightness Conditions  
Table 3.8 The basic information of HC-SR04  
Table 3.9 Relation of I/O at special value  
Table 5.1 parameters of brushed motor MG540  
Table 5.2 Control logic of H-bridge  
Table 5.3 Parameters of NMOS NTMFS4833N  
Table 5.4 Performance of the Algorithm in Different Wetness Conditions  
Table 5.5 Performance of the Algorithm in Different Brightness Conditions  
Table 5.6 Color Threshold of red cardboard  
Table 7.1 Consumption record of the team

## 1

# Introduction

In this project, the main purpose is to design robots to complete two patios with different tasks. Both patios are located next to the East Lake of the university and have different paths. In patio 1, beginning from the arrow at the starting point, the robot would follow a colored tile to a bridge that crossed two tiles. After locating the bridge, the robot should be able to cross it while remaining in the bridge's center. When crossing the bridge, the robot could follow the colored tiles at the other end. After traveling through an arch at the end of the colored tiles, the robot would eventually come to a halt. As for patio 2, firstly, the robot should do shape matching for three different shapes, circle, triangle, and square. For each different shape, they indicate different paths on a diamond pattern and different shapes on the end of the diamond to find. After passing the correct shape, the robot should move to the release point and release a tennis ball into a designated basket. Finally, the robot should stop at the planter area and transmit a message to a laptop.

To fulfill the objectives mentioned above, we divided the project into various sub-tasks and assigned individual duties. Several direct responsibility personnel is in charge of operating different modules and communicating between multiple elements. The NUCLEO-L432KC module and the OpenMV module are the two primary modules. The UART communication protocol is used to communicate between these two devices. Several distinct pieces accomplish the robot's job efficiently under two primary sections.

Here is our Gantt chart of the project schedule, and a division chart.

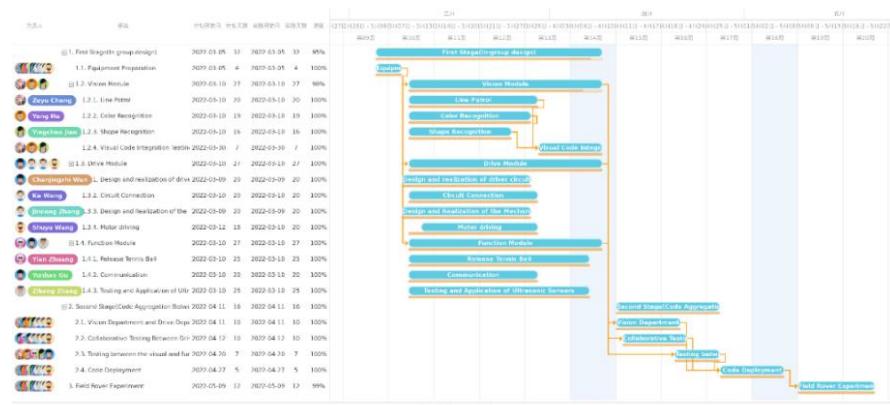


Figure 1.1 Gantt chart of the project schedule

## 1 Introduction

Classification	Content	Person in Charge
Project Preparation	Design of the Network Diagram, Gantt Chart, Division of Task Chart	Chang Zeyu
Function Group	Distance Detection	Zhang Ziheng
	Ball Releasing	Zhuang Yian
	Communication	Gu Yunhao
Vision Group	Road Patrolling	Chang Zeyu
	Shape Matching	Jian Yingchao
	Color Recognition	Hu Yang
Auto and Motors Group	Mechanics Design	Zhang Jincong
	Driver Circuit Design And PCB Design	Wen Chenjingzhi
	Electronic Compass And PID Algorithm	Wang Shuyu
	Pulse Width Modulation	Wang Ke
System Intergration	3D Printing (Outer Shell, Cloud Terrace.etc )	Chang Zeyu
	UART Implementation of OpenMV and Mbed Data Transfer	Chang Zeyu, Wang Shuyu, Jian Yingchao, Hu yang
	OpenMV and Driver Board Information Transfer Design	Zhang Jincong, Wen Chenjingzhi, Wang Shuyu
Report Writing	Sonar, Communication, Pitching System Trigger Design	Zhang Ziheng, Zhuang Yian, Gu Yunhao
	Individual Part	All Team Members
	Group Part	Zhang Jincong, Jian Yingchao, Zhang Ziheng
Video	Integration and Typography	Chang Zeyu, Wen Chenjingzhi, Gu Yunhao, Zhuang Yian
	Demo Video Making	Hu Yang
	Individual Part	All Team Members
	Integration and Typography (Initial)	Chang Zeyu
Group Notebook	Integration and Typography (Final)	Wang Shuyu

**Table 1.1** Division chart

# 2

## Overall system design

### 2.1. Task Breakdown

Depending on the detailed description of the task in the introduction, our team subdivided the task into the following functions to be implemented after careful and careful consideration.

#### Patio 1:

- **Path Tracing:** follows the white line on the ground.
- **Obstacle avoidance:** detect obstacles and react
- **Turning.**
- **Straight-line travel.**

#### Patio 2:

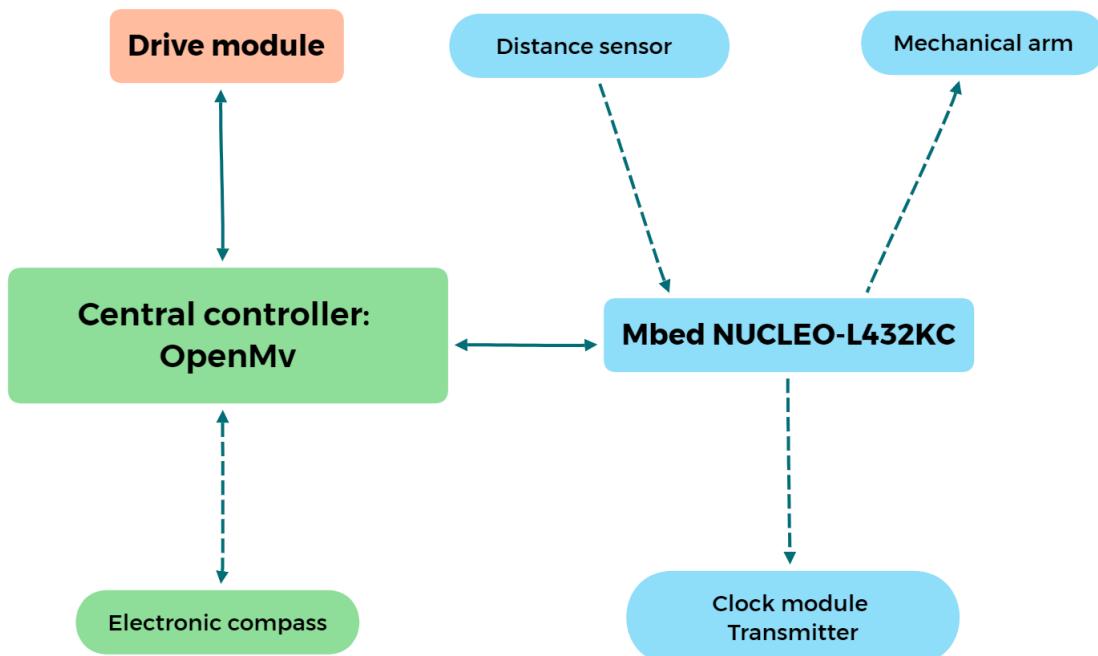
- **Shape Matching:** identify and react to specific shapes
- **Color Recognition:** identify and respond to a specific color
- **Wireless Communication:** wireless communication with computer
- **Tennis release:** put the tennis ball in the basket.

### 2.2 System Structure

To complete the tasks, our team divide the rover into three main parts – The drive module, OpenMV module, and function module. The drive module mainly contains the body of the rover and the drive circuit, which is used to control the movement of the rover. The OpenMV is the core of our design. Most environmental perception parts, including path tracing, shape recognition and color recognition, are realized based on OpenMV. In addition, an electronic compass is also attached to it to provide accurate direction during the patio. The function module is based on the Mbed NUCLEO-L432KC, and mainly

contains an ultrasonic distance sensor HC-SR04, a technical arm to release the tennis ball and a wireless communication module. The wireless communication module consists of a transmitter and a clock, communicating with NUCLEO-L432KC by UART protocol.

After research and discussion, we decide to use the OpenMV as the central controller because it has various powerful pins. All the collected data will be transmitted to it so that it can give corresponding instructions to the function and drive module. The interaction between the OpenMV and the driving circuit utilizes the simplex pulse signal PWM. Additionally, since L432KC needs to exchange data with the OpenMV, we choose the UART protocol for real-time and full-duplex communication.

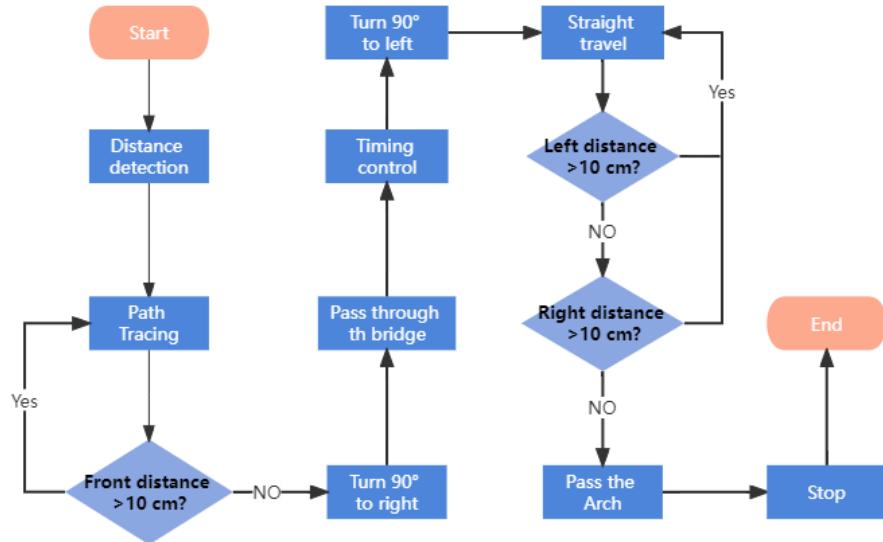


**Figure 2.1.** Overall system structure

### 2.3 Detailed design

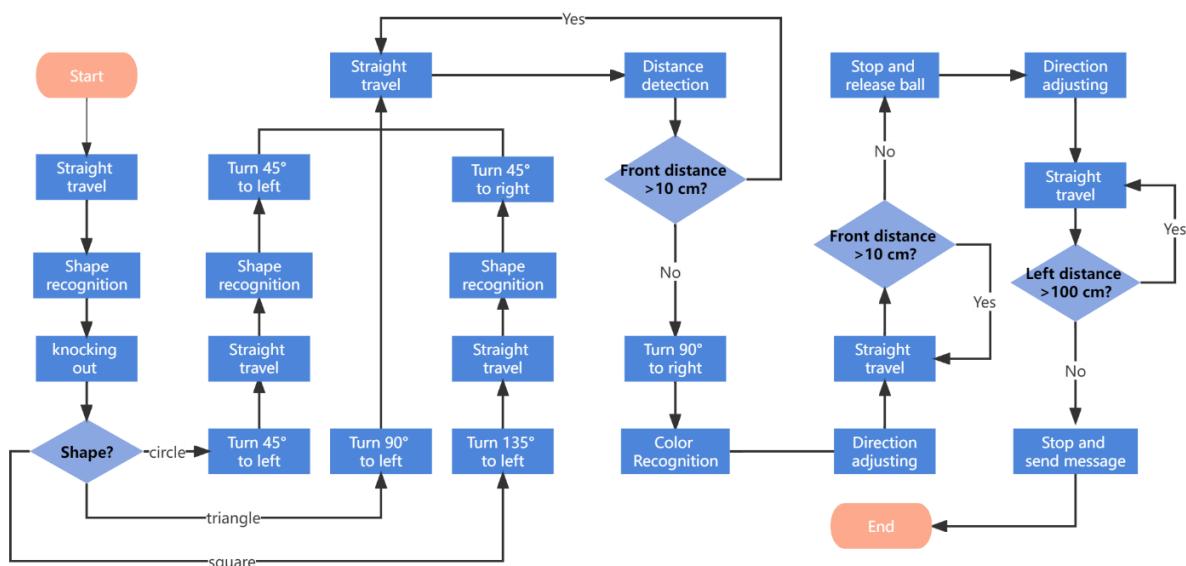
In part 1, the process to achieve all the tasks are shown in the following figure. Generally, our team mainly use path recognition and distance detection as the main solution for the tasks. In the beginning, the rover follows the line on the ground using path recognition. Next, with the distance sensor and the electronic compass, it turns right and passes through the bridge. Finally, it moves straight forward, passes the arch, and stops after the distance sensor detects the arch.

## 2 Overall system design



**Figure 2.2.** Flow chart of patio 1

In patio 2, the rover recognizes the specific shape and executes the corresponding route at first. Then, with the help of color recognition and an electronic compass, it will make the center point of the marker coincide with the image to determine the direction. After arriving at the specific location, the machine arm releases the ball, and the rover turns to the proper direction by electronic compass. At last, when it enters the designated area, the return data of the distance sensor on the left makes the wireless communication module send the required information.



**Figure 2.3** Flow chart of patio 2

# 3

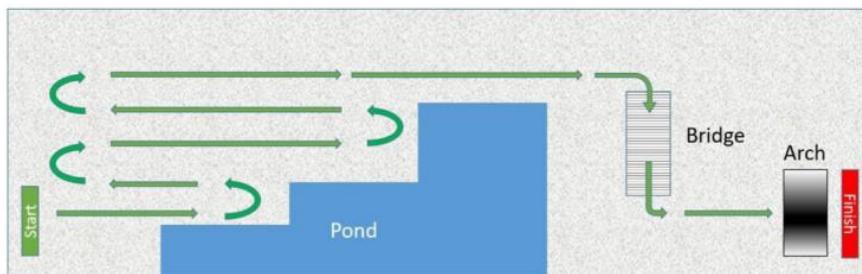
## Subsystem Design

### 3.1 Auto and Motors Modules

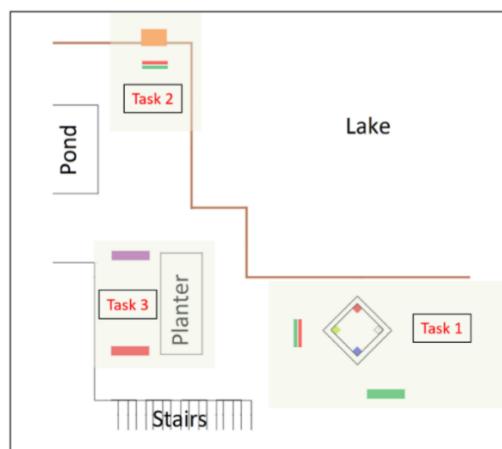
#### 3.1.1 Requirements of Performance

##### Analysis of The Patios

Although the rover has to solve different problems in Patio 1 and Patio 2 respectively, the two Patios share similar road conditions. Both Patio has uneven paved surfaces covered with cobblestones.



**Figure 3.1** The schematic diagram of Patio 1



**Figure 3.2.** The schematic diagram of Patio 2

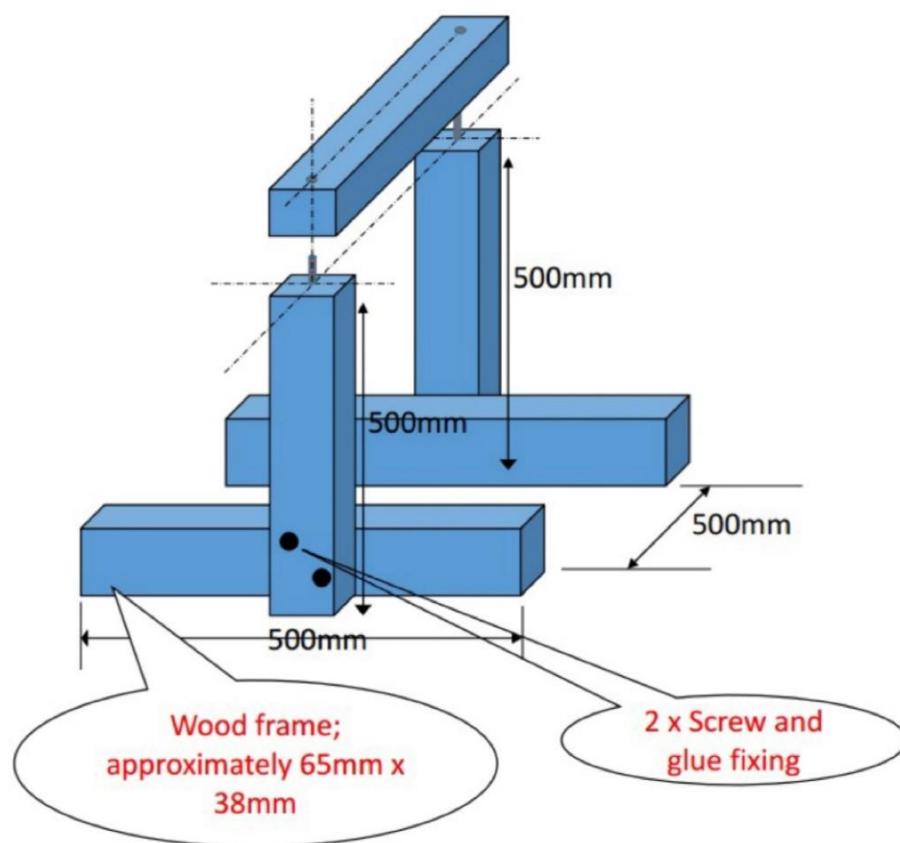
### 3 Subsystem Design

In Patio 1, the rover needs to cross a bridge and an arch [1]. The bridge in Patio 1 is approximately 150mm high and has a ramp angle of approximately 14 degrees.



**Figure 3.3** The shape and dimensions of the Bridge

The width and height of the arch are approximately 500mm.



**Figure 3.4** The shape and dimensions of the Arch

In Patio 2, the rover needs to move a tennis ball into a roughly 250mm high bucket.

#### Analysis of The Rover

A range of devices, including batteries, need to be carried on the rover. The upper layer of the rover requires a mechanical claw to grip the tennis ball, and the center of gravity of the rover is mainly dependent on the position and height

of the claw. The bucket used in Patio 2 has a high altitude and requires a specific height for the rover.

#### The Requirements of The Rover

The rover structure is limited according to Patio's requirements and the specificity of the site. The rover must be able to negotiate rough surfaces steadily so that bumps do not cause damage to the electronics. Besides, the width of the rover must be less than 500mm to pass the bridge in Patio 1. And the height of the rover must not be too high or too low to go through the arch and ensure that the claw reaches the edge of the bucket. Due to the presence of the mechanical claw, the center of mass of the rover may be high and not in the center of the body. Careful consideration needs to be given to the layout of the electronics to avoid sending it tumbling during the uphill and downhill phases of the bridge.

#### 3.1.2 Ideas of Design—Zhang Jincong

##### The Material of The Rover

###### Plastic

Plastics are polymeric compounds made from monomers and polymerized by addition or condensation reactions. Plastics have an important place in our daily lives and are used in many fields in an extensive range of applications. Plastics have the advantages of high corrosion resistance, low cost, lightness, and insulation. The disadvantage is that the production process is not environmentally friendly and is difficult to recycle. More damaging is the fact that commercially available plastics are generally not as strong and are prone to twisting and distortion when subjected to external impacts.



**Figure 3.5** Plastic granules

### Aluminum Alloy

Aluminum alloys are alloys based on aluminum with a certain amount of other alloying elements and are a light metal material. Aluminum alloys are more rigid than steel and have a very low density. They also have good electrical and thermal conductivity, corrosion resistance, and workability. For this reason, aluminum alloys have a wide range of applications in transport and even aerospace.



**Figure 3.6** Aluminum alloys

### Choice

Both patios put the strength of the rover's structure to the test. Vibrations from bumpy roads can damage the body structure. The advantages of plastic are its cheapness, lightness, and versatility, while the disadvantages are its fragility and poor load-bearing capacity. In contrast, aluminum alloys have the advantage of much higher strength and reliability. The burdens are that it is much heavier and more expensive than plastic. A table is drawn below to facilitate the choice between the two materials.

	Plastic	Aluminium Alloy
Benefits	Cheap Light Versatility	Strong Reliable
Drawbacks	Fragile Poor Load-bearing Capacity	Heavy Expensive

**Table 3.1.** Comparison between plastic and aluminum

Considering that the strength of the rover's body was important for the successful completion of the test, we chose aluminum as the raw material for the rover body. Although plastic is light and cheap, the strength of aluminum is

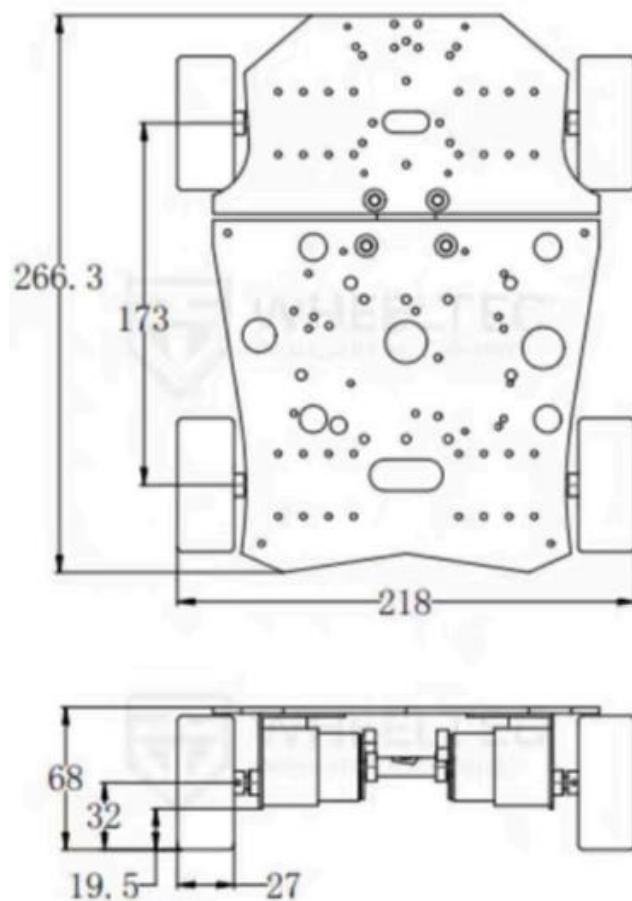
much more practical for us. This choice also means that the rover will be quite heavy, so powerful motors will be needed to ensure that the rover can move properly.

## The Principle of Driving—Zhang Jincong

### Wheel

The wheel is a significant invention in human history. Almost all vehicles have been constructed on the wheel. This is because the wheeled structure supports the body and, at the same time, ensures that the vehicle can be driven. The wheel is often used in complex and demanding conditions, where it is subjected to various deformations, loads, forces, and high and low temperatures while driving.

For the rover, there are also many solutions based on wheeled structures. The figure below shows a regularly wheeled rover.



**Figure 3.7** The design of a regularly wheeled rover

In general, a rover of wheeled construction has four wheels. This means that there are different ways of driving wheeled structures.

### 2WD

2WD stands for 2-wheel-drive, which means that only two of the four wheels are driven by the electric motor, while the other two wheels have no power. This

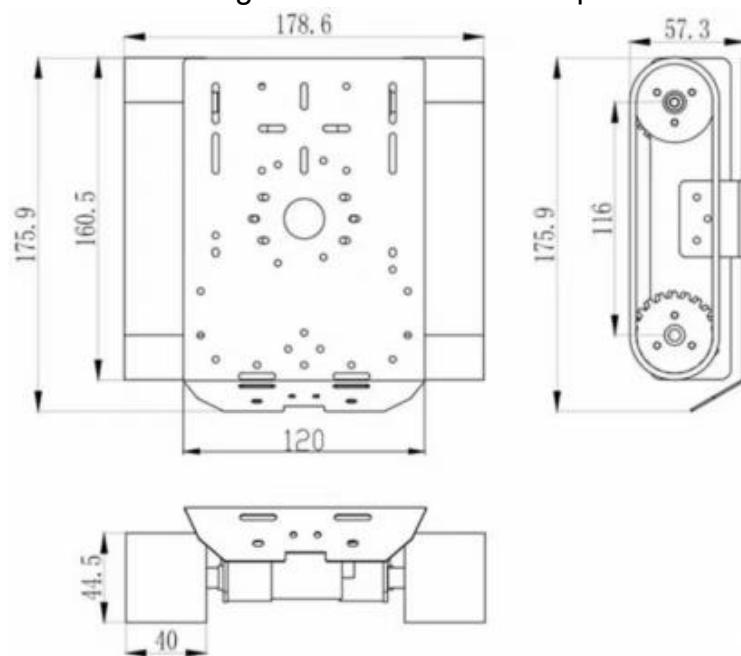
construction simplifies the complexity of the mechanics to a certain extent. But such a structure also means that the power is not evenly distributed over the four wheels. The rover can get stuck when driving over rough terrain, and the drive wheels can slip on hills and fail to move forward.

#### 4WD

4WD stands for 4-wheel-drive, which means that all four wheels of the rover have a source of power. 4WD is achieved in two ways, either by distributing the motor's power output evenly between the front and rear via a driveshaft or by equipping each of the four wheels with an electric motor. The former approach can be mechanically complex due to the additional physical structure of the drive shaft. This solution makes the mechanical design of the rover more unstable and increases the chances of failure. The second approach would add significant overhead, as the motors are not cheap. Also, this solution places a more substantial burden on the microcontroller, as all four motors need to be controlled by the microcontroller. The power consumption of the two different motors is also not to be underestimated.

#### Track

The track is a flexible chain ring driven by the main wheel and surrounding the main wheel, the load wheel, the inducer wheel, and the carrier wheel. On the side, in contact with the ground, there are reinforced anti-slip bars to improve the robustness of the track plates and the adhesion of the tracks to the ground. Tracks are often used on vehicles such as tanks that cope with extreme road conditions. This is because the track construction allows the vehicle to easily negotiate obstacles that would be difficult for a wheeled structure. It also performs well off-road. The figure below shows a simple track structure.



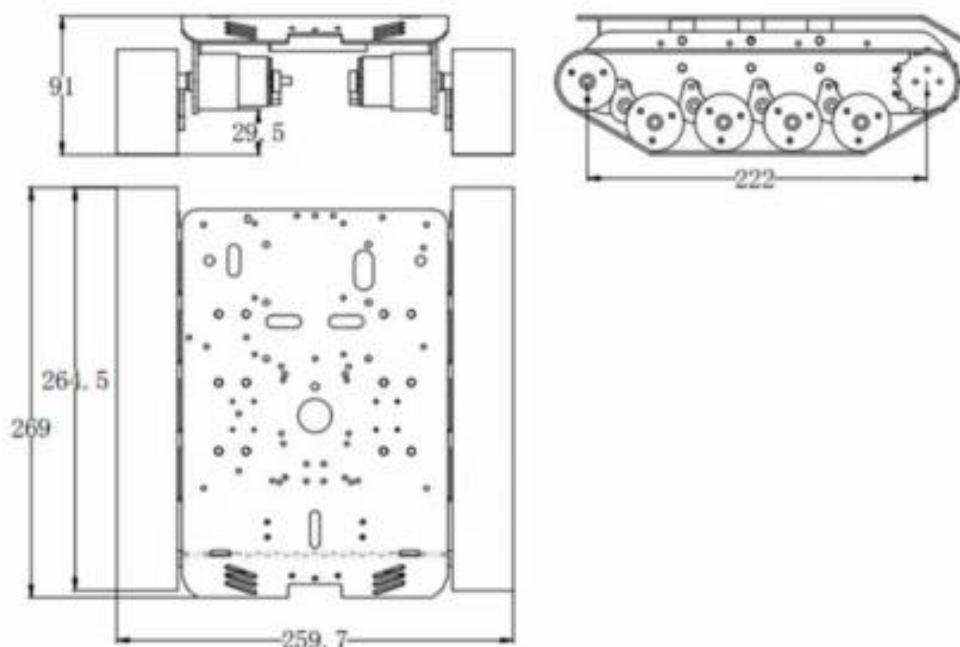
**Figure 3.8** The design of a simple track rover

#### The Simple Solution

Figure 3.7 shows a diagram of a simple track structure rover. In this structure, the rover has a drive wheel directly driven by an electric motor and a driven wheel on each side. This structure is straightforward and requires a small budget. However, the overly simple construction also prevents the tracks from reaching their full potential. There may not be a significant advantage over wheeled structures in practical applications.

#### The Complex Solution

In addition to the simple track construction described above, dealers offer a more complex design, which is shown below.



**Figure 3.9** The design of a complex track rover

In this design, the rover has one driving wheel, five driven wheels, and four damping springs on each side. The height of the rover in this design is 91 mm, and the width is 259.7 mm. This rover size can successfully pass through the Arch and Bridge in Patio 1. The presence of four pairs of load-bearing wheels makes this structure very solid and more stable when driving over rough roads. Overcoming obstacles is much easier. Eight damping springs absorb bumps in the road and provide a more stable working environment for the electronics on the rover.

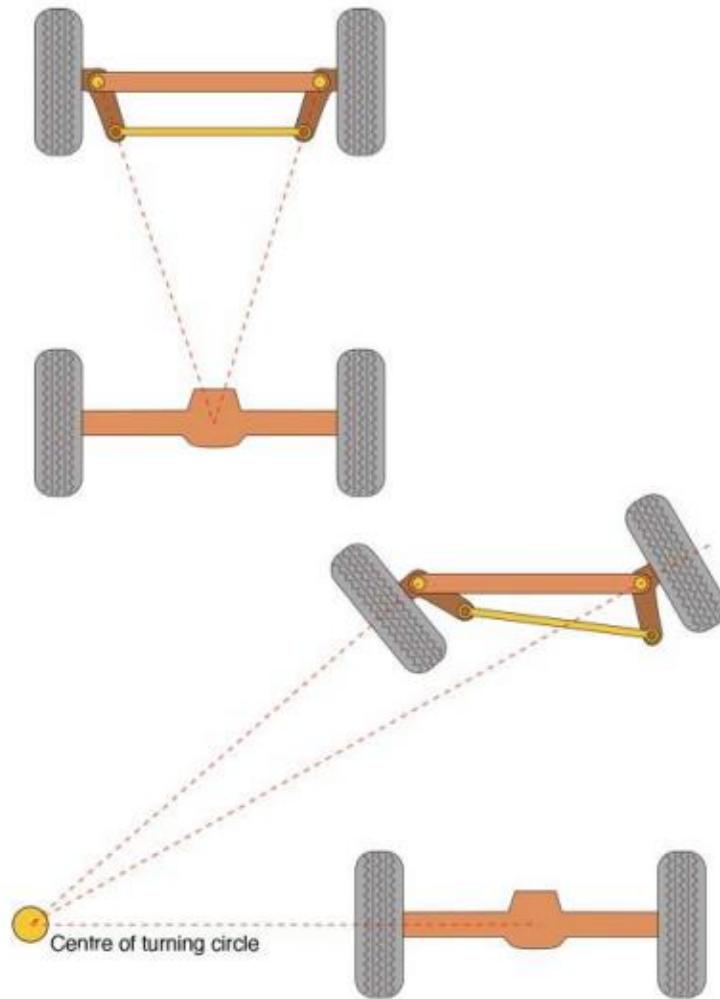
#### Steering Principle

In addition, steering is also one of the rover's essential functions to forward and reverse. The steering of the rover can be achieved in two ways. The first can only be applied to wheeled structures; that is, the rover is steered by mechanically deflecting the angle of the two steering wheels. The second,

which applies to both wheeled and tracked structures, is achieved by controlling the speed of the wheels on both sides of the rover.

#### Ackermann Structure

The Ackermann structure is a type of the first type of steering. The steering is achieved by moving the linkage between the two steering wheels by employing a steering engine, thus changing the angle of the steering wheels.



**Figure 3.10** The schematic diagram of the Ackermann structure

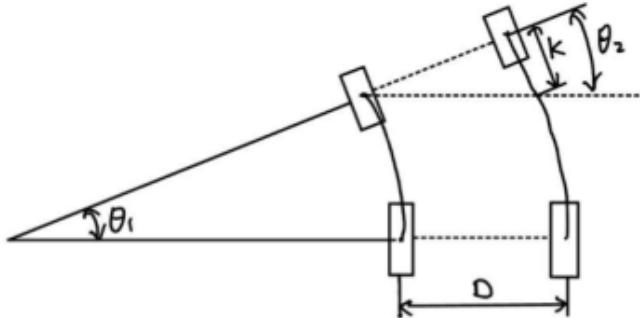
This structure is suitable for steering needs but adds more mechanics and a steering engine. This also means more possibilities for faults, and the microcontroller needs to be allocated a few pins for controlling the steering engine. This may not be the most logical choice for practical applications.

#### Differential

The differential speed principle applies equally to wheeled and tracked structures. This is a structure in which steering is achieved by controlling the

speed of the wheels or tracks on each side of the rover separately. Such a structure does not require any additional construction, only changes to the code that controls the motor speed.

Next, we will discuss the differential principle in detail.



**Figure 3.11** Simplified model of the differential principle of motion

The forward direction in the diagram is the positive X-axis direction, the left translation direction is the positive Y-axis direction, and the counterclockwise direction is the positive Z-axis direction. The spacing between the two-wheeled is D. The velocities of the rover's X and Z axes are:  $V_X$  and  $V_Z$ , respectively, and the velocities of the rover's left and right wheels are  $V_L$  and  $V_R$ , respectively. Assuming that the rover travels a distance in a left forward direction, let the distance traveled by the rover's right wheel over the left wheel be approximated as K, using the point on the rover's wheel as a reference point for extending the reference line, finally get:  $\theta_1 = \theta_2$ . Since this  $\Delta t$  is small, the amount of change in angle  $\theta_1$  is also small, and hence the approximation equation:

$$\theta_2 \approx \sin \theta_2 = \frac{K}{D}$$

From the mathematical analysis the following equation can be obtained:

$$K = (V_R - V_L) * \Delta t, \omega = \frac{\theta_1}{\Delta t}$$

From the above formulas and equations, the results of the kinematic solution can be solved: velocity  $V_X = (V_L + V_R)/2$  in the X-axis direction of the carriage velocity  $V_Z = (-V_L + V_R)/D$  in the Z-axis direction of the carriage.

The kinematic inverse solution is obtained by direct backpropagation from the positive solution: velocity of the left wheel of the rover

$$V_L = V_X - (V_Z * D)/2$$

and velocity of the right wheel of the rover

$$V_R = V_X + (V_Z * D)/2.$$

#### Choice

The wheeled construction is characterized by its cheapness, ease of maintenance, and variety. There is a choice of wheels of different sizes, thicknesses, and shapes on the commercial market. The different types of

wheels can vary enormously in their performance in the same situation. The wheeled construction also allows the rover to move at very high speeds. In a mechanical problem, repairs can be carried out very quickly. The downside is that it can be very bumpy on uneven surfaces, which can be dangerous for the electronics on the rover. There is also a risk that the rover will not pass over the bumps properly and get stuck somewhere. In contrast, tracked construction solves this problem almost perfectly. And tracked construction is inherently more reliable and durable, offering more excellent performance on rough terrain. However, the tracked construction is destined to be slow and very expensive. In order to compare the advantages and disadvantages of the two structures, we have created the following table.

	<b>Wheel</b>	<b>Track</b>
<b>Advantages</b>	Fast Cheap Various Form Easy in Mechanics	Steady Durable Off-road
<b>Disadvantages</b>	Bumpy Poor Off-road-performance	Slow Expensive

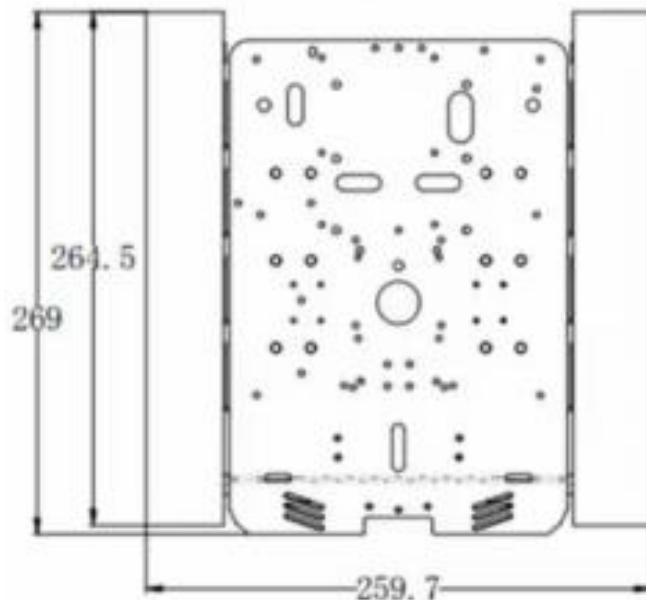
**Table 3.2** Comparison between wheel and track

A rover with a tracked construction is chosen to provide the electronics with as stable an operating environment possible. And of the two tracked structures, we opted for the more complex one. This is because the four pairs of extra shock springs and the four pairs of extra load-bearing wheels will give the rover a considerable boost in stability and ability to overcome obstacles. Although this represents a high cost, we believe it is worthwhile and necessary. For tracked structures, steering can only be achieved utilizing the differential principle.

## **Chassis**

### **The Chassis Chosen**

The design of the chassis we have chosen is shown in the diagram below.



**Figure 3.12** The chassis chosen

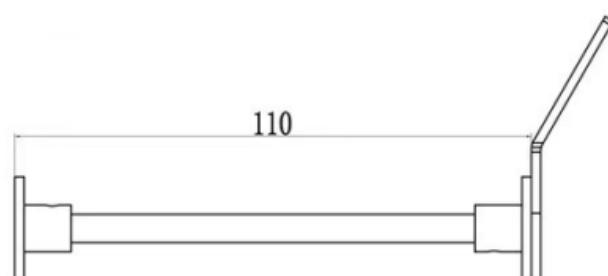
The cavities in the diagram represent cavities in corresponding positions on the physical object.

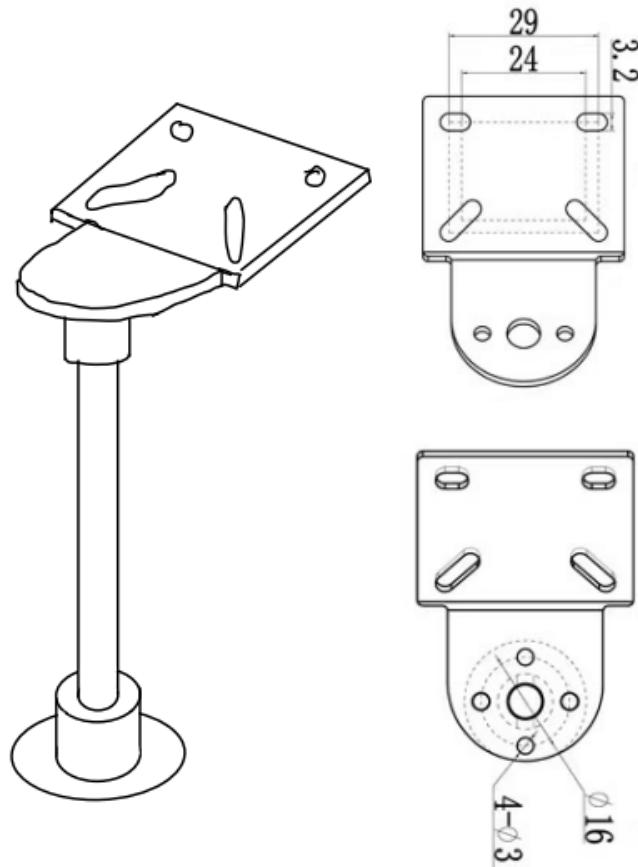
### The Reasons

In the previous discussion, we decided to go for an aluminum rover, so this chassis is also aluminum. Thanks to the machinability of the aluminum alloy, there are many holes in the chassis for screwing in other mechanical structures. The aluminum chassis has a certain amount of weight to it. If there are many different mechanical or electronic structures in the chassis, the whole rover's center of mass will be on the high side. A heavier chassis will reduce the overall center of mass of the rover to a certain extent, making the ride more stable and less likely to tumble. The most important advantage of the aluminum chassis is its strength, which effectively protects the rover from damage during driving. These are the reasons why we chose this chassis.

## Mount of The Camera

### The Metal Mount





**Figure 3.13** The diagram and dimension of the metal mount

### The Design of This Mount

The picture shows a camera mount made of metal. It is possible to keep the height of the camera above the chassis at 110mm. The camera angle to the ground can be adjusted by bending the metal plate. The camera needs to be attached to the mount by screws and studs. The length of the studs needs to be greater than the length of the camera pins so that the camera pins can be connected.

### The Analysis

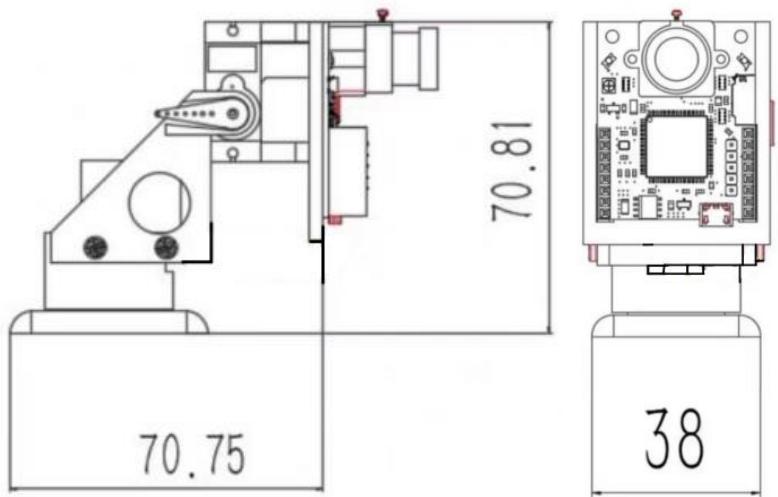
In practical tests, it is found that this construction of the camera mount has many flaws. Firstly, the metal plate that needs to be bent in order to adjust the camera angle is very sturdy and requires a lot of force to do so. It is almost impossible to adjust the angle of the camera while it is in use, as this would damage the camera. In order to adjust the angle of the camera, the camera should be removed, and the metal plate is supposed to change before putting the camera back on. This is a time-consuming and laborious process. Secondly, the height of the camera mount off the ground is not adjustable. This height is fixed at approximately 200 mm, and in tests, the camera cannot recognize the ground. This height is apparently too high. The last point is that the mount's

design makes it impossible to easily connect the camera pins to the microcontroller, making it cumbersome to install, remove and repair. There is also a high risk of breaking the camera pins.

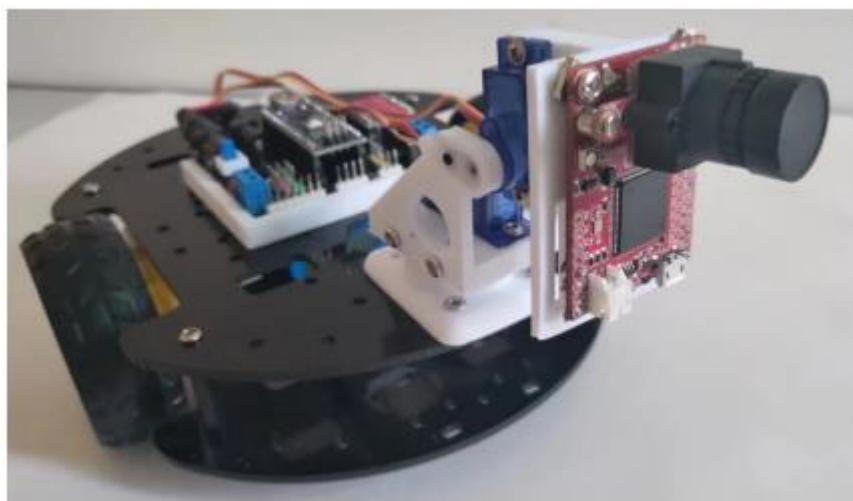
## The Plastic Holder

### Design of This Holder

This plastic camera mount consists of several resin parts manufactured by 3D printing and two steering engines that are responsible for turning the mount. The specific design is shown in the diagram below.



**Figure 3.14** The diagram and dimension of the plastic mount



**Figure 3.15** The picture of the plastic mount

It is easy to see that the size of this mount is not much bigger than the camera. The top of the mount is about 71mm high from the base. The length of the mount is about 71mm, and the width is about 38mm. In practice, it was

found that the camera module has a lens height of approximately 150mm above the ground. And the camera can be rotated in both directions by controlling the steering engines. The vertical field of view covers from directly above the rover to directly below it, while the horizontal field of view covers 360 degrees.

#### Analysis

During tests, it could be concluded that the height of the camera on this mount is quite suitable for the ground and that the camera can recognize the floor very well when it is facing at an angle. The angle of the mount can be adjusted manually in addition to the steering engine's control. If the microcontroller does not have enough pins to connect the two steering engines, you can choose to adjust the camera angle manually, which will also reduce some code work.

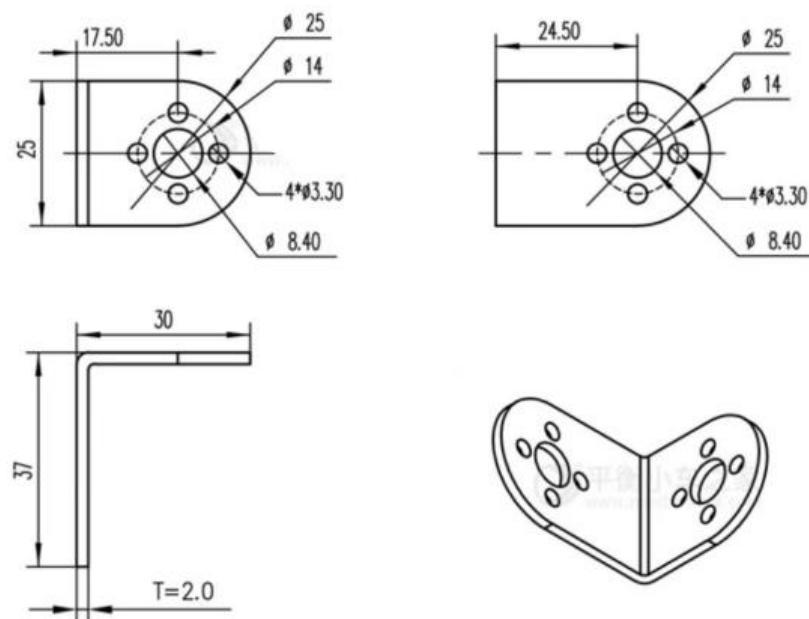
#### The Choice

The second plastic mount is a much more reasonable option. An overhead can be saved by 3D printing, and there is a high degree of flexibility in making the required structural adjustments by modifying the design drawings. The size of this camera mount is also just right for visual processing. The flexible adjustment of the angle gives a great deal of convenience during practical use. The camera pins are much easier to connect. There is no doubt that this mount is ideally suited to our needs.

### Holders of Other Components

#### The L-shaped Holder

The diagram below shows the design and dimensions of the L-shaped holder.



**Figure 3.16** The diagram and dimension of the L-shaped holder

### The Long-U-shaped Holder

The diagram below shows the design and dimensions of the Long-U-shaped holder.

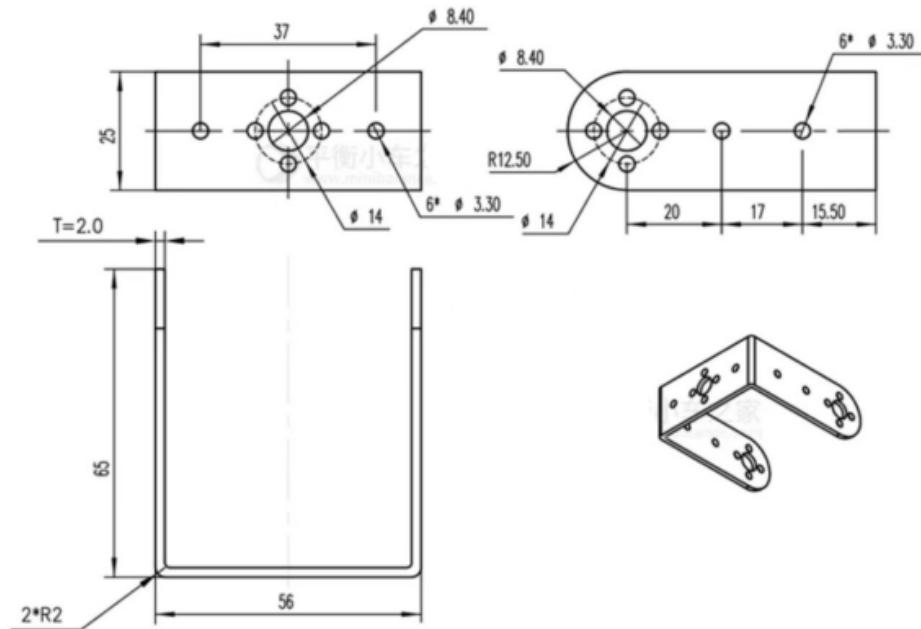


Figure 3.17 The diagram and dimension of the Long-U-shaped holder

### The Multi-function Holder

The diagram below shows the design and dimensions of the multi-function holder.

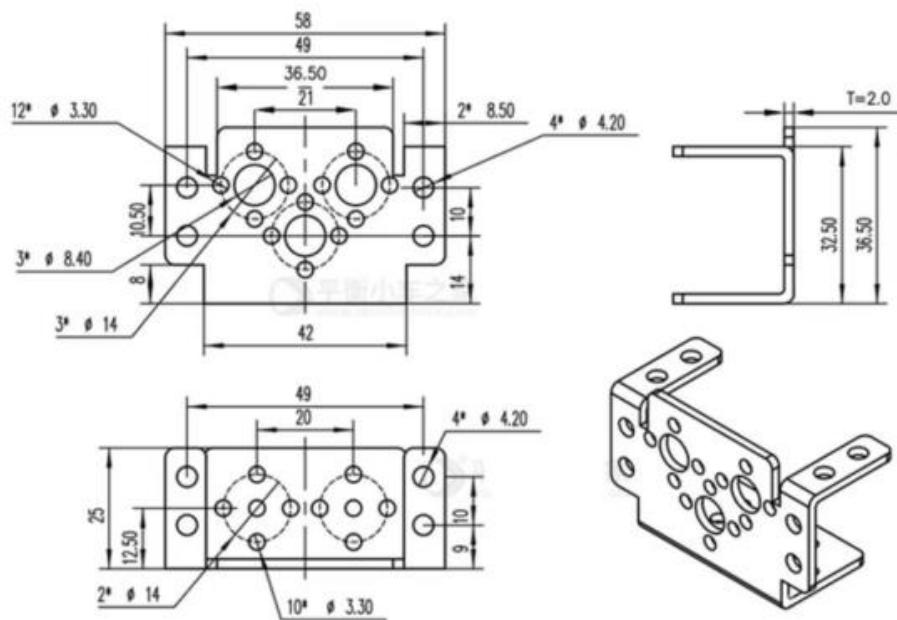


Figure 3.18 The diagram and dimension of the multi-function holder

#### Considerations About Holders

All three of these brackets can be made by 3D printing, which saves an expense and relieves some of the burdens on the rover. Depending on the application conditions, all three holders can be screwed onto the rover chassis and mounted in different positions. As the implementation of the rover function also requires other electronics such as three ultrasonic sonars and a wireless communication module, the location of these components needs to be carefully considered. For example, the three ultrasonic modules have to be oriented in each of the three directions of the rover, and there must be no obstructions in front of them. The antennas of the wireless communication modules must also not have any other obstacles in front of them. This requires that the corresponding brackets are mounted in the right place. For example, the ultrasonic mount is mounted on the outside of the rover.

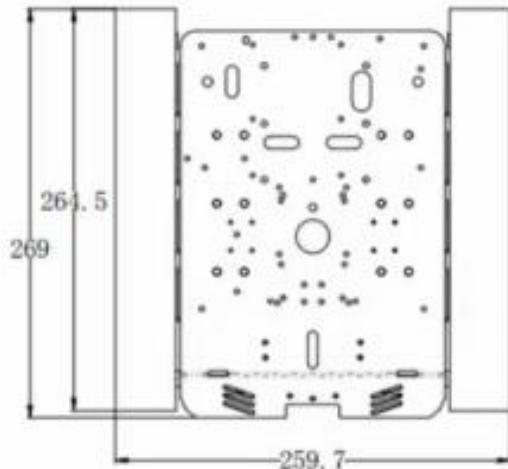
## Heightening

#### The Reason for Heightening

In Patio 2, the rover needs to move a tennis ball into a basket with a height of approximately 250mm. We decided to grab the tennis ball with a mechanical claw and release the claw when the tennis ball was moved above the basket. But our rover chassis is only about 91mm above the ground, which means that if the mechanical claws were mounted on the chassis, they would not reach the edge of the basket because of the lack of height. In addition, the electronics that need to be installed in the rover's chassis include cameras, ultrasonic modules, wireless communication modules, and microcontrollers. These components and the cables required to connect them take up a large amount of space. This does not leave enough space for the mechanical jaws. Therefore, if a chassis could be added to the rover, much more space could be made available for other electronics, and the height of the jaws could reach the edge of the basket.

#### Methodology

We chose to add a chassis made of aluminum identical to the rover's chassis. The dimensions and design drawings for the planned additional chassis are shown below.



**Figure 3.19** The diagram and dimensions of the new chassis

We decided to connect the two chassis employing four studs of 60 mm length and four studs of 40 mm length. This means that the new chassis is mounted 100mm above the original chassis.



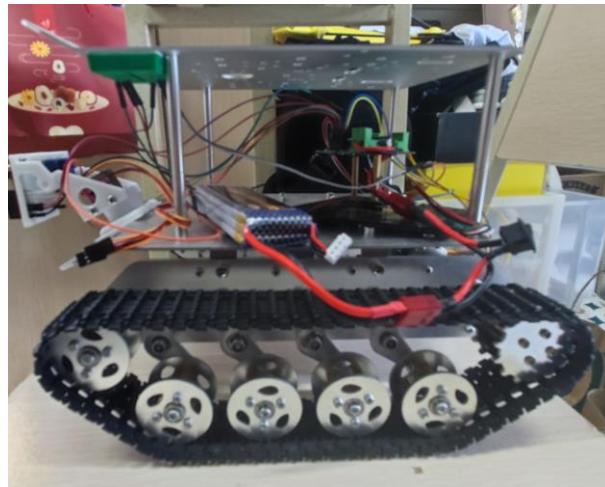
**Figure 3.20** Studs of different length

The heightened rover has a lot of space to place the electronics. The mechanical jaws are also high enough to perform the tasks of the Patio2 when mounted on the newly added chassis.

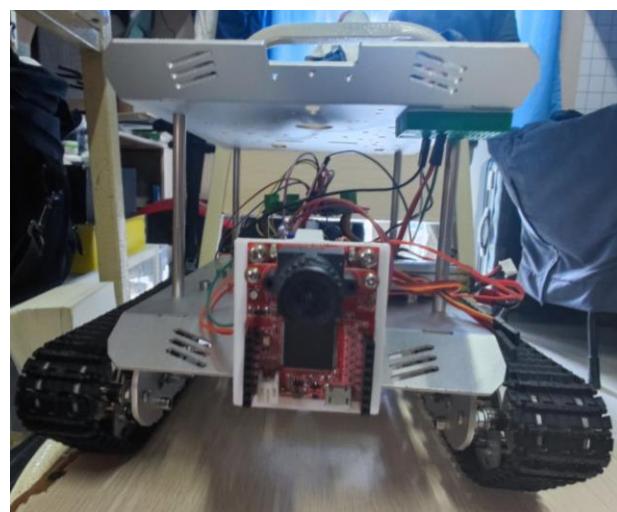
## Auxiliary Equipment

In addition to the basic parts such as screws, studs, and nuts, which will be used extensively, other tools such as double-sided tape, electrical tape, AB glue, and zip ties are also used extensively to simplify the installation and dismantling process.

## Results



**Figure 3.21** The broadside of the rover



**Figure 3.22** The front of the rover



**Figure 3.23** The top of the rover

### 3.1.3 Motor Choice—Wen Chenjingzhi

A motor is a device that supplies torque based on the laws of electromagnetic induction. It uses an energized coil (stator) to generate a rotating magnetic field that acts on a closed metal frame (rotor) to create magnetoelectric rotation, converting electrical energy into mechanical energy and generating torque to drive the rover. The two common types of motor are known as brushed motors and brushless motors, which have their advantages and disadvantages. Brushed motors have the advantage of being cheap and easy to control, but their speed control is less precise. On the other hand, Brushless motors can be controlled precisely and have high energy efficiency, but have the disadvantage of being complex and expensive to control [2]. For this project, we decided to use a brushed motor with a feedback design to improve accuracy, considering the overall budget. In terms of driving a heavy load, we have chosen the MG540, a high-powered motor with the following essential parameters.

Working voltage	Motor type	Block running current	Rated current	Rated torque
12V	Brushed	>8A	1.44mA	2.6kgf.cm

Table 3.3 parameters of brushed motor MG540 [3]

#### Control strategy

The DC brushed motor has two input pins. When a positive voltage is applied, the motor will turn clockwise; when a negative voltage is applied, the motor will turn in the opposite direction [2]. Within the voltage tolerance of the motor, the higher this voltage is, the faster the motor will turn; as mentioned above, the rover adopts a two-wheel differential drive strategy, which requires that the left and right wheels of the rover have variable direction and variable speed capabilities [4]. From the above principles, it can be concluded that the motor drive circuit needs to have the ability to control the positive and negative value and the magnitude of the voltage input, where the positive and negative voltage input can be governed by the H-bridge circuit, and the voltage input magnitude can be controlled by the duty cycle of the PWM wave.

### 3.1.4 H bridge Design—Wen Chenjingzhi

Since the motor operates at 12 V, the standard microcontroller has a high-level voltage output of 5 V or 3.3 V. This voltage cannot fully drive the motor. We choose to use a controllable external power supply of 12V and a circuit to control this external power supply to drive the motor. The H-bridge circuit is a typical circuit used to manage the external power input to the motor, which is connected to the motor via four switches. The schematic and control logic is as follows.

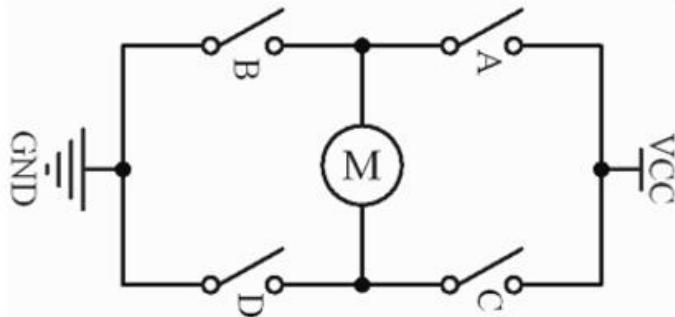


Figure 3.24 Schematic diagram of H-bridge [5]

A	B	C	D	STATE
ON	OFF	OFF	ON	CW
OFF	ON	ON	OFF	CCW
ON	ON	OFF	OFF	BRAKE
OFF	OFF	ON	ON	BRAKE
OFF	OFF	OFF	OFF	COAST

Table 3.4 Control logic of H-bridge [5]

Note that this circuit should avoid simultaneous conduction of AB or CD, in which case VCC would be directly connected to GND, shorting out the power supply.

### MOSFET

In circuits, MOSFETs are often used as switches, and their operating principle is as follows.

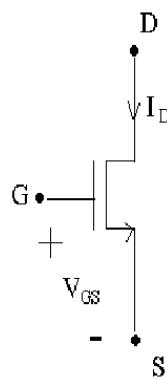


Figure 3.25 Schematic diagram of NMOS [6]

The NMOS has three terminals, drain(D), gate(G), and source(S). When the voltage of  $V_{GS}$  is greater than the threshold voltage ( $V_{th}$ ), the drain current( $i_d$ ) becomes more prominent as the  $V_{DS}$  becomes larger until it reaches the

saturation region. in this case, the switch is turned on. When the voltage of V<sub>GS</sub> is more minor than V<sub>TH</sub>, the ds terminal is disconnected, and the id does not change with V<sub>DS</sub>, at which point the switch is disconnected. In summary, NMOS can be seen as a switch controlled by the gate voltage [6].

### MOSFET selection principle

A complete H-bridge circuit requires four NMOSs. The first parameter to be considered is the drain-to-source voltage withstand value (V(BR)DSS), this value indicates that the MOSFET will be burnt out when the VDS is more significant than this value. In practice, when the motor stops suddenly, a voltage more critical than the operating voltage will be generated. This gives the principle that the withstand voltage should be at least twice the operating voltage. In this circuit, this value is at least 24 V. Then, there is the continuous drain current (Id), this value indicates the maximum value at the Id continuously. However, this value is measured under ideal thermal conditions. In practice, the withstand drain current will be less than this ideal value. Thus, this value is chosen to be as large as possible. The third parameter to be concerned with is the resistance over drain and source when the MOSFET is on (R<sub>DSON</sub>). When the MOSFET is working, the more excellent the internal resistance, the greater the voltage drop between the drain and source. This would cause a more significant power consumption and heat generation and is more likely to cause the MOSFET to be damaged. When choosing NMOS, the principle is to select the smaller on-state resistance the better. In addition, when using the MOSFET as a switch, it should note that the MOSFET should be thoroughly conducted. Otherwise, the on-state resistance will also increase, and leads to overheating. The last parameter is the thermal resistance(R<sub>θJA</sub>), this factor can be used to estimate the temperature rise of a MOSFET at a given current. The parameters of the NMOS NTMFS4833N we chose are displayed below.

V <sub>(BR)DSS</sub>	V <sub>GS</sub>	I <sub>d</sub>	R <sub>DSON</sub>	R <sub>θJA</sub>	V <sub>TH</sub>
30V	±20V	191A	2mΩ@10V	137.8°C/W	1.5V-2.5V

**Table 3.5** Parameters of NMOS NTMFS4833N [7]

The temperature increase can be estimated by

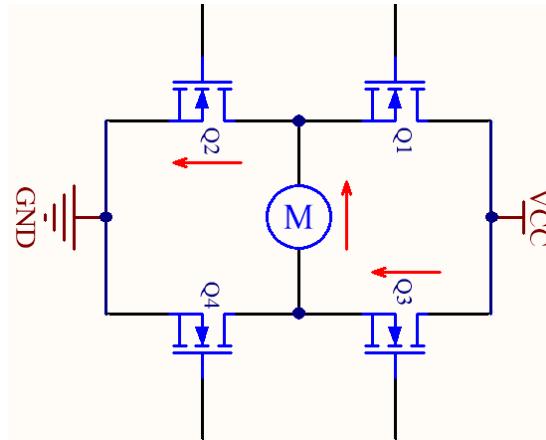
$$T = T_E + I^2 \times R_{DSON} \times R\theta JA = 25^\circ C + (8A)^2 \times 137.8 = 42.6^\circ C$$

Where TE denotes the environment temperature. This formula indicates that the MOSFET is within the Operating junction temperature of -55 to 150 [8].

### H-bridge driver

This NMOS has a maximum threshold voltage of 2.5V, which the high-level output of the microcontroller can be controlled. However, in the H-bridge circuit,

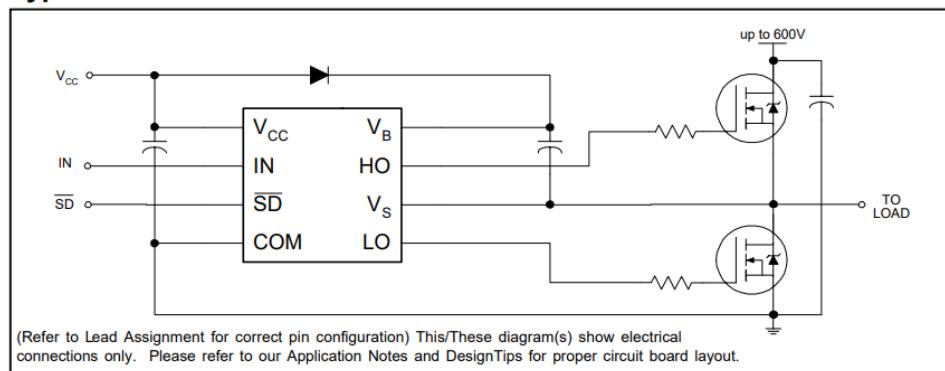
this voltage cannot conduct a current path consisting of two NMOSs. The reason is given as follows based on the given H-bridge diagram. Assume that both the upper right and lower left NMOSs are connected with a shorted load when the upper right NMOS is conducted. At this point, the voltage of VS is approximately equal to the voltage of VD due to the small internal resistance of the conduction of NMOS. To turn on the NMOS, the VGS should be greater than VTH, which is equivalent to that Vg is greater than VTH + VS. In this circuit, this voltage is  $12 + 2.5 = 14.5\text{v}$ .



**Figure 3.26** H-bridge consists of NMOS [5]

Whereas the output from the microcontroller cannot provide such a high voltage. A half-bridge driver circuit is applied to provide this voltage. The recommended circuit diagram for the commonly used half-bridge driver chip IR2104S is given in the datasheet. This circuit consists of two parts, the chip and the bootstrap circuit.

#### Typical Connection



**Figure 3.27** Connection of H-bridge driver circuit [9]

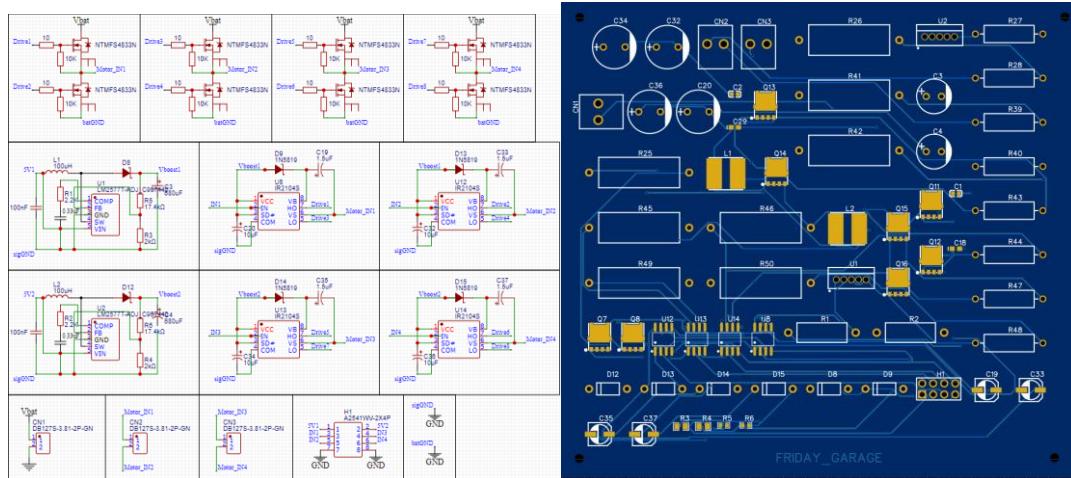
This is the typical connection of the half-bridge driver IR2104S, where the VCC is the power input, IN and SD are the control input, the HO and LO are connected directly to the gate of the two MOSFETs, and the VB and VS are used to construct the bootstrap circuit, and the COM pin is connected directly to the ground [9].

#### Boot-strap circuit

The boot-strap circuit drives the H-bridge following the given steps. When the course starts to operate, the PWM wave is used to generate the control signal. As the left upper and lower MOSFET control signals are low and high, respectively, the upper MOSFET is switched off, and the lower MOSFET is switched on. At this moment, the bootstrap capacitor is charged, and the voltage across the capacitor is  $VCC-GND=12v$ . When the upper MOSFET is on, and the lower MOSFET is off, the bootstrap capacitor starts to discharge. Due to the diode, the capacitor discharges to the gate terminal of the upper MOSFET so that the voltage difference between the two gates and the source is 12v. When the capacitor discharges fully over time,  $VG$  will finally equal  $VCC$ . Assuming the MOSFET is still on, it gives that  $VD$  is equal to the  $VBAT$ . In this case, if  $VGS$  is still greater than  $VTH$ , the MOSFET remains on, but as  $VGS$  decreases,  $RDS(ON)$  will increase, and heat dissipation rises. If  $VGS$  is less than  $VTH$ , the upper MOSFET is turned off again. To keep the MOSFET at the on-state, the capacitor needs to be continuously charged and discharged. Thus, a PWM wave is used for controlling [9].

Here the 12V  $VCC$  is provided by the boost converter circuit, where it converts the 3.3 VCC output of the microcontroller to 12V for the half bridge driver and the  $VBAT$  is given by a 5-C discharge rate battery.

The final construction of the driver circuit design and PCB allocation are shown below.



**Figure 3.28** Overall design of driver circuit and its PCB allocation

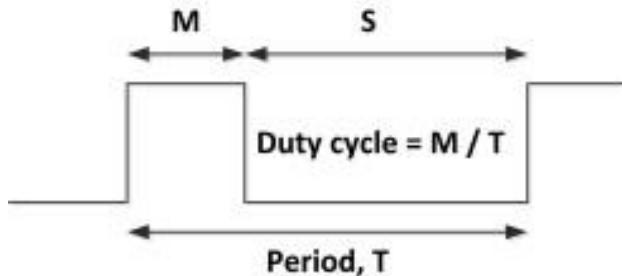
Where the  $V_{BAT}$  is given by a 5-C discharge rate battery.

#### 3.1.5 Pulse Width Modulation—Wang Ke

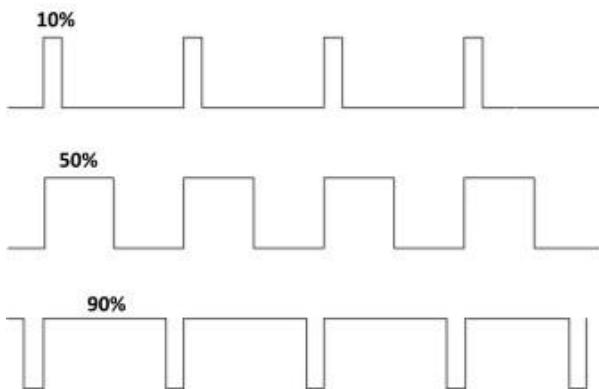
##### The Theory of PWM

Pulse Width Modulation is a digital coding method for analog signal levels. The level of a specific analog signal can be encoded by modulating the duty cycle of a square wave with a high-resolution counter. The duty cycle is the

proportion of the on-time relative to the total time within a pulse cycle.



**Figure 3.29** A typical PWM signal



**Figure 3.30** PWM signals with different duty cycles

The PWM signal is a digital signal. This is because the full amplitude DC supply is either completely present or completely absent at any point in the PWM signal. These are the two states ON and OFF in the PWM signal. The voltage or current source is applied to the analog load in a repetitive sequence of ON or OFF pulses. ON is when the DC supply is applied to the load, and OFF is when the supply is disconnected. Therefore, any analog value can be encoded using PWM as long as the bandwidth is sufficient.

One advantage of PWM is that the signals from the processor to the controlled system are in digital form after digital to analog conversion. The effect of noise can be minimized. Noise can only affect the digital signal if it is strong enough to change logic 1 to logic 0 or logic 0 to logic 1.

The increased resistance to noise is another advantage of PWM over analog control, and this is the main reason PWM is sometimes used for communication. Switching from analog signals to PWM can significantly extend the communication distance. At the receiving end, modulated high-frequency square waves can be filtered out and the signal reduced to analog form utilizing an appropriate RC or LC network.

#### The Usage of PWM

The PWM signal is used in many parts of the rover's construction. In the drive part of the rover, the microcontroller needs to output two PWM signals to

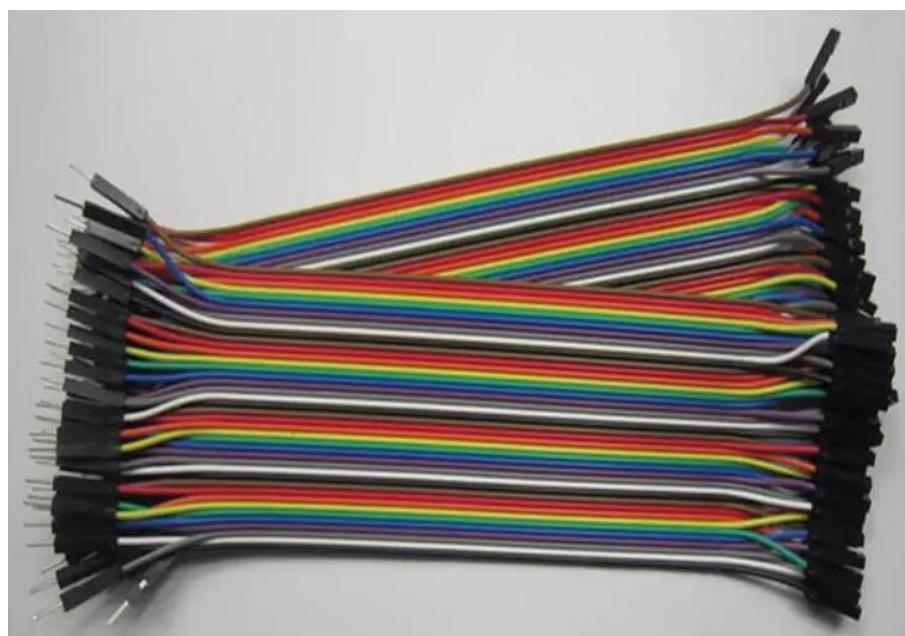
act as ENABLE signals to control the drive board. Each ENABLE signal in the drive board controls the start and speed of one motor.

The amplitude is usually fixed by the logic 1 level of the microcontroller output, which depends on the power supply voltage. In some applications, it may be necessary to use external circuitry to increase the amplitude. In the control of the rover, we will be using the output logic 1 level of the microcontroller which is +3.3 V. The frequency depends on the application. In this project, we will generate a PWM signal with a frequency of 40 kHz.

#### 3.1.6 DuPont Thread

##### The Introduction of DuPont Thread

DuPont thread is a sewing thread with special effects produced by the American company DuPont.



**Figure 3.31** DuPont thread

DuPont cables are available with both male and female connectors. As shown above, the connector with the exposed pins on the left is the male connector and the one on the right is the female connector. The DuPont cable is easy to use and only requires easy plugging and unplugging to connect the various pins.

##### Why DuPont Thread?

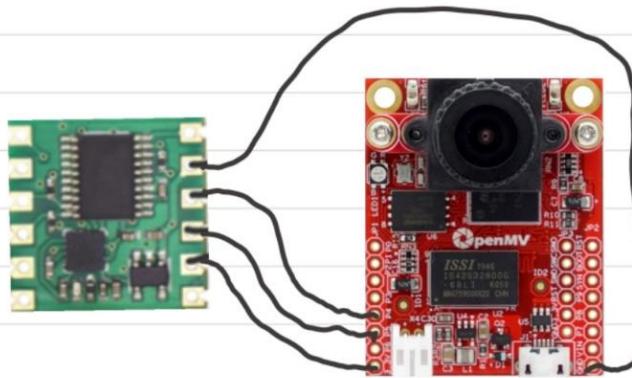
The rover's design, assembly, and debugging process required many adjustments to the pin-to-pin connections. We, therefore, need to connect the pins by cable rather than by soldering before finalizing the design. DuPont wire is used extensively in rover. In contrast, solder connections are hardly ever used, except for the mounting of components on the driver board. This is a great convenience for us.

### 3.1.5 Electronic Compass—Wang Shuyu

#### Electronic compass

##### Electronic compass implementation

In our task, we need to get the car to travel in a fixed direction or turn to a specific angle. Therefore, using an accurate electronic compass can effectively correct the travel direction of the rover.



**Figure 3.32** Connection between HMC5883L and OpenMV

In this project, we choose hmc5883l as the electronic compass of our robot car. HMC5883L is a three-axis magnetic field sensor, an electronic compass module that communicates using the IIC protocol. The traditional compass uses a magnetized magnetic needle to sense the earth's magnetic field. The electronic compass is the same, except that a magnetoresistive sensor replaces the magnetic needle, and then the perceived geomagnetic information is converted into a digital signal and output to the central processing unit. In order to get the value returned by the electronic compass, we need to read the data stored in the register through the IIC protocol. The original data representing the X, Y and Z axes are stored in different addresses. The data of each axis is 16 bits in total and stored in two registers. After reading the register's data, we can get the components in the x and y directions through the calibration algorithm.

Our Python program returns the angel of the heading direction of our rover when the formula calls the function of the compass.

$$\theta(\text{angle}) = \arctan(y - y_{\text{offset}} / x - x_{\text{offset}}) * 180/\pi$$

#### Problems of electronic compass

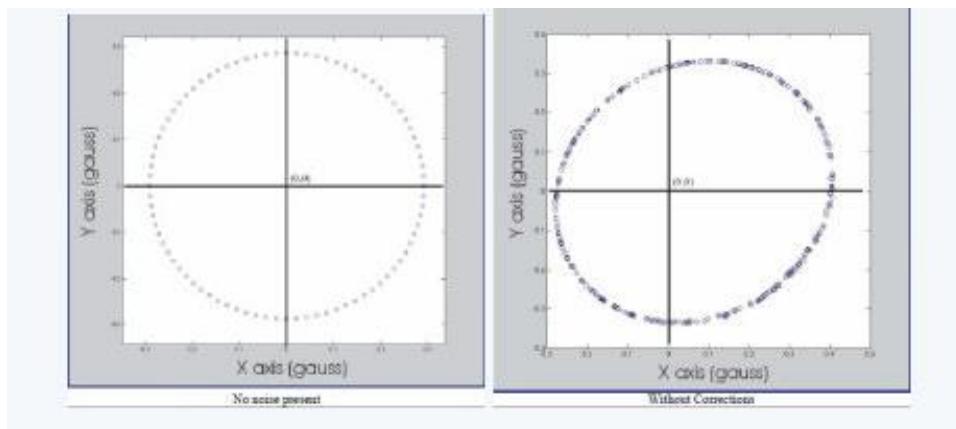
In both patio 1 and patio 2, when the robot car is turning, we need a precise and fast return angel value from our electronic compass. First, we need to calibrate our electronic compass before using it. The initial XY value is directly generated by the local electromagnetic field, which is usually different from the magnetic field of the test site. Thus, we need to get the  $y_{\text{offset}}$  and  $x_{\text{offset}}$  on

the formula before to get the right direction.

We can take the same point as the circle's centre, rotate the electronic compass at a horizontal and uniform speed, record the values in the X and Y direction, and then apply the formula to calibrate our compass.

$$y_{offset} = (y_{max} + y_{min})/2$$

$$x_{offset} = (x_{max} + x_{min})/2$$



**Figure 3.33** Calibration of electronic compass

Another problem is that the sensor's response speed seems not very sensitive. In our tests, when we turn the compass, there is a latency of about 0.2s to 0.4s between the turning and the sensor returning the correct angle value. To solve the problem, when the car turns, we program it to turn to a specific angle and slow down the car to reduce the impact of delay.

#### IIC protocol

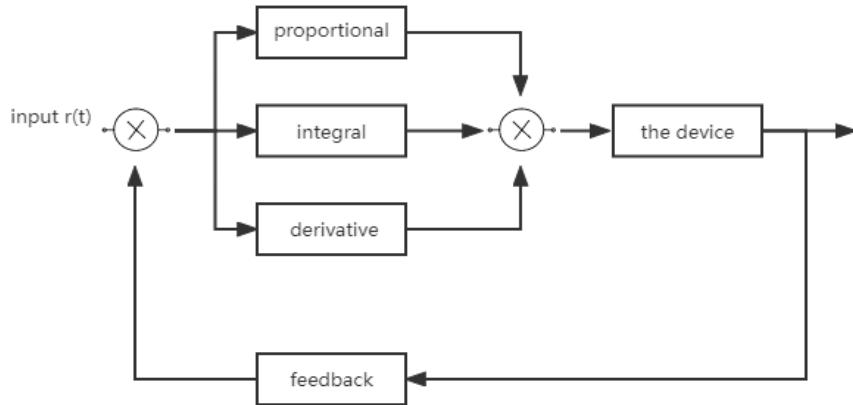
IIC is divided into two buses: SDA (data line) and SCL (clock line). Devices requiring communication are mounted on these two buses. Devices like hmc5883l have their own IIC addresses. There are 8 general address bits (7 bits are address bits and 1 bit is direction bit). The heading angle of the car is transmitted to OpenMV by IIC communication.

### 3.1.6 PID Algorithm—Wang Shuyu

#### Technical content

We found that the car cannot run straight when we apply the same input for both motors, so we decided to use the PID algorithm to correct the head direction of the car. PID algorithm is a common control algorithm to keep the device stable. Proportional, integral and derivative. "P" represents the current information, which can correct the deviation and make the response quickly. "I" represents the information accumulated in the past. It can eliminate the static error and improve the static characteristics of the system. "D" represents the future information, forcing the process at the beginning of the process, reducing overshoot at the end of the process, overcoming oscillation, improving the

system's stability and accelerating the system's transition process.



**Figure 3.34** PID Algorithm Diagram

#### Problems of PID algorithm

Initially, I planned to use the wheel rotation speed as the PID algorithm's feedback. Suppose the speed of the wheel rotation can be obtained. In that case, the speed of the robot car and the speed difference between the wheels on both sides can be calculated very accurately, and the vehicle can move in a straight line and turn very accurately. We can use the hall sensor on the motor to obtain the speed and then calculate the car's speed. However, after several experiments, I found that this method is not feasible and time-consuming to implement. We need four pins on the OpenMV to get feedback from one motor and 8 pins for both wheels to use the hall sensor. There are not enough pins on the OpenMV for the driving part, so we have to give up this idea.

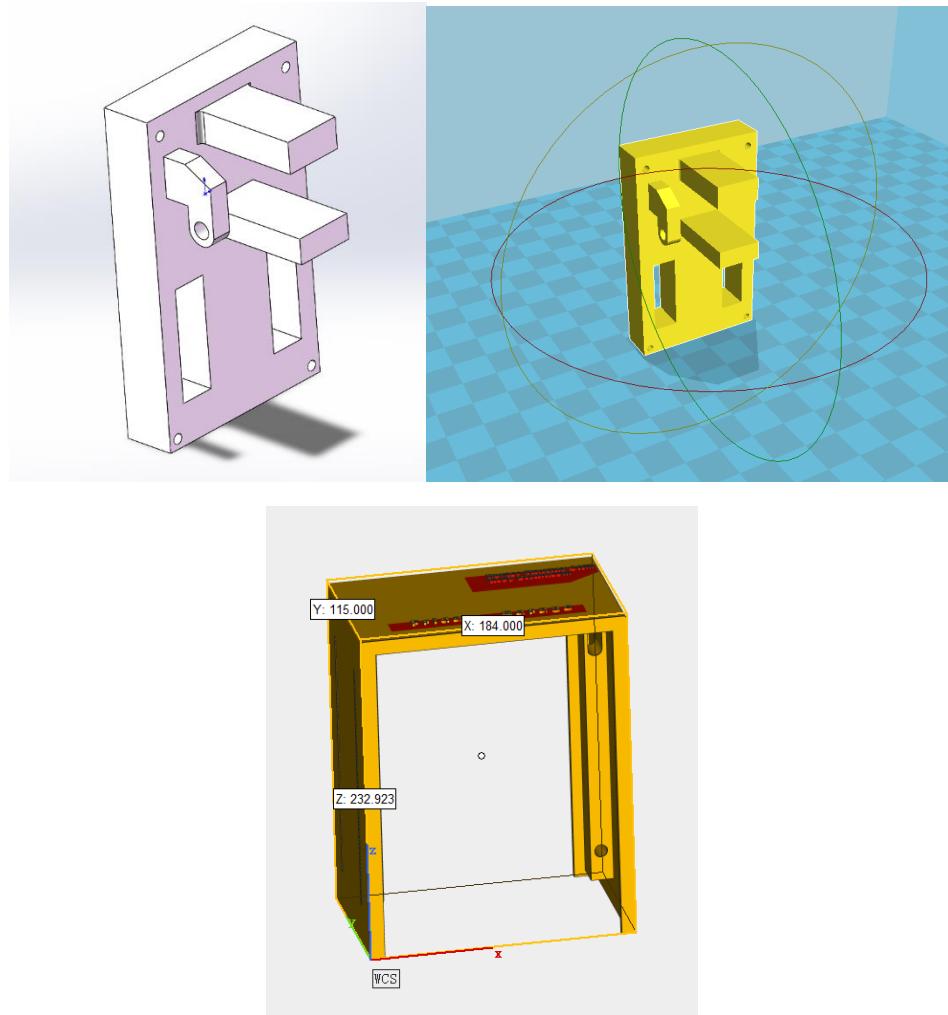
I choose to use an electronic compass for the feedback, and it can also help the rover run straight and turn in the right direction.

#### 3.1.7 3D Printing—Chang Zeyu

3D printing was used to manufacture the outer shell and parts of the rover to control consumption. The overall budget and cost of the project are critical in designing a project [10]. Besides, the material of the shell and parts involved in the design of the rover also need to be carefully considered. An excellent project does not mean that the materials used in the designed product are the best, but rather that the cost of producing the product is the most reasonable [10]. In other words, the material and structure of the product can meet the product's needs.

Costs were controlled throughout the project. 3D printing was considered a suitable option because the school's 3D printer was available to use, which meant that no money needed to be spent on 3D printing. Therefore, some parts of the rover, such as the terrace that holds the OpenMV, the accessories that hold the mechanical claws and the sonar detection were produced through 3D

printing. Below is one of the components and shell produced by 3D printing.

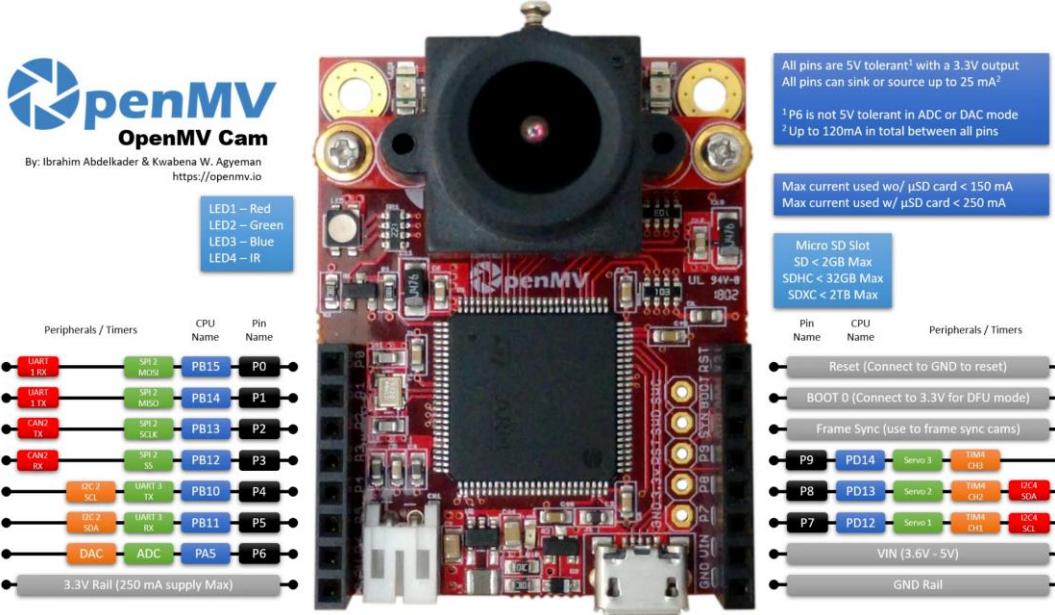


**Figure 3.35** Model of the Terrace and Outer Shell Constructed by Computer Software

## 3.2 Vision Modules

### 3.2.1 The OpenMV Cam

OpenMV is an integrated circuit board with machine vision modules. It is characterized by small size, low power consumption and open source. Machine vision algorithms can be implemented through MicroPython. Because of Python's advanced data structures, complex data is easy to process by programming. It integrates the STM32F427 processor as the core and the OV7725 camera chip as the image input source. C language implements the core algorithm of machine vision efficiently in hardware. Since many Python modules are designed in advance, it is easy to call mature modules during actual programming, which makes it very easy and efficient to operate.



**Figure 3.36** The pin configuration of the OpenMV Cam

## Applications of OpenMV

**Colour Tracking:** OpenMV can detect up to 16 colours in an image at once, with each colour having an unlimited number of colour blocks. Each colour block's position, size, centre, and orientation are determined by OpenMV. OpenMV Cam may be designed to follow the sun, line tracking, target tracking, and more using colour tracking.

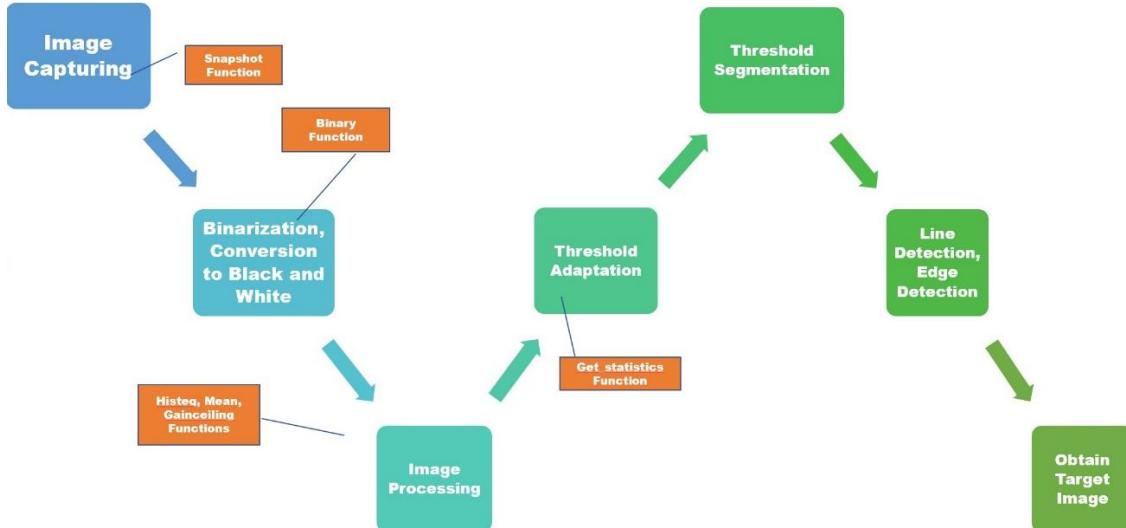
**Marker Tracking:** OpenMV Cam can be used to detect groups of colours instead of individual colours. That is, for two or more tags of a particular colour, OpenMV retrieves the contents of the tag object.

**QR Code Detection/Decoding:** OpenMV Cam can read QR codes in its field of view. Through QR code detection/decoding, the intelligent robot can read the tag in the environment.

**Shape Matching:** OpenMV Cam can match shapes by Hough Transform. It can quickly detect infinitely long lines at almost full frame rate. Moreover, it is possible to find lines that are not infinitely long. The detection accuracy of the circle is very high.

### 3.2.2 Path Tracing System—Chang Zeyu

The line patrol is applied throughout patio 1. The road tracking function is to detect the specified route and ensure that the rover can keep travelling within the range of the route [11]. OpenMV and the drive module were jointly utilized to achieve the line patrol task. In the designed OpenMV system, OpenMV continuously captures pictures. Then the system analyses the captured photos and could obtain a binary image.



**Figure 3.37** Image Processing Flowchart

After the processed image is obtained, the direction the rover needs to travel will be calculated by the PID algorithm in the drive module, and the rover can make corresponding adjustments. It is worth noting that the colour threshold was used throughout the route pathing process, which is related to OpenMV's unique LAB colour space. This section mainly focuses on the methods of path tracing, and the detailed settings of the LAB colour space are presented in the colour trace section.

## Road Capturing

Considering that the clarity of OpenMV is only 4800 pixels and can be affected by light. Therefore, three methods were adopted to meet the tracing requirements. These are threshold adaption, threshold segmentation, and line detection.

### Threshold Adaptation

The Threshold Adaptive module is designed to read the colour information of the route and measure its average threshold [12]. As opposed to the fixed thresholding operation in OpenMV, the threshold value is different for each pixel point in the image in adaptive thresholding, determined by a point-weighted average of the image pixels in its field [12].

Advantages of this:

- OpenMV is heavily influenced by ambient light and it is possible that the thresholds set in advance may not be suitable for the actual game or application site and that the thresholds taken in advance may not work if mixed colours or extreme conditions are encountered [12]
- Local image areas with different brightness, contrast, and texture correspond to local binarization thresholds [13].
- This avoids the need for tedious manual debugging.

The principle of the adaptive thresholding algorithm is a two-dimensional Gaussian distribution with the following equation.

$$f(x, y) = \left(2\pi\sigma_1\sigma_2\sqrt{1-\rho^2}\right)^{-1} \exp \left[ -\frac{1}{2(1-\rho^2)} \left( \frac{(x-\mu_1)^2}{\sigma_1^2} - \frac{2\rho(x-\mu_1)(y-\mu_2)}{\sigma_1\sigma_2} + \frac{(y-\mu_2)^2}{\sigma_2^2} \right) \right]$$

The Gaussian probability function relies on the two-dimensional coordinates of the image, i.e. (x,y). The centre of the obtained image is used as the origin of the coordinates for sampling [13], from which the coordinates of any position can be obtained.

(-1, 1)	(0, 1)	(1, 1)
(-1, 0)	(0, 0)	(1, 0)
(-1, -1)	(0, -1)	(1, -1)

**Figure 3.38** Example of 3\*3 Gaussian Function Template

A Gaussian filter template can be generated based on the relative coordinates of the points obtained [13]. Each has a weight, also known as a coefficient, obtained by discretizing the Gaussian function. The value of the standard deviation  $\sigma$  can be chosen arbitrarily, which determines the effect of the system on the image [13]. If  $\sigma$  is small, the smoothing of the image is not so noticeable; conversely, if  $\sigma$  is large, the smoothing of the image is more prominent. Besides, there are subtle differences in the thresholds obtained for different values. In OpenMV, the standard deviation can be determined by the function

```
statistics.stdev()
```

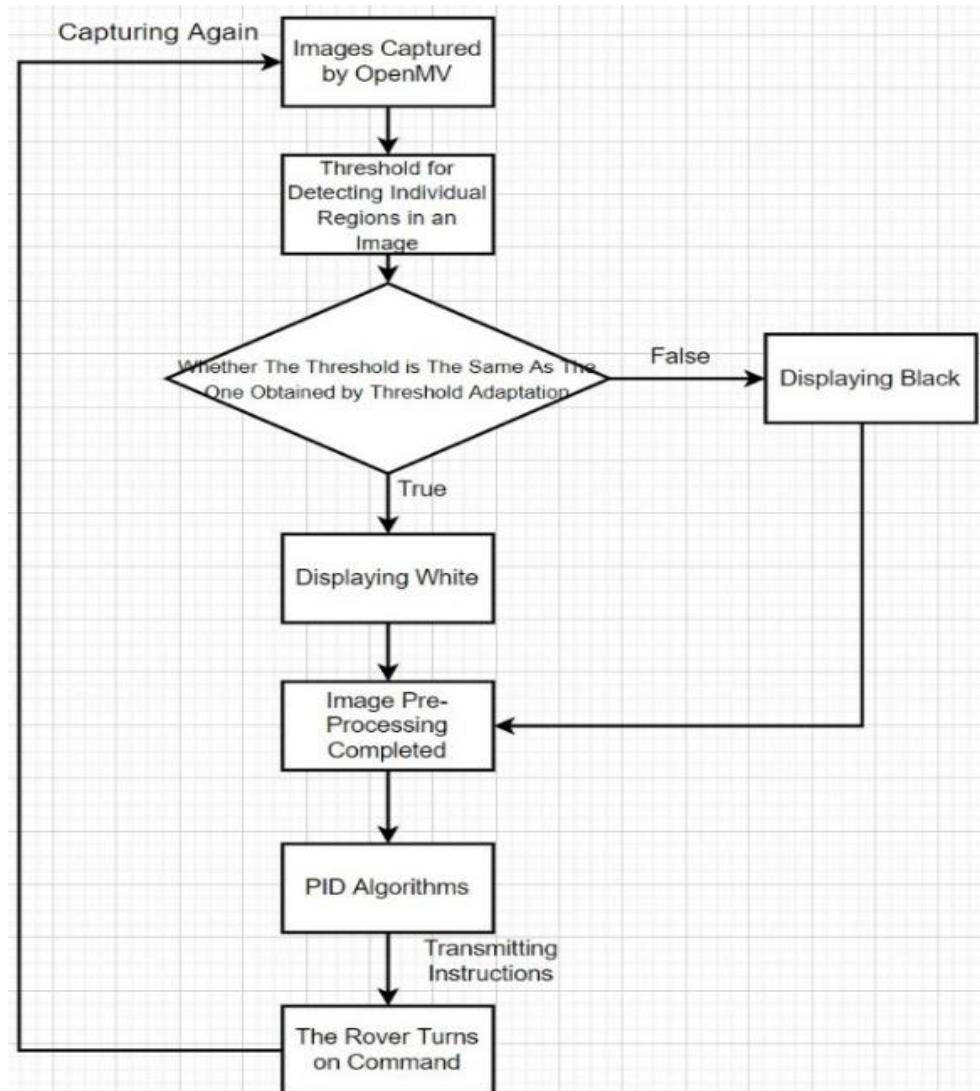
accompanied by the function

```
statistics.mode()
```

The plurality of thresholds in the target image is obtained. This method greatly improves the accuracy of the thresholds and deals well with the inevitable noise in the external environment. Also, as weather conditions could impact the performance of OpenMV, the values obtained for threshold adaptation in multiple conditions were designed to place in an array to improve OpenMV's performance in capturing routes in all weather conditions.

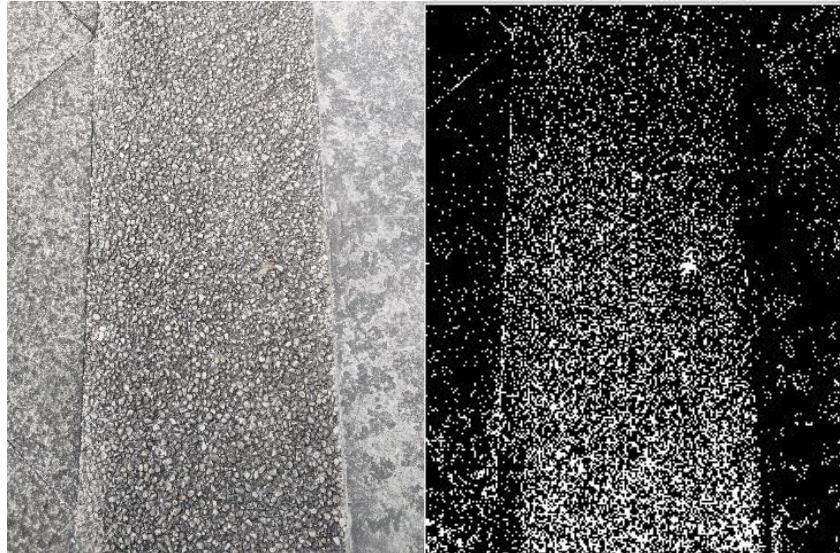
### Threshold Segmentation

Thresholding, a region-based image segmentation technique, is integral to implementing patrol lines and is based on the principle of dividing image pixel points into some classes. Image thresholding aims to separate the collection of pixels according to grey levels, with each subset obtained forming a region corresponding to the actual scene [14,15]. Parts have consistent properties within each area, while neighbouring areas do not have such consistent properties. According to this principle, it is theoretically possible to distinguish the road to be driven from its surroundings.



**Figure 3.39** Threshold Segmentation System Design

The set threshold range will appear white after processing and the remaining subset of thresholds will appear black [14, 15].



**Figure 3.40** Before Processing (Left) After Processing (Right)

The threshold segmentation method simplifies the analysis and processing steps. Stable performance is ensured and tracing requirements are met. Moreover, this method could provide timely feedback on the rover's travel. Combined with the function

```
histeq(); mean(2)
```

The resulting images were sharpened and de-noised to improve image processing accuracy, which could be applied to overcome the problem of the roads on the test site being similar in colour to the surrounding gravel roads.

#### Line Detection

Line detection is used to identify road edges, which assist the rover in the steering operation during the patrol. Line detection is based on the principle of the Hough transform, which is a feature extraction with an ample parameter space [16]. Although the images captured by OpenMV may be blurred, the algorithm can decide based on the local maximum in the accumulator space. Such a mechanism is more like a weighted decision [17]. Remarkably, the Hough transform algorithm has a unique step that allows it to find the parameters of a graph in a complex parameter space, which can significantly improve the recognition rate of roads [16, 17].

The data for the Hough transform is derived from the processing of images collected by OpenMV by the Sobel filter. The core is the Sobel operator, widely used for edge detection [18]. The application of the Sobel operator to edge detection is based on two 3\*3 kernels, which act in the horizontal and vertical directions, respectively.

-1	0	1
-2	0	2
-1	0	1

-1	-2	-1
0	0	0
1	2	1

**Figure 3.41** Sobel Operator, Horizontal (Left), Vertical (Right)

When processing an image, the Sobel operator performs a differential operation on the pixel values in the horizontal and vertical directions of the image, which in mathematics can be written as:

$$grad(I) = \left[ \frac{\partial I}{\partial x}, \frac{\partial I}{\partial y} \right]^T$$

This effectively reduces the interference caused by noise and helps the feature extraction of the Hough transform [18].

The effective identification of road edges means that rovers can better plan and adjust the trajectory and are less likely to deviate from their route. In addition, the ground colour of the rover's route is not uniform, so the use of edge line recognition can overcome this problem and ensure that the rover travels appropriately.

## Optimal Processing of Patrol Identification

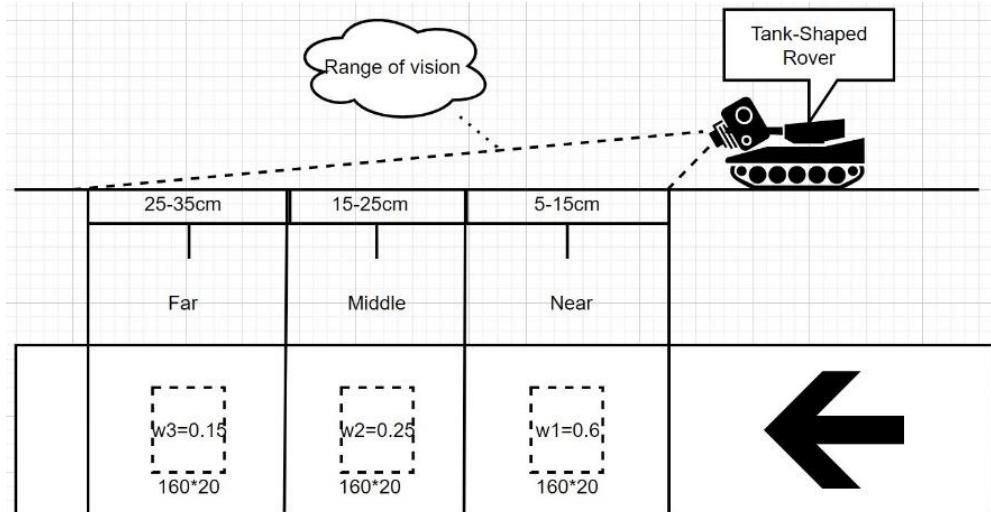
In the actual test, the rover's patrol performance is greatly affected by the external environment. For example, in the case of strong sunlight and a large inclination angle, the rover could change the colour of the road in the image captured by OpenMV. While on rainy days, the reflection of light by water on the road can also affect the threshold adaptation process. It is therefore crucial to design a roving rover for all weather conditions.

### Threshold Setting with Weights Attached

The threshold adaptation module changes from the threshold recognition of a single region to the threshold recognition of multiple areas with different weights to improve the recognition of the route. And throughout the rover's travel, the code is designed to allow OpenMV to continuously perform the process of threshold adaptation. In detail, the field of view that the camera can capture is divided into three areas: near, middle, and far. OpenMV can take a threshold reading in each of the three regions and the three results will be weighted and summed with weights of 0.6, 0.25, and 0.15, respectively. Then

### 3 Subsystem Design

the rover could perform the threshold segmentation by the computed threshold results.



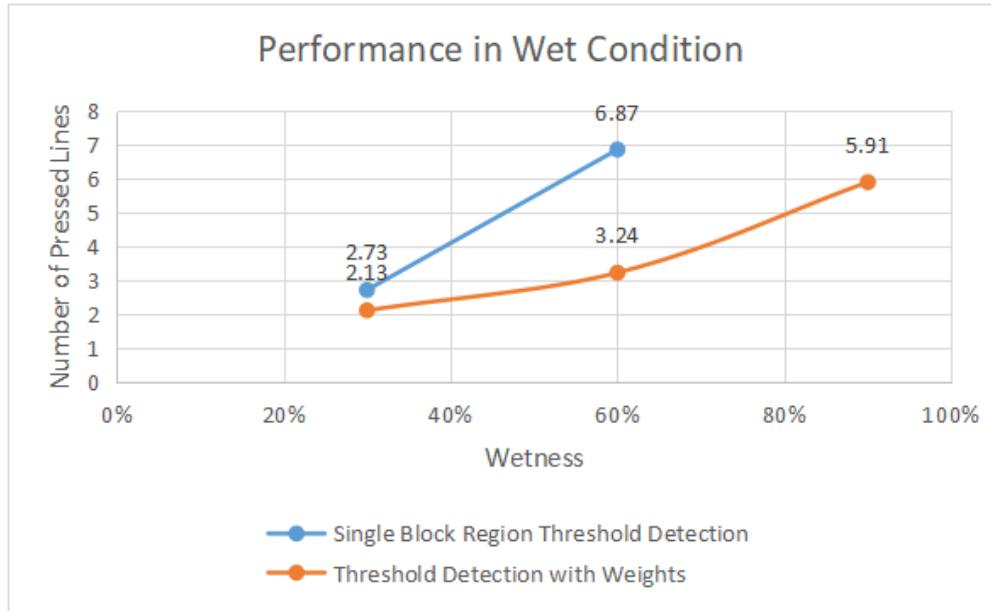
**Figure 3.42** Weight Identification

The rover conducted a line inspection test after rain to verify the optimization effect. The non-weight-adjusted threshold-segmented rover was used as the control group. The weight-adjusted threshold-segmented rover was the experimental group to carry out the experiments. The experiments were carried out at the same period, with the rovers having the same speed and at the same starting point, and the performance of the patrol was represented by the number of times the rover pressed the line. The experiment was repeated ten times and the results were averaged. The following are the results of the experiments.

Wetness (ratio of watered area to total area)	Single Block Region Threshold Detection	Threshold Detection with Weights
Five Hours After Rainfall (30%)	2.73	2.13
Two Hours After Rainfall (60%)	6.87	3.24
Just Raining / Raining (90%)	N/A	5.91

**Table 3.6** Performance of the Algorithm in Different Wetness Conditions

During testing, we found that Single Block Region Threshold Detection failed to recognize when humidity conditions reached 90%. Furthermore, the Threshold Detection with Weights showed better performance than the usual threshold detection in all humidity conditions and the line graph was used to indicate a more visual comparison.



**Figure 3.43** Performance in Different Wet Conditions

Similarly, the performance shown under different lighting conditions was also used to test the optimization effect.

Brightness (Depending on the harshness of the sun)	Single Block Region Threshold Detection	Threshold Detection with Weights
Morning (40%)	1.45	1.24
Afternoon (60%)	7.12	4.62
Noon (100%)	10.93	7.69

**Table 3.7** Performance of the Algorithm in Different Brightness Conditions

Threshold Detection with Weights also showed better patrol results in light interference conditions. When the two sets of experiments are compared, it can be concluded that light interference is more disruptive to OpenMV during the patrol.



**Figure 3.44** Performance in Different Light Conditions

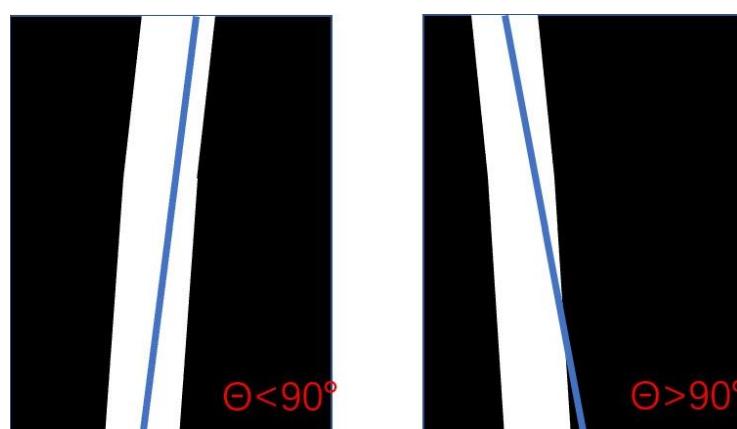
## The PID Algorithm Realizes the Control of the Rover

The PID algorithm is used to determine the steering operation of the rover. This section mainly explains the mechanism of the PID algorithm to judge the steering, and the detailed algorithm content and principle are presented in the drive module.

The PID algorithm can present a straight line in the image based on a threshold segmentation, representing how the rover is steered. In the algorithm, the slope and intercept of the straight line are expressed as

$$\text{theta\_pid}, \text{rho\_pid}$$

The straight line is supposed to stay in the centre of the picture. The effect of the PID algorithm processing is shown below.



**Figure 3.45** PID Algorithm for Image Processing

The system will then send adjustment instructions to the rover to achieve  $\theta = 90^\circ$ .

#### 3.2.3 Shape Matching—Jian Yingchao

In Patio 2, preassigned paths at the beginning are determined by three different shapes. For each shape, they are matched to one specific route and then different tasks will be completed. If the shape to be tested is a circle, the designed robot will move to the upper tip of the ground diamond. If the shape is a triangle, the robot will run to the left tip of the diamond. If the shape is square, the robot will move to the lower tip of the diamond. When the direction is determined, the corresponding shape behind the stand will be knocked out. Therefore, the visual module should have the ability to detect three types of shapes. In this part, the author will introduce the shape recognition method adopted by our group and related technical details.

#### General Process

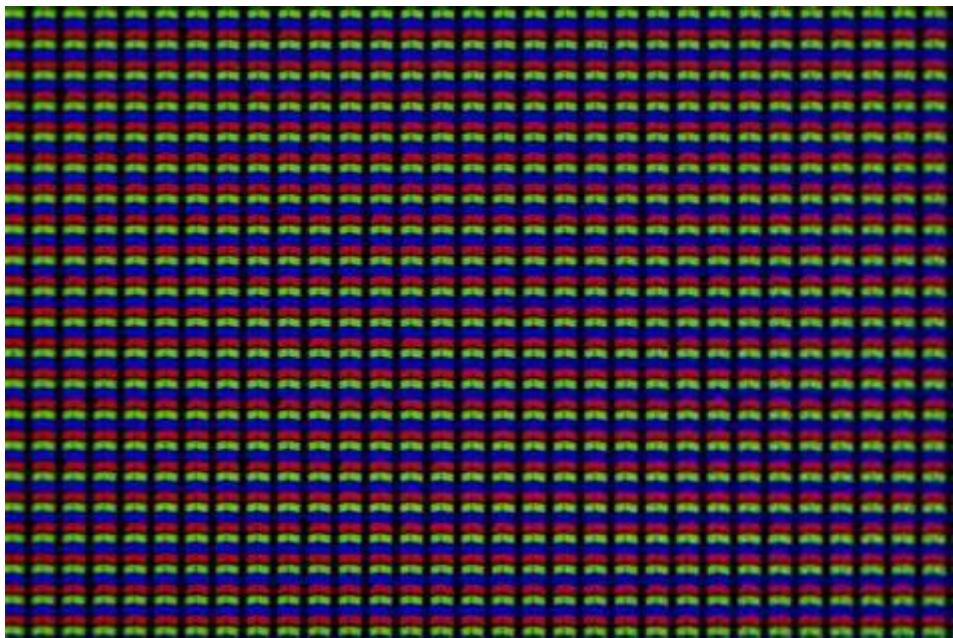
Since there may be several interfering shapes and lines in the real situation, the area to be tested should be determined with an accurate detection size of the detected shape. This requires the segmentation of the region of interest (ROI) for detection in the process of shape matching. This is achieved by determining the colour threshold of the detected area in advance and then finding the biggest colour block, which is the desired ROI. Because the shape to be detected is placed inside the border, the border may affect the detection of the shape, that is, the border may be detected as a square and cause interference. The solution to this problem is to ensure that the ROI of shape detection is smaller than the border. A smaller ROI can be achieved by reducing the number of pixels in the colour block, which does not include borders.

As the robot gets closer to the shape, the proportion of the desired colour block in the entire space changes. The detection position of the robot can be controlled by determining the pixel points of the colour block, and the robot can stop moving in a suitable position range. To verify the shapes, the Hough transform will be adopted. Since different shapes have different geometric features, shapes can be distinguished by mathematical models. Due to the different probability of successful detection in the three shapes, the detection accuracy can be improved by setting the detection sequence. The technical details are described below.

#### Pixel background knowledge

The smallest unit in a visual display or digital image is the pixel [19], which is short for picture elements. Pixels are the basis of computer screens. Each pixel is made up of red, blue, and green lighting elements, which are combined

in various ways and intensities to produce millions of different colours [19].



**Figure 3.46** The LCD Screen Pixel Pattern Supermacro

Pixels are the units used to convert binary code data into a picture on a computer screen. Each pixel on an RGB monitor receives a code that instructs it on how to show a specific colour. The code is made up of a triplet of eight-digit values encoded in binary code since each pixel has red, green, and blue lighting elements. This instructs each of the three-colour elements to display at what intensity, and when all three colours are merged, the desired colour is displayed. Each pixel represents a single colour, and the colours are combined to form an image [19].

Three bytes are allotted for each colour of the RGB scale in the conventional 24-bit colour systems used for practically all PC monitors and smartphone displays, resulting in a total of 16,777,216 colour variants. A 30-bit deep colour system allocates 10 bits to each of red, green, and blue, yielding 1.073 billion colour combinations. However, since the human eye's sensitivity to colour is limited, a variety of colours may cause a colour banding issue rather than getting more information [19].

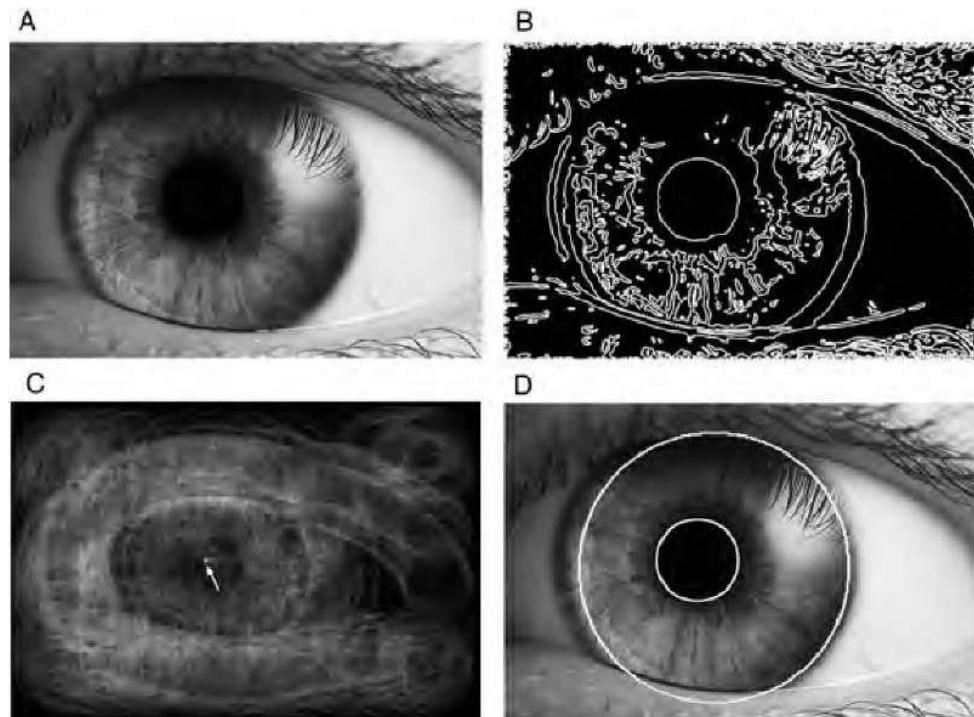
To get an accurate ROI for eliminating all distractions, pixels are used to partition all areas to get ROI containing shapes. OpenMV has a number of designed microPython modules. A function in the image module is used.

***image.find\_blobs(thresholds, x\_stride = 2, y\_stride = 1)***

where value of the input ***thresholds*** represents the threshold at which we want to find colour blocks. For value of the input ***x\_stride***, it is used to find the smallest width pixel in the *x* direction of the colour block, which defaults to 2. For value of the input ***y\_stride***, it is used to detect the smallest width pixel in the *y* direction of the colour block, which defaults to 1.

## Hough Transform

The Hough transform is a technique for separating characteristics of a certain shape inside a picture [19]. The classical Hough transform is most typically employed for the detection of regular curves such as lines, circles, and ellipses since it needs the required characteristics to be provided in some parametric form [20,21].



**Figure 3.47** Round pupil detected by Hough transform

Circle detection and line detection in Hough transform will be used in designing the algorithm.

## Line Detection

In Cartesian coordinates, the coordinates of a line can be determined by two parameters. Parameters  $a$  and  $b$  are used to represent the equation of a line.

$$y = a \cdot x + b$$

However, this form cannot represent vertical lines. Therefore, the form of the Hough transform:

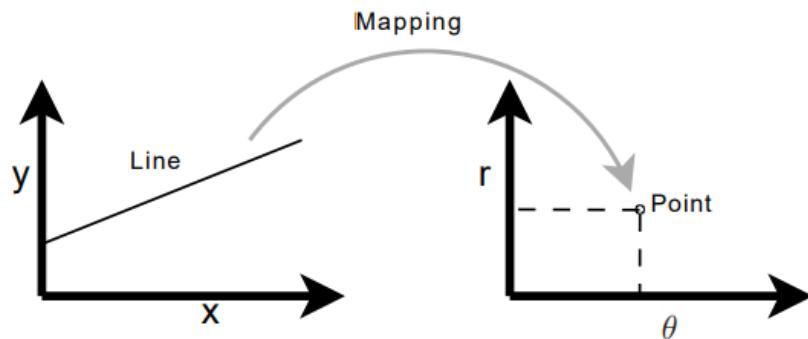
$$r = x \cdot \cos \theta + y \cdot \sin \theta$$

which can be rewritten to resemble the representation of the Cartesian coordinate system:

$$y = -\frac{\cos \theta}{\sin \theta} \cdot x + \frac{r}{\sin \theta}$$

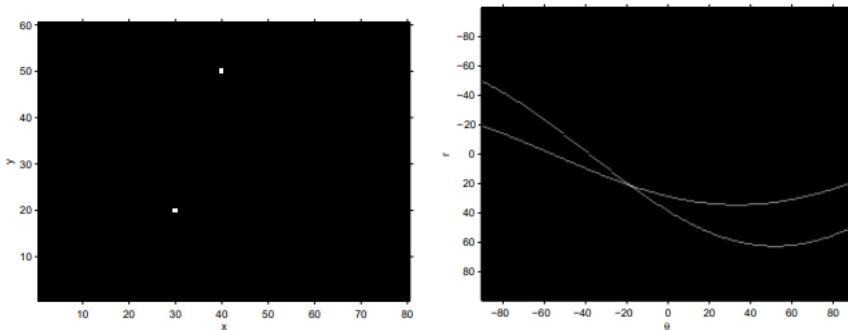
### 3 Subsystem Design

The parameters  $\theta$  and  $r$  are the angle of the line and the distance from the origin, respectively. Both of  $\theta$  and  $r$  are bounded in a range,  $\theta \in [0^\circ, 360^\circ]$ ,  $r \geq 0$ . The Hough space of lines has two dimensions, and a line in Cartesian coordinate system can be directly represented by one point  $(\theta, r)$ .



**Figure 3.48** A transformation of a line from Cartesian coordinate to  $\theta - r$  coordinate

If there is a point in the Cartesian plane  $(x_1, y_1)$  and  $(x_2, y_2)$ , the graph in  $\theta - r$  plane shows here.



**Figure 3.49** A transformation of two points from Cartesian coordinate to  $\theta - r$  coordinate

Therefore, a line in a Cartesian coordinate system should be an intersection on the  $\theta - r$  plane after the Hough transformation, which helps determine lines.

### Circle Detection

In Cartesian coordinates, the coordinates of circle can be determined by equation:

$$(x - a)^2 + (y - b)^2 = r^2$$

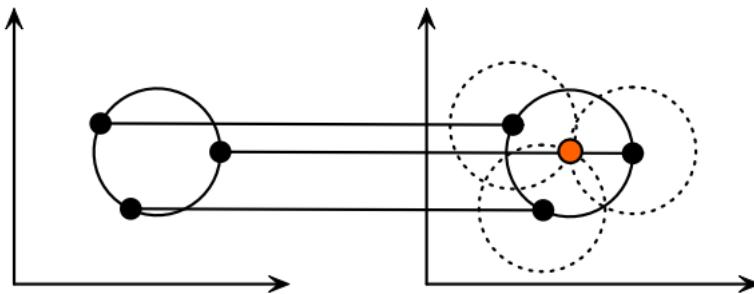
In another form, all points  $(x, y)$  on the circle must satisfy the following conditions:

$$x = a + r \cos \theta$$

$$y = b + r \sin \theta$$

when the value of angle  $\theta$  changes from  $0^\circ$  to  $360^\circ$ , all points  $(x, y)$  will form a circle.

Parameter triplets  $(a, b, r)$  can determine a circle. In the real situation, the radius  $r$  is not determined. Therefore, in actual situation,  $r$  will have a range, and all circles will satisfy this radius range. Circles will be detected  $r$  from smallest to largest. All points in ROI will produce a circle with radius  $r$ . If these points have only one common intersection, they form a circle in Cartesian coordinates.



**Figure 3.50** points in ROI produce a circle with radius  $r$

Some functions designed in **image** module can identify straight lines and circles quickly and effectively

***image.find\_lines(roi, x\_stride = 2, y\_stride = 1)***

where **roi** is the range of interest. **x\_stride** and **y\_stride** represent the number of pixels skipped on the  $x$  and  $y$  axes during the Hough transformation.

***img.find\_circles(x\_margin = 10, y\_margin = 10, r\_margin = 10)***  
**x\_margin, y\_margin, r\_margin** represent the value of the maximum  $x$   $y$   $r$  that can merge circles. **r\_min = 2, r\_max = 100, r\_step = 2** show that the range of  $r$  is from 2 to 100, and each detection radius corresponds to  $r + 2$ .

### 3.2.4 Color Recognition Module—Hu Yang

The main task of this part is responsible for completing the path planning from task one to task two in patio2.

Initially, an infrared rangefinder capable of transmitting and receiving infrared light was chosen as the tool for ranging and positioning. However, in practice, it was found to have many limitations and was not very good for the task of ranging. The main working principle of the infrared rangefinder is to send and receive the reflected signal back and determine the distance between therover and the obstacle by calculating the time interval [22]. Therefore, in the short distance measurement, because the infrared propagation speed is very fast

and the time interval between sending and receiving is very short, it will increase the impact of the delay on the measurement accuracy, resulting in the measured distance does not match the actual. And because the railing is not a wall, and cannot fully rebound infrared, which even leads to in some cases do not get back the distance. Second, because the rover is tested outdoors, it will be affected by many environmental factors, which will also make the measurement results inaccurate. Infrared distance measurement also has the disadvantage of poor directionality [22]. Even if a specific distance is measured, it is not possible to determine the direction of the object and whether the rover is following the expected route. If the direction of the rover is shifted, even if it is only a small angle at the beginning, it will be very different from the correct path after travelling for some time. And it is also difficult to correct the position of obstacles through the algorithm, and it also leads to a huge workload. Therefore, the infrared distance measurement is not suitable for the project of small vehicles.

After that, the idea of placing markers and using the OpenMV camera for color recognition or QR code recognition was proposed. The specific implementation plan is to place markers at the second task location of patio2, and the rover will rotate slowly in place at the specific location until the markers appear in the camera. After determining that the marker appears will adjust the body direction left and right until the marker stops rotating in the centre of the image and proceed in a straight line in this direction. If the marker is found to be off-centre in the image, the rover stops and reorients the body again so that the marker is back to the centre of the image and continues to move forward. Repeat the above steps until the destination point is reached and the body direction is adjusted to control the gate to throw down the tennis ball. This not only can indirectly solve the problem of measuring distance, but also can make the trajectory of the cart follow the expected direction exactly, and even if there is deviation, it can be corrected instantly. In practice, the markers must be easily identifiable and not similar to the surrounding environment, so it is necessary to choose markers with high distinction from the environment to reduce this effect. Secondly, this solution is susceptible to light conditions. Different seasons, time periods or weather conditions can lead to different light intensities. Under different lighting conditions, the colour thresholds of the markers are different, and if only tested according to the standard measured indoors, the markers will not be identified in many cases. Therefore, tests need to be performed and data recorded in the field at different times to ensure that the carts are up to the task in all conditions.

## Marker Selection

Markers are the key to completing route planning. A good marker not only saves a lot of testing time but also increases the success rate of the task. Markers need to be easily identifiable and better able to combat environmental

factors. Based on this feature two types of markers can be selected, plain blocks of colour and images with strong colour contrast themselves, such as QR codes. Both types of markers have their advantages and disadvantages.

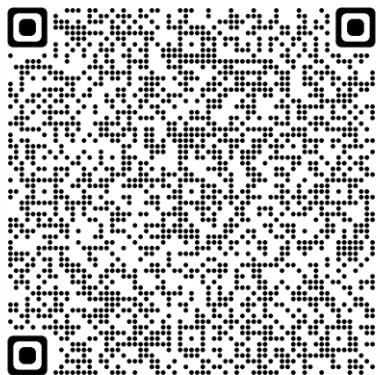


Figure 3.51 QR code



Figure 3.52 Red blocks

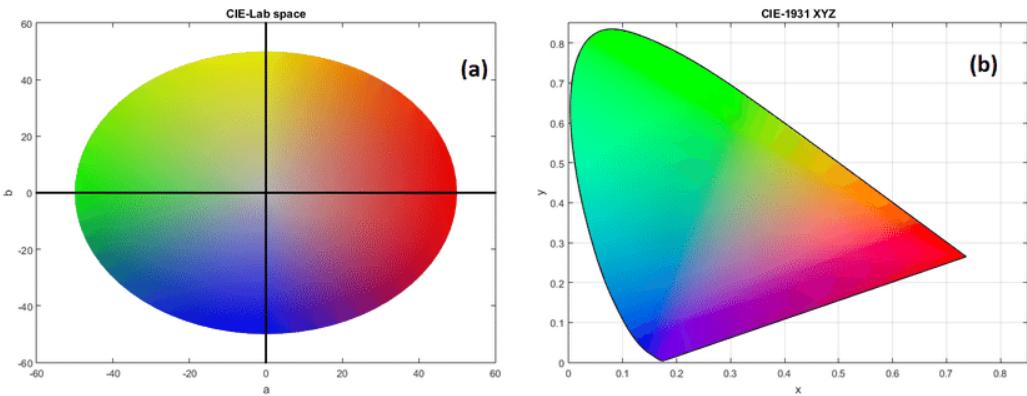
After a simple test, the QR code is better able to fight against environmental interference and has a little higher probability of being recognized in different environments due to its strong contrast, but this requires a higher pixel count for the camera. Because if the pixels are low, the camera cannot capture the exact details of the QR code when it is far away, and it even looks like a small black colour block. This can render the program completely useless and therefore requires a replacement camera with higher pixels. However, this option was rejected due to cost considerations. When tested, the solid colour block also performed well, as it was recognized faster than the QR code, but was more susceptible to the light factor. This can only be achieved by more field measurements to record the colour thresholds under different conditions. At the same time, since the recognition is purely colour-based, the framework of the code becomes very simple and stable, and only the colour thresholds have to be changed for each test. By observing the environment around the task site, it was found that the colours were almost exclusively green, white and grey. Red is highly distinguishable from these colours, so rectangular cardboard with red colour was chosen as the marker to help complete the task.

## Lab Colour Space

After choosing a marker, it is necessary to consider how to find the desired colour from the many environmental disturbances. lab colour space is chosen to distinguish colours, which is a colour opposition space with three dimensions: L, a, and b. Dimension L represents the brightness of a colour, and a and b represent the contrast of a colour [23, 24]. The dimension L represents the brightness of a colour, while a and b represent the contrast of colour, using these three dimensions to accurately describe the colours seen by the human

eye [H3]. lab colour space describes colours more intuitively than other types

of colour spaces. It has a stronger perceived uniformity, enhances the description of colours by a high match between luminance L and human luminance perception, and can accomplish the work of colour balance by adjusting only the a and b channels [23].



**Figure 3.53** Lab colour space [23]

By measuring colour, it can be found that the colour is black when L is zero and white when L is 100 [23, 24]. The colour corresponding to dimension changes from red to green when it goes from positive to negative, and the colour corresponding to dimension b changes from yellow to blue when it goes from positive to negative [23]. The minimum and maximum values of each dimension corresponding to each colour are the colour threshold of this colour. Of course, the colour thresholds can vary depending on the lighting conditions, so the colour thresholds measured indoors cannot be applied outdoors or cannot accurately describe the colours outdoors. For example, the colour threshold for red cardboard measured indoors is (54, 77, 38, 85, -21, 62). However, if the same cardboard is measured outdoors in the same weather and at the same distance, the colour thresholds will change. The range of dimension L will increase and the dimensions a and b will be shifted to the left accordingly. This is also in line with human visual intuition that the colour seen under bright light will not only be distinguished from the colour seen normally in terms of brightness but also the colour will change. Therefore, OpenMV made several measurements of the red cardboard under different conditions and finally determined the average colour thresholds in normal weather, when it rains, and when it is cloudy, as well as the colour thresholds for different time periods under the same weather conditions. Figure 1 is a graph of the colour thresholds for all conditions.

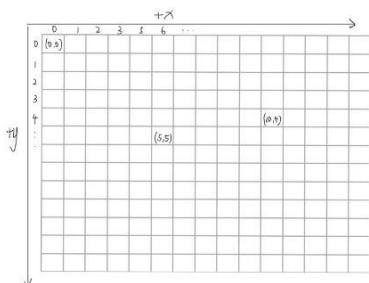
Color Threshold	Sunny	Cloudy
<b>8: 00-9: 00</b>	(54, 77, 38, 85, -21, 62)	(43, 65, 35, 79, -22, 64)
<b>12: 00-13: 00</b>	(58, 90, 44, 89, -26, 78)	(44, 88, 42, 84, -28, 77)
<b>17: 00-18: 00</b>	(34, 62, 32, 83, -18, 81)	(18, 54, 39, 79, -17, 69)

**Figure 3.54** Colour Threshold of red cardboard

## Calculating the centre of the marker

After the marker can be accurately identified, it is necessary to ensure that the direction of travel of the rover is always toward the marker. The coordinates of the marker's centre point are calculated from the coordinates of the marker's pixel position in the image. If this coordinate coincides exactly with the coordinate of the image centroid, the rover is considered to have not deviated from the expected route.

This scheme needs to get the specific coordinates of the pixels from OpenMV, which can be used to get a frame of the image by `sensor.snapshot()` function and then get the coordinates of the pixels occupied by the marker according to the pixel coordinate system of OpenMV [25]. However, not all of these pixel coordinates are useful, and if all of them are kept, they not only take up storage space but also increase the computational effort, making the operation slow [25]. Theoretically, to calculate the centre of the marker, we only need to know the position coordinates of the outer contour. And because the marker used is a regular rectangle, so through the camera no matter which direction to observe, the resulting graph is still a regular quadrilateral. Using this can greatly simplify the calculation and quickly find the centre point of the marker. The general method is to find the midpoint of each contour line and then connect the corresponding midpoints, and the intersection of the two centre lines is the centre point of the figure.

**Figure 3.55** the pixel coordinate system of OpenMV

However, when using the code to implement it, we found that OpenMV cannot get the centre point directly by connecting the lines, so we can calculate half of the sum of the horizontal coordinates and half of the sum of the vertical coordinates of the midpoints of the opposite sides respectively, and the result is the coordinates of the centre point. Meanwhile, when testing the trolley, it was found that the centroid of the marker could not always coincide with the image centroid. Therefore, after testing, the orientation of the rover was considered correct if the centroid of the marker was within a circle with a radius of ten-pixel points centered on the image centroid. When the car reaches the target position for a tennis throw, it needs to use the sonar to judge the distance and stop moving forward.

#### 3.2.5 Communication Protocol—Chang Zeyu, Hu Yang, Jian Yingchao, Wang Shuyu

In this project, OpenMV is used as the master board of the trolley to directly control the vision and drive modules of the trolley and indirectly control the gate and communication. Mbed board is used to receive commands from the master board and directly control the operation of the sonar, gate, etc. The master device needs to receive the distance information generated by the sonar from the slave devices, and send specific data to control the operation of the gate and the transmission of the communication devices. To achieve bi-directional communication between OpenMV and MbedL432KC, it is necessary to choose a communication protocol that they both support. At the same time, since both the master and the controlled board need to send and receive information in real time, half-duplex communication and simplex communication are not in line with the requirements [26]. So, both SPI and UART protocols, which support full-duplex communication, can be chosen as the main protocols for inter-board communication in the project.

UART is a universal asynchronous transceiver, which is capable of converting data between serial and parallel communication, so it is often used as an interface to other devices [26, 27]. It requires at least three wires, RX, TX, and GND, to communicate; RX is responsible for sending data, TX is responsible for receiving data, and GND is responsible for ground balancing potential [27]. The sending and receiving of data are mainly related to the high and low potentials on the TX and RX lines. When the microcontroller needs to send data, the TX line will send a 0 as the start bit to tell the receiver that there is data to be sent [27]. The TX line sends a 0 as the start bit to tell the receiver that there is data to be sent, and sends and receives information according to the pre-defined number of bits, and ends with a high level at the end of a frame [27]. The advantage of UART is that it requires fewer data lines and no clock signal. It is only necessary to set the same baud rate and the number of bits of data on the receiver and the sender in advance to complete the operation of sending and receiving data. The disadvantage of UART is that it does not

### 3 Subsystem Design

support the transmission of multiple hosts and slaves, but there is always only one host and one slave in this project.

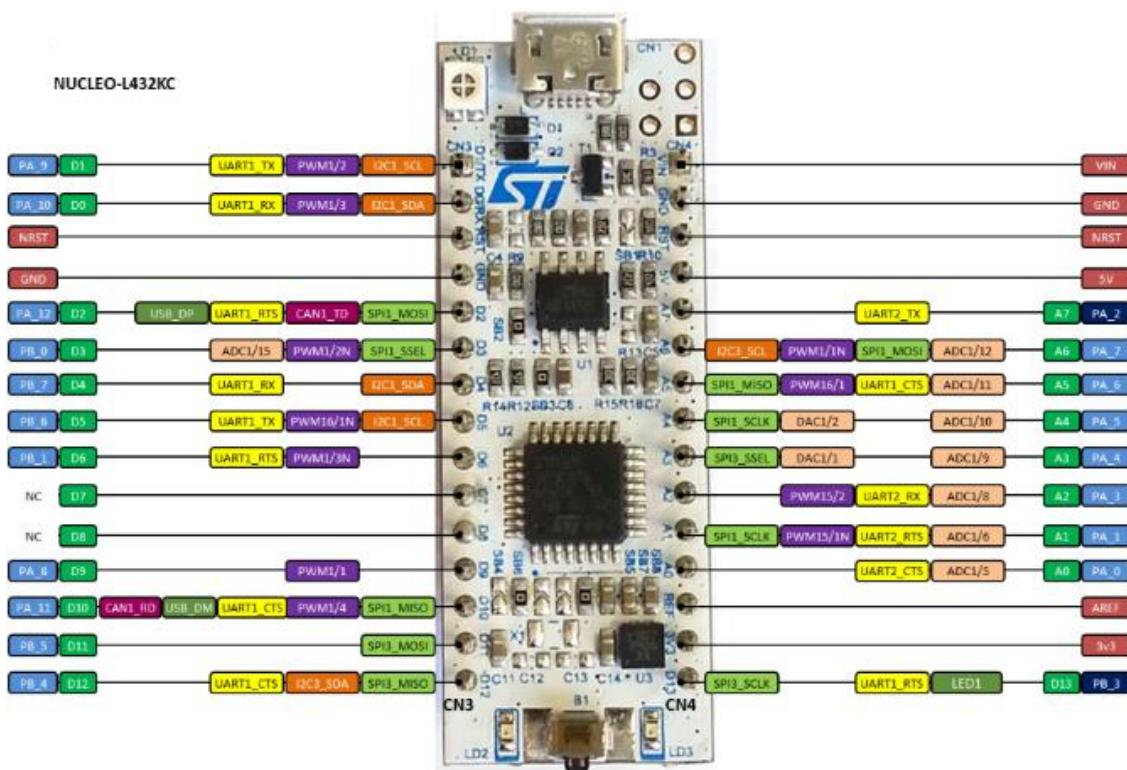
SPI is a serial peripheral interface, a communication bus for synchronous full duplex communication [26]. It requires four lines, MISO, MOSI, SCLK, and CS, for communication. MISO and MOSI are responsible for master and slave device input and output data. SCLK is the clock signal generated by the master device. CS is the enable signal for the master device to control the slave device. During data transfer, the master device controls to generate the time signal and the slave device finishes reading the data at the rising or falling edge of the time signal [26]. The main advantage of SPI is that the communication rate can reach several megabytes or even tens of megabytes per second. But for this project is not very useful, because the data to be transmitted does not need to use a large rate. SPI also supports communication between a master device and multiple slave devices, but there is no need to use multiple slave devices in this design.

After considering the above-mentioned UART and SPI, UART, which occupies less interface and has a more stable transmission and simpler code, is chosen as the communication protocol between OpenMV and Mbed.

## 3.3 Function Modules

### 3.3.1 Demo Board NUCLEO-L432KC

The demo board chose to finish the project is the NUCLEO-L432KC. NUCLEO-L432KC is a Nucleo board developed by STMicroelectronics [28]. It contains a CPU, a microcontroller STM32, flash, SRAM, Timers General Purpose and any other functions we need [28]. The communication, micro servo and sonar modules in our project need to be implemented based on NUCLEO-L432KC and the Mbed online editor.



**Figure 3.56** The pins of NUCLEO-L432KC

The figure shows the pins and the corresponding functions. In the project, we use some analogue pins and digital pins as the input and output of the modules. A PWM pin is needed to control the micro servo. In addition, UART pins are used for serial communication with OpenMV.

### 3.3.2 Sonar-based Distance Sensor HC-SR04 –Zhang Ziheng

In the whole project, it is necessary to get the distance between the rover and other objects in many situations. However, it is difficult to obtain accurate distance based on OpenMV visual recognition. After relevant research, we

### 3 Subsystem Design

decide to use the ultrasonic distance sensor HC-SR04. Compared with other distance sensors, HC-SR04 has a better performance in various aspects. The maximum measurement distance of HC-SR04 can reach 400cm with a 3mm accuracy [29]. It is tiny and weighs only 13g so it can be carried by the rover with no difficulty. Additionally, it is not easily disturbed by the environment so it has a stable return distance value.

The figure 3.57 illustrates the working principle of the HC-SR04. To be specific, this ultrasonic distance sensor will send an ultrasonic wave when detecting. In the next stage, the reflected wave by the detecting object will be received by the HC-SR04. Because the speed of the ultrasonic wave is fixed, we only need to record the time from its emission to its return, and then we can get the distance between the sensor and the object to be measured.



Figure 3.57 HC-SR04

Operating Voltage	5V DC
Operating Current	15mA
Operating Frequency	40KHz
Min Range	2cm
Max Range	400cm
Accuracy	3mm
Dimension	45 x 20 x 15mm

Table 3.8 The basic information of HC-SR04

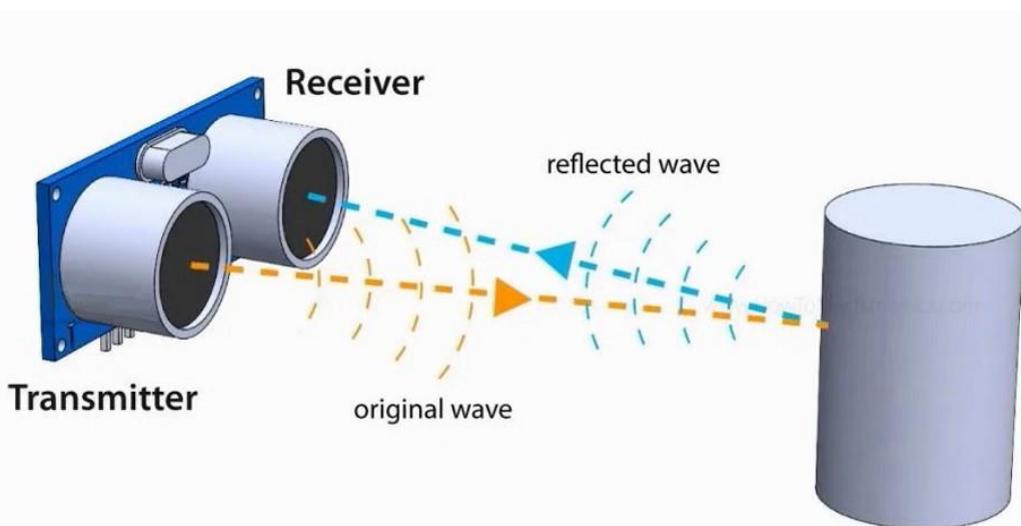
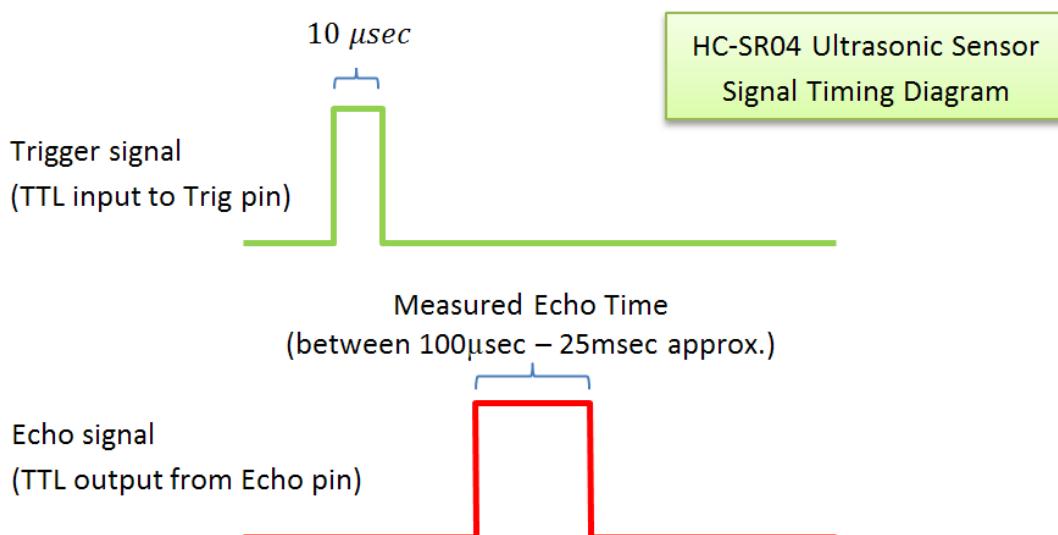


Figure 3.58 The working principle of HC-SR04

As shown in the figure 3.57, HC-SR04 has 4 pins which are  $V_{cc}$ , Trig, Echo and GND respectively.  $V_{cc}$  should be connected to a 5V DC input as the power supply and the GND should be connected to the ground. Trig and Echo are the input and output of the device respectively. The detailed operation method and the internal working principle of HC-SR04 are shown in the figure 3.58 [29]. To activate HC-SR04, we need to set the Trig pin on a High State for 10  $\mu$ s. After that, the internal of HC-SR04 will sends out an ultrasonic pulse (eight cycles at 40Khz) [29]. When the sensor detects the echo sent before, it will output an Echo signal at the Echo pin. From the measured echo time (the time of the signal in high state), we can calculate the distance as the following formula:

$$Distance = (Measured\ Echo\ Time \times 1000000)/58$$

In our design, the use of HC-SR04 will be based on the Mbed and NUCLEO-L432KC. After the calculation results are obtained, the data will be transmitted from the microcontroller to our main control equipment OpenMV. In the next stage, the data will be used in the route decision-making process of the rover.



**Figure 3.59** The signal timing diagram of HC-SR04

### 3.3.3 Wireless Communication—Gu Yunhao

In the last task, the robot should stop when it enters the planter area, and then a message containing the team name, team members' names and time of day must be sent to a laptop at 433MHz. To realize the communication between robot and laptop, a wireless transceiver is necessary. In this task, an HC-12 wireless transceiver is used to undertake this work. Additionally, getting the time of day is the other problem that needs to solve. At first, time information is

provided by the microcontroller used because it has an internal ticker. However, considering the limitation and difficulties we meet, using another module to get time is more suitable. Therefore, a DS1302 real time clock chip is added to the circuit to get the time of day and then return time data to HC-12 to send to the laptop. In the following content, the reason for choosing the component and the theory of the device will be introduced briefly.

#### HC-12 wireless transceiver

HC-12 wireless transceiver has a wide working frequency band from 433.4 MHz to 473MHz and the maximum sending power is 100mW [30]. At 5000bqs, HC-12 wireless transceiver can ensure -116dBm sensitivity and at most one thousand meters communication distance in open and wide places [30]. Not only the frequency but the sensitivity of this component is suitable for the requirement of this task. HC-12 can work in an environment whose humidity is from 10% to 90% [30]. The lake and water pool near the text ground cannot have a bad influence on its work.

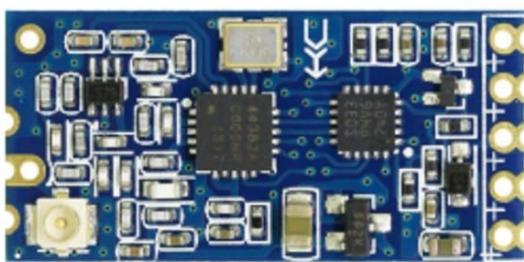


Figure 3.60 HC-12 wireless transceiver

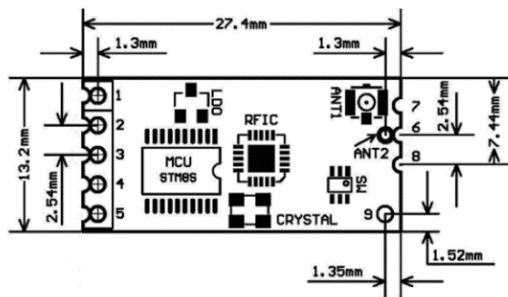
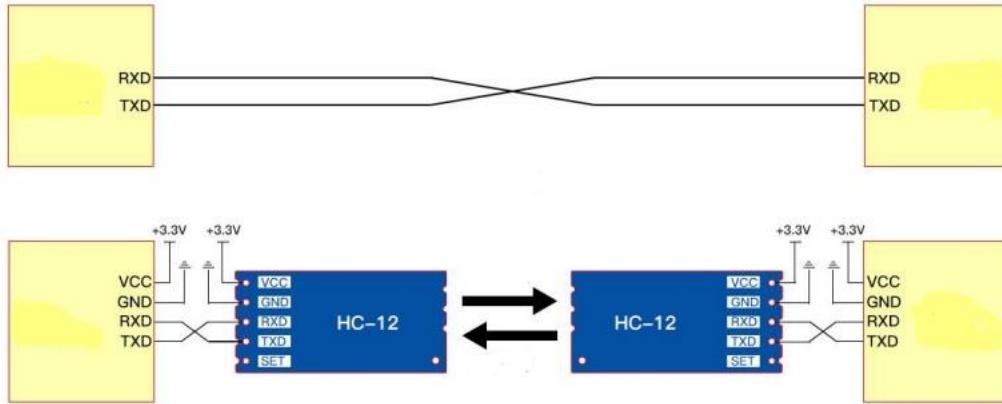


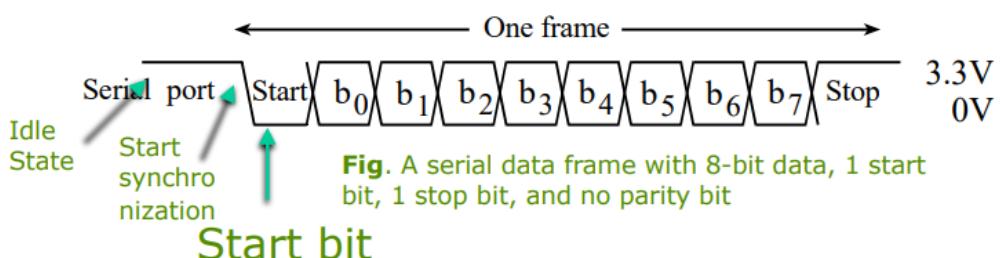
Figure 3.61 HC-12 schematic diagram

HC-12 wireless transceiver is used to replace a half-duplex physical communication line. HC-12 is also a type of half-duplex component, which means the HC-12 wireless transceiver cannot send and receive the message at the same time. Wireless transceivers cannot build wireless communication lines alone because laptops cannot recognize the information from it independently, two wireless transceivers are used in the whole machine. One wireless transceiver connected with the microcontroller will send the message from the microcontroller and the other wireless transceiver connected with the laptop as the receiver, will receive the information from the microcontroller. Since the pins of the wireless transceiver are not designed as a USB port so it cannot connect directly to the laptop. An additional USB-to-TTL converter is needed to ensure the laptop can figure out a wireless transceiver.



**Figure 3.62** half-duplex physical communication line and HC-12 wireless communication line

HC-12 wireless transceiver is supported by UART (Universal Synchronous Asynchronous Receiver Transmitter) protocol. UART protocol is widely applied in many fields of wireless communication because it is simple to operate, does not need clock signal and can add parity for error checking. A wireless transceiver which is supported by the UART protocol has two communication ports used to receive messages. In HC-12 wireless, pin TXD is the “TX” port and pin RXD is the “RX” port. In UART transmission, data is always sent at the length of 8 bits. Because UART is a kind of asynchronous communication, there is no clock signal to control the line. Start bits and stop bit are used during data transmission. The start bit locates in the first bit of the whole data, to control the line open and the stop bit usually indicates the end of the data to control the communication line close. Between the start bit and stop bit, there is the 8-bit length message that should be transmitted. In UART protocol, the parity bit is not necessary. If wants used to detect the error, both transmitter and receiver must know the appearance of parity before communication. Parity includes odd and even parity. According to the type of the parity to judge ports: “TX” and “RX”. The function of the “TX” port is to send messages and “RX” pe error or not.



**Figure 3.63** The general form of UART communication

HC-12 has four types of serial port transmission modes [30]. The factory module of HC-12 wireless is FU3, which is a normal mode and fit the most situation. FU1 and FU2 modes focus on low power dissipation and FU4 mode

is major used for long distance and a little data transmission. In this task, HC-12 works in FU3 mode at a 9600 baud rate and frequency around 433MHz.

#### DS1302 real time clock chip

In the beginning, the ticker integrated into the microcontroller is used to get information about the time of day. However, the major question is the power supply. The intrinsic function of the real time clock contained in the microcontroller is stored in the seconds from January first in 1970 then uses a ticker to calculate the increasing seconds as time goes by. Only by providing power for the microcontroller, the ticker can work otherwise the stored seconds must be changed because the number of seconds loses synchronization with the real time. Although some alternative plans such as setting a suitable delay in the program, considering the accuracy of seconds and initial seconds must be changed every time, this plan is abandoned. DS1302 replaces the previous work of the microcontrollers. It can get the time data independently and has a battery rabbet to provide power for its ticker, which matches our needs.

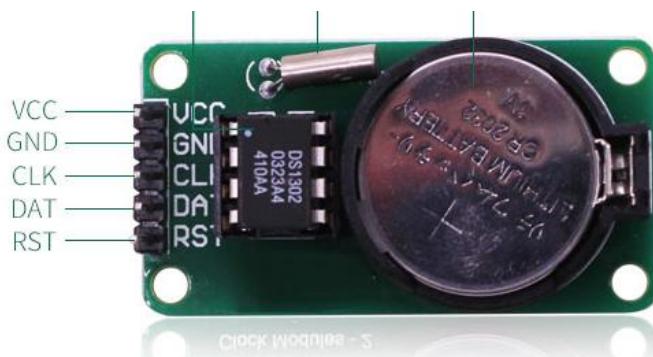


Figure 3.64 DS1302

DS1302 integrate a real time clock and 31 bits of static RAM [31]. It can provide real seconds, minutes, hours, days, dates, months and years data. The major function of DS1302 in this circuit is getting the seconds, minutes and hours data and then transmitting them to HC-12. DS1302 can build a simple serial communication through connected RES, I/O and SCLK lines with the microcontroller. The data can be transmitted at the speed of one byte or 31 bytes at a time. The working power dissipation of DS1302 is very low. The power to maintain its data and clock is less than 1mW.

The theory of DS1302 is similar to the microcontroller. There is a ticker is integrated into the DS1302 chip as well. An initial time data is provided for its register in advance and time data will increase automatically as time goes through the internal ticker. Through serial communication, the relevant data is sent to the microcontroller. A battery supplies power for its ticker, which allows DS1302 to refresh time data by itself without power supplied from the microcontroller. DS1302 simplifies the program, and the preparation work and does not take too much load for the whole circuit.

### 3.3.4 Tennis Ball Release Device—Zhuang Yian

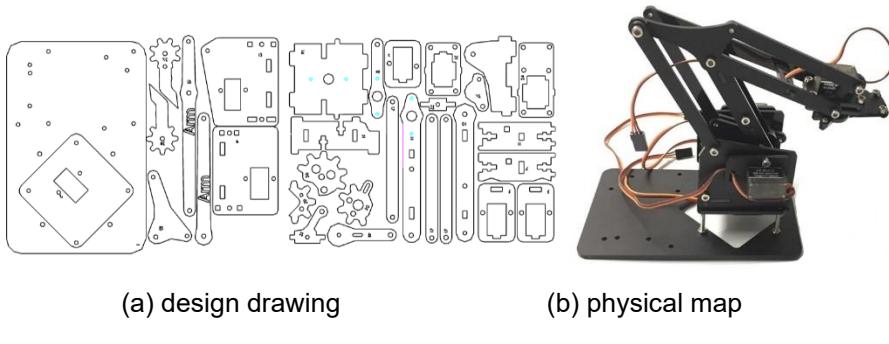
The fundamental design scheme is to make a device that meets the basic function of manually controlling grasping and placing objects in three-dimensional space. Therefore, a mechanical robot arm is applied in the project to release the tennis ball into the basket.

In terms of material selection, the team initially preferred their mechanical arm to be made of metal due to its durability and great mechanical strength. However, as vehicle assembly proceeded, the team found the rover could hardly bear the load and large size. Conversely, wooden material is too fragile, a compromise on this issue would be acrylic (a.k.a. PMMA or Polymethyl methacrylate), which is favoured for excellence in both practicality and price.

#### Structure

In general, it is a four-degree mechanical robot arm, the structure of which is divided into two parts: the arm and the claw.

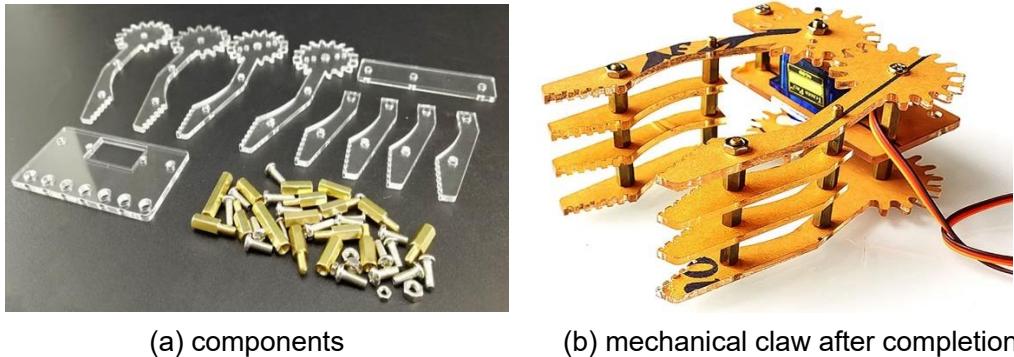
The arm consists of a base, left arm, right arm and joint. The left arm is in control of claw's motion in the vertical direction (up and down). The cap of the micro servo is connected to a rack post. When the micro servo on the left rotates counterclockwise, the claw is expected to rise, and when it rotates clockwise, the claw is expected to descend. The right arm is in control of the claw's motion in the horizontal direction (forward and backward). Similarly, the end of the cap is connected to a rack post. When the micro servo on the right rotates counterclockwise, the claw goes ahead, and when it rotates clockwise, the claw moves back. The whole structure can rotate on the horizontal plane around the base, although this might not be functional in the project. The design drawing and physical map of the arm part are provided in Figure 3.65.



**Figure 3.65** structure of arm part

The claw is composed of 2 left claws, 2 right claws, 4 half claws and two plate fixers. Since there is only one degree of freedom, the structure of the claw part is much less complicated compared to the arm part. The micro servo cap is fixed to the upper right claw. As it rotates counterclockwise, the claw swings open and as it rotates clockwise, the claw swings close. In this connection flare angle of the claw equals twice the

rotating angle. The photograph of all components and mechanical claw after completion of installation is provided in Figure 3.66.

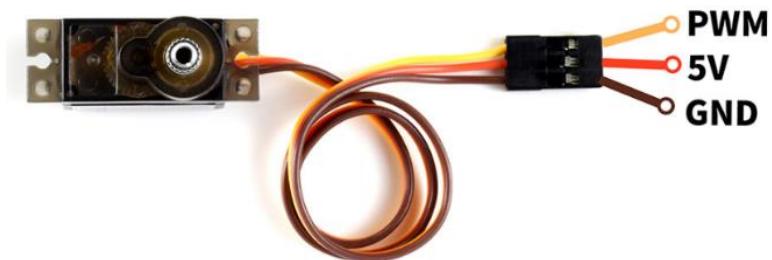


**Figure 3.66** structure of claw part

## How to Control

Based on hardware, the Mbed microprocessor L432KC, micro servo SG90, micro servo MG90s and power module are the main elements, and the motion control system is constructed by building the hardware platform and designing the software control program, where the hardware platform is stated in the previous block.

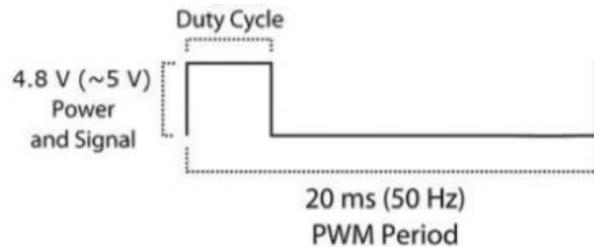
Generally, the motion of the whole system is dominated by the rotating angle of the micro servo motor. The micro servo motor is mainly composed of a shell, a circuit board, a drive motor, a reduction gear and a position detection element. When the microcontroller sends a signal to it, the integrated circuit on the circuit board drives the coreless motor to rotate and sends a signal back from the position detector to determine whether the target position has been reached.



**Figure 3.67** external appearance of micro servo motor

For the sake of control, unlike normal DC motors, servo motors usually have an additional pin aside from the two power pins (Vcc and GND) which is the signal pin. The signal pin is used to control the micro servo motor, turning its shaft to any desired angle. It has a geared output shaft which can be electrically controlled to turn one degree at a time. The control signal of the micro servo motor is a PWM signal with a period of 20ms, in which the pulse width is from 0.5ms-2.5ms, and the corresponding position of the micro servo motor is 0-180

degrees, which varies linearly. In the program, simply modify the duty cycle of the PWM signal output by the microcontroller to realize the position control of each joint.



**Figure 3.68** example of PWM control signal

Linear relation between input pulse width and output degree:

$$\frac{\text{output degree}}{180^\circ} = \frac{\text{pulse width} - 0.5\text{ms}}{2\text{ms}}$$

Pulse Width	Output Degree	Duty Ratio
0.5ms	0°	2.5%
1.0ms	45°	5%
1.5ms	90°	7.5%
2.0ms	135°	10%
2.5ms	180°	12.5%

**Table 3.9** Relation of I/O at a special value

## Optimization

The micro servo motor has a high current requirement, so when more than one servo motor is used with one single microcontroller, it is important to connect their power connections to an external power supply as the Mbed may not be able to source the current needed for the servo. Since there will be more than one servo needed in this project, it is necessary to apply an external power source to supply micro servo motors.

# 4

# System Integration

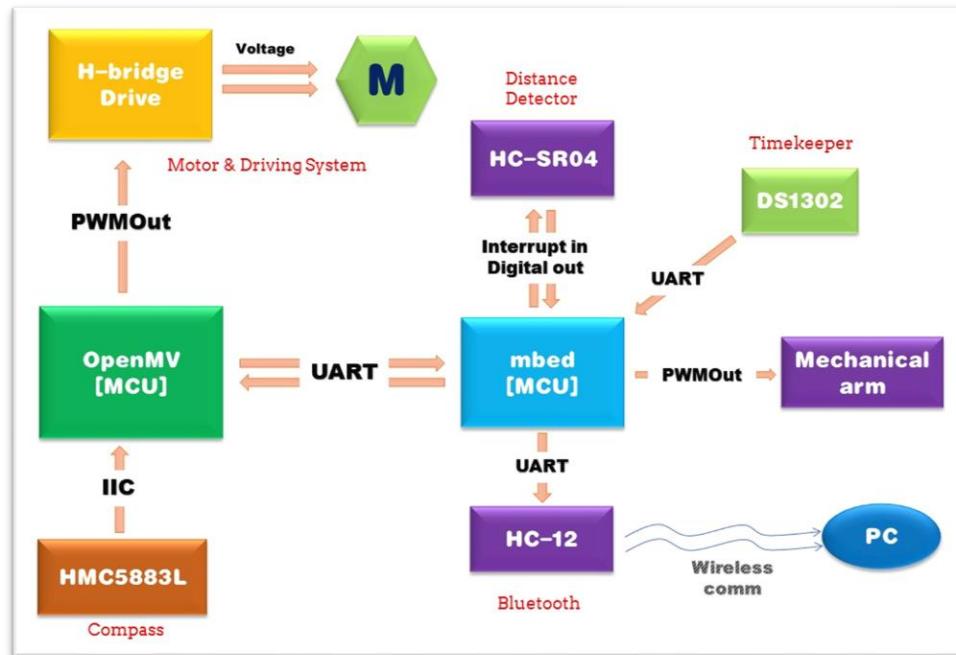
## Hardware

After most subsystems are in the completed state, the team starts to carry out the hardware integration of Rover. The total hardware of the rover includes OpenMV, NUCLEO-L432KC, drive circuit, distance sensor, electronic compass, mechanical arm, wireless communication module, motor, outer shell, and batteries. As the overall system design discussed, the drive circuit, Mbed board, and electronic compass will be connected to OpenMV using wires. The same relationship also happens between the Mbed board and distance sensor, mechanical arm, and wireless communication module. However, their installation positions are also different because of the characteristics of different hardware. In order to ensure the height of the robot arm, it is placed on the top of the rover. The distance sensors are installed on the front, left, and right of the outer shells of the rover respectively, whose wires pass through the holes on the shell specially designed. The front side of the rover is empty to provide a space for OpenMV. In addition, to minimize the interference, the antenna of the communication module is exposed outside the outer shell. All hardware is fixed to the rover with screws and brackets. The wires between different components are soldered onto pins and PCBs. The battery and power bank are connected to the OpenMV, NUCLEO-L432KC, drive circuit to provide different voltage. Additionally, fuses are used in these connections to avoid overloading. Finally, the 3D printing shell is inserted into the rover to ensure the aesthetics and stability of internal components.

## Software

In the two patios, the two main modules are OpenMV module and the NUCLEO-L432KC module. The programming language of OpenMV is micro-python, while the C language was used to program NUCLEO-L432KC. OpenMV is the core control component of the entire architecture in this project, and Mbed is the auxiliary control component. The communication protocol between the two modules is UART. The Mbed and the HC-SR04 distance detector are linked by a digital in and interrupt in a statement. The timekeeper

DS1302 and the Bluetooth HC-12 are linked to the Mbed using the UART interface. A wireless connection connects the Bluetooth HC-12 to a personal computer. PWMOut connects the mechanical arm to the Mbed. The IIC communication protocol is used to link the compass HMC5883L to OpenMV.



**Figure 4.1** System integration with all involved modules

The PWMOut is used by OpenMV to operate the H-bridge drive on the drive board. The H-bridge Drive regulates the motor's functioning by changing the voltage.

# 5

## Results & Discussion

In this project, the rover had to complete a series of tasks in Patio 1 and Patio 2. Each task requires the cooperation of the various components of the rover. The features of each part need to meet certain specifications in the process. The final performance of the rover will also be discussed in detail.

### The Design in Mechanics

Although plastic has the advantage of being cheap and lightweight, we chose the more expensive, heavier, and stronger aluminum alloy in order to keep the rover stable on rough roads and to provide a stable working environment for the rover's electronics.

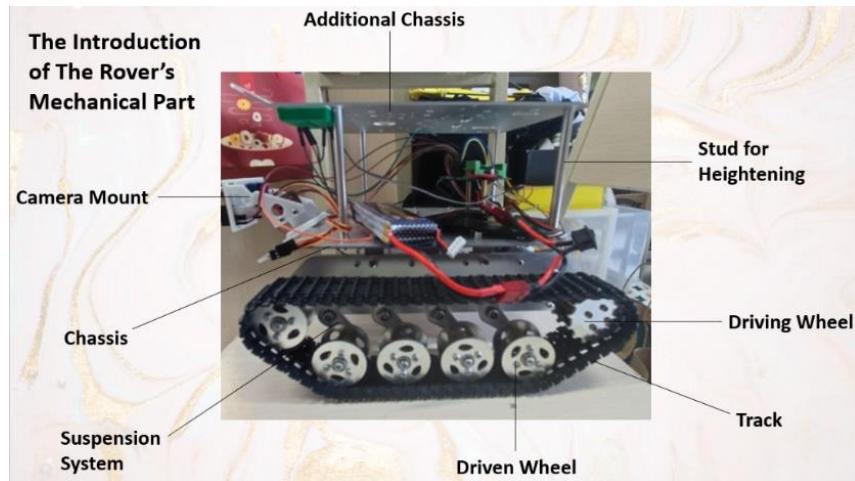
Wheeled construction allows for faster movement, lower costs, simpler mechanics, and a wide range of detailing options. However, the mechanical design of the rover needs to focus on stability when traveling, so we prefer the more reliable and stable tracked construction with more excellent off-road capability. Differential steering can be implemented without any unnecessary components, reducing the complexity of the rover and ensuring high steering efficiency while reducing the possibility of failure. However, the sturdy construction and strong off-road capabilities are still insufficient to keep the rover stable on the road. For this reason, we have added eight springs to the track of the rover to absorb the vibrations caused by unevenness in the road. In comparative tests, the stability gains of the eight springs are visible. Although the rover still wobbles considerably on rough terrain, this level of disturbance does not hinder the task. The addition of springs is certainly an efficient improvement.

We chose a plastic mount for the OpenMV that can be made with a 3D printer. This mount can be rotated 360 degrees horizontally and 90 degrees vertically. The rotation of the mount is controlled by two steering engines, which allows us to give the rover the flexibility to adjust the angle of the camera for different tasks and to obtain the ideal working environment. In practical tests, we found that the camera mounted on this stand had a very suitable position. However, if we needed to use the steering engines to adjust the camera angle, we would need four extra pins, which is unacceptable for our microcontroller,

## 5 Results & Discussion

which has a very limited number of pins and cannot be wasted on steering engines control. The good thing is that the angle of the mount can be adjusted manually so that the camera angle can be flexibly adjusted even without using steering engines.

The finalized result is shown in the photo below.



**Figure 5.1** The finalized design of the rover's mechanics

## Auto Modules

Considering that the rover has a heavier weight, we need two powerful motors to ensure that the rover is driven correctly. We chose the MG540, which operates at 12 V and can operate at a steady 30 W.

Working voltage	Motor type	Block running current	Rated current	Rated torque
12V	Brushed	>8A	1.44mA	2.6kgf.cm

**Table 5.1** parameters of brushed motor MG540 [3]

This is essential for the rover to drive correctly. If the power of the motors is too low, the rover may not be able to drive on slopes or over cobbled surfaces. The control of the two motors is done via five pins each, two of which are high and ground, and the other three are PWM inputs and two-variable high and low levels, respectively. This makes it very easy to control the speed and direction of rotation of the motors and reduces the workload of the code to a large extent. This control process is realized via the H-bridge.

## 5 Results & Discussion

A	B	C	D	STATE
ON	OFF	OFF	ON	CW
OFF	ON	ON	OFF	CCW
ON	ON	OFF	OFF	BRAKE
OFF	OFF	ON	ON	BRAKE
OFF	OFF	OFF	OFF	COAST

**Table 5.2** Control logic of H-bridge[5]

A complete H-bridge is composed of four NMOSs, and in order to avoid burning out of the MOSFETs during the drive of high-power motors, the most suitable NMOS, NTMFFS4833N, was chosen. this MOSFET can operate stably at voltages of up to 30 V and can perform at a temperature rise of about

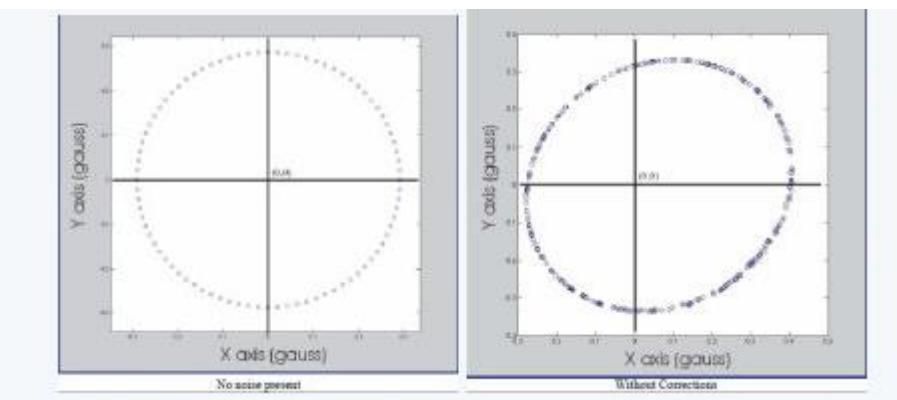
$$T = T_E + I^2 \times R_{DS(ON)} \times R\theta JA = 25^\circ C + (8A)^2 \times 137.8 = 42.6^\circ C$$

at maximum load, well below the upper limit of 150 degrees Celsius.

$V_{(BR)DSS}$	$V_{GS}$	$I_d$	$R_{DS(ON)}$	$R\theta JA$	$V_{TH}$
30V	$\pm 20V$	191A	2m $\Omega$ @10V	137.8 $^\circ C/W$	1.5V-2.5V

**Table 5.3** Parameters of NMOS NTMFFS4833N [7]

In order to correct the speed and direction of travel of the motors on both sides at any time during the rover's journey, we chose to use a PID algorithm and an electronic compass to do the job. The reason for abandoning the Hall sensors that come with the motors is that they require eight extra pins to control, but our microcontroller did not have enough pins. In the end, we found that the electronic compass achieved an excellent calibration and did not require many pins.



**Figure 5.2** Calibration of electronic compass

## 5 Results & Discussion

However, there are some problems with the practical use of the compass. The first is that the initial bearing value of the electronic compass is confirmed by the electromagnetic field of the site where it is used, which results in a different bearing value for actual use than the one obtained during the development process. We need to confirm the offset values in each direction to get the correct angle. The second thing is that the compass we chose is not very responsive, and when the compass orientation is changed, it takes about 0.2 to 0.4 seconds to return to a new bearing. As this delay is generated by the electronic compass itself, we cannot optimize this directly. Therefore, when the rover needs to turn, we slow down the motor to compensate for the delay caused by the insensitivity of the compass.

## Vision Modules

Thanks to the power of OpenMV, we could implement color tracking, marker tracking, QR code detection, and shape matching, all of which were needed in Patio 1 and Patio 2. However, we found that external conditions had a significant impact on OpenMV's image recognition capabilities. When the sunlight is too strong, or there is a large angle of inclination, the colors of the roads identified by OpenMV can change significantly. When it is rainy, the light reflected by the water on the road can also affect the threshold adaptation process. To improve OpenMV's recognition of the road surface, we split the camera capture into three regions by taking a threshold reading and weighting the sum of these three regions. The calculated thresholds can be used for the new threshold segmentation. After repeating the optimization ten times, we found that better performance was obtained for humidity conditions, being able to handle humidity from 60% to 90%. The same approach can also optimize recognition in bright light conditions. After optimization, we found that the performance gains achieved by this optimization were significant, even more so than for the humidity condition.

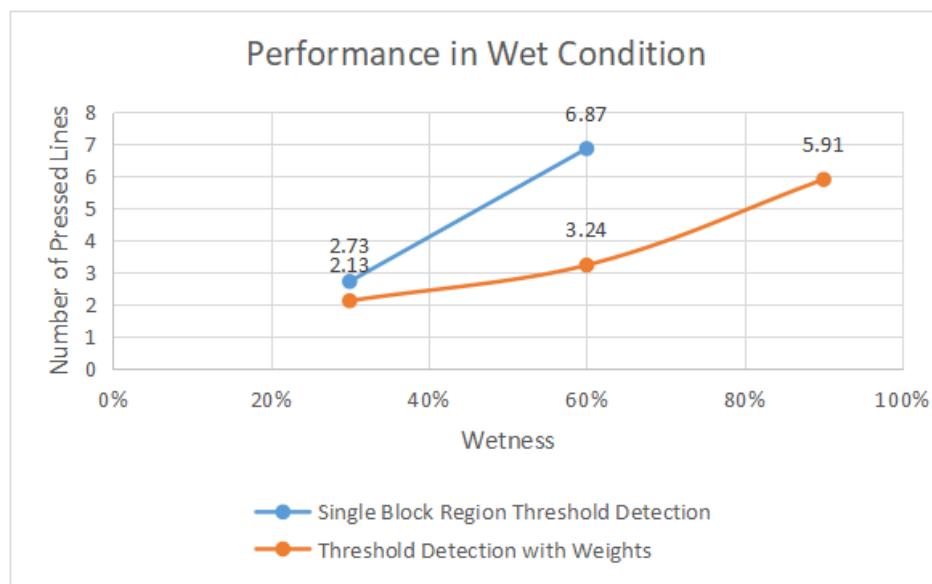
Wetness (ratio of watered area to total area)	Single Block Region Threshold Detection	Threshold Detection with Weights
Five Hours After Rainfall (30%)	2.73	2.13
Two Hours After Rainfall (60%)	6.87	3.24
Just Raining / Raining (90%)	N/A	5.91

Table 5.4 Performance of the Algorithm in Different Wetness Conditions

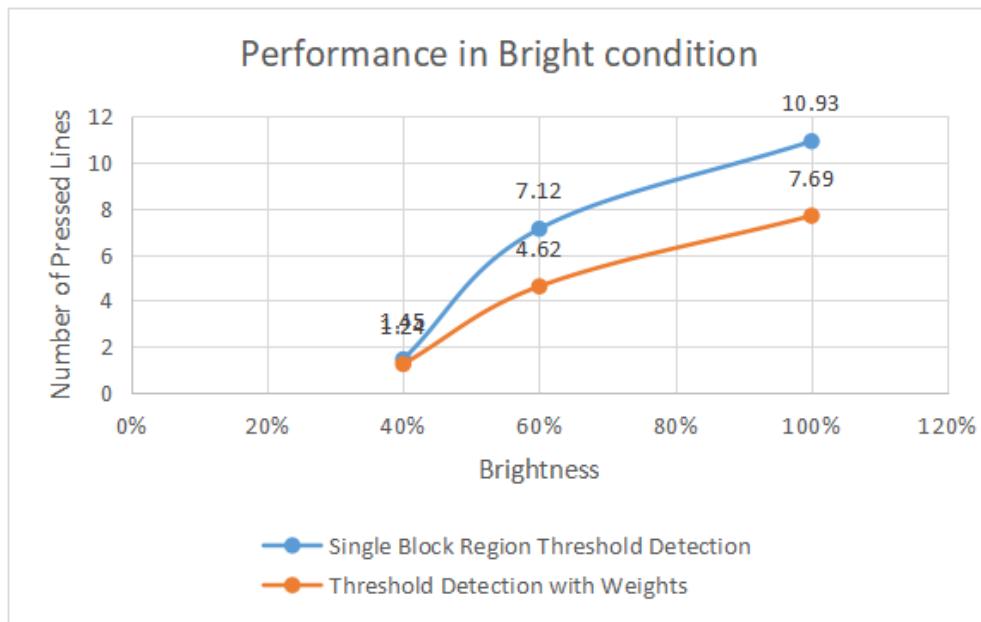
## 5 Results & Discussion

Brightness (Depending on the harshness of the sun)	Single Block Region Threshold Detection	Threshold Detection with Weights
Morning (40%)	1.45	1.24
Afternoon (60%)	7.12	4.62
Noon (100%)	10.93	7.69

**Table 5.5** Performance of the Algorithm in Different Brightness Conditions



**Figure 5.3** Performance in Different Wet Conditions



**Figure 5.4** Performance in Different Light Conditions

In the implementation of the color recognition function, we found that OpenMV's recognition of colors is easily influenced by the intensity of light. Tests carried out in different seasons, periods and weather conditions can vary greatly, resulting in very different color thresholds for markers. Therefore, we identified markers under different lighting conditions and recorded the thresholds for each of these conditions to ensure that the markers were identified regardless of the conditions under which the test was conducted.

The marker must be a color block or a QR code with strong color contrast. Our tests conclude that QR codes are more resistant to environmental interference. Still, our camera has low pixels and cannot correctly identify QR codes at longer distances. We, therefore, had to choose pure blocks of color with a high degree of differentiation from the environment as markers, such as red.

Color Threshold	Sunny	Cloudy
<b>8: 00-9: 00</b>	(54, 77, 38, 85, -21, 62)	(43, 65, 35, 79, -22, 64)
<b>12: 00-13: 00</b>	(58, 90, 44, 89, -26, 78)	(44, 88, 42, 84, -28, 77)
<b>17: 00-18: 00</b>	(34, 62, 32, 83, -18, 81)	(18, 54, 39, 79, -17, 69)

**Table 5.6** Color Threshold of red cardboard

However, in practice, we have found that even plain blocks of color cannot be effectively identified by OpenMV due to the distance. We intended to use OpenMV to identify the marker and guide the rover to the bucket where the tennis ball was to be dropped, but if the marker could not be identified from a distance, this technique was not suitable for our rover. So, we chose to use the distance sensor to guide the rover to a stop next to the bucket rather than using visual recognition.

## Distance Sensing

As a complement to the OpenMV visual recognition, we decided to add three HC-SR04 distance sensors to the rover. These three sensors allow us to know how far the rover is from the front, left, and right. The distance measured can be used as a basis for determining whether the rover is about to hit a surrounding obstacle. In Patio 2, we used the distance sensor to determine whether the rover had reached the target bucket. However, the HC-SR04 did not perform very well in our rover test. The measured distances were inaccurate by a few centimeters. When the distance exceeded 1 meter, the error became

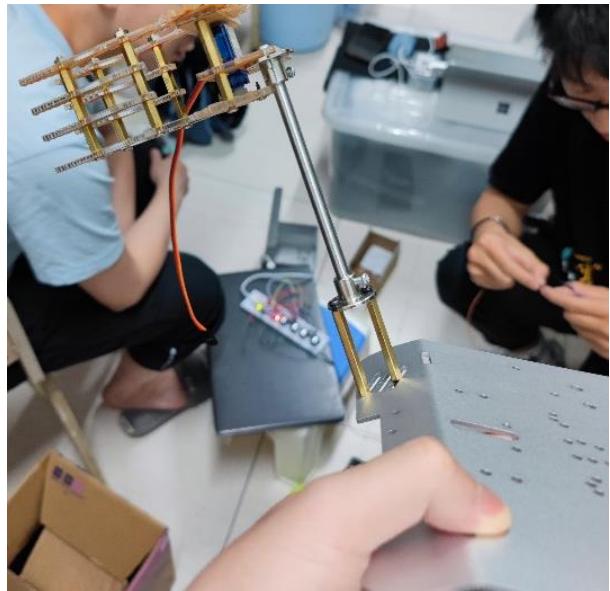
even more significant. This error is not ideal for the range of distances we need to measure and could lead to collisions due to inaccurate judgment of distances. After repeated testing and analysis, we speculate that this is because the ultrasonic wave emitted by the sensor will be reflected by objects in non-target direction. Therefore, the sensor will get the wrong distance. The greater the distance to be measured, the greater the error. As a result, we set 10cm as the standard for determining the judgment condition, which is a relatively small and acceptable error and will not affect the completion of the rover's task.

## Wireless Communication

We chose the HC-12 wireless transceiver for the task of wireless communication. The HC-12 has been tested to work correctly in places like the lake where the humidity is very high, and its 100mW transmitting power allows us to communicate with it properly in open areas. One task required sending the time during the test; we had planned to use the ticker that came with the microcontroller itself. However, this ticker needs to be powered continuously for proper operation, even when we are not in the test rover. This is very cumbersome and impractical. For this reason, we have chosen the DS1302 as a new clock module, which can obtain real-time time independently without having to keep the power supply on. So, although this choice adds an extra piece of electronics and requires more wiring and code work, it is the better choice for our needs, and the corresponding sacrifice is worth it.

## Ball Releasing

In order to place the tennis ball in the bucket, the basic idea is to use a mechanical claw to grab and release the tennis ball. To ensure the stability of this process, we initially planned to use a mechanical claw made of metal. However, this system made of metal was too heavy for the rover to maintain a reasonable distribution of the center of gravity. We had to make some compromises and switch to acrylic, which we found in our tests to be more stable in gripping the tennis ball but could be dislodged in the event of violent shaking. This is where the mechanical design of the rover comes in handy, with a track structure and aluminum chassis to minimize bumps. The mechanical jaws have a more stable working condition.



**Figure 5.5** The installation of the mechanical claw

In the previous design of the rover's mechanics, we added a new chassis with 100mm long studs in the hope that this improvement would give the jaws enough height to reach the edge of the bucket. However, our tests found that the raised claw was still not high enough. We, therefore, used the previously discarded camera mount and four studs to raise it again. This time the height was increased by 105 mm, and after testing the claw was high enough for the task. When the rover is driven over bumps, it remains stable and the tennis ball does not fall off accidentally.

# 6

## Conclusion

From this project, a rover has been designed to meet the requirement for two different patios. To meet the objectives, we divided the project into various sub-tasks and assigned responsibility to personnel. There are also a few direct-responsibility personnel regulating various modules and connecting with various sections. To keep the rover stable on bumpy roads and offer a steady working environment for the rover's electronics, the aluminum alloy was employed. A more durable and stable tracked structure with more robust off-road performance was employed since the rover must focus on stability when traveling. The camera capture was separated into three areas by obtaining a threshold reading and weighting the total of these three regions to enhance OpenMV's detection of the road surface. PID algorithm and an electronic compass were chosen to adjust the speed and direction of movement of the motors on both sides throughout the rover's running. In order to make the whole project go more smoothly during the actual inspection, the two paths should be tested several times after the robot is designed.

# References

- [1] Abdullah Al-Khalidi, TDPS\_ProjectHandbook\_2021-2022. [Lecture Slides]. (2021/22, Spring). UESTC3010 Team Design and Project Skills. Chengdu, China: College of the Glasgow. Accessed: March.5th, 2022. [Online]. Available: <https://moodle.gla.ac.uk/course/view.php>
- [2] G. C. R. Sincero, J. Cros and P. Viarouge, "Efficient simulation method for comparison of brush and brushless DC motors for light traction application," *2009 13th European Conference on Power Electronics and Applications*, 2009, pp. 1-10.
- [3] Y. Kambara and K. Ohnishi, "A design method of observers for bilateral control using dc brushed motor," *IECON 2015 - 41st Annual Conference of the IEEE Industrial Electronics Society*, 2015, pp. 000938-000943, doi: 10.1109/IECON.2015.7392220. two-wheel differential control
- [4] H. Yu, "Steering Control and Path Planning of Two-Wheel Vehicle with Hazard Avoidance," *2021 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2021, pp. 2037-2042, doi: 10.1109/SMC52423.2021.9658618.
- [5] A. Wagh, N. Sheth and J. Joshi, "Implementation of MOSFETs in a closed loop H-Bridge motor driver using charge pumps," *2015 Annual IEEE India Conference (INDICON)*, 2015, pp. 1-5, doi: 10.1109/INDICON.2015.7443724.
- [6] Lianping Hou, Chapter 5 Controllable Switches. [Lecture Slides] (2021/22, Spring). UESTC3022: Power Electronics (2021-22). Sichuan, Chengdu: UESTC, Glasgow College. Accessed: May. 6, 2022. [Online]. Available: <https://moodle.gla.ac.uk/course/view.php?id=27163>
- [7] ON Semiconductor, "NTMFS4833N: MOSFET, Single N-Channel 30V, 191A, 2mΩ." onsemi.com. (2006). Accessed: March. 6. 2022. [Online]. Available: <https://www.onsemi.com/products/discrete-power-modules/mosfets/ntmfs4833n>
- [8] Lianping Hou, Chapter 6 Heatsinks. [Lecture Slides] (2021/22, Spring). UESTC3022: Power Electronics (2021-22). Sichuan, Chengdu: UESTC, Glasgow College. Accessed: May. 6, 2022. [Online]. Available: <https://moodle.gla.ac.uk/course/view.php?id=27163>
- [9] International Rectifier, "IR2104S" infineon.com. (2001). Accessed: March. 6. 2022. [Online]. Available: <https://www.infineon.com/cms/en/product/power/gate-driver-ics/ir2104s/>
- [10] Duncan Bremner, Design for Manufacturing(DFM). [Lecture Slides]. (2021/22, Spring). UESTC3031 Engineering Project Management and Finance. Chengdu, China: College of the Glasgow. Accessed: May.5th, 2022. [Online]. Available: <https://moodle.gla.ac.uk/course/view.php?id=27169>
- [11] Abdullah Al-Khalidi, TDPS\_ProjectHandbook\_2021-2022. [Lecture Slides]. (2021/22, Spring). UESTC3010 Team Design and Project Skills. Chengdu, China: College of the Glasgow. Accessed: March.5th, 2022. [Online]. Available: <https://moodle.gla.ac.uk/course/view.php>
- [12] K. K. Tomchuk and V. Y. Volkov, "Analysis of the Quality of Thresholding for Object

## References

- Detection in Images," 2020 Wave Electronics and its Application in Information and Telecommunication Systems (WECONF), 2020, pp. 1-5, doi: 10.1109/WECONF48837.2020.9131491.
- [13] X. Xie, W. Huang, H. H. Wang and Z. Liu, "Image de-noising algorithm based on Gaussian mixture model and adaptive threshold modeling," *2017 International Conference on Inventive Computing and Informatics (ICICI)*, 2017, pp. 226-229, doi: 10.1109/ICICI.2017.8365343.
- [14] Y. Zhang and Z. Xia, "Research on the Image Segmentation Based on Improved Threshold Extractions," *2018 IEEE 3rd International Conference on Cloud Computing and Internet of Things (CCIOT)*, 2018, pp. 386-389, doi: 10.1109/CCIOT45285.2018.9032505.
- [15] S. Song and T. Gao, "Research on image segmentation algorithm based on threshold," *2021 13th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA)*, 2021, pp. 306-308, doi: 10.1109/ICMTMA52658.2021.00071.
- [16] M. Bober and J. Kittler, "A Hough transform based hierarchical algorithm for motion segmentation and estimation," *IEE Colloquium on Hough Transforms*, 1993, pp. 12/1-12/4.
- [17] P. Maya and C. Tharini, "Performance Analysis of Lane Detection Algorithm using Partial Hough Transform," *2020 21st International Arab Conference on Information Technology (ACIT)*, 2020, pp. 1-4, doi: 10.1109/ACIT50332.2020.9300083.
- [18] T. Sanida, A. Sideris and M. Dasygenis, "A Heterogeneous Implementation of the Sobel Edge Detection Filter Using OpenCL," *2020 9th International Conference on Modern Circuits and Systems Technologies (MOCAST)*, 2020, pp. 1-4, doi: 10.1109/MOCAST49295.2020.9200249.
- [19] <https://www.stocksy.com/pixelstories>, "Lcd screen pixel pattern supermacro by Pixel Stories for Stocksy United," Stocksy United, 2016. <https://www.stocksy.com/1251089/lcd-screen-pixel-pattern-supermacro> (accessed May 15, 2022).
- [20] E. R. Davies, Machine vision: theory, algorithms, practicalities. Amsterdam; Oxford: Morgan Kaufmann, 2005.
- [21] "[Hough Transform] Circle detection and space reduction – Lipman's Artificial Intelligence Directory." <http://laid.delanover.com/hough-transform-circle-detection-and-space-reduction/> (accessed May 15, 2022).
- [22] A. Rakshit and A. Chatterjee, "A Microcontroller-Based IR Range Finder System With Dynamic Range Enhancement," in *IEEE Sensors Journal*, vol. 10, no. 10, pp. 1635-1636, Oct. 2010, doi: 10.1109/JSEN.2010.2048204.
- [23] Bagdasar, Ovidiu & Birlutiu, Adriana & Chen, Minsi & Popa, Ioan-Lucian. (2017). Qualitative case study methodology: Automatic design and correction of ceramic colors. 699-703. 10.1109/ICSTCC.2017.8107118.
- [24] F. Dai, B. Fan and Y. Peng, "An image haze removal algorithm based on blockwise processing using LAB color space and bilateral filtering," *2018 Chinese Control And Decision Conference (CCDC)*, 2018, pp. 5945-5948, doi:

## References

- 10.1109/CCDC.2018.8408172.
- [25] Z. Wei-Peng, C. Li-Sha, L. Er-Min and Z. Feng-Chun, "Design and Production of Tracking System based on OpenMV Image Recognition," 2020 IEEE 5th Information Technology and Mechatronics Engineering Conference (ITOEC), 2020, pp. 1198-1202, doi: 10.1109/ITOEC49072.2020.9141560.
- [26] M. Poorani and R. Kurunjimalar, "Design implementation of UART and SPI in single FGPA," 2016 10th International Conference on Intelligent Systems and Control (ISCO), 2016, pp. 1-5, doi: 10.1109/ISCO.2016.7726983.
- [27] R. Szabó and A. Gontean, "The development process of an UART chip on FPGA for driving eMbedded devices," 2015 IEEE 21st International Symposium for Design and Technology in Electronic Packaging (SIITME), 2015, pp. 175-179, doi: 10.1109/SIITME.2015.7342320.
- [28] Mbed, "NUCLEO-L432KC" os.Mbed.com. (2020). Accessed: May. 11, 2022. [Online]. Available: <https://os.Mbed.com/platforms/ST-Nucleo-L432KC/>
- [29] Mbed, "HC-SR04" os.Mbed.com. (2019). Accessed: May. 11, 2022. [Online]. Available: <https://os.Mbed.com/components/HC-SR04/>
- [30] Guangzhou HC, "HC-12 wireless 433MHz serial module users' manual" [Online]. Available: <http://www.hc01.com/download/>
- [31] Guangzhou Zhou Ligong Single chip Micyoco "DS1302 trickle charge clock chip" [Online]. Available: <http://www.zlgmcu.com>

## Appendix A

**Table 7.1 Consumption record of the team**

Unit: RMB

Component	Price	Tax Rate	Quantity	Budget	Cost
OpenMV4plush7	355.45	1%	1	350	359.00
Vehicle Chasis	305.74	1%	1	300	309.30
3500mah 5c Battery	112.77	1%	1	100	113.90
PCB Board Printing	49.00	/	1	60	49.00
Double-sided Hole Board	1.98	1%	4	5	8.00
Micro Servo SG90	6.1	/	2	13	12.20
Mechanical Gripper	17.80	/	1	15	17.80
Wireless Transceiver	18.24	13%	1	15	20.61
RTC Chip	3.02	/	1	5	3.02
HC-SR04	5.53	/	3	15	16.59
Car Shell	Free	/	1	50	0
Color Cardboard(Beacon)	5	/	1	5	5.00
Cardboard Box(Beacon)	Free	/	1	5	0
Dupont Line	1.81	13%	1	2	2.05
<b>Total</b>	<b>/</b>	<b>/</b>	<b>/</b>	<b>940</b>	<b>916.47</b>

# Appendix B

## A.1 Road Patrolling

```

GRayscale THRESHOLD = [(0, 64), (35, 0, -23, 22, -33, 40)]

# Each roi is (x, y, w, h), and the line detection algorithm will try to find the center of mass of the largest blob in each roi.
# Then the x position of the center of mass is averaged with different weights, where the largest weight is assigned to the roi near the bottom of the image.
# The smaller weight is assigned to the next roi, and so on.
ROIS = [ # [ROI, weight]
        (0, 100, 160, 20, 0.6),
        (0, 050, 160, 20, 0.25),
        (0, 000, 160, 20, 0.15)
    ]
#roi represents the three sampling regions, (x,y,w,h,weight), representing the rectangle with width w and height h at the top left vertex (x,y), respectively
#weight is the weight of the current rectangle. Note that the QQVGA image size used in this routine is 160x120, roi that is, the image is divided into three
#The thresholds of the three rectangles should be adjusted according to the actual situation, and the weight of the rectangle closest to the robot's field of

weight sum = 0
for r in ROIS: weight_sum += r[4] # r[4] is the roi weight.

sensor.reset()
sensor.set_pixformat(sensor.GRAYSCALE)

sensor.set_framesize(sensor.QQVGA)

sensor.skip_frames(30)
sensor.set_auto_gain(False)
sensor.set_auto_whitebal(False)
clock = time.clock()

while (True):
    clock.tick()
    img = sensor.snapshot() |

    centroid sum = 0
    #利用颜色识别分别寻找三个矩形区域内的线段
    for r in ROIS:
        blobs = img.find_blobs(GRayscale THRESHOLD, roi=r[0:4], merge=True)
        # r[0:4] is roi tuple.
        #Find the line in the field of view, merge=true, merge the found image areas into one

        if blobs:
            largest_blob = 0
            for i in range(len(blobs)):
                #There may be more than one color block (line segment block) found in the target area, find the largest one and use it as the target line in this
                if blobs[i].pixels() > most pixels:
                    most_pixels = blobs[i].pixels()
                    #merged_blobs[i][4] is the total number of pixels in this color block, and if the total number of pixels in this color block is greater than
                    #most pixels, then this region is taken as the color block with the largest pixel count.
                    largest_blob = i

                img.draw_rectangle(blobs[largest_blob].rect())
                img.draw_cross(blobs[largest_blob].cx(),
                              blobs[largest_blob].cy())

                centroid_sum += blobs[largest_blob].cx() * r[4] # r[4] is the roi weight.
                # calculate centroid_sum, centroid_sum is equal to the x-coordinate value of the centroid of the largest color block of each region multiplied by
                center_pos = (centroid_sum / weight_sum) # Determine center of line.

                # Convert center_pos to a declination. We are using a non-linear operation, so the more we deviate from a straight line, the stronger the response.
                # Nonlinear operations are good for the output of such algorithms to cause a response "trigger".
                deflection_angle = 0

                # This will limit the angular output to about -45 to 45 degrees. (Not exactly -45 to 45 degrees).
                deflection_angle = -math.atan((center_pos-80)/60)

                deflection_angle = math.degrees(deflection_angle)

                if deflection angle==0:
                    run(40,40)
                elif deflection_angle > 0:
                    run(40+deflection_angle/10,40-deflection_angle/10)
                elif deflection angle < 0:
                    run(40-deflection_angle/10,40+deflection_angle/10)

def get_gray(num=100):
    #print("Get the average threshold within roi")
    ROI=(50,50,20,20)
    gray_mean=0
    for i in range(num):
        img = sensor.snapshot()
        img.draw_rectangle(ROI)
        img = sensor.snapshot()
        gray=img.get_statistics(roi=ROI)
        print(gray[0])
        gray_mean=gray_mean+gray.mean()
    #print("The {} sampling was performed and the average grayscale threshold obtained was: {}".format(num,gray_mean/num))
    return gray_mean/num

print(get_gray(100))

```

## Appendix B

### A.2 Shape Matching

```
enable_lens_corr = True # turn on/off for straighter lines

import sensor, image, time
sensor.reset()
sensor.set_pixformat(sensor.RGB565)
sensor.set_framesize(sensor.QVGA)
sensor.skip_frames(time = 2000)
clock = time.clock()
b=[(26, 60, -27, 21, -21, 36)]
min_degree = 0
max_degree = 179

while(True):
    clock.tick()
    img = sensor.snapshot()
    if enable_lens_corr: img.lens_corr(1.8)

    i=0
    s=0
    c=()
    r=()
    blobs = img.find_blobs(b,x_stride=2, y_stride=10, invert=True, area_threshold=10, merge=True, margin=4)
    for blob in blobs:
        print(blob.rect())
        img.draw_rectangle(blob.rect(), color=(255, 0, 0))
        if s < blob.pixels():
            s=blob.pixels()
            t=blob.rect()

    lst = list(t)
    lst[0]=lst[0]+20
    lst[1]=lst[1]+14
    lst[2]=lst[2]-40
    lst[3]=lst[3]-28
    tuple_1 = tuple(lst)
    print('ROI is: !')
    print(tuple_1)

    for c in img.find_circles(roi=tuple_1,threshold = 4500,
                               x_margin = 10, y_margin = 10,
                               r_margin = 10,r_min = 20,
                               r_max = 100, r_step = 2):
        img.draw_circle(c.x(), c.y(), c.r(), color = (255,0,0))
        print(c)
        print("circle")
    if c:
        break

    for r in img.find_rects(roi=tuple_1,threshold = 35000):
        img.draw_rectangle(r.rect(), color = (255, 0, 0))
        for p in r.corners(): img.draw_circle(p[0], p[1], 5, color = (0, 255, 0))
        print(r)
        print("square")
    if r:
        break

    for l in img.find_lines(roi=tuple_1,threshold = 1000, theta_margin = 25, rho_margin = 25):
        if (min_degree <= l.theta()) and (l.theta() <= max_degree):
            i+=1
            img.draw_line(l.line(), color = (255, 0, 0))
            print(l)
    if i==3:
        print("triangle")
        break

    print("FPS %f" % clock.fps())
```

### A.3 Pario 1 Function Group

```
1 #include "mbed.h"
2 #include "platform/mbed_thread.h"
3 #include <cstdio>
4
5 Serial pc(USBTX, USBRX);
6 Serial device(D1, D0); // tx, rx
7 InterruptIn echo(A5);
8 InterruptIn echo1(A1);
9 InterruptIn echo2(A4);
10 DigitalOut trigger2(D7);
11 DigitalOut trigger(D2);
12 DigitalOut trigger1(D3);
13
14 Timer timer;
15 Timer t;
16 Timer ti;
17
18 void up(){
19     t.start();
20 }
21 void down(){
22     t.stop();
23 }
24 void startTimer() {
25     timer.start();
26 }
27 void stopTimer() {
28     timer.stop();
29 }
30
31 void startT() {
32     ti.start();
33 }
34 void stopT() {
35     ti.stop();
36 }
37
38
39 int i;
40 int main() {
41     echo.rise(&startTimer);
42     echo.fall(&stopTimer);
43     echo1.rise(&up);
44     echo1.fall(&down);
45     echo2.rise(&startT);
46     echo2.fall(&stopT);
47
48     for(i=1;i<10000;i++) {
49         trigger = 1;
50         wait_us(10);
51         trigger = 0;
52         float distance = timer.read() * 1000000 / 58;
53         timer.reset(); // reset the timer to 0
54         pc.printf("Distance: %5.1f cm\r", distance);
55         float distance1=100.0;
56         float distance2=100.0;
```

## Appendix B

```
57     int rev;
58     rev=device.getc();
59     pc.printf("%d\r",rev);
60     if (rev=='2'){
61         trigger1 = 1;
62         wait_us(10);
63         trigger1 = 0;
64         distance1 = t.read() * 1000000/ 58;
65         t.reset(); // reset the timer to 0
66         pc.printf("Distance1: %5.1f cm\r", distance1);
67
68         trigger2 = 1;
69         wait_us(10);
70         trigger2 = 0;
71         distance2 = ti.read() * 1000000/ 58;
72         ti.reset(); // reset the timer to 0
73         pc.printf("Distance2: %5.1f cm\r", distance2);
74     }
75     distance=int(distance);
76     device.putc(distance);
77     if (distance1<=10 || distance2<=10){
78         device.putc('s');
79     }
80     wait(1);
81     }
82 }
83 }
```

## Appendix B

### A.4 Pario 2 Function Group

```
1 #include "mbed.h"
2 #include "platform/mbed_thread.h"
3 #include <cstdio>
4 SPI MySPI(D11,D12,D13);
5 DigitalOut cs(A3);
6
7 Serial pc(USBTX, USBRX);
8 Serial device(D1, D0); // tx, rx
9 PwmOut claw(D6);
10
11 InterruptIn echo(A5);
12 InterruptIn echo1(A1);
13 InterruptIn echo2(A4);
14 DigitalOut trigger2(D7);
15 DigitalOut trigger(D2);
16 DigitalOut trigger1(D3);
17
18 Timer timer;
19 Timer t;
20 Timer ti;
21
22 void up(){
23     t.start();
24 }
25 void down(){
26     t.stop();
27 }
28 void startTimer() {
```

## Appendix B

```
29     timer.start();
30 }
31 void stopTimer() {
32     timer.stop();
33 }
34 void startT() {
35     ti.start();
36 }
37 void stopT() {
38     ti.stop();
39 }
40
41
42 int i;
43 int main() {
44     echo.rise(&startTimer);
45     echo.fall(&stopTimer);
46     echol1.rise(&up);
47     echol1.fall(&down);
48     MySPI.format(8,0);
49     MySPI.frequency(1000000);
50     cs=0;
51     i=1;
52     char rev;
53
54     if(i==1){
55         claw.period(0.02f); // 20 micro second period
56         claw.write(0.108f); // 10.8% duty cycle
57
58         i++;
59     }
60     for(i=1;i<10000;i++) {
61         trigger = 1;
62         wait_us(10);
63         trigger = 0;
64         wait(1);
65         float distance = timer.read() * 1000000 / 58;
66         timer.reset(); // reset the timer to 0
67         pc.printf("Distance1: %5.1f cm      ", distance);
68
69         trigger1 = 1;
70         wait_us(10);
71         trigger1 = 0;
72         wait(1);
73         float distance1 = t.read() * 1000000/ 58;
74         t.reset(); // reset the timer to 0
75         pc.printf("Distance2: %5.1f cm\r", distance1);
76         device.putc(distance);
77         device.putc(distance1);
78         rev=device.getc();
79         pc.printf("%d\r", rev);
80         if(rev=='2'){
81             MySPI.write(5);
82             cs=1;
83         }
84     }
}
```