

COPYRIGHT INTERESTING CO



**MR.TUBE**  
REPORT VER1.0

[GROUP 3]

Interesting © Company TM 2018

INTERESTING

Interesting<sup>©</sup> Company <sup>TM</sup> 2018

# Contents

1.	INTRODUCTION .....	1
1.1	Project Introduction .....	1
1.2	Achievement .....	1
1.2.1	Low Cost.....	1
1.2.2	Game Characteristics .....	1
1.2.3	Mechanical Design and Manufacture .....	1
1.2.4	Software Design and Implement .....	1
1.2.5	Debugging Process .....	2
1.3	Game's Background .....	2
1.4	Summary .....	2
1.5	Acknowledgement .....	2
2	MECHANICAL DESIGN .....	3
2.1	Design Process.....	3
2.1.1	Summary of Requirements .....	3
2.1.2	Analysis of Requirements .....	3
2.1.3	Feedback (Project Q&A).....	3
2.1.4	Design Ideas .....	3
2.1.5	Early Design.....	4
2.1.5.1	Layer 1.....	4
2.1.5.2	Layer 2.....	5
2.1.5.3	Layer 3.....	6
2.1.5.4	Layer 4.....	7
2.1.5.5	Layer 5.....	8
2.1.5.6	Initial Mechanical Sketch .....	9
2.1.5.7	Appearance Part Description List.....	10
2.2	Final Design .....	11
2.2.1	Sub-systems .....	11
2.2.1.1	Game Sub-system .....	11
2.2.1.2	Recovery Sub-system .....	12
2.2.1.3	Authentication Card Reader Sub-system .....	12
2.2.1.4	Reward Sub-system.....	13
2.2.1.5	Power Supply Sub-system.....	14
2.2.1.6	Power Sub-system.....	14
2.2.2	Final Three-view drawing.....	15
2.2.3	Dimensional Drawing .....	16
2.2.4	Operation Procedure .....	20
2.2.4.1	Player Mode .....	20
2.2.4.2	Debugging Mode.....	20
3	DIGITAL SYSTEM DESIGN .....	21
3.1	Introduction .....	21
3.2	Function.....	21
3.3	Code Structure .....	21
3.4	ASM chart and State Machine chart .....	25
3.5	Block diagrams .....	26

# INTERESTING

2

3.6	Pin Planning.....	26
4	SOFTWARE DESIGN .....	27
4.1	MBED.....	27
4.1.1	Introduction .....	27
4.1.2	Functions of MBED.....	27
4.1.3	The programming language (C++) .....	27
4.1.4	Structure .....	28
4.1.4.2	Maintenance Mode.....	28
4.1.4.3	User Mode.....	29
4.1.5	Pseudocode.....	29
4.1.6	Important Functions .....	30
4.2	PC.....	30
4.2.1	PC Functions.....	30
4.2.2	GUI Design.....	30
4.2.2.1	WPF framework .....	30
4.2.2.2	Using C# .....	31
4.2.2.3	Using Xaml .....	33
4.2.2.4	Design Style .....	33
4.2.2.5	Window and pages.....	33
4.2.2.6	Design pattern --- MVVM.....	33
4.2.2.7	Features of the application:.....	34
4.2.2.8	Operation Procedure .....	34
4.2.3	Maintenance Mode.....	34
4.2.3.1	Maintenance Mode Functions .....	34
4.2.3.2	Operation Procedure .....	35
4.2.3.3	Command Menu and the Internal Operating Principle .....	36
4.2.4	Operation Mode .....	36
4.2.4.1	Operation Mode Functions .....	36
4.2.4.2	Interface .....	37
4.2.4.3	Operation Procedure .....	38
4.2.5	Important Functions .....	39
4.2.5.1	Language switching.....	39
4.2.5.2	Database .....	41
4.2.5.3	Audio Helper .....	45
4.2.6	Innovative Functions.....	46
4.2.6.1	Ranking list and the slide animation .....	46
4.2.6.2	Display the progress of the game: .....	52
4.3	Protocol .....	52
4.3.1	PC to MBED .....	52
4.3.2	MBED to FPGA .....	55
4.4	Strength and Weakness .....	56
4.4.1.	Strength.....	56
4.4.2.	Weakness .....	56
5	SENSOR SYSTEM .....	57
5.1	Manipulation .....	57
5.1.1	Color Sensor .....	57
5.1.2	Distance Sensor.....	59
5.1.3	Ultrasonic Distance Sensors.....	59

# INTERESTING

5.1.4	Infrared Sensor .....	60
5.1.5	Photoelectric Gate Sensor .....	60
5.1.6	Reed Pipe Sensor .....	60
5.2	Tests and Data .....	61
5.2.1	Color Sensor .....	61
5.2.2	Distance Sensor .....	62
5.2.3	Ultrasonic Distance Sensors .....	62
5.2.4	Infrared Sensor .....	62
5.2.5	Photoelectric Gate Sensor .....	62
5.2.6	Reed Pipe Sensor .....	62
5.3	Circuit Diagrams and Layout .....	63
6	MARKETING AND PUBLIC RELATION .....	64
6.1.	Overview .....	64
6.2.	Market assessment .....	64
6.2.1.	Demand analysis .....	64
6.2.2.	Advantages of physical games .....	64
6.2.3.	Market saturation analysis .....	65
6.2.4.	Product features .....	65
6.2.5.	SWOT .....	66
6.3.	Poster and Propaganda film .....	66
6.3.1.	Poster .....	66
6.3.2.	Propaganda film .....	68
6.4.	Website .....	70
6.5.	Exhibition Stand .....	71
6.6.	Team Member Biography .....	72
7.	PROJECT MANAGEMENT .....	73
7.1.	Project Plan .....	73
7.2.	Risk Management .....	75
7.3.	Teamwork .....	76
7.4.	Meetings .....	77
7.5.	Experience and Lessons .....	78
8.	APPENDIX .....	79
8.1.	Meeting Minutes .....	79
8.2.	Risk Management .....	90
8.3.	Chart Index .....	92
8.4.	About this Document .....	94

3

INTERESTING

Interesting<sup>©</sup> Company <sup>TM</sup> 2018

# 1. INTRODUCTION

## 1.1 Project Introduction

During this project, all the members (six) of Interesting Consulting Company will undoubtedly try their best to get the contract with Queue-Interact in a public competition for designing and delivering a robotic alien with an automated and entertaining system that will work in an interactive manner with those who are having to spend a long time in the queues for the park's attractions.

In order to get the final contract, our company decided to design a moving robot which contains a relatively simple game based on an attractive background stories and all the members attended the competitive tender on 29 November.

## 1.2 Achievement

### 1.2.1 Low Cost

From my perspective, the most significant property of a new product is cost-effective. As long as we are able to manufacture the best quality products at the lowest price, we can definitely win the bid of the competition. Then we were determined to produce a robot with most of the parts were totally hand-made and controlled the cost within the range of 200-400 yuan. As a result, we spent 245 yuan for this project, which was one of the lowest costs among the whole 16 groups in Xidian campus.

### 1.2.2 Game Characteristics

- **Fast**

Because people are waiting in lines, they don't have too much time to play the game and treat it just as a way to Relieve Boredom.

- **Easy**

If the rule of the game is too complicated to understand, the individuals will be not interested in this game due to the circumstance. They are extremely whiny and tired during the queuing process.

- **Stop at any time**

If their queuing time is finished, they can stop the game immediately.

### 1.2.3 Mechanical Design and Manufacture

First and the most important, we had perfectly balanced the proportion of intelligence and automation. After finishing the game, recovery sub-system and messenger wire will put the blocks in the block compartment at the preset place on the track each time, which is the symbol of automation. Then, the machine can communicate and interact with the user through phonetic system, and when we input certain instruction into the machine, it will act in different ways. This ability to process and analyze information represent intelligence.

Then, we designed the multi-layer system to maximize the utilization of space and divided the whole robot into five layers.

Power sub-system provided the potential to move purposefully, such as finding a person or avoiding an obstacle.

### 1.2.4 Software Design and Implement

The implementation of software is divided into three parts: GUI (Graphical user interface) on Windows platform, Relational database, MBED digital system. The GUI is based on WPF, providing interactive information to the user with high visual experience, plentiful functions and comparative stability. The database stores the information of the users using Access. User can select to see the rank, time and other information on the UI. The MBED digital system is the kernel of our product. It controls all the sensors, like the color sensor, distance sensor and ultrasonic sensor, analyses the data using special algorithms and send signals which includes the testing data and warning signals to PC.

# INTERESTING

## 1.2.5 Debugging Process

We must create a mode to check if all the parts of the system are operating normally, so that we established a debugging mode to detect the errors and deal with the problems in order to guarantee a good user experience and a perfect working system.

## 1.3 Game's Background

This is a story about repairing the tube, so we call the robot “Mr.Tube”. There are some damaged places in the tube, then we need to repair or replace the tube. We will provide a series of pictures of tube in stretchable latex and users must place the tube in the game area according to the picture. When you win the game, a bottle of water will be as a prize for the users.

## 1.4 Summary

This report will contain the detailed and rigorous descriptions of the robot to introduce the system that we created. The document will cover the mechanical design, software design (MBED, FPGA and C#), sensor system, marketing, web and project management.

Chapter 2 will minutely provide the five layers and six sub-systems of the robot, as well as the design procedure. Chapter 3 will describe the digital system in detail, with the help of a various of charts and block diagram, as well as some codes. The contents and operational logic of MBED (as well as PC) and five kinds of sensors will be included in Chapter 4 and Chapter 5 respectively. At last, the summary of website and marketing will be in Chapter 6 and the description of project management will be in Chapter 7.

## 1.5 Acknowledgement

Thanks for all the selfless help and effective guidance from the teachers and professor in the laboratory. Their own duties are as following:

Prof. Yanhui Chen Project Manager

Weidong Xu Project Management Lecturer and support staff

Huanfeng Liu C#, MBED, digital system, sensor Lecturer and support staff

## 2 MECHANICAL DESIGN

### 2.1 Design Process

#### 2.1.1 Summary of Requirements

When the allocation of the group had been published, we had our first group meeting immediately. During the meeting, our group member discussed the requirements together in PPT that the teacher gave us, then we considered what our products would be under the rule of the course. We discussed for about one hour and found five notable things corresponding to the mechanical design.

- We are required to make extensive use of a variety of sensors to develop a system that can operate with a level of autonomy.
- The final product should be a robotic alien that can physically interact with the user based on an entertainment function at the theme park during a long time in the queues.
- The user must remain at least 10cm from the machine.
- LED indicators should be used.
- A card will identify the user and give 16 different combinations.

#### 2.1.2 Analysis of Requirements

After we had seen the PPT, we started to think about the products based on the requirements and everyone would give their own opinions and ask some questions. The following points are the final results of our meeting.

In order to use a variety of sensors, we will consider color sensors, distance sensors, ultrasonic distance sensors, infrared sensor for human body, photoelectric gate sensor and reed pipe sensor. On the basis of the long time's queue, the game can't take too long and needs to be operated easily, the rules of the game should be relatively easy to understand. In addition, the game should be able to stop at any time in order to avoid affecting the queue due to the long detention time.

We will use LED indicators to decorate and remind users during the process of reading ID card and taking the prize.

We cut out 4 small gaps in a card to represent 16 results with binary numbers (0000 to 1111). With the requirements of designing a robotic alien and interacting physically with the user, we decide to manufacture a robot in pirate style with a track at the bottom in order to move forward and backward.

Layering the robots to achieve various functions is a better way.

Moreover, we also proposed some questions.

1. What is the limited size of the robot ?
2. What is the true meaning of operating a level of autonomy ?
3. How can we connect the motor to the shaft ?

#### 2.1.3 Feedback (Project Q&A)

Through the communication with teachers in the project Q&A session and the meeting, the problems had been solved.

1. The limited size of the robot is 30\*30\*50 (cm).
2. Machines respond differently as the environment and circumstances change. For example, a recovery system can be used to perform the autonomy, no matter how the order of balls alter, the recovery system will rearrange the order in the same way each time.
3. Refer the structures of the steel and the concrete, fixing with steel wire and hot melt adhesive.

#### 2.1.4 Design Ideas

The points which are able to represent our design ideas based on the previous analysis and feedback were determined as following:

- **Layering**

In order to make full use of space, we need to layer the robots with the plank. Meanwhile, different layers will contain different elements and have different purposes.

- **Saving cost**

# INTERESTING

All product's parts should be manufactured by hands, only in this way can we minimize the cost and optimize the cost performance ratio.

- **Physical interact**

The robot contains a function of moving and has a potential of finding users through the sensors.

- **Easily disassemble**

The back cover can be opened in order to check each layer and debug the whole systems.

## 2.1.5 Early Design

According to our design ideas, we decide to divide the system into five layers and six sub-systems, i.e. game sub-system, recovery sub-system, authentication card reader sub-system, reward sub-system, power supply sub-system and power sub-system.

### 2.1.5.1 Layer 1

It refers to the game sub-system. After the identity authentication, the messenger wire will be released by the motors which control the gear to release the game board. At the same time, the track driven by the motor under the block cabin will move the block cabin to the position of the pick-up port, then the user can take out the "pipe" in the block cabin and put it on the game board for game playing.

In the same way, when the game ends and the nine color sensors' results are completed, the motor that controls the game board will drive the gear to roll back the messenger wire so that the game board can be hatched. Meanwhile, the track driven by the motor will also move the block cabin to the position of the recovery track.

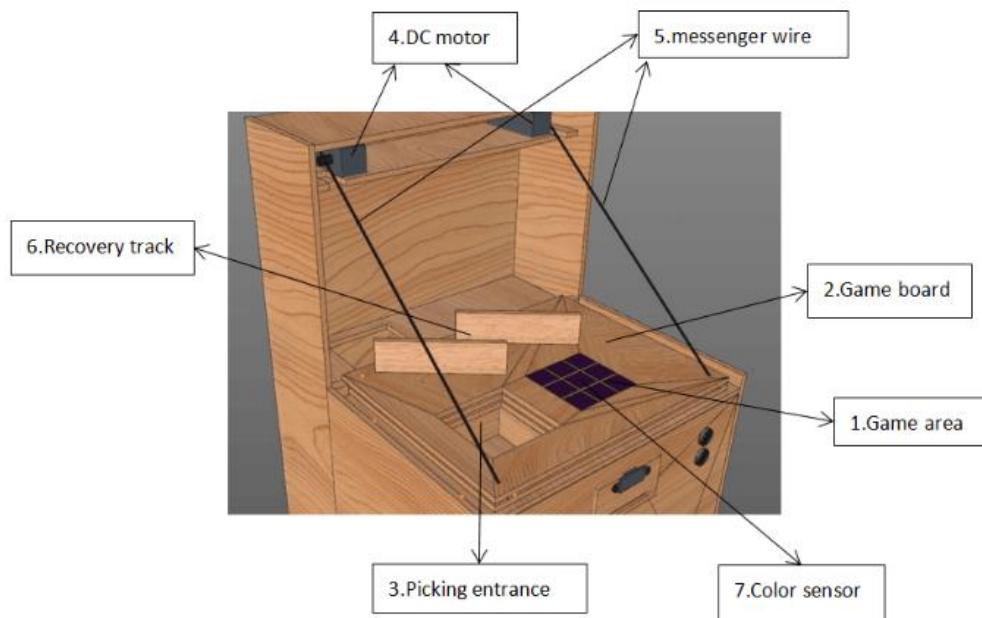


Figure 2-1 C4D model of game sub-system

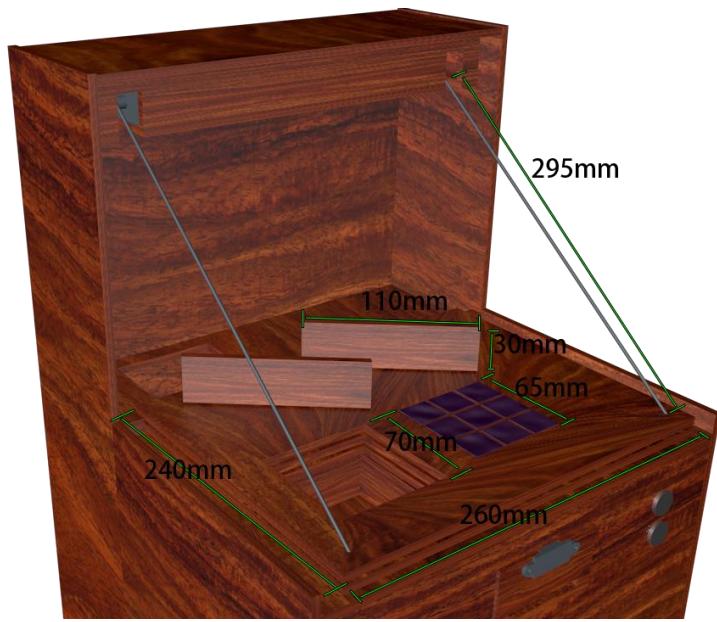


Figure 2-2 Dimension drawing of game sub-system

### 2.1.5.2 Layer 2

It refers to the recovery sub-system and the authentication card reader sub-system.

First, we discuss the recovery sub-system. After the cabin is closed and the block cabin is located. The wooden blocks placed on the game board will slide into the recovery track. Meanwhile, the track under the block cabin will move the block to the position of the recovery track, so that the block can enter the block through the recovery track.

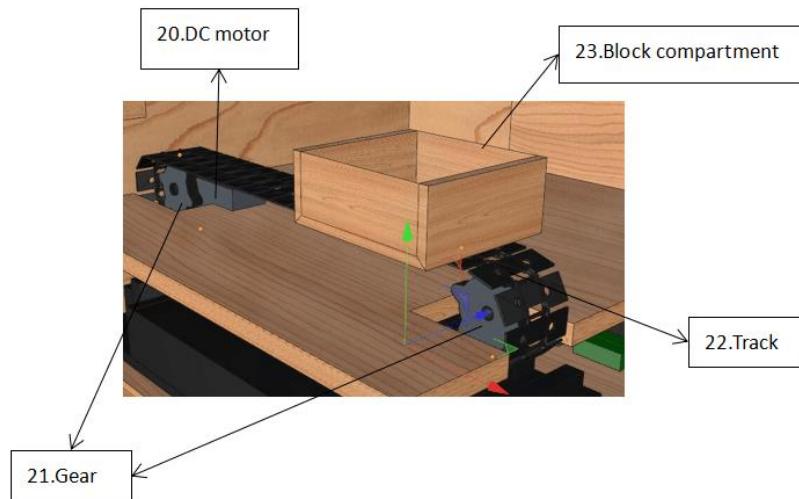


Figure 2-3 C4D model of recovery sub-system

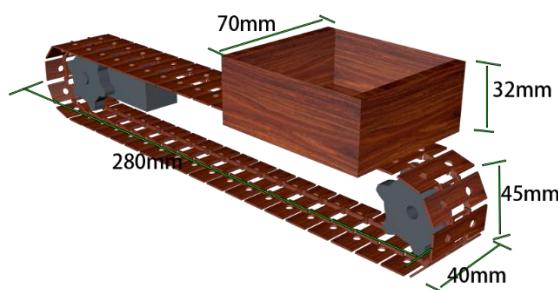


Figure 2-4 Dimension drawing of recovery sub-system

# INTERESTING

Then, we discuss the authentication card reader sub-system. When the user puts his ID card into the reading bayonet, the number of the card will be defined by detecting the gap in the ID card with the optoelectronic sensor. (there is a gap: 0, there is not a gap:1. 1111 is the number of debugging worker, 0001--1111 is the user number).

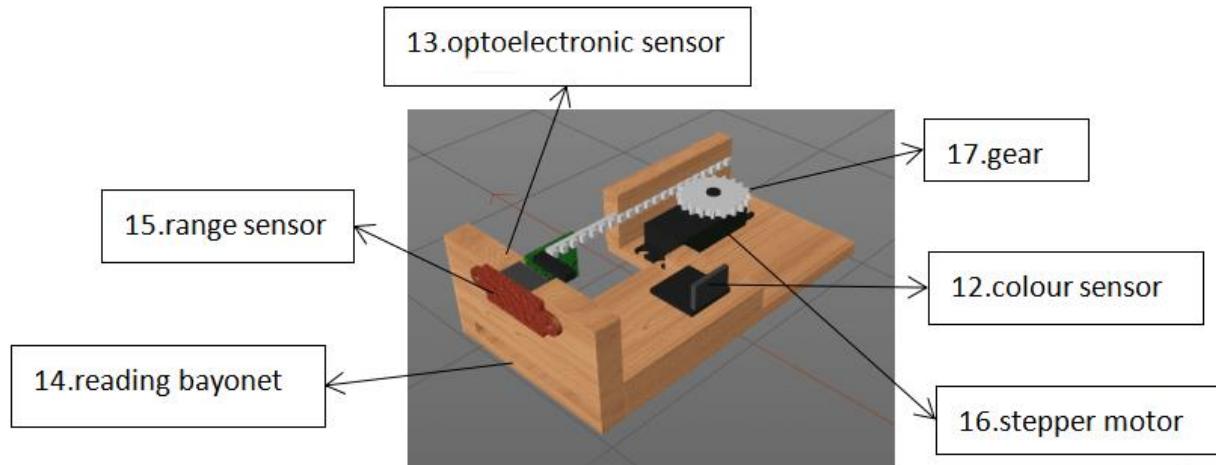


Figure 2-5 C4D model of authentication card reader sub-system

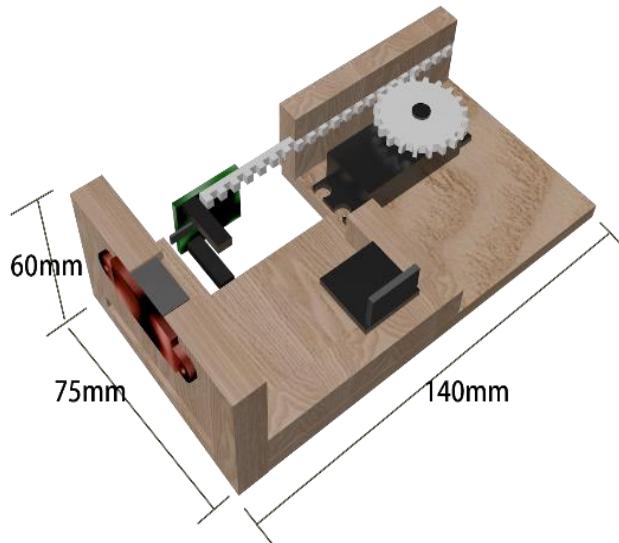


Figure 2-6 Dimension drawing of authentication card reader sub-system

## 2.1.5.3 Layer 3

It refers to the reward sub-system. When the color sensor detects that the user has put all the "pipes" in the correct position, the rewards hatch will be opened at a certain Angle under the control of the steering gear, so that a bottle of water falls from the storage track to the water intake, and the user will get his reward from the water intake.

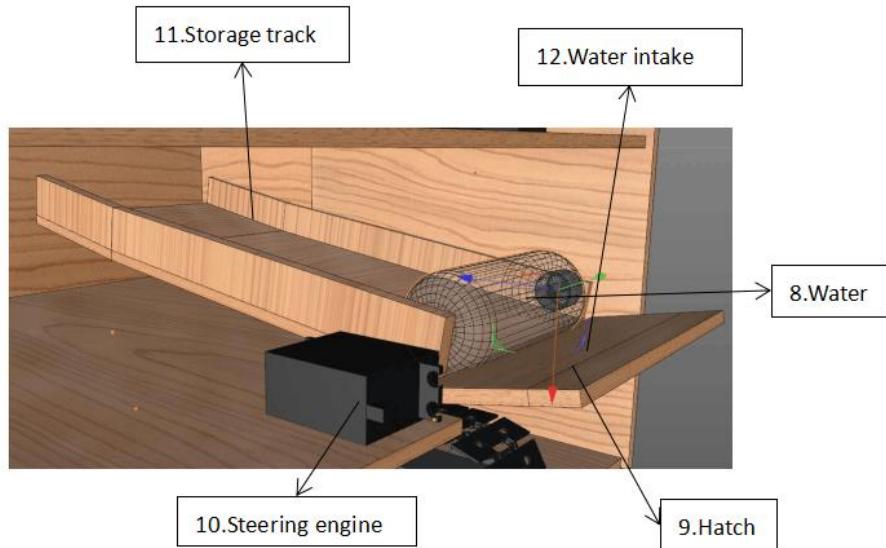


Figure 2-7 C4D model of reward sub-system

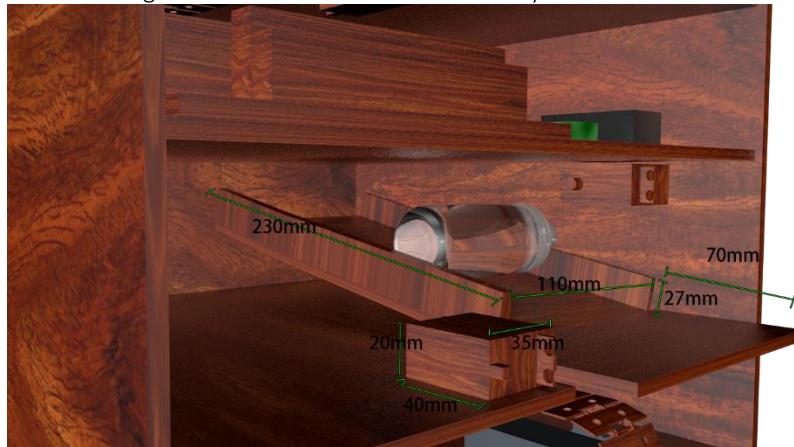


Figure 2-8 Dimension drawing of reward sub-system

#### 2.1.5.4 Layer 4

It refers to the power supply sub-system. This product is equipped with an internal power supply. We use 18650 rechargeable battery for conveniently changing.

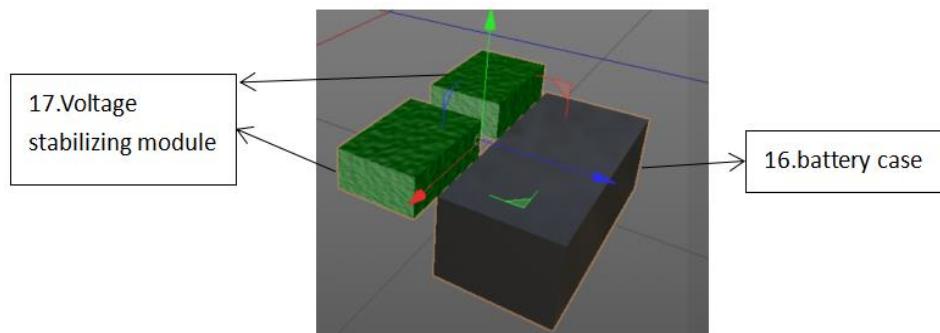


Figure 2-9 C4D model of power supply sub-system

# INTERESTING

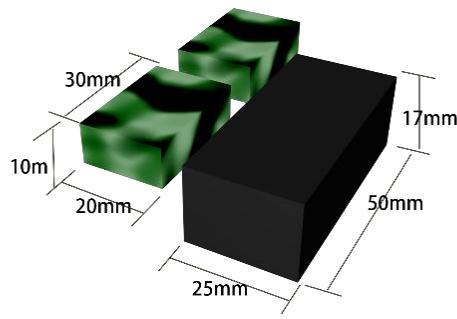


Figure 2-10 Dimension drawing of power supply sub-system

## 2.1.5.5 Layer 5

It refers to the track system. Two gears constitute a set of tracks, and the four gears of the two tracks are respectively controlled by four dc motors. The two gears of the same set of tracks keep the same speed at all times.

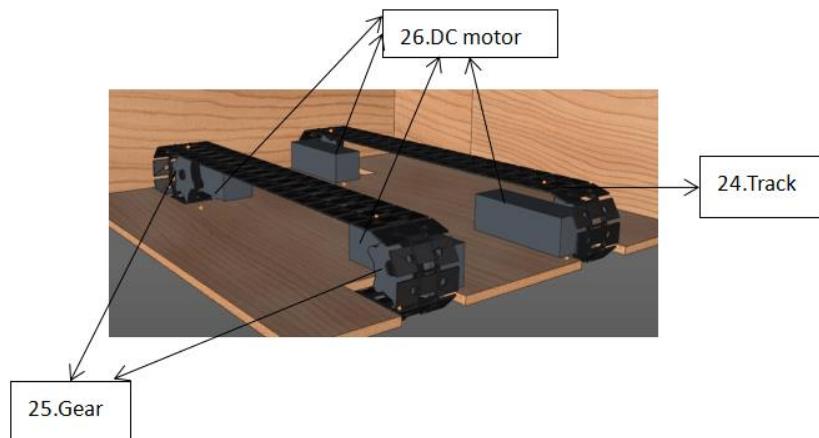


Figure 2-11 C4D model of power sub-system

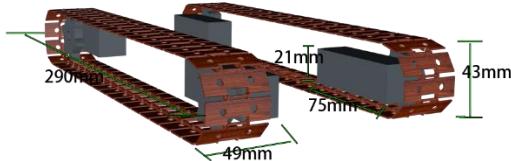


Figure 2-12 Dimension drawing of power sub-system

#### 2.1.5.6 Initial Mechanical Sketch

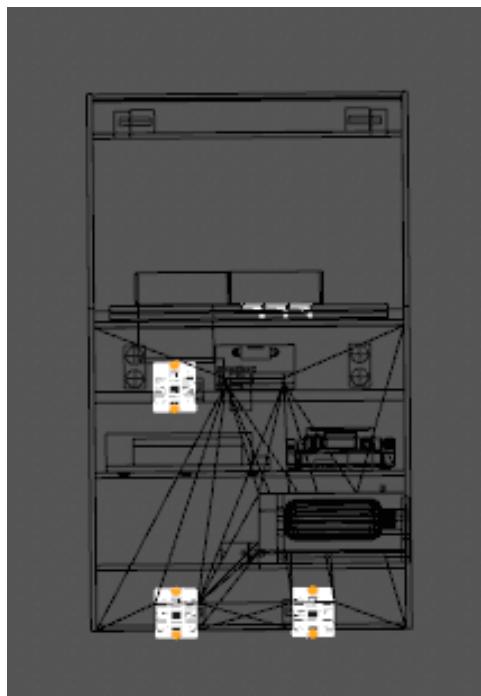


Figure 2-13 Front View

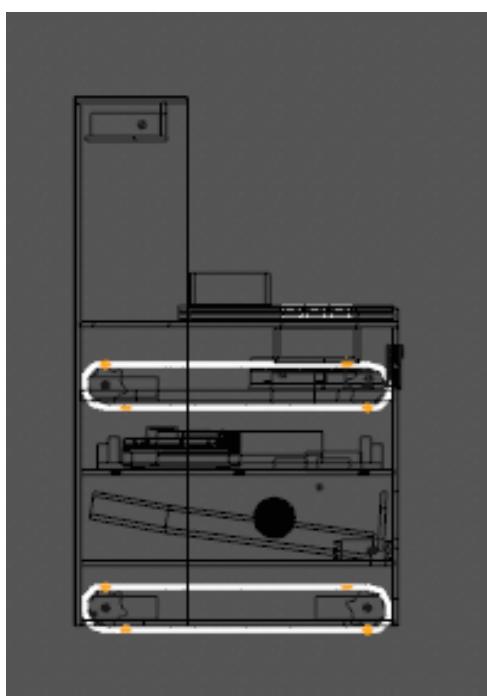


Figure 2-14 Right View

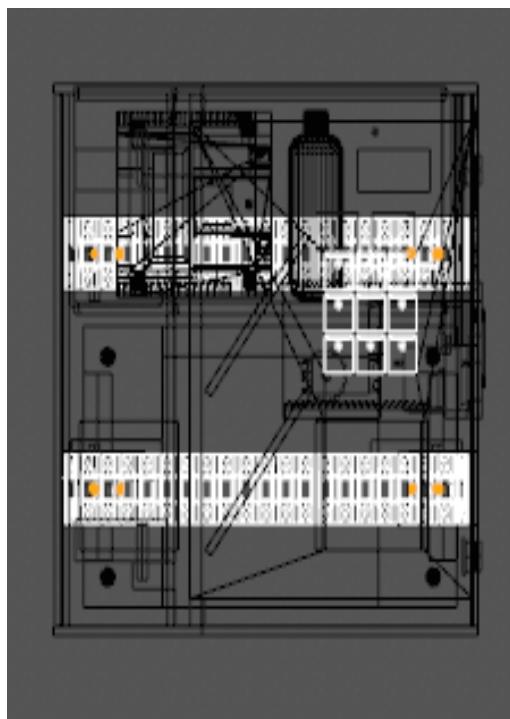


Figure 2-15 Vertical View

# INTERESTING

## 2.1.5.7 Appearance Part Description List

No	Components	Number	Function
1	Game area	1	User put the block according to the picture and then repair the tube
2	Game board	1	A wooden board that contains the game area, recovery track and block compartment
3	Picking entrance	1	User take the board in the block compartment
4	DC motor	2	Provide the power to lift the board through messenger wire
5	Messenger wire	2	Use the power from the motor to lift the board
6	Recovery track	1	During the process of the lifting of the board, due to the gravity, the block compartment will glide from the recovery track, the back to the picking entrance through the track in the recovery module
7	Color sensor	9	Check if the result is correct
8	Water	5-7	As a reward
9	Hatch	1	Open a certain degree and let the bottle fall down
10	Steering engine	1	Control the hatch in order to open a certain degree
11	Storage track	1	Place the water
12	Water intake	1	The user takes the reward (water) from this place
13	Optoelectronic sensor	1	The serial number of the card is determined by detecting the notch on the id card
14	Reading bayonet	1	The user puts the card into authentication module the from the port
15	Range sensor	1	Detect the distance between the user and the device, and activate the interactive system when the user is within 10cm
16	Stepper motor	1	Drive gear
17	Gear	1	The rotation of the gear drives the movement of the gear bar and controls the position of the optoelectronic sensor so that it can test the notch on the card in turn.
18	Battery case	1	Place the battery
19	Voltage stabilizing module	1	Keep the output voltage basically unchanged when the input voltage fluctuates or the load changes
20	DC motor	2	Provide power for the gear
21	Gear	2	Control the movement of the track by the rotation
22	Track	1	Put the bock compartment into the picking entrance through recovery track
23	Block compartment	1	Place the block
24	Track	2	Control the movement of the robot
25	Gear	4	Control the movement of the track by the rotation
26	DC motor	4	Provide power for the gear

## 2.2 Final Design

### 2.2.1 Sub-systems

#### 2.2.1.1 Game Sub-system



Figure 2-16 Motor from the bottom

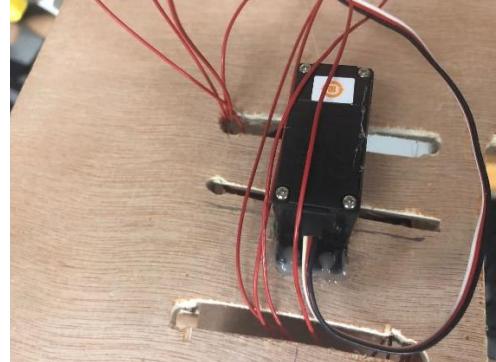


Figure 2-17 Color sensors from the top

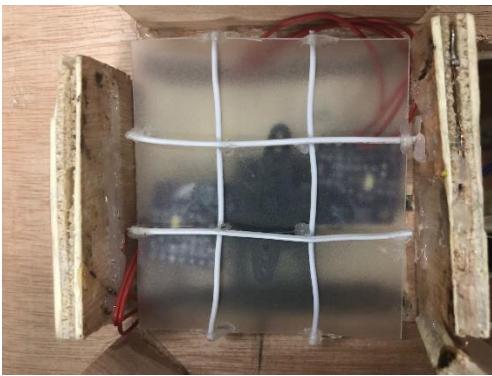


Figure 2-18 Game board

Instead of nine color sensors fixed at the bottom of the game area, we only use two color sensors fixed on the opposite two rotating axles of the motor by hot melt adhesive. The reason we can't use nine color sensors is that we are unable to complete the reuse of the color sensors. The best solution for this problem is that we control two motors to rotate three times, so as to check eight places except the centre of squared paper for practicing calligraphy.

We choose plastic transparent cloth to represent the game board by cutting it into a square, and use DuPont lines and melt adhesive to divide it into nine small squares. Then we need to change our technical drawings because the central square can't be used anymore. To be specific, we can only use eight places.

The rotation angle must be estimated precisely by multiple metering as a result of the interplay between two pipes with different colors.

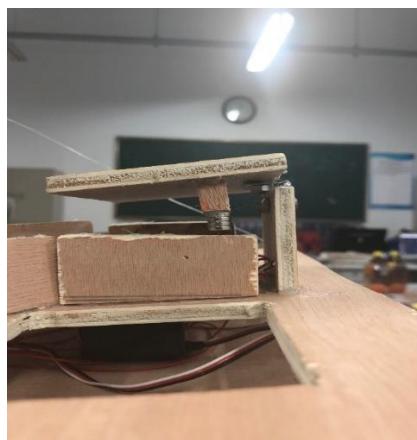


Figure 2-19 The cover

# INTERESTING

Additionally, another modification is that we add a cover on the top of the game area. There are two advantages by doing this:

## 1. Shading

We need to obscure the light source, whether is the natural light (sun) or non-natural light, to make the read of color sensors more accurate, only in this way can we obtain a correct game result.

## 2. As a switch

When we open the cover, the game will begin. Similarly, when we close the cover, the motor will begin to drive the shaft to rotate and the color sensors will check the results of “pipe”. All of these are achieved by the reed pipe sensor.

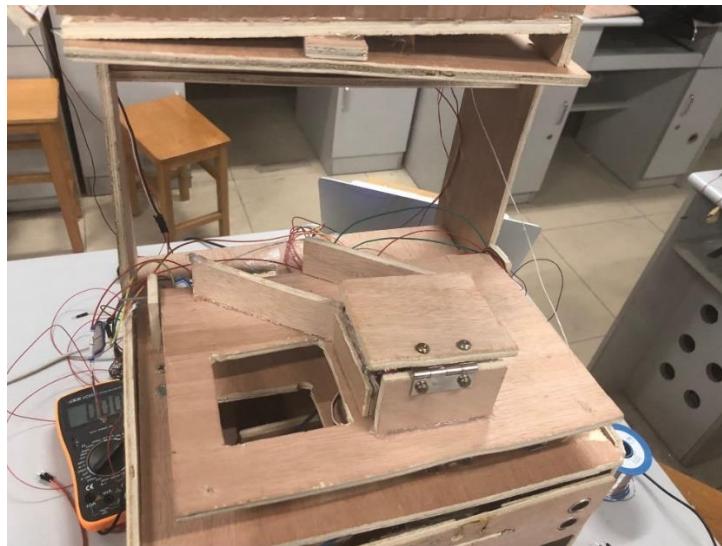


Figure 2-20 Game sub-system

### 2.2.1.2 Recovery Sub-system

The block compartment was made by five blocks with hot melt adhesive because we can't find such a structure that could fit our dimension. The manufacturing process was a little bit rough, but it works efficiently.

The track was bought online, and we assembled it by ourselves. With the help of the gears and DC motors, it could do the job after we defined an appropriate moving distance and velocity.

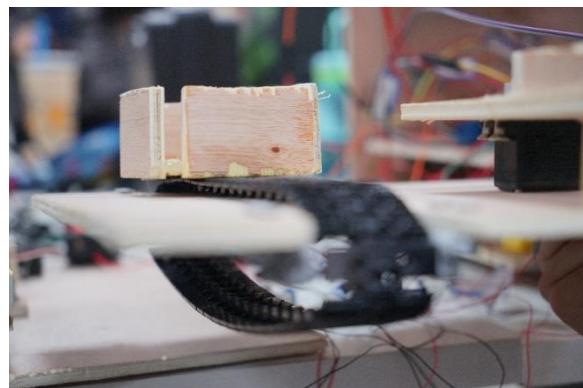


Figure 2-21 Recovery sub-system

### 2.2.1.3 Authentication Card Reader Sub-system

This part is the same as we designed early. The ID card slot was cut by perforators and knife and we decorated by the sticker.

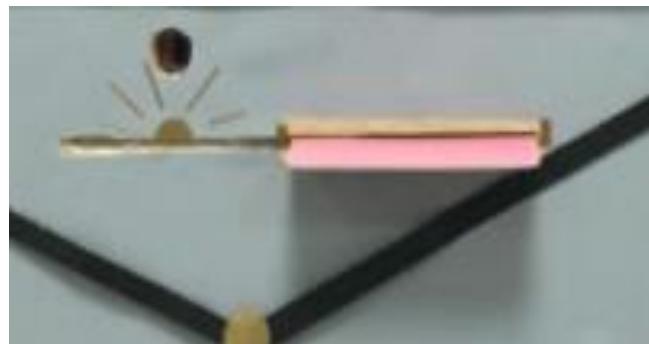


Figure 2-22 ID card slot

The motor will drive the optoelectronic sensor to move three times and check if there is a gap and define it is a user or a debugging worker. It is similar to other parts, we also utilized wooden board and hot melt adhesive for the place of ID card.  
(A gap means a 0, 1111 means an operator)

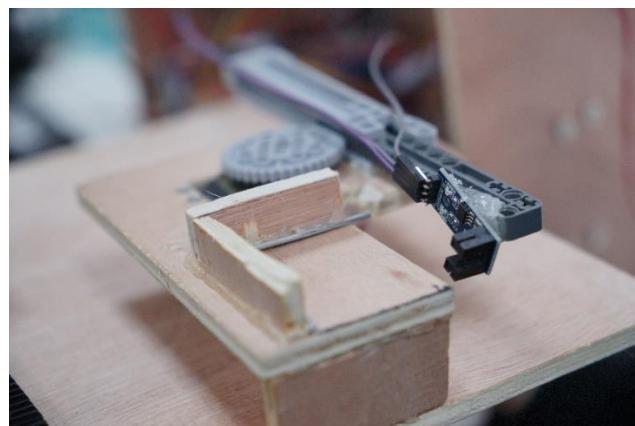


Figure 2-23 Authentication card reader sub-system

#### 2.2.1.4 Reward Sub-system

When the user finishes the game and win the prize, the hatch will be released automatically by the steering engine which is beside the hatch. We put a row of LED indicators on the board and there are two advantages.

1. It is a beautiful decoration and we believe that users will have a good mood when they take the prize out.
2. It can shine, and users will clearly see the prize.

Screws and nuts were placed to connect and fix the storage track and the board behind it. Perforators and screwdrivers are necessary.



Figure 2-24 Reward Sub-system

# INTERESTING

## 2.2.1.5 Power Supply Sub-system

Two rechargeable batteries and two voltage stabilizing modules compose this module. It is one of the most important parts of the machine because the function of this sub-system is to charge the whole robot conveniently and efficiently.

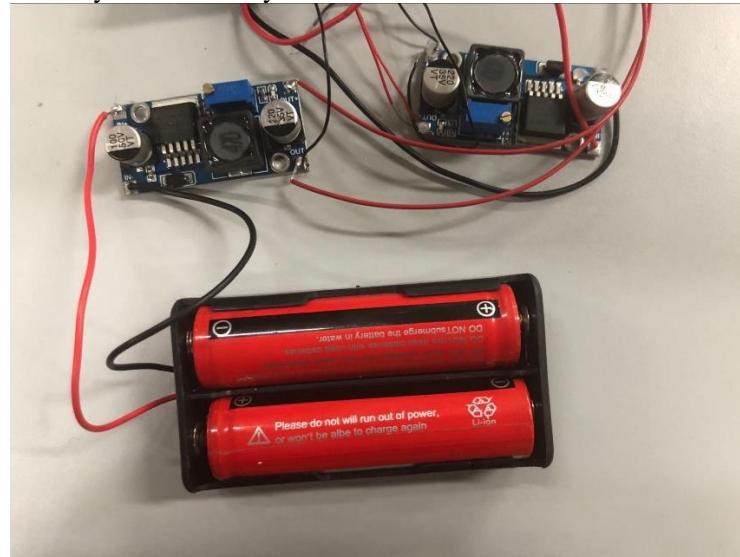


Figure 2-25 Power supply sub-system

## 2.2.1.6 Power Sub-system

At the beginning, we decided to use ultrasonic distance sensors and infrared sensor to control the track at the bottom.

The white circle at lower right corner is infrared sensor which is used for detecting whether it is a person or not. If it is, the machine will begin to work, if it is not, the machine will ignore it and find the new thing to detect.

The two-black circle at higher left corner are ultrasonic distance sensors which are used to detect whether there is an obstacle or not. If there is, the machine will adjust the speed of track on both sides in order to turn right or left due to the velocity difference. If there isn't, the machine will begin to work. By the way, in our user interface, there are two eyes to represent it, if there is a stuff at the right, no matter it is a person or an obstacle, two eyes will look at the right direction. The same to the left direction.

But unfortunately, the robot is too heavy to be driven by the track system (two tracks). Another reason is the voltage is too small, we tried to use a device for lowering voltage, but it didn't work. To be specific, we can just treat it as a prototype (model machine) and this is only an ideal function.



Figure 2-26 Ultrasonic distance sensors and infrared sensor



Figure 2-27 Power sub-system

### 2.2.2 Final Three-view drawing



Figure 2-28 Right view



Figure 2-29 Front view



Figure 2-30 Vertical view

# INTERESTING

## 2.2.3 Dimensional Drawing

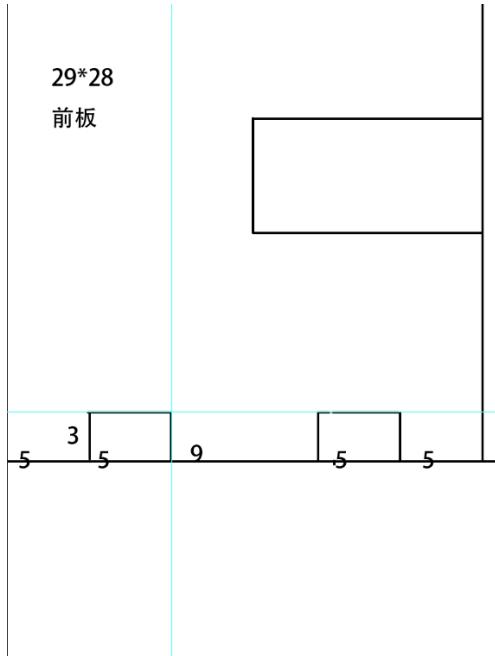


Figure 2-31 Track (small)

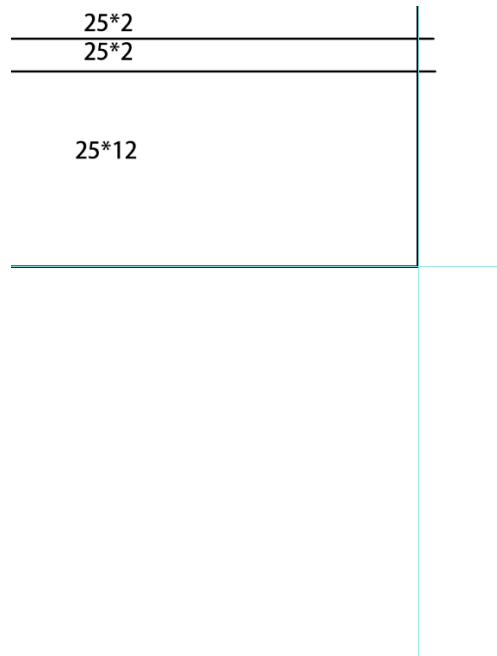


Figure 2-32 Forward board (small)

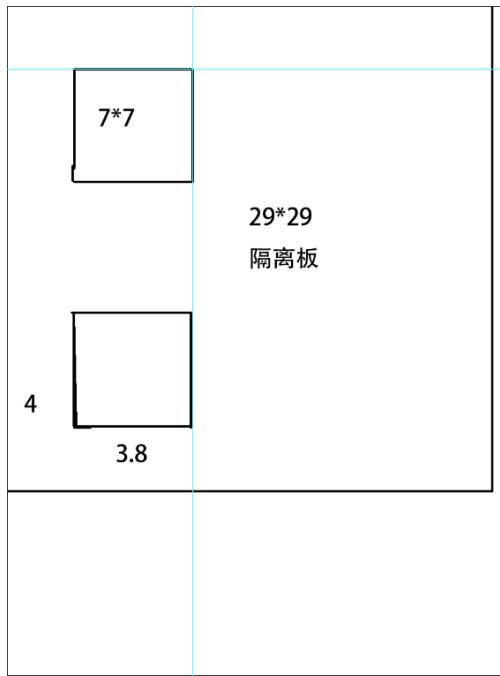


Figure 2-33 Top baffle's front 1 (small)

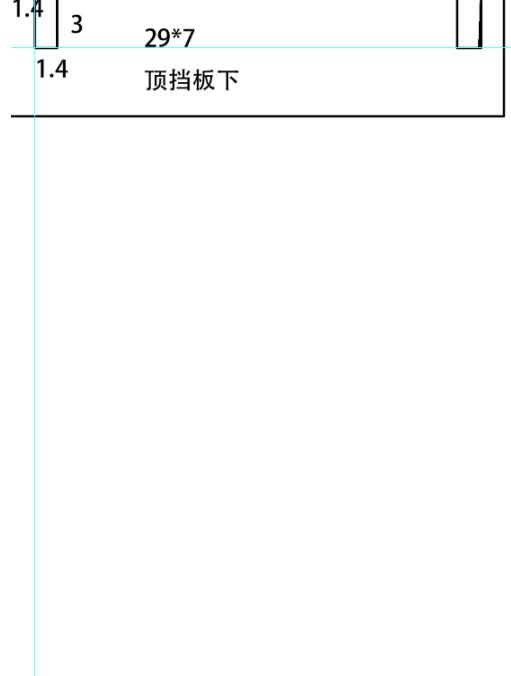
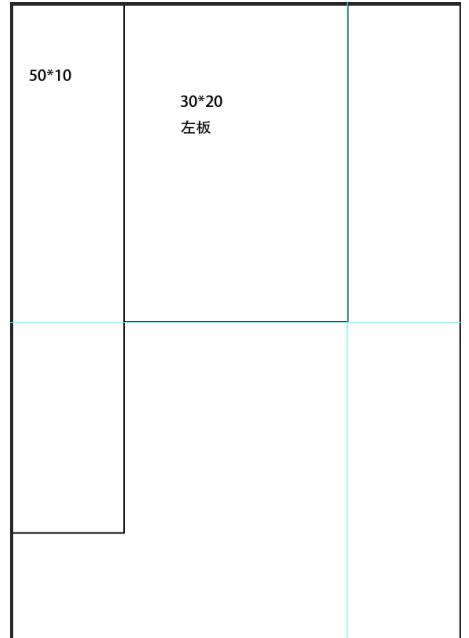
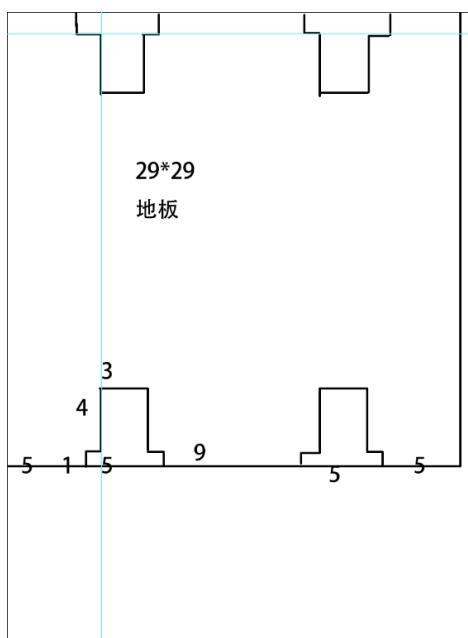
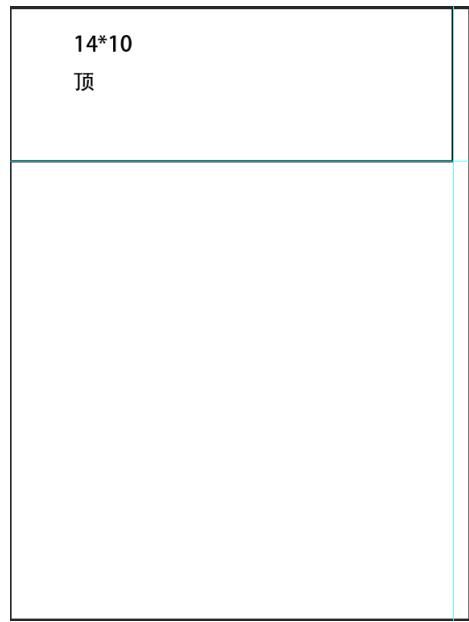
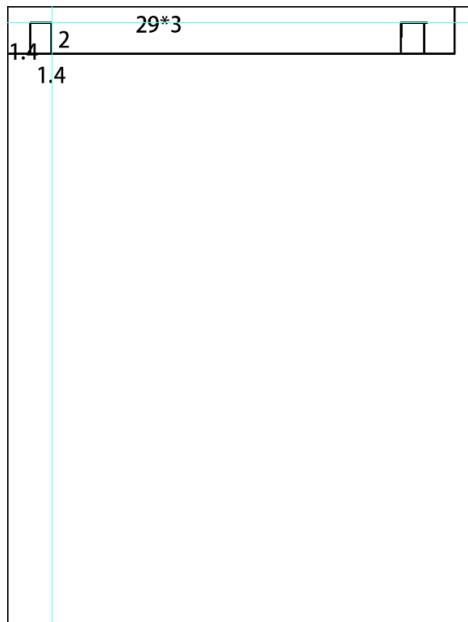


Figure 2-34 Isolation board (small)



# INTERESTING

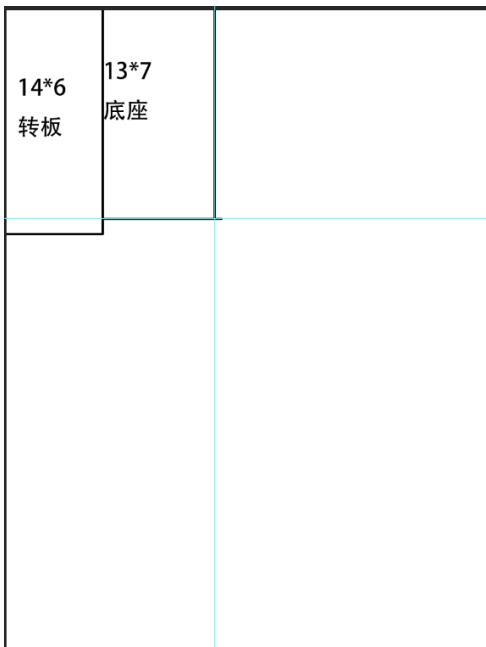


Figure 2-39 Operation panel (small)

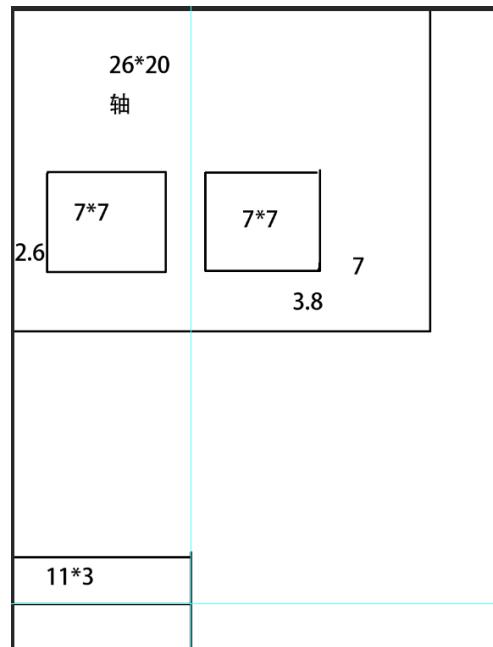


Figure 2-40 Rotating board and base (small)

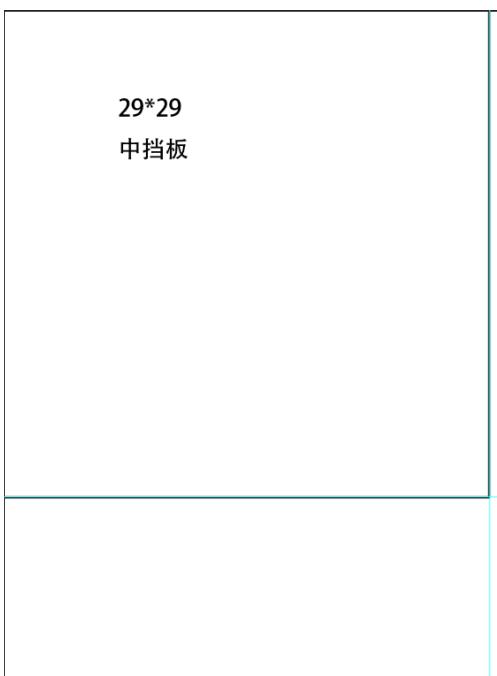


Figure 2-41 Middle baffle 1 (small)

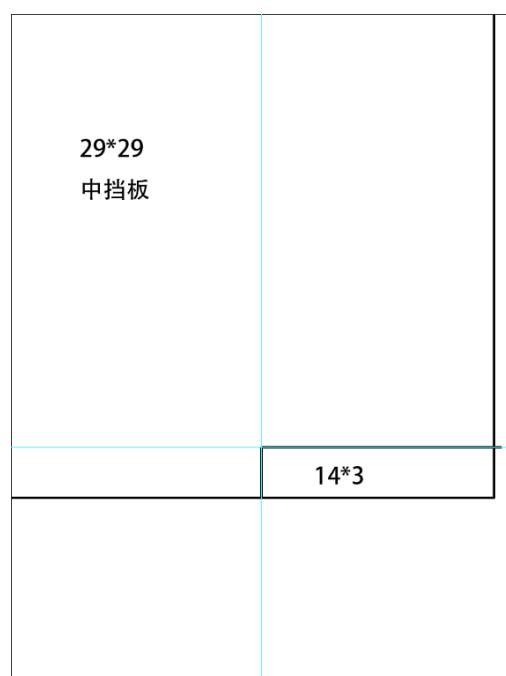


Figure 2-42 Middle baffle 2 (small)

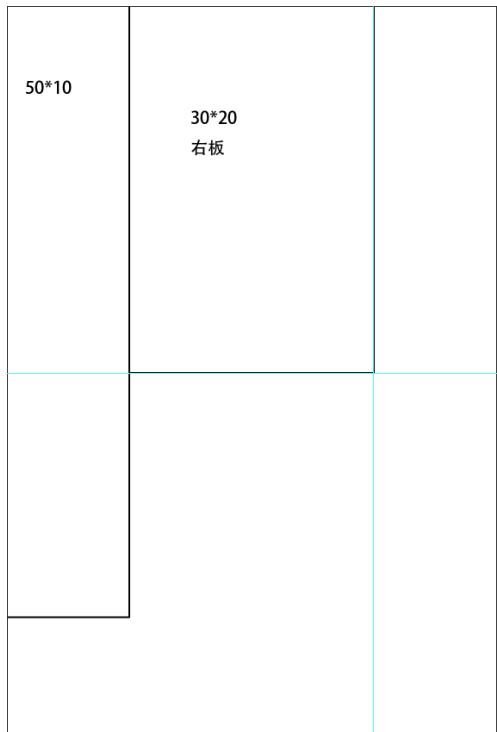


Figure 2-43 Overhead gauge (small)

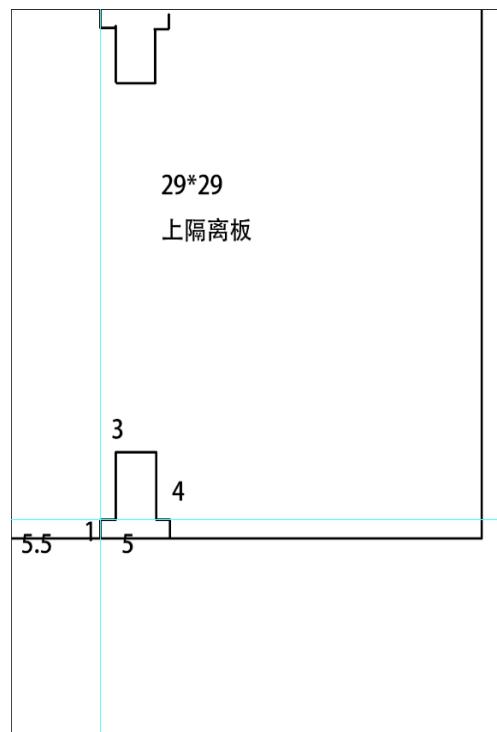


Figure 2-44 Right board (big)

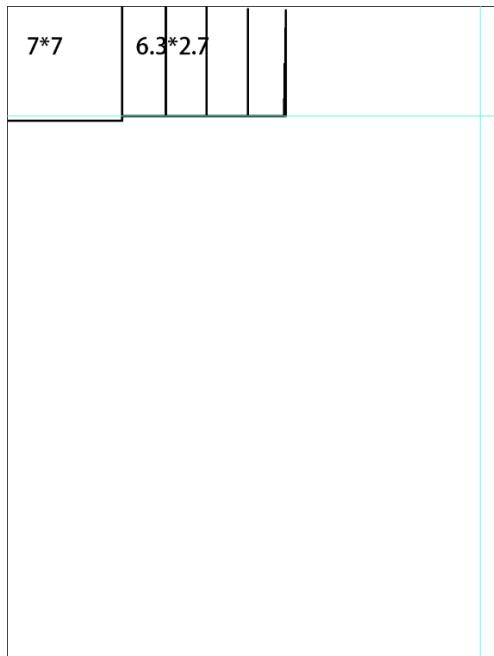


Figure 2-45 Back board (big)

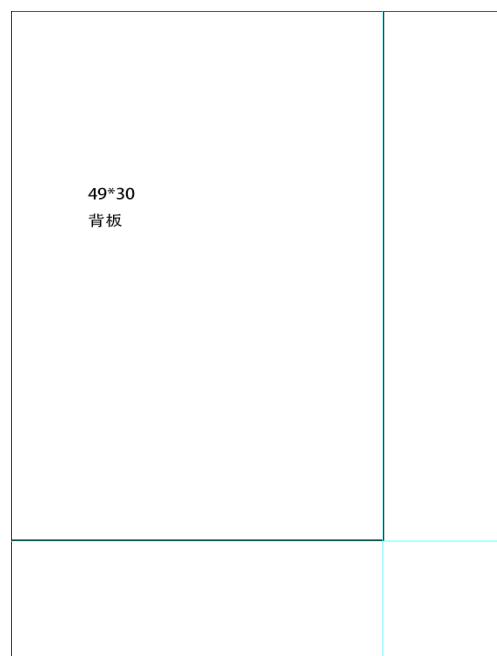


Figure 2-46 Box (small)

# INTERESTING

## 2.2.4 Operation Procedure

Step 1: When the user is 10cm from the device, the ultrasonic distance sensor on the device will detect the user's position, and the robot will move to the user's face and request the user to help repair the "pipe". At this time, the display will display the prompt of the user to insert the card.

Step 2: The player needs to insert his ID card into the card-reading port and the device will automatically complete your identification.

(In order to give users and developers a better experience, GUI is designed with two modes: player mode and debugging mode, which is used for the maintenance and debugging of the machine. The mode depends on the number of your ID card. Only users with a number of 0000 can enter debugging mode.)

### 2.2.4.1 Player Mode

Step 3: After identification, the screen will display the user's identity information. At the same time, the game cabin will be opened slowly. The user only needs to click the start button to start the game.

Step 4: The user needs to place the object blocks of different colors representing the pipeline in the correct position in the game area, so as to connect the water pipe inside the robot. In the game, the game drawings and the player's current time are displayed on the screen.

Step5: After the game, the equipment will automatically detect whether the pipeline placed by the player is correct. If the result is correct, the robot will send the user a bottle of water to thank him, and then close the cabin to find the next user.

### 2.2.4.2 Debugging Mode

Step 3: After detecting the identity of the debugger, the device will enter the debugging mode.

Step 4: At the same time, the display screen will enter the debugging interface. There will be three options on the screen: "Component debugging", "drawing setup", "background music".

Step 5: After debugging, click "exit debugging" in the lower right corner of the screen to exit debugging.

This is just a simple summary of debugging mode and the detailed description and analysis will be in part 4 software design

### 3 DIGITAL SYSTEM DESIGN

#### 3.1 Introduction

##### FPGA

FPGA (field-programmable Gate Array) is a product of further development on the basis of programmable devices such as PAL, GAL and CPLD. It is a semi - custom circuit in the field of ASIC. The appearance of FPGA not only solves the deficiency of custom circuit, but also overcomes the shortcoming of limited gate circuit number of original programmable devices.

FPGA uses hardware description language (Verilog or VHDL) to complete circuit design, so that developers can quickly download the program to FPGA for testing after a simple integration and layout. This design method is the mainstream of modern IC design verification technology. The FPGA comes with editable elements (such as AND, OR, XOR, NOT) that make circuit design faster AND more efficient.

We can connect logic blocks inside the FPGA through editable connections as needed, as if a circuit test board were placed on a chip. The logic blocks and connections of a finished FPGA can be changed according to the designer, so the FPGA can complete the required logical functions.

##### DC motor

Dc motor is a motor that converts dc electric energy into mechanical energy. Because of its good speed regulation performance, it is widely used in electric power drive. In our project, we used dc motor to drive the belt and gear.

##### Servo

Servo motor is the main device for FPGA control. Servo motor can guarantee the speed's and the position's accuracy, and the voltage signal can also be converted into torque and speed in order to drive the controlled object. Servo motor rotor's speed is controlled by input signal, and can react quickly, in the automatic control system, as an executing element, and has the characteristics of small electromechanical time constant, high linearity and starting voltage, etc. The received electrical signals are able to be converted into the motor shaft angular displacement or angular velocity output.

There are two categories, dc and ac servo motor. Their main characteristics are that when the signal voltage is zero, there is no rotation phenomenon even if the torque or the speed is increased by the operator.

#### 3.2 Function

FPGA is the bridge between MBED and servo motor. FPGA controls servo motor rotation based on signals sent from MBED. According to the protocol between FPGA and MBED, FPGA converts the signal from MBED into one of the 8 pre-set states in FPGA, and different states correspond to different angles of servo motor rotation. This mechanism realizes that MBED controls servo motor rotation through FPGA.

#### 3.3 Code Structure

##### counter

The key of the control for servo motors is to determine the ratio between the clock period and the PWM signal.

It is quite similar to the theory of frequency divider. The frequency of the clock cycles assumed as 1M Hz, but the frequency of the PWM signal is 50Hz. Consequently, we should use a large number of clk pulse to compose the PWM signal and determine the number of high-level and low-level respectively.

1. // Cummulator
2. reg [n-1: 0] Q, R;
3. reg [n-1: 0] Q\_1, R\_1;
4. reg [n-1: 0] Q\_2, R\_2;

# INTERESTING

## states store

A 3-bit register stores 8 binary Numbers from 000 to 111, each of them represents a pulse width signal, which allows servo motor to rotate at different angles.

```
1. // States store
2. // 8 states in all
3. reg [2:0] state;
4. reg [2:0] state_1;
5. reg [2:0] state_2;
6. wire ld, cz;
```

wire ld\_1, cz\_1;

## Signals

Internal signals are responsible for communication within the circuit

```
1. // signals
2. // ld: indicate when need to reset the cummulator
3. // cz: indicate the current output
4. assign ld = (Q >= (M - 1));
5. assign cz = (Q < (R - 1)) | ld;
6. assign ld_1 = (Q_1 >= (M - 1));
7. assign cz_1 = (Q_1 < (R_1 - 1)) | ld_1;
8. assign ld_2 = (Q_2 >= (M - 1));
9. assign cz_2 = (Q_2 < (R_2 - 1)) | ld_2;
```

## FSM

The angle of the servo is controlled according to the bandwidth of the pwm. The input takes the forms of a pulse width signal that varies between 1mS and 2mS. Varying the input signal between 1mS and 2mS gives a working angle of 90 degrees. The pulse is 50Hz.

A wave form example:

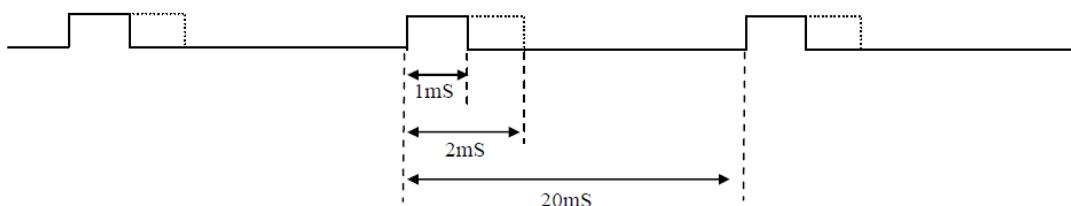


Figure 3-1 PWM explain

As it is indicated by the code, there are three output pwms, thus three FSM, which have 8 states. Each state is an angel of the servo.

The method for changing the angle of one servo:

1. Select the servo (00, 01, 11)
2. Select the angle (000 – 111)

```
1. //FSM
2. always @ (posedge clk)
3. begin
4.   if (ld) begin
5.     Q <= 0;
6.     pwm1 <= cz;
7.     if (set == 2'b00)
8.       begin
9.         state <= selector;
10.      end
11.    end
12.   else begin
13.     Q <= Q + 1;
14.     pwm1 <= cz;
15.   end
16. end
17.
18. always @ (posedge clk)
19. begin
20.   if (ld_1) begin
21.     Q_1 <= 0;
22.     pwm2 <= cz_1;
23.     if (set == 2'b01)
24.       begin
25.         state_1 <= selector;
26.       end
27.     end
28.   else begin
29.     Q_1 <= Q_1 + 1;
30.     pwm2 <= cz_1;
31.   end
32. end
33.
34. always @ (posedge clk)
35. begin
36.   if (ld_2) begin
37.     Q_2 <= 0;
38.     pwm3 <= cz_2;
39.     if (set == 2'b11)
40.       begin
41.         state_2 <= selector;
42.       end
43.     end
44.   else begin
45.     Q_2 <= Q_2 + 1;
46.     pwm3 <= cz_2;
47.   end
48. end
49.
50.
51. always @(state)
52. begin
```

# INTERESTING

```
53. case (state)
54.   0: R = K1;
55.   1: R = K9;
56.   2: R = K10;
57.   3: R = K11;
58.   4: R = K12;
59.   5: R = K6;
60.   6: R = K7;
61.   7: R = K8;
62. endcase
63. end
64.
65. always @(state_1)
66. begin
67.   case (state_1)
68.     0: R_1 = K1;
69.     1: R_1 = K12;
70.     2: R_1 = K3;
71.     3: R_1 = K13;
72.     4: R_1 = K14;
73.     5: R_1 = K6;
74.     6: R_1 = K7;
75.     7: R_1 = K8;
76.
77.   endcase
78. end
79.
80. always @(state_2)
81. begin
82.   case (state_2)
83.     0: R_2 = K1;
84.     1: R_2 = K2;
85.     2: R_2 = K3;
86.     3: R_2 = K4;
87.     4: R_2 = K5;
88.     5: R_2 = K6;
89.     6: R_2 = K7;
90.     7: R_2 = K8;
91.
92.   endcase
93. end
94.
95. endmodule
```

### 3.4 ASM chart and State Machine chart

The eight states of FPGA can jump to each other. The following charts just gives an example of jumping from 000 to seven other states, but in fact, each state can jump to seven other states:

000->001,010,011,100,101,110,111

001->000,010,011,100,101,110,111

010->000,001,011,100,101,110,111

011->000,001,010,100,101,110,111

100->000,001,010,011,101,110,111

101->000,001,010,011,100,110,111

110->000,001,010,011,100,101,111

111->000,001,010,011,100,101,110

#### ASM chart

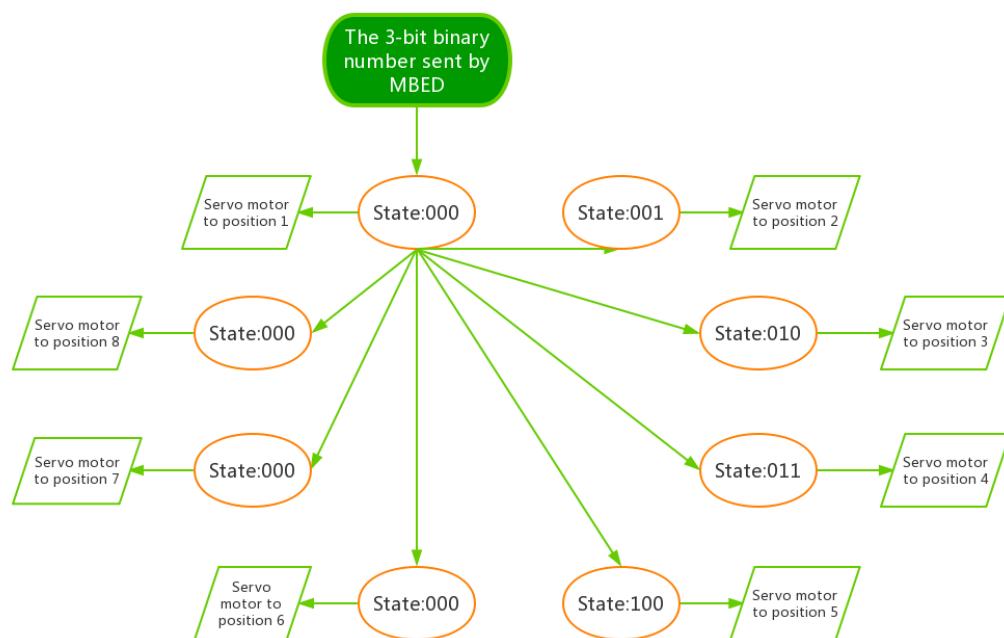


figure 3-2 ASM Chart

#### State Machine Chart

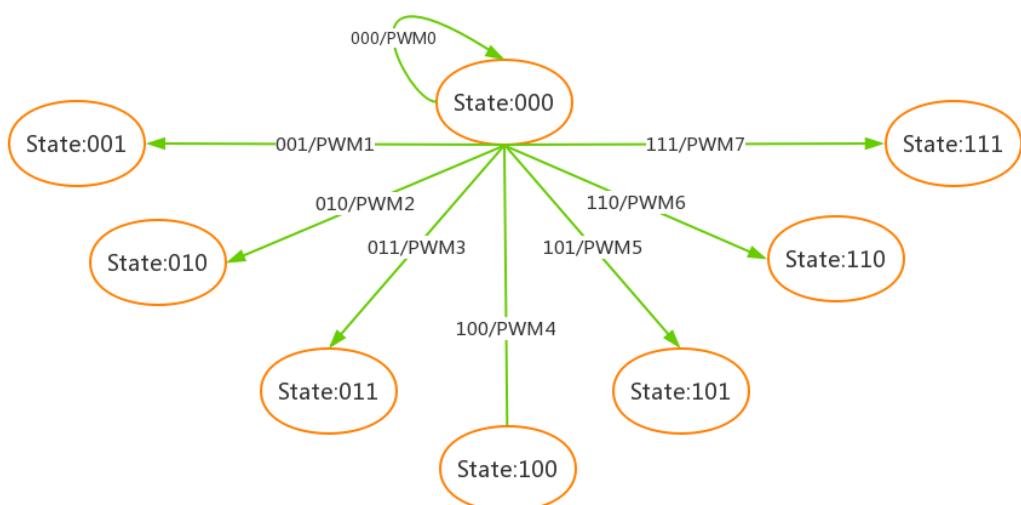


Figure 3-3 State Machine Chart

# INTERESTING

## 3.5 Block diagrams

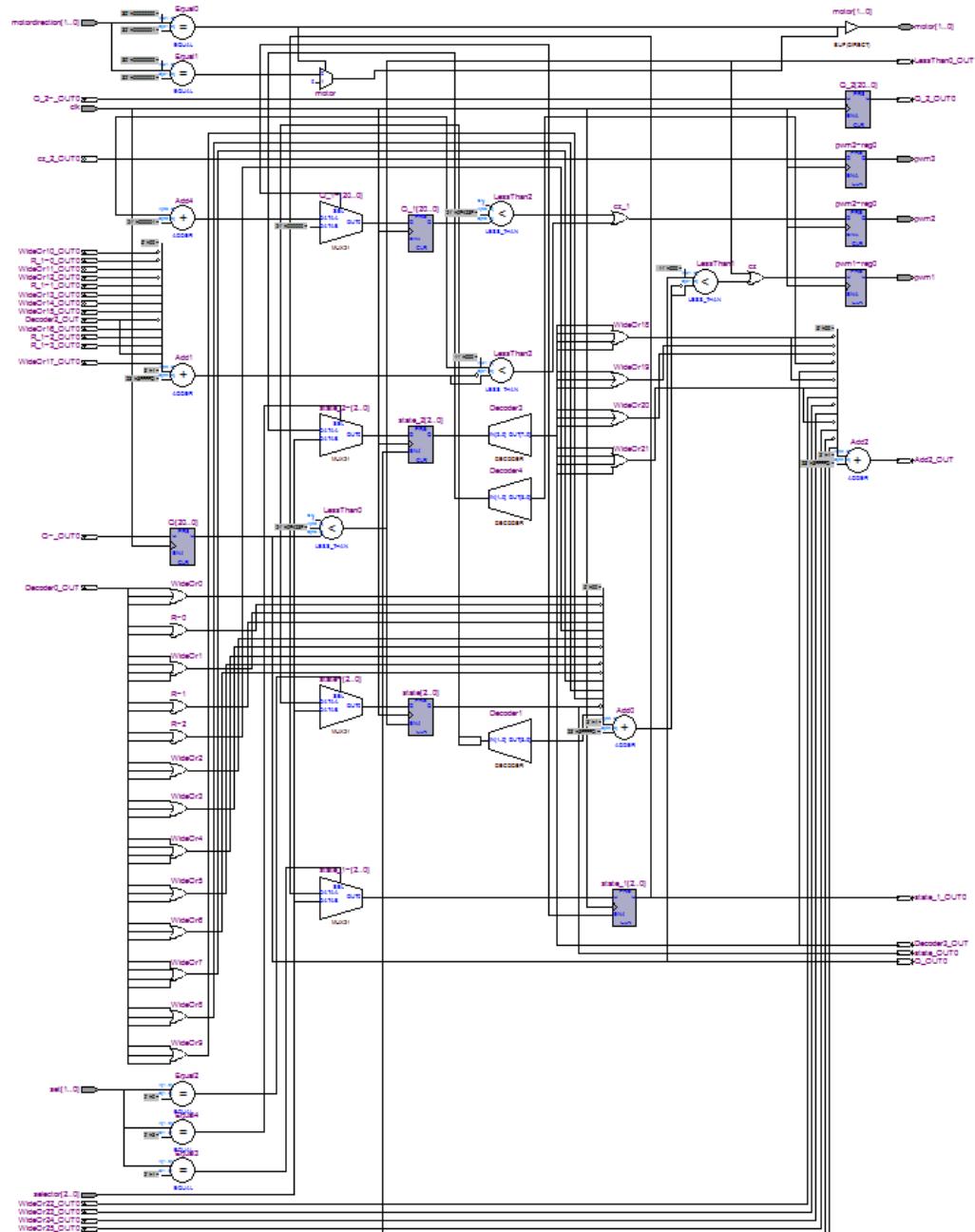


figure 3-4 RTL Diagram

## 3.6 Pin Planning

	Node Name	Direction	Location	I/O Bank	VREF Group	I/O Standard	Reserved	
1	clk	Input	PIN_L1	2	B2_N1	3.3-V LVTTL (def...		
2	pwm1	Output	PIN_A13	4	B4_N1	3.3-V LVTTL (def...		
3	pwm2	Output	PIN_H12	4	B4_N1	3.3-V LVTTL (def...		
4	pwm3	Output	PIN_B13	4	B4_N1	3.3-V LVTTL (def...		
5	selector[2]	Input	PIN_J19	5	B5_N1	3.3-V LVTTL (def...		
6	selector[1]	Input	PIN_J18	5	B5_N1	3.3-V LVTTL (def...		
7	selector[0]	Input	PIN_L19	5	B5_N1	3.3-V LVTTL (def...		
8	set[1]	Input	PIN_K20	5	B5_N1	3.3-V LVTTL (def...		
9	set[0]	Input	PIN_L18	5	B5_N1	3.3-V LVTTL (def...		
10	<<new node>>							

Figure 3-5 Pin Planning

## 4 SOFTWARE DESIGN

### 4.1MBED

#### 4.1.1 Introduction

Mbed is a 32-bit ARM cortex-m SCM based networking device platform and operating system. After logging into the MBED website, you can write programs in the development environment provided by MBED. After the program is written, we download the program to the microcontroller. This development mode avoids the compatibility problems caused by different programming software and greatly improves the development efficiency.

#### Applications

Applications on the Mbed platform can be developed using the Mbed online IDE (a free online code editor and compiler). Only one Web browser needs to be installed on the local PC because the project is compiled on the cloud, that is, on the remote server, using the ARMCCC C/C++ compiler. The Mbed IDE provides the ability for private workspaces to import, export, and share code with distributed Mercurial version control, and it can also be used for code document generation. Applications can also be developed using other development environments, such as Keil., the IAR embedded workbench and Eclipse, which uses GCC ARM embedded tools.

#### MBED OS

MPES operating systems provide McB/C/C++ software platforms and tools for running microcontroller firmware on IOT devices. It consists of core libraries that provide microcontroller peripheral drivers, networks, RTOS and runtime environments. It can also build tools, test and debug scripts. Component databases provide driver libraries for components and services that can be connected to microcontrollers to build the final product.

#### 4.1.2 Functions of MBED

MBED is the bridge of this system. It connects PC, FPGA and various sensors just like a transportation hub. Besides, it also performs lots of data operations.

##### Basic Functions:

- **The control sensor reads the data.** For example: control the color sensor to read the color, control the distance sensor to read the distance.
- **Receive detection data from the sensor.** For example: receive the color data read by the color sensor, receive the distance data returned by the distance sensor, and receive the id card serial number returned by the photoelectric contact sensor.
- **Send detection data and important signals to PC.** For example: send the testing data to the PC in debugging mode; Send the game results to the PC in user mode and send warning signals to the PC that the user is not within the range.
- **Receive operation instructions from PC.** For example: execute instructions sent by maintenance workers in debugging mode; Receive reset signal from PC in user mode.
- **8-bit signal is sent to the FPGA to make the FPGA control servo to perform the corresponding operation.** For example, when the identity information is read, the servo is controlled to rotate three times to drive the photoelectric sensor to read the serial number on the identity card.

#### 4.1.3 The programming language (C++)

C++ is the inheritance of C language. It can not only carry out the procedural programming of C language, but also the object-oriented programming characterized by abstract data type and polymorphism.

C++ not only has the practical characteristics of efficient computer operation, but also aims to improve the programming quality of large-scale programs and the problem description ability of programming language.

# INTERESTING

## 4.1.4 Structure

### 4.1.4.1 Two Modes

There are two modes in our operating system: **User Mode** and **Maintenance Mode**. User Mode is for customers to interact with the device and play the game. The maintenance mode is specially designed for maintenance personnel to test all the components and repair the device breakdown.

The original state of the device is the standby search state. When the distance sensor detects that a user is coming, the PC will prompt the user to insert an ID card for identification. The identity serial number 0-14 is the player identity, and the identity serial number 15 is the exclusive identity of the maintenance personnel. Users with an identity sequence number less than 15 will enter User Mode, and users with an identity sequence number of 15 will enter Maintenance Mode. After exiting from both modes, the device will enter the standby search state again.

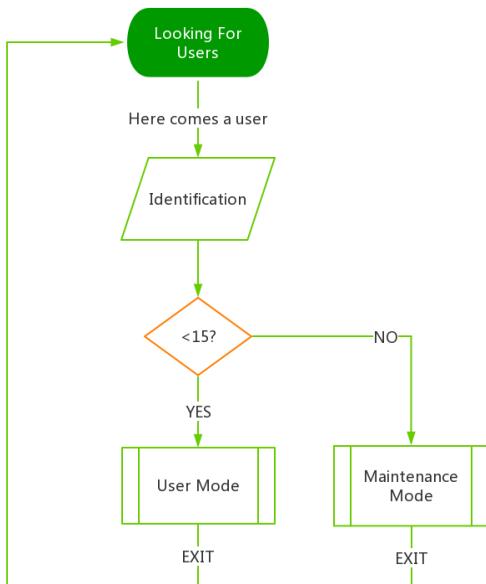


Figure 4-1 original state Logic

### 4.1.4.2 Maintenance Mode

When the user enters maintenance mode, MBED will always wait for the operation instructions entered by the maintenance staff. When MBED receives the instruction, it will control the relevant parts to complete the operation required by the instruction and return the corresponding data, and then continue to wait for the instruction input by the maintenance staff. MBED repeats the loop until the user exits the mode.

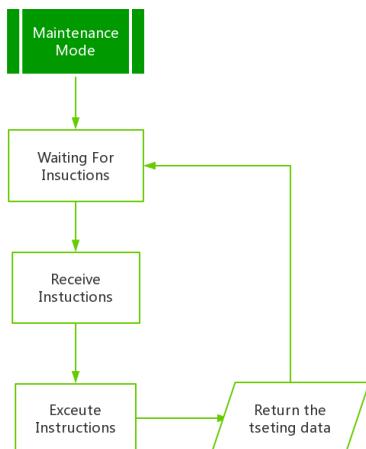


Figure 4-2 Maintenance Mode

#### 4.1.4.3 User Mode

When the user enters User Mode, MBED waits for the user to click the "start" button on the interface. When MBED receives the start signal from PC, MBED enters the game state.

In the game state, MBED will control the distance sensor to detect the distance between the device and the user. When the user is too far or too close to the device, MBED sends a signal to the PC and starts the timer. The machine automatically resets when the timer exceeds 15 seconds. During the timing process, if the user returns to the specified range, the game will continue.

When MBED detects that the user has closed the game box, servos will be started to drive the color sensor to detect the game results. If the user succeeds, MBED will control servo to open the door of the reward cabin and send rewards to the user. If the user fails, the device resets directly to find the next user.

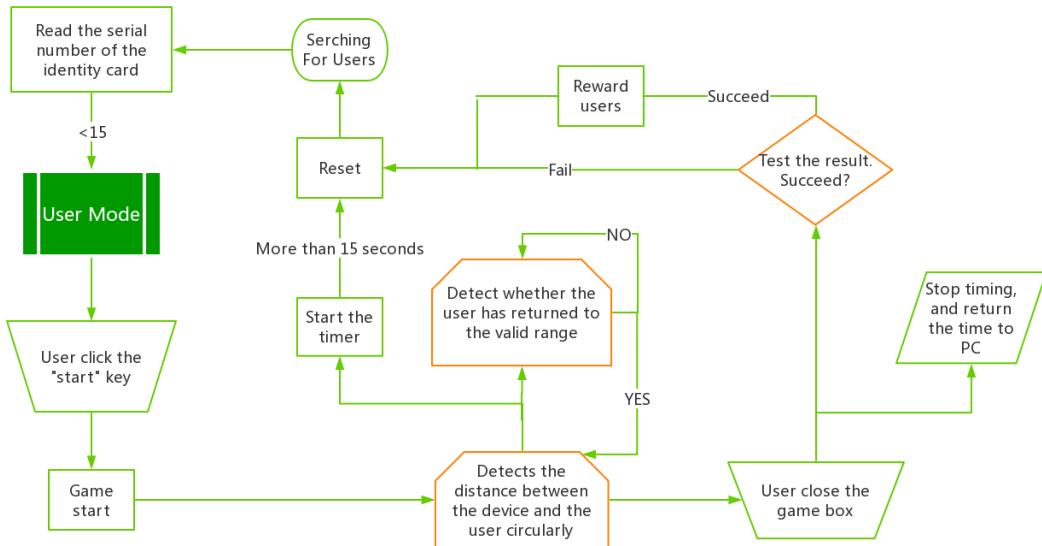


Figure 4-3 User Mode

#### 4.1.5 Pseudocode

```

1. void GameLogicControl()
2. {
3.     while(1)
4.     {
5.         wait(1);
6.
7.         FindUser();
8.
9.         if(!User)
10.        {
11.            TurnToMaintenance();
12.            continue;
13.        }
14.
15.        Preparing();
16.
17.        GameStart();
18.
19.        PlayGame();
20.
21.        if(Out of range)
22.        {
23.            Reset();
  
```

# INTERESTING

```
24.         continue;
25.     }
26.
27.     if(GameEnd())
28.     {
29.         Reward();
30.     }
31.
32.     Reset();
33. }
34. }
```

## 4.1.6 Important Functions

### Maintenance Mode

- **void doumo()**: Execute the user's instructions.
- **void GameComplete()**: //end the game when the cover of the game box is turned off and the magnet switch is turned off
- **void helperfunc()**: Show the command list for the Maintenance maintenance worker

### User Mode:

- **void fpgafunc ()**: This function can be called to communicate with the FPGA through the serial ports
- **bool Identification()**: Read the identity information on the identity card.
- **void PlayGame()**: Detect if the user is out of specified range. When the distance sensor detects that the distance between the user and the device exceeds the specified range, the following command will be executed:MBED will set isoutofrange to 1, and then start the timer, warning voice, and interface prompt. When the distance sensor detects that the user enters the specified range again, the game continues.
- **bool GameEnd ()**: Send the game result to PC.
- **void Reset ()**: Reset the whole device and enter the initial state.

## 4.2 PC

### 4.2.1 PC Functions

The software system running on PC is the main interactive carrier between the users and our device, and also takes responsibility of processing the users 'operation to manipulate the whole process of our system.

#### Basic Functions:

- **Interaction with users**: the interface displays all human-computer interaction information. Include: guide information, game results, leaderboard.
- **Control process**: user control of the game process will be completed by the interface. For example, the game does not start until the player presses the "start" key. The device does not reset until the player presses "complete" key.
- **Receive and process the signal fromMBED**. For example, whenMBED sends a signal that the user is too far or too close to the device, the PC will start the alarm system.
- **Send instructions toMBED**. For example: in maintenance mode, instructions from the maintenance staff are sent toMBED by PC, and then executed byMBED.

### 4.2.2 GUI Design

#### 4.2.2.1 WPF framework

There are lots of different languages, frames and tools to design a GUI on the Windows platform, and when talking about cross-platform, many good tools like Qt, Xamarin provide better solutions.

WPF is not a cross-platform, but it is one of the best toolkits on Windows platform. Another well-known code base is Windows Forms, while many full-featured desktop applications have been successfully built using Windows Form, the fact of the matter is that this programming model is asymmetrical. Simply, the `System.Windows.Forms.dll` and `System.Drawing.dll` do not provide direct support for many additional technologies required to build a feature-rich desktop application. To illustrate, consider the toolkits you need to build a complete GUI using Windows Forms.

Desired Functionality	Technology
Building windows with controls	Windows Forms
2D graphics support	GDI+ ( <code>System.Drawing.dll</code> )
3D graphics support	DIRECTX APIs
Support for streaming video	Windows Media Player APIs
Support for flow-style documents	Programmatic manipulation of PDF files

Figure 1

As illustrated in Figure 1, a Windows Forms developer must pull in types from a number of unrelated APIs and object models. It is the crux why WPF is published.

WPF (introduced with .NET 3.0) was purposely created to merge these previously unrelated programming tasks into a single unified object model.

Desired Functionality	Technology
Building forms with controls	WPF
2D graphics support	WPF
3D graphics support	WPF
Support for streaming video	WPF
Support for flow-style documents	WPF

Figure 2

The obvious benefit here is that .NET programmers now have a single, symmetrical API for all common GUI desktop programming needs. That is better and even beneficial to learn as a beginner. What's more, WPF provides a new design pattern, known as MVVM(Model-View-ViewModel), which successfully separates the work of graphic designer and background code programmer. Introducing the XAML, a XML-based grammar to describe the outlook, a graphic designer even does not need to know anything about the .NET framework or C#. All he needs to do is just designing the performance on the canvas on Visual Studio Blend, then save his work into a .xaml file and hand it to the programmer. It also releases the burden of the programmers, because they only need to think about the logic, the data structure and algorithms. Actually, WPF is simplifying the complex UI programming.

#### 4.2.2.2 Using C#

Microsoft's .NET platform (and the related C# programming language) was formally introduced circa 2002 and have quickly become a mainstay of modern-day software development. The grammar of C# is largely based on C/C++ and Java, making programmers more easily to learn this

# INTERESTING

language from the experience of what they have learned. C# also has a number of features traditionally found in various functional languages such as lambda expressions and anonymous types.

Well, the main reason is that C# applies much easier way to make applications on Windows. Full featured APIs make it possible to design complex applications without knowing much of the frame of Windows. What's more, the completed documents on MSDN satisfy whatever we want to implement a specific function into our application, such as the serial port to embed serial communication (using System.IO.Ports. SerialPort object), the audio helper (using System.Windows.Media namespace).

From my perspective, C# is practically useful language, as it adds whatever you want into its features. It's hybrids from various kinds of languages and become strong and flexible in many ways. Additionally, Microsoft's IDE, Visual Studio, is well designed to satisfy the features of C#, making it the most efficient language on software design of Windows platform. (In fact, it is the IL makes it such easy to build applications of Windows family, though)

.NET Namespace	Meaning in Life
System.Reflection System.Reflection.Emit	These namespaces define types that support runtime type discovery as well as dynamic creation of types.
System.Runtime.InteropServices	This namespace provides facilities to allow .NET types to interact with unmanaged code (e.g., C-based DLLs and COM servers) and vice versa.
System.Drawing System.Windows.Forms	These namespaces define types used to build desktop applications using .NET's original UI toolkit (Windows Forms).
System.Windows System.Windows.Controls System.Windows.Shapes	The System.Windows namespace is the root for several namespaces that represent the Windows Presentation Foundation (WPF) UI toolkit.
System.Linq System.Xml.Linq System.Data.DataSetExtensions	These namespaces define types used when programming against the LINQ API.
System.Web	This is one of many namespaces that allow you to build ASP.NET web applications.
System.ServiceModel	This is one of many namespaces used to build distributed applications using the Windows Communication Foundation API.
System.Workflow.Runtime System.Workflow.Activities	These are two of many namespaces that define types used to build "workflow-enabled" applications using the Windows Workflow Foundation API.
System.Threading System.Threading.Tasks	This namespace defines numerous types to build multithreaded applications that can distribute workloads across multiple CPUs.
System.Security	Security is an integrated aspect of the .NET universe. In the security-centric namespaces, you find numerous types dealing with permissions, cryptography, and so on.
System.Xml	The XML-centric namespaces contain numerous types used to interact with XML data.

However, the various features also extend the period on learning C# and make it even impossible to develop a complex software in a short time. The good news is that .Net documents are well edited, so that we only need to find the specific part we want and learn. All APIs can be searched on MSDN with precise definition and useful examples.

WPF uses C# in its code base, just as Windows Forms. It is possible to only program with C#, though WPF provides better solution of using C# and Xaml. Well, in fact the Xaml file will be ultimately transformed into C# code, then IL, but Xaml is better on graphical design, and as I have mentioned, separating the work of designer and programmer.

#### 4.2.2.3 Using Xaml

Just like the designation of a web page, the xml-based grammar is much more efficient in the graphical design. Xaml is likely the same as xml, but new features are added, like databinding and WPF styles.

Writing Xaml needs not to know C# and programming knowledge. All in all, it is only a language to describe the outlook of the application. Quite simple but useful.

#### 4.2.2.4 Design Style

The WPF style is another feature in UI design. In WPF, all controls, like the button, the textblock and the label have a style, the style describes what the control looks like. Thus, by designing the styles, we can customize the controls as we want.

The styles also decrease the code to decorate the controls, as a style in an independent block, we add a style be change the style member of the control in Xaml.

#### 4.2.2.5 Window and pages

The program will be in one window, known as MainWiondow, which contains several pages, such as the login page to register in the game, the game page in the user mode and the developer page in operation mode.

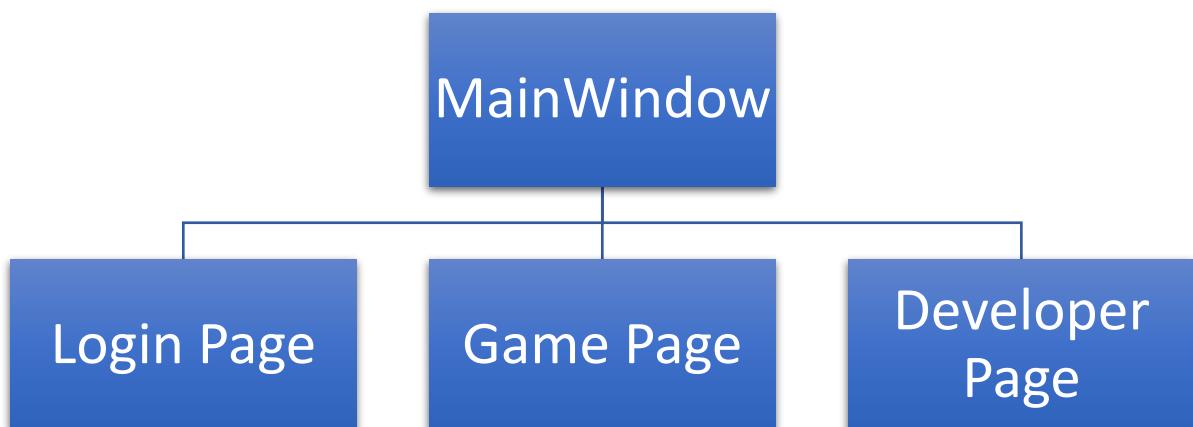


Figure 4-4 4.2.2.5 Window and pages

#### 4.2.2.6 Design pattern --- MVVM

As WPF separate the GUI design into two parts, it also invites a new design pattern. The MVVM defines the outlook as the View, which means it does not have any logics about the core functions of the application, it is just a view to the user. Model is the core functions, which are independent of the view, so that it can only concentrate on the functions and can be the base of any view. The thing that link the Model and View into one entity is the ViewModel, it is a specific design of the view and model. Just like a tube, which can receive the message from the UI and send it to the core, then get the message back to the UI.

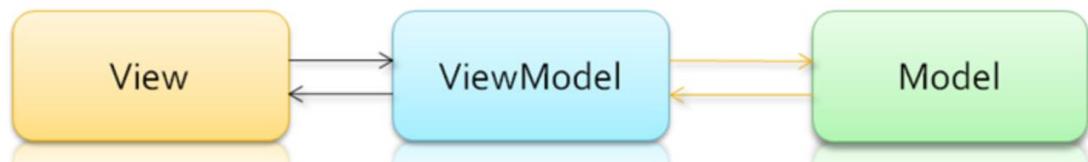


Figure 4-5 Design pattern --- MVVM

A quite important part is the design of the.viewmodel. The main technic is the databinding between the view and.viewmodel, and we will use lots of bindings in our application.

# INTERESTING

## 4.2.2.7 Features of the application:

Our application system has the following outstanding characteristics and advantages

- Two modes
- Audio helper
- Switch of languages
- Simple database
- Display the progress of the game
- Communicate with mbed

## 4.2.2.8 Operation Procedure

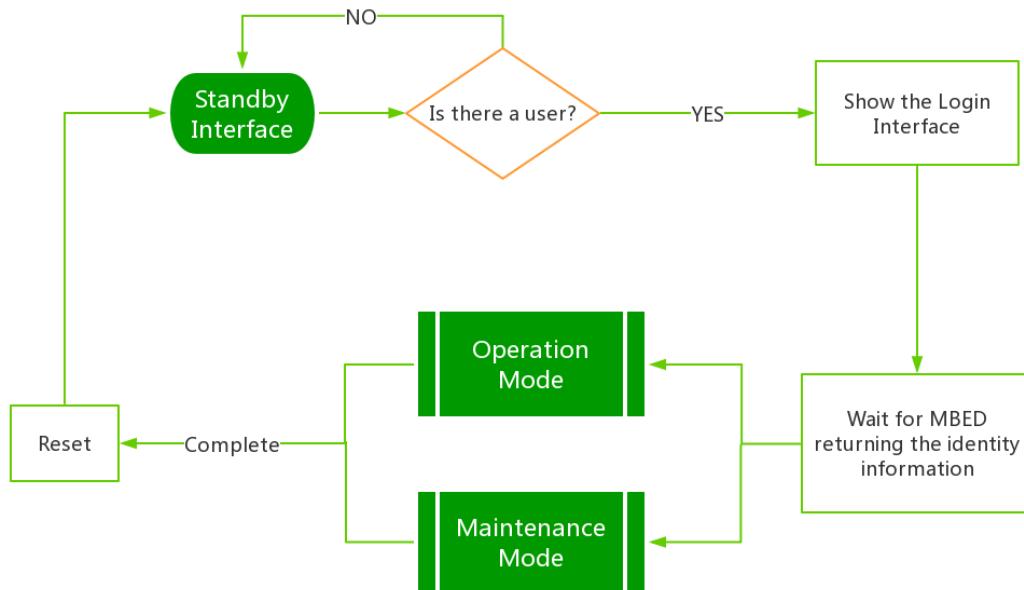


Figure 4-6 Operation Procedure

## 4.2.3 Maintenance Mode

### 4.2.3.1 Maintenance Mode Functions

#### Basic Functions:

- The maintainer enters the instruction in this mode and the PC sends the instruction to MBED to execute.
- Receive the return value from MBED and display the return value in a command line window.

Maintenance mode is produced to detect whether the components are working properly or not. For example, when the instruction of testing motor is input by maintenance stuffs, the program transforms PC language into MBED language by the protocol between PC and MBED and send the instruction to MBED to execute it. It is worth mentioning that one instruction cannot be sent until the last process is finished.

Besides, the maintenance mode can also receive the feedback from sensors. In maintenance mode, when the maintenance staff enters the instruction of the test sensor, the sensor will read the data once and return the current data to the PC side for display. Maintenance personnel can determine whether there is a problem with the sensor device or the connection line by observing whether the return value is within the expected range.

As for the extensions of the maintenance mode, it can also read the identity information and other statistics in the database, which is designed for the companies that buy our equipment to analyse customer information and make adjustment to their sales and management.

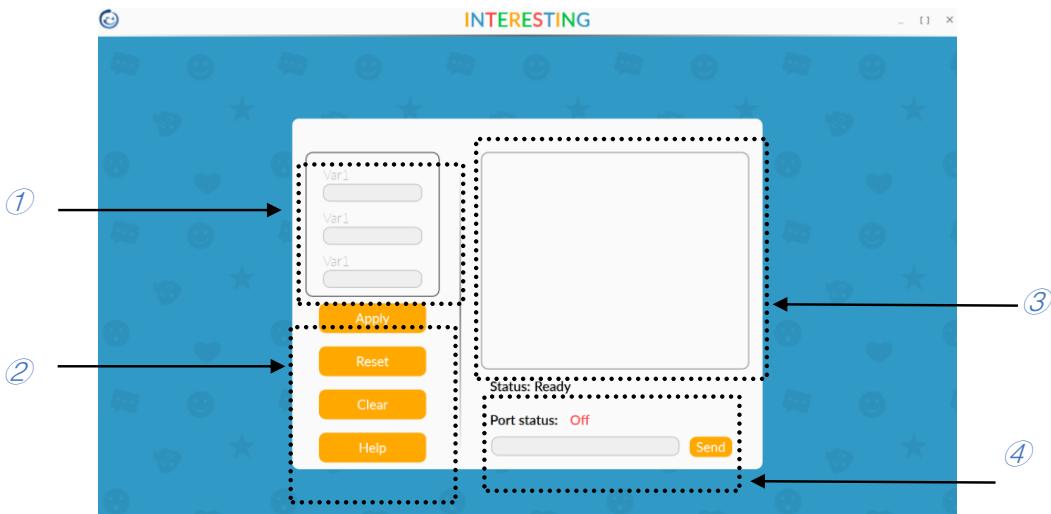


Figure 4-7 Interface

**① Parameter input area:** When debugging instructions are executed, maintenance personnel enter relevant parameters there. For example, maintenance personnel can input the servo rotation angle here when debugging servos.

**② Instruction operation area:** Maintenance personnel can complete special debugging instructions through the buttons in this area of the motor.

(**Apply:** Enter the parameters. **Reset:** Forces the device to Reset. **Clear:** Clear the contents of the command line window. **Help:** view the Help menu.)

**③ Command-line window area:** The results returned from MBED will be displayed in this area each time you enter debugging instructions.

**④ Command sending area:** The area where the maintenance staff enters debugging instructions. (see the command menu for details)

#### 4.2.3.2 Operation Procedure

In maintenance mode, the user should input debugging instructions first, and then the PC will send instructions to MBED. MBED will return the data read by the debugged device after executing the instruction, and the PC will accept the data and display the data to the command line window for maintenance personnel to view. This mode repeats the loop until the user exits.

If the user enters an invalid instruction, the system will prompt the user for an error and wait for the user to input another instruction. Users can click the Help button on the interface or enter "?" to view the command menu which shows all the valid instructions.

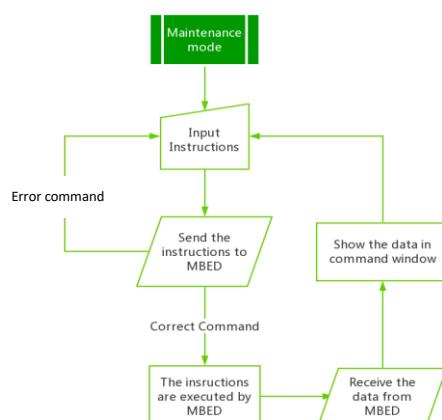


Figure 4-8 maintenance mode

# INTERESTING

## 4.2.3.3 Command Menu and the Internal Operating Principle

The debugging command that the maintainer can input is shown below:

Command	Function	Internal Operating Principle
?	View the help menu	The whole valid command list will be displayed on the command line window.
r	Test the color sensor	The color sensor reads the data once and the current data is displayed in the command line window.
s	Test the distance sensor	The distance sensor reads the data once and the current data is displayed in the command line window.
p	Lift hatch	The motor will retract the wires to pull up the hatch cover
d	Test the ultrasonic sensor	The ultrasonic sensor reads the data once and the current data is displayed in the command line window.
a	Test the reward system	MBED will control servo to open the door of the reward cabin and send rewards to the user.
b	Test photoelectric sensor	The opto-optical contact sensor will read the data four times driven by servo to simulate the function of reading the identity serial number.
t	Test conveyor belt function	The motor drives the conveyor belt to transport the recovery tank to the designated position.
q	Exit	Exit maintenance mode.

## 4.2.4 Operation Mode

### 4.2.4.1 Operation Mode Functions

Operation Mode is the main mode used by users. In this mode, the device will coordinate the operation of various parts under the MBED system, and guide users to play games, so as to achieve the purpose of recreation.

At the same time, our device also has many important functions to help users get a better game experience. For example: language switching system can guide users of different languages to play games; Database and ranking system can enable users to know their ranking in real time and stimulate their desire to compete. Voice help system can more clearly guide users to launch the game; The animation effect of the interface makes the interface we designed more pleasing to the eye.

#### 4.2.4.2 Interface

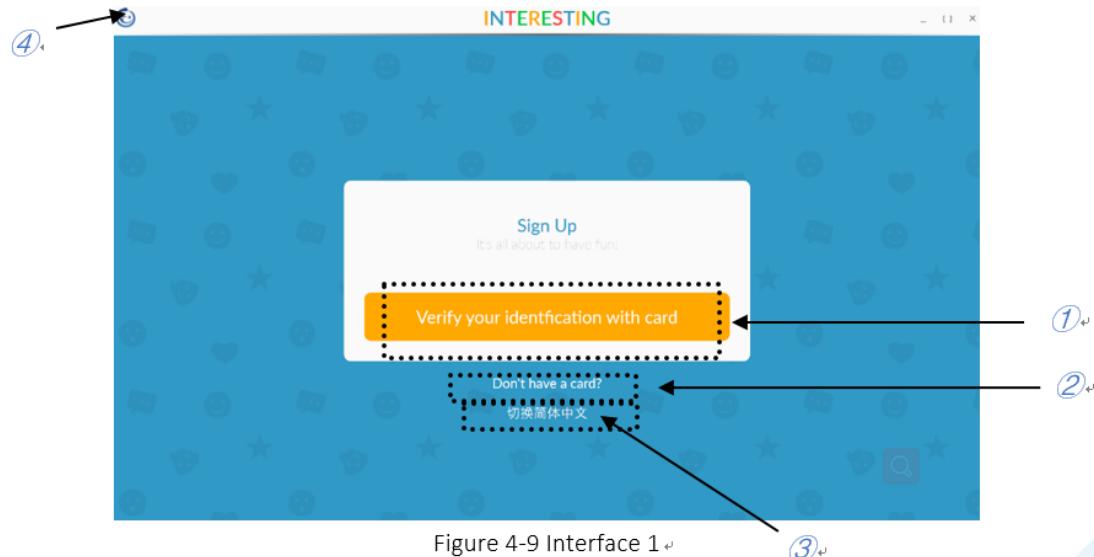


Figure 4-9 Interface 1

**1. Identification:** After clicking the button, the user can start the internal identification system of the device to identify the serial number on the identity card, so as to read the user's identity.

**2. Register:** Users who do not have an ID card can click this button to register the identity serial number at the official website.

**3. Language Switching:** The device's operating system is available in both Chinese and English, and users can click the button to switch to their own language.

**4. Ranking list:** Click this button, the user can view the player's ranking

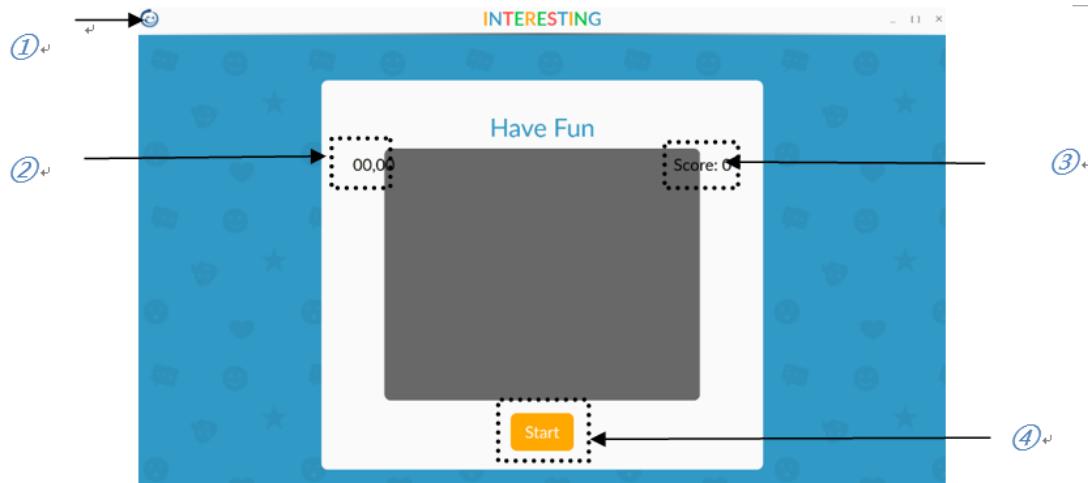


Figure 4-10 Interface2

**1. Ranking list:** Click here to view the ranking list.

**2. Time:** This area displays user times.

**3. Score:** This area displays user scores.

**4. Start/End:** Click this button to start/end the game

# INTERESTING

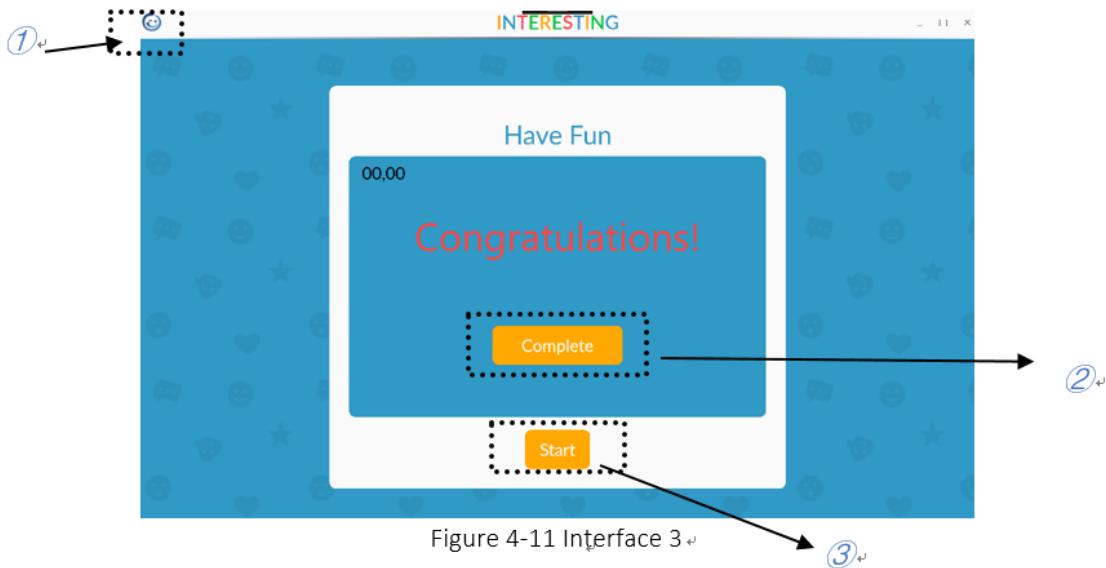


Figure 4-11 Interface 3

1. **Ranking list:** Click here to view the ranking list.
2. **Complete:** Complete and exit this game.
3. **Start/End:** The key in this interface fails and becomes grey

#### 4.2.4.3 Operation Procedure

When the user enters operation mode, the PC will first display the game interface. The user needs to press the "start" button to start the game.

During the game, the PC can receive the warning signal from MBED (the user is not in the valid range). Once the PC receives an alert signal, it will start the timer and prompts the user to return to the valid range through the display prompt interface and voice prompt. If the user come back within 15 seconds, the timer goes to zero and the game continues. If the user does not return after 15 seconds, the device will assume that the user has left and automatically reset.

After the game completion, the PC waits for the game result sent by MBED. If the game results is success, the system will save the user's score into the user information table in the database and refresh the ranking list, then jump to the completion interface, and the device will automatically reset after the user clicks the "complete" button. If the game fails, the device is reset directly.

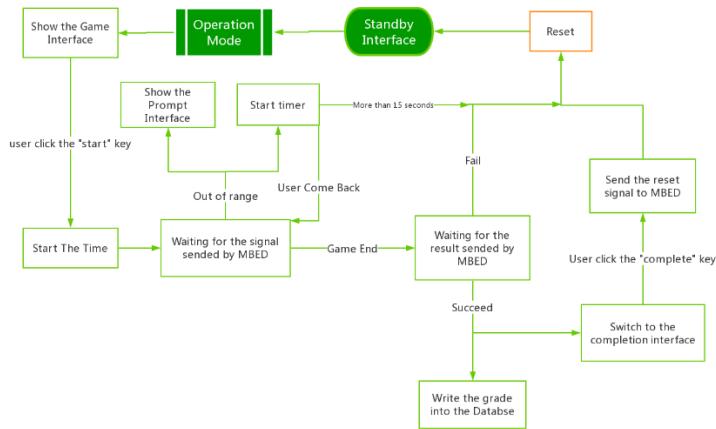
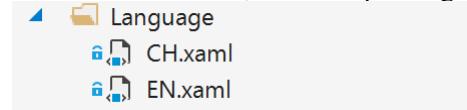


Figure 4-9 Operation Procedure

## 4.2.5 Important Functions

### 4.2.5.1 Language switching

Our system also provides language selection before the identification. As WPF has sources, which can be dynamically loaded. We create two source dictionaries to store the English and Chinese texts of one function. Then in the UI, each control which displays a text will dynamically load the resources in the dictionaries. The viewmodel decides which dictionary to load at run time, as it is depending on the user decision.



The sources in the dictionary:

```
<ResourceDictionary xmlns="http://schemas.microsoft.com/winfx/2006/xaml/presentation"
    xmlns:x="http://schemas.microsoft.com/winfx/2006/xaml"
    xmlns:local="clr-namespace:Example1"
    xmlns:system="clr-namespace:System;assembly=mscorlib">

    <system:String x:Key="kLogin">Sign Up</system:String>
    <system:String x:Key="kHavefun">It's all about to have fun!</system:String>
    <system:String x:Key="kIdentification">Verify your identification with card</system:String>
    <system:String x:Key="kNocard">Don't have a card?</system:String>
    <system:String x:Key="kLanguage">切换简体中文</system:String>
    <system:String x:Key="kApply">Apply</system:String>
    <system:String x:Key="kReset">Reset</system:String>
    <system:String x:Key="kClear">Clear</system:String>
    <system:String x:Key="kHelp">Help</system:String>
    <system:String x:Key="kSend">Send</system:String>
    <system:String x:Key="kStatus">Status:</system:String>
    <system:String x:Key="kStart">Start</system:String>
    <system:String x:Key="kWorking">Working</system:String>
    <system:String x:Key="kPortStatus">Port status:</system:String>
    <system:String x:Key="kPortOn">On</system:String>
    <system:String x:Key="kPortOff">Off</system:String>
    <system:String x:Key="kVerifying">Verifying</system:String>
    <system:String x:Key="kOutOfRange">You are out of Range!</system:String>

</ResourceDictionary>
```

Implement the dynamical load behaviour in Xaml:

```
Content="{DynamicResource kIdentification}"
```

The resource name is the name of the key defined in the dictionary.

To change the current language, add a button to the page and bind its command to the change language function in the viewmodel:

```
1. <Button Margin="0" Content="{DynamicResource kLanguage}" Style="{StaticResource TextButton}" Command="{Binding ChangeLanguageCommand}"/>
```

The button binds to a command called “ChangeLanguageCommand” in the viewmodel.

```
1. //Change current language
2. ChangeLanguageCommand = new RelayCommand(ChangeLanguage);
```

It is actually a function in the viewmodel, and inside the function:

```
1. /// <summary>
```

# INTERESTING

```
2.     /// Change the current language
3.     /// </summary>
4.     private void ChangeLanguage()
5.     {
6.         ResourceDictionary dict = new ResourceDictionary();
7.
8.         if (_CurrentLang == "ZH")
9.         {
10.             dict.Source = new Uri($"pack://application:,,,/Language/EN.xaml");
11.             _CurrentLang = "EN";
12.         }
13.         else
14.         {
15.             dict.Source = new Uri($"pack://application:,,,/Language/CH.xaml");
16.             _CurrentLang = "ZH";
17.         }
18.
19.         Application.Current.Resources.MergedDictionaries[0] = dict;
20.     }
```

It changes a private variable, which stores the current language, and loads the specified dictionary.



Figure 4-10 the UI after the language change

#### 4.2.5.2 Database

##### Access Database:

The database saves the name, age, gender, ID, time, the rank and other information of the users.

Each time the player completes a game, the last game score of the player in the user information table is called up and compared with this game score. If the score this time is a record high, it will be saved into the user information table and refresh the ranking, otherwise it will not be saved. In our game, the shorter the time spent, the higher the score.

ID	user_name	age	gender	time	rank	Park_Type
0	Allen	20	male	20	5	Marine Park
1	Kevin	20	male		5	Zoo
10	James	24	male	24	7	Movie Theme Theme Park
11	Charles	25	male	56	14	Comic Theme Park
12	Hale	26	male	56	13	Literature Theme Park
13	Steve	27	male	34	9	History Theme Park
14	David	28	male	30	8	Science Theme Park
15	Rose	32	female	22	6	Folk culture park
2	Richard	20	male	39	10	Folk culture park
3	Daniel	21	female	50	11	History Theme Park
4	Sophia	30	female	15	4	Science Theme Park
5	Andrew	40	male	54	12	Comic Theme Park
6	Hale	60	male	89	15	Movie Theme Theme Park
7	John	80	male	15	3	Zoo
8	Lucy	22	female	14	2	Marine Park
9	Amy	23	female	0	0	Literature Theme Park

Figure 4-11 User Information Table

rank	ID	user_name	time
0	9	Amy	0
1	1	Kevin	5
10	2	Richard	39
11	3	Daniel	50
12	5	Andrew	54
13	12	Hale	56
14	11	Charles	56
15	6	Hale	89
2	8	Lucy	14
3	7	John	15
4	4	Sophia	15
5	0	Allen	20
6	15	Rose	22
7	10	James	24
8	14	David	30
9	13	Steve	34

Figure 4-12 Ranking List

##### Customer analysis

In addition to the user's basic information, each card stores the approximate age of each user. Writing the approximate age of each player into the user information table does not violate the privacy of users, but helps us complete the research on the age distribution of players. Every time a player logs into the game, the number of players of the corresponding age group will be increased by 1 in the table of statistical age distribution of players in the database. In this way, when many players have used our product, the database will record the age distribution of all players. So that we can make targeted adjustments and improvements to the product. In addition,

# INTERESTING

Besides, each id card also records the type of park where the device is located, so that we can count the number of people using our device in different types of parks. This data can be used to analyse the popularity of our equipment in different types of parks, so that we can adjust the equipment placement strategy reasonably to obtain more benefits.

The following figure shows the statistical results obtained after 100 experiments:

park_type	user_volume
Comic Theme Park	10
Folk culture park	0
History Theme Park	1
Literature Theme Park	3
Marine Park	22
Movie Theme Park	22
Science Theme Park	41
Zoo	1

Figure 4-13 Statistics on the number of players in different types of parks

age_group	age	user_volume
Children	Under the age of 12	24
Middle-aged	30-50	32
Teenagers	12-18	13
The elderly	Older than 50	16
Youth	18-30	15

Figure 4-14 Statistics on the number of players of different ages

In order to achieve this function, the process of the program is as shown below: after a user carries out identification, the age group and park type of the user are first read, then the age group and the original number of users of the park type are read from the database, and finally one user is added on the basis of the number of users and then written into the database. The flow chart is shown below:

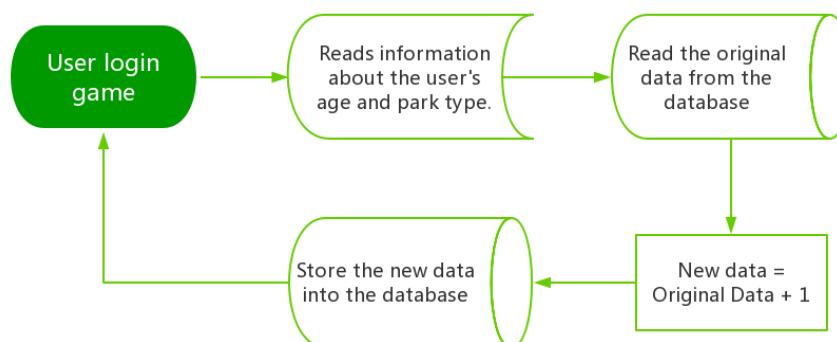


Figure 4-15 The process of customer analysis

The key code in the program is shown below:

Read user information

```
/// <summary>
/// Extract user information: display user information, and count the number of parks and ages
/// </summary>
public void select_user(string ID)
{
    string sql = "select age, Park_Type from user_information where ID= '" + ID + "' ";
    OleDbDataAdapter dbDataAdapter = new OleDbDataAdapter(sql, oleDb);
    DataTable dt = new DataTable();
    dbDataAdapter.Fill(dt);
    int age = Convert.ToInt16(dt.Rows[0].ItemArray[0]);
    string age_group;
    if (age < 12)
        age_group = "Children";
    else if (12 < age && age < 18)
        age_group = "Teenagers";
    else if (18 < age && age < 30)
        age_group = "Youth";
    else if (30 < age && age < 50)
        age_group = "Middle-aged";
    else
        age_group = "The elderly";
    string park_type = dt.Rows[0].ItemArray[1].ToString();
    Change_park(park_type);
    Change_age(age_group);
}
```

Write into the table of statistics on the number of players in different types of parks:

```
/// <summary>
/// Count the number of customers in various parks
/// </summary>
public bool Change_park(string park_type)
{
    string sql = "select user_volume from park_survey where park_type= '" + park_type + "' ";
    OleDbDataAdapter dbDataAdapter = new OleDbDataAdapter(sql, oleDb);
    DataTable dt = new DataTable();
    dbDataAdapter.Fill(dt);
    int user_volume = Convert.ToInt16(dt.Rows[0].ItemArray[0]) + 1;
    string sql1 = "update park_survey set user_volume=' " + user_volume + " ' where park_type = '" + park_type + "' ";
    OleDbCommand oleDbCommand = new OleDbCommand(sql1, oleDb);
    int j = oleDbCommand.ExecuteNonQuery();
    return j > 0;
}
```

Write into the table of statistics on the number of players of different ages:

```
/// <summary>
/// Count the number of customers of all ages
/// </summary>
public bool Change_age(string age_group)
{
    string sql = "select user_volume from age_survey where age_group= '" + age_group + "' ";
    OleDbDataAdapter dbDataAdapter = new OleDbDataAdapter(sql, oleDb);
    DataTable dt = new DataTable();
    dbDataAdapter.Fill(dt);
    int user_volume = Convert.ToInt16(dt.Rows[0].ItemArray[0]) + 1;
    string sql1 = "update age_survey set user_volume=' " + user_volume + " ' where age_group= '" + age_group + "' ";
    OleDbCommand oleDbCommand = new OleDbCommand(sql1, oleDb);
    int j = oleDbCommand.ExecuteNonQuery();
    return j > 0;
}
```

### Interface:

Any time the user wants to see the rank list, no matter whether he has played the game, he can open the rank list by clicking the application icon on the left top of the UI, the blue smile face. The user who clicks the button before the game will see the rank of former players, and after the game, he will find the rank changed because of his new data. That is to say, the rank list will be updated every time he clicks the icon.

The way we achieve it is to design a database control, which has the APIs on push and clone of the

# INTERESTING

data from the database. The rank will be executed after the clone and the rank list will be downloaded into the table “ranking”.

The database is the classic database type, the Microsoft Access database. We first establish the database and its class in the control. The class has the basic functions on connection to the database, it will store the data we want in its public lists, then we read the list and allocate the data in the legend list in UI.

The public members:

```
#region public members

public List<string> IDs = new List<string>();
public List<string> Names = new List<string>();
public List<int> Ages = new List<int>();
public List<string> Genders = new List<string>();
public List<int> Times = new List<int>();
public List<string> Ranks = new List<string>();

#endregion
```

The constructor of this class:

```
/// <summary>
/// Constructor
/// </summary>
public Access()
{
    oleDb.Open();
    Findrank();
}
```

The constructor opens the connection and ranks the data each time the object is instantiated. Then we can write the database control, which contains the instance of Access object.

```
class DataBaseControl
{
    #region public members

    public Access access { get; set; }

    #endregion

    #region ctor

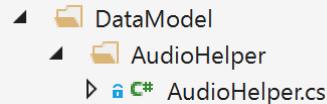
    public DataBaseControl()
    {
        access = new Access();
    }

    #endregion
}
```

#### 4.2.5.3 Audio Helper

In each step, the audio helper will help the user continue the process to complete the game. It also has two languages, English and Chinese. The implementation is simple, which records the sound in mp3 and loads the specific sound at each progress.

The control in the UI to play the sound is a MediaElement in WPF. We get the instance in the viewmodel, then operate on it using instance methods.



The constructor finds the MediaElement in the page and changes the source directory to where the mp3 file stores:

```
1. #region Constructor
2. /// <summary>
3. /// Get the mediaelement of the window
4. /// </summary>
5. public AudioHelper()
6. {
7.     Helper = (MediaElement)Application.Current.MainWindow.FindName("audiohelper");
8.     CurrentDirectory = AppDomain.CurrentDomain.BaseDirectory + "Audios\\";
9. }
10. #endregion
```

One instance of audio play:

```
1. public static void PleaseInsertCard()
2. {
3.     if (mLang == "EN")
4.     {
5.         var uri = CurrentDirectory + OnVerifying;
6.         Helper.Source = new Uri(uri, UriKind.Absolute);
7.     }
8.     else if(mLang == "ZH")
9.     {
10.        var uri = CurrentDirectory + OnVerifying;
11.        Helper.Source = new Uri(uri, UriKind.Absolute);
12.    }
13.    while (!Helper.IsLoaded);
14.    Helper.Play();
15. }
```

# INTERESTING

## 4.2.6. Innovative Functions

### 4.2.6.1 Ranking list and the slide animation

The control contains the instance, then each time we read the data by using the syntax: DataBaseControl.access.[type]. The list contains the rank is designed in the main window, for we need to be able to see the list no matter which page we are in.

We make a new control name SideMenuControl:

```
<!--The content of the main window-->
<Border Background="{StaticResource SkyBlueBrush}">
    <Grid>
        <Grid.ColumnDefinitions>
            <ColumnDefinition Width="Auto"/>
            <ColumnDefinition Width="*"/>
        </Grid.ColumnDefinitions>
        <Grid.Background>
            <ImageBrush TileMode="Tile" ImageSource="images/background/emoji->
        </Grid.Background>
        <MediaElement Grid.Column="1" Visibility="Collapsed" x:Name="audiohel
        <Frame Grid.Column="1" x:Name="frame" NavigationUIVisibility="Hidden"
              

        <local:SideMenuControl x:Name="sidemenu"
            local:LegendListAnimationAttachedProperty.Value ="{Binding
                ApplicationViewModel.SideMenuVisible,
                Source={x:Static local:ViewModelLocator.Instance}}}>
            <local:ChatListControl/>
        </local:SideMenuControl>
    </Grid>
</Border>
```

From the code below, we design the content of the main window in two columns, the left column contains the list, and the right column contains the page (the Frame element). Because we need to control it using the button. We create the attached property so that we can access the value in all the controls.

We first write the base abstract class, BaseAttachedProperty.

```
/// <summary>
/// A base attached property to replace the vanilla WPF attached property
/// </summary>
/// <typeparam name="Parent">The parent class to be the attached property</typeparam>
/// <typeparam name="Property">The type of this attached property</typeparam>
public abstract class BaseAttachedProperty<Parent, Property>...
```

Then we write the functional base class of the animation element attached property to make the menu appears and disappears like sliding.

---

```
/// Base attached property for all element animations
/// </summary>
/// <typeparam name="Parent"></typeparam>
public abstract class BaseAnimationAttachedProperty<Parent> : BaseAttachedProperty<Parent, bool>
    where Parent : BaseAttachedProperty<Parent, bool>, new()
{

    public bool FirstLoad { get; set; } = true;

    public override void OnValueUpdated(DependencyObject sender, object value)
    {
        if (!(sender is FrameworkElement element))
            return;

        if (sender.GetValue(ValueProperty) == value && !FirstLoad)
            return;

        if (FirstLoad)
        {
            RoutedEventHandler onLoad = null;
            onLoad = (s, e) =>
            {
                (s as FrameworkElement).Loaded -= onLoad;
                OnAnimation(element, (bool)value, FirstLoad);
                FirstLoad = false;
            };
            element.Loaded += onLoad;
        }
        else
        {
            OnAnimation(element, (bool)value, FirstLoad);
        }
    }
}
```

The attached property is a bool type as we have acquired, so that we can bind to it to control the behaviour of sliding in and sliding out.

After the preparation, we can write our SideMenuAttached property which is based on this abstract base class.

```
/// <summary>
/// The attached property on legend list, which controls the slide in and slide out
/// </summary>
public class LegendListAnimationAttachedProperty : BaseAnimationAttachedProperty<LegendListAnimationAttachedProperty>
{
    public override async void OnAnimation(FrameworkElement element, bool value, bool FirstLoad)
    {
        if(value)
        {
            await element.SlideAndFadeInFromLeftAsync(FirstLoad ? 0f : 0.9f);
        }
        else
        {
            await element.SlideAndFadeOutToLeftAsync(FirstLoad ? 0f : 0.9f);
        }
    }
}
```

Override the animation () method to add the animation, then each time we change the value of this property, the right animation will be displayed.

This property is banded to the SideMenuVisible property in the ApplicationViewModel.

# INTERESTING

```
public class ApplicationViewModel : BaseViewModel
{
    /// <summary>
    /// True if the side menu should be shown
    /// </summary>
    public bool SideMenuVisible { get; set; } = false;
}
```

The left thing we need to do is to bind the command, which changes the SideMenuVisible property. The SideMenuCommand:

```
SideMenuCommand = new RelayCommand(() =>
{
    IoC.Get<ApplicationViewModel>().SideMenuVisible ^= true;
    var SideMenu = Application.Current.MainWindow.FindName("sidemenu") as SideMenuControl;
    Application.Current.Dispatcher.Invoke(() => SideMenu.DataContext = new ChatListViewModel());
});
```

We XOR the value to true, so that each time we click the button, the value will be changed to its compliant. The following code does this: Find the sidemenu element in the main window, then change its datacontext to a new viewmodel, so that we update the data. In fact, we can write the command to be a parameterized command and send the element itself, then we do not need to find it in the function.

The way we access the SideMenuVisible is using a new design pattern, the IoC(Inversion of Control). Then we apply the dependency injection:

```
public static class IoC
{
    public static IKernel Kernel { get; private set; } = new StandardKernel();

    public static void Setup()
    {
        BindViewModels();
    }

    private static void BindViewModels()
    {
        Kernel.Bind<ApplicationViewModel>().ToConstant(new ApplicationViewModel());
    }

    public static T Get<T>()
    {
        return Kernel.Get<T>();
    }
}
```

Also, the ViewModelLoactor:

```
public class ViewModelLocator
{
    public static ViewModelLocator Instance { get; private set; } = new ViewModelLocator();

    public static ApplicationViewModel ApplicationViewModel => IoC.Get<ApplicationViewModel>();

}
```

The IoC needs to be initiated before the window, so that it can supply the dependency to the elements. Thus, we need to edit the App class:

```

/// <summary>
/// Interaction logic for App.xaml
/// </summary>
public partial class App : Application
{
    protected override void OnStartup(StartupEventArgs e)
    {
        base.OnStartup(e);

        IoC.Setup();

        Current.MainWindow = new MainWindow();
        Current.MainWindow.Show();
    }
}

```

Then we turn to the SideMenuControl. This control is actually consisted by list items. The data of the list is the items, and the data of the items is the data clone from the database.

The xaml code of the list:

```

<Grid d:DataContext="{x:Static local:ChatListDesignModel.Instance}"
      Background="{StaticResource ForeGroundLightBrush}">
    <ScrollViewer VerticalScrollBarVisibility="Auto">
        <ItemsControl ItemsSource="{Binding Items}">
            <ItemsControl.ItemTemplate>
                <DataTemplate>
                    <local:ChatListItemControl/>
                </DataTemplate>
            </ItemsControl.ItemTemplate>
        </ItemsControl>
    </ScrollViewer>
</Grid>

```

It is rather simple to understand: we create the list and add the item as datatemplate. Each datatemplate is banded to the Items property in the listviewmodel. Then the Items is a list of the itemviewmodels.

```
public List<ChatListItemViewModel> Items { get; set; }
```

The Itemviewmodel has the property which contains the data from the database:

```

public int Age { get; set; }

public string Gender { get; set; }

public string Rank { get; set; }

public int Time { get; set; }

```

We bind to such properties in the xaml, so that each item will contain the data of one user:

# INTERESTING

```
    </Grid.ColumnDefinitions>
<Border Grid.Column="0" Padding="8">
    <Border Background="{Binding ProfilePictureRGB, Converter={local:StringToRGBValueConverter}}"
        Width="{Binding ActualHeight, RelativeSource={RelativeSource Mode=Self}}"
        CornerRadius="{Binding ActualHeight, RelativeSource={RelativeSource Mode=Self}}">
        <TextBlock Text="{Binding Initials}"
            VerticalAlignment="Center"
            HorizontalAlignment="Center"
            FontSize="{StaticResource LargeFontSize}"
            FontFamily="{StaticResource Lato-regular}"
            Foreground="{StaticResource ForeGroundLightBrush}"/>
    </Border>
</Border>
<Border Grid.Column="1" Padding="0 0 8 0">
    <StackPanel VerticalAlignment="Center">
        <TextBlock Text="{Binding Name}"
            Foreground="{StaticResource WordBlueBrush}"
            FontSize="{StaticResource SmallFontSize}"
            FontFamily="{StaticResource Lato-regular}"/>
        <TextBlock Text="{Binding Time}"
            Foreground="{StaticResource ForeGroundMainBrush}"
            FontSize="{StaticResource SmallFontSize}"
            FontFamily="{StaticResource Lato-regular}"
            TextTrimming="CharacterEllipsis"/>
    </StackPanel>
</Border>
```

This part of the xaml is cut from the design of the item, you can see the core function to bind the property to the property of the controls. Now the item looks like:



And the list contains multiple items:

9	Kasumi 0	0
1	Tom 5	1
4	Makise 5	2
8	Amy 14	3
0	Tommy 20	4
15	Summer 22	5
10	Yuki 24	6
14	Ella	7

Figure 4-16 The scroll viewer

The number in the bubble is the ID and the number on the right side is the rank. The scroll viewer could have been made more beautiful, though.

Additionally, you can see the colors are different, we made it by first create a list of strings which contains the RGB code of the colors we like. Then, randomly choose the color to the items when we create them. We write a new property for the list item called ProfilePictureRGB and binds it to the background of the bubbles.

The rgb array:

```
/// <summary>
/// The data of rgb values
/// </summary>
public string[] rgbarray = { "3099c5", "fe4503", "ffa800", "ac1680", "08f2c5", "b9e5c7", "5cff05", "bc6af4",
"ee5c9f", "c59aaa", "ef5678"};
```

The random instance should be designed as static object, so that each time we are peeking numbers from the same source.

```
#region private members

private static Random mrandom = new Random();

#endregion
```

The constructor of the item control:

```
public ChatListItemViewModel(string name, string rank, int time, string ID)
{
    Name = name;
    Rank = rank;
    Time = time;
    Initials = ID;
    ProfilePictureRGB = rgbarray[mrandom.Next(rgbarray.Length)];
}
```

In the game, we can check the rank list:

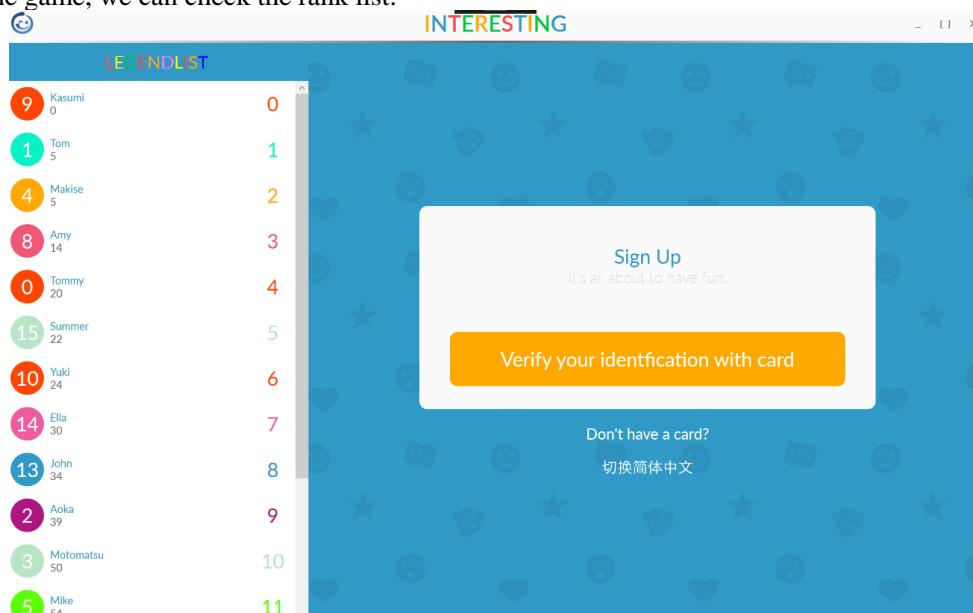


Figure 4-17 Rank List in The Game

# INTERESTING

#### **4.2.6.2 Display the progress of the game:**

The UI will be controlled according to the message sent from mbed, and the mbed sends the message of the current game status, thus it is possible to display the status by decoding the message from the mbed and change the display.

In the design, all the message from mbed will be saved in two variables, and the status message is another one variable:

```
/// <summary> The message between mbed  
public string Message { set; get; }  
  
/// <summary> The message from the port  
public string ReceMessage { set; get; } = "Ready";  
  
/// <summary> The message to display on the textbox  
public string DisplayMessage { set; get; } = string.Empty;
```

The “ReceMessage” stores last message, “Message” stores all messages, and “DisplayMessage” stores the message to display.

### 4.3 Protocol

### 4.3.1 PC toMBED

The serial communication between PC and mbed uses the library in the System.IO.Ports.SerialPort assembly:

```
public class SerialPort : System.ComponentModel.Component
```

This assembly makes it rather simple to access the serial port driver. Details can be found on MSDN(<https://docs.microsoft.com/en-us/dotnet/api/system.io.ports.serialport?view=netframework-4.7.2>)

And the mostly used functions are:

**Write(String)** Writes the specified string to the serial port.

**Read(Char[], Int32, Int32)** Reads a number of characters from the [SerialPort](#) input buffer and writes them into an array of characters at a given offset.

Also, an event is also used:

**DataReceived** Indicates that data has been received through a port represented by the [SerialPort](#) object.

## The identity serial number protocol

The following is the protocol for converting the identity serial number read by MBED into an ID in PC.

```
1. namespace Example1
2. {
3.     class CommunicationCommands
4.     {
5.         /// <summary>
6.         /// the protocol between pc and mbed
7.         /// </summary>
8.         #region User ID messages
9.     }
```

```

10. public static string UserID01 { get; } = "us001";
11. public static string UserID02 { get; } = "us002";
12. public static string UserID03 { get; } = "us003";
13. public static string UserID04 { get; } = "us004";
14. public static string UserID05 { get; } = "us005";
15. public static string UserID06 { get; } = "us006";
16. public static string UserID07 { get; } = "us007";
17. public static string UserID08 { get; } = "us008";
18. public static string UserID09 { get; } = "us009";
19. public static string UserID10 { get; } = "us010";
20. public static string UserID11 { get; } = "us011";
21. public static string UserID12 { get; } = "us012";
22. public static string UserID13 { get; } = "us013";
23. public static string UserID14 { get; } = "us014";
24. public static string UserID15 { get; } = "us015";
25. public static string UserID00 { get; } = "us000";
26.
27. #endregion
28.
29. #region The # of the map
30.
31. public static string Map01 { get; } = "mp001";
32. public static string Map02 { get; } = "mp002";
33. public static string Map03 { get; } = "mp003";
34. public static string Map04 { get; } = "mp004";
35. public static string Map00 { get; } = "mp000";
36.
37. #endregion
38. }
39. }
```

### Debug commands protocol

The following is the protocol for converting instructions entered by the maintainer in the interface of the PC's maintenance mode into commands executed by MBED.

```

1. namespace Example1
2. {
3.   class DebugCommands
4.   {
5.     public static string Debug { get; } = "debug";
6.     public static string ReadDistanceSensor { get; } = "rdist";
7.     public static string ReadRGBSensor { get; } = "rrgbs";
8.     public static string MoveServo { get; } = "servo";
9.     public static string Back { get; } = "dtolo";
10.   }
11. }
```

### Game commands protocol

The following is the protocol for PC to communicate with MBED during the game.

```

1. namespace Example1
2. {
3.   /// <summary>
4.   /// The commands on game screen
5.   /// </summary>
```

# INTERESTING

```
6. class GameCommands
7. {
8.     public static string GameStart { get; } = "start";
9.     public static string GameEnd { get; } = "end_";
10.    public static string GamePause { get; } = "pause";
11.    public static string OutRange { get; } = "outrn";
12.
13.    /// <summary>
14.    /// When the user get back from out of range
15.    /// </summary>
16.    public static string GetBack { get; } = "gtbac";
17.
18.    /// <summary>
19.    /// Back to the login Page
20.    /// </summary>
21.    public static string Back { get; } = "gtolo";
22.    public static string CountUp { get; } = "conup";
23.    public static string CountDown { get; } = "condn";
24.
25.    /// <summary>
26.    /// receive when the game end and the user complete the game successfully
27.    /// </summary>
28.    public static string GameSucceed { get; } = "gmsuc";
29.
30.    /// <summary>
31.    /// receive when the game end and the user complete the game successfully
32.    /// </summary>
33.    public static string GameFailed { get; } = "gmfal";
34.
35.    /// <summary>
36.    /// When the user complete the behavior of fetching the gift, click the button and send the message
37.    /// </summary>
38.    public static string GiftFetched { get; } = "compl";
39. }
40. }
```

## Login commands protocol

The following protocol converts signals sent by the PC into commands executed byMBED during login.

```
1. namespace Example1
2. {
3.     /// <summary>
4.     /// The commands on login screen
5.     /// </summary>
6.     class LoginCommands
7.     {
8.         public static string IDVerify { get; } = "verif";
9.         public static string Login { get; } = "login";
10.        public static string DisplayScreen { get; } = "displ";
11.        public static string Root { get; } = "root_";
12.        public static string PlayAnime { get; } = "plyan";
13.        public static string PauseAnime { get; } = "pauan";
14.        public static string StopAnime { get; } = "stpan";
15.        public static string PlayMidAnime { get; } = "plmid";
```

```

16.     public static string PlayLeftAnime { get; } = "pllft";
17.     public static string PlayRightAnime { get; } = "plrit";
18.     public static string ToFirstPage { get; } = "frstp";
19. }
20. }
```

#### 4.3.2 MBED to FPGA

MBED and FPGA are connected by a 3-bit data line. MBED sends a 3-bit binary number to the FPGA, and each binary number represents the state of an FPGA. Different states output different PWM signal with different bandwidth to control the rotation of servo motor.

**Instruction Table:**

Instruction	Action
000	Turn the motor 0 degrees
001	Turn the motor 30 degrees
010	Turn the motor 45 degrees
011	Turn the motor 60 degrees
100	Turn the motor 90 degrees
101	Turn the motor 120 degrees
110	Turn the motor 150 degrees
111	Turn the motor 180 degrees

#### Debug commands protocol

The following is the protocol for converting signals sent by MBED into commands executed by FPGA.

```

1. void fpgafunc()
2. {
3.     myPort.printf("The FPGA can set the servo to 0(1), 30(2), 60(3), 90(4), 120(5), 150(6),
4.                 180(7), 45(8) degrees.\n");
5.     myPort.printf("Enter the number: defaule :(1)\n");
6.     char a = myPort.getc();
7.     switch(a)
8.     {
9.         case '1': servo.Turn_0(); break;
10.        case '2': servo.Turn_30(); break;
11.        case '3': servo.Turn_60(); break;
12.        case '4': servo.Turn_90(); break;
13.        case '5': servo.Turn_120(); break;
14.        case '6': servo.Turn_150(); break;
15.        case '7': servo.Turn_180(); break;
16.        case '8': servo.Turn_45(); break;
17.        default: servo.Turn_0(); break;
18.    }
19.    wait(0.5);
}
```

# INTERESTING

## 4.4 Strength and Weakness

### 4.4.1. Strength

#### PC

- The interface style is simple and beautiful. Added animation effects to make the interface more enjoyable.
- All elements use relative coordinates. No matter how the window size changes, the boundary of the interface will not be blank.
- We have added three small subpages to the game interface, making one game interface usable as three interfaces. This design reduces the repetition between different pages and reduces the developer's workload without affecting the functionality and aesthetics of the page.

#### MBED

- The overall structure is rigorous, there will be no process stuck on the problem.
- Clear connections between multiple processes without logic errors.
- The algorithm is optimized and perfect, which makes the whole device run smoothly.

### 4.4.2. Weakness

#### PC

- The interface design is not lively enough to impress users.
- As the core of the whole project isMBED, the function of the interface is relatively brief.

#### MBED

- We don't useMBED to achieve any innovative functions
- The communication betweenMBED and FPGA is parallel communication, which results in excessive pin consumption ofMBED.

## 5 SENSOR SYSTEM

### 5.1 Manipulation

#### 5.1.1 Color Sensor

With the functions on the website, we can obtain the values of R, G, B and C in order to define a certain color, which corresponds to red, green, blue and clear respectively. However, a particular problem is that the four values are extremely easy to be affected by the environment and distance, such as the light in the room and the material between the color sensor and the detected color. First, reading RGB values with color sensor.

```
void rgbCharge::ReadInData(int number, int r, int g, int b)
{
    _resultarray[number][0] = r;
    _resultarray[number][1] = g;
    _resultarray[number][2] = b;
}
```

Figure 5-1 Read RGB values

Then, through several times of estimation, we found that the proportion of R, G and B were similar each time and we defined the sum of these three values were 255, which meant that we could find two limited value (upper and lower limits) for R, G and B respectively.

```
// Read rgb to the port
void rgbsensorfunc()
{
    colorsensor.enablePowerAndRGBC();
    colorsensor.setIntegrationTime(100);
    wait(0.1);
    int rgb_readings[4];
    //wait(0.5);
    colorsensor.getAllColors( rgb_readings );
    //myPort.printf( "red: %d, green: %d, blue: %d, clear: %d\n", rgb_readings[0], rgb_
    int red = rgb_readings[1]*255/(rgb_readings[1]+rgb_readings[2]+rgb_readings[3]);
    int green = rgb_readings[2]*255/(rgb_readings[1]+rgb_readings[2]+rgb_readings[3]);
    int blue = rgb_readings[3]*255/(rgb_readings[1]+rgb_readings[2]+rgb_readings[3]);
    myPort.printf("red: %d, green: %d, blue: %d\n",red,green,blue);
    myPort.printf("clear: %d\n",rgb_readings[0]);
    //colorsensor.disablePowerAndRGBC();
    colorsensor2.getAllColors( rgb_readings );
    //myPort.printf( "red: %d, green: %d, blue: %d, clear: %d\n", rgb_readings[0], rgb_
    red = rgb_readings[1]*255/(rgb_readings[1]+rgb_readings[2]+rgb_readings[3]);
    green = rgb_readings[2]*255/(rgb_readings[1]+rgb_readings[2]+rgb_readings[3]);
    blue = rgb_readings[3]*255/(rgb_readings[1]+rgb_readings[2]+rgb_readings[3]);
    myPort.printf("red: %d, green: %d, blue: %d\n",red,green,blue);
    myPort.printf("clear: %d\n",rgb_readings[0]);
}
```

Figure 5-2 Read RGB values with proportions

According to this property, we can distinguish three colors (blue, orange and colourless) by allocating a range to R, G and B separately for each color and there would be no coincidence among three conditions. But unfortunately, the circumstances were too changeable and unstable to control, when the condition altered slightly, the values would change remarkably. Therefore, we needed another solution to cope with this problem.

# INTERESTING

```
int rgbCharge::FindColorBeta(int r, int g, int b)
{
    int red = r*255/(r+g+b);
    int green = g*255/(r+g+b);
    int blue = b*255/(r+g+b);
    if(red >= 89 && red <= 110 && blue >= 74 && blue <= 94)
        return 1;
    else if(red >= 40 && red <= 77 && blue <= 140 && blue >= 108)
        return 3;
    else if(red >= 60 && red <= 88 && green >= 70 && green <= 90 && blue <= 108 && blue >= 85)
        return 0;
    else
        return 2;
}
```

Figure 5-3 Define three colors

We decided to input and read the map data (R, G and B values) of color into the array before beginning the game each time.

```
void rgbCharge::GetMapData(int number, int r, int g, int b, int value, int mapnumber)
{
    if(mapnumber * 8 > _size) ArrayResize(mapnumber * 8);

    int temp = mapnumber - 1;

    _MapData[temp * 8 + number][0] = r;
    _MapData[temp * 8 + number][1] = g;
    _MapData[temp * 8 + number][2] = b;
    _MapData[temp * 8 + number][3] = value;
}
```

Figure 5-4 Read the map before the game

Finally, we compared the results we got from the color sensors with the previous map's results and defined a limited range. If the results were in the range, the user won the game. If they were not, it would lose the game.

```
bool rgbCharge::Decide(int mapnumber)
{
    int temp = mapnumber - 1;

    for(int i = 0; i != 8; i++)
    {
        if(_resultarray[i][0] >= _MapData[temp * 8 + i][0] - 9 &&
           _resultarray[i][0] <= _MapData[temp * 8 + i][0] + 9 &&
           _resultarray[i][1] >= _MapData[temp * 8 + i][1] - 9 &&
           _resultarray[i][1] <= _MapData[temp * 8 + i][1] + 9 &&
           _resultarray[i][2] >= _MapData[temp * 8 + i][2] - 9 &&
           _resultarray[i][2] <= _MapData[temp * 8 + i][2] + 9)
            continue;
        else
            return false;
    }

    return true;
}
```

Figure 5-5 Final version of color sensor

### 5.1.2 Distance Sensor

This is how we read the distance to the port. The requirement is that the user need to be at least 10cm from the machine.

```
//Read distance to the port
void distancesensorfunc()
{
    float reading;
    reading = rf;
    myPort.printf("Read %4.1f cm\n", reading);
}
```

Figure 5-6 Read distance

### 5.1.3 Ultrasonic Distance Sensors

First, we read the distance from right and left. The purpose of this sensor is to control the direction of the power sub-system (the track). If there is an obstacle in the right side, it will turn left. But the return value is unstable and sometimes -1. So we need a new method.

```
void UltalDistance()
{
    ReadUltralDistance();
    myPort.printf("The distance of left: %d\n", UltraArray[0]);
    myPort.printf("The distance of right: %d\n", UltraArray[1]);
}
```

Figure 5-7 Get distance by ultrasonic

This is a perfect solution guaranteeing the sensors' stability for the above problem. We measure the data for three times and calculate the mean value to use.

```
void ReadUltralDistance()
{
    int temp1[3],temp2[3];
    for(int i = 0; i != 3; i++)
    {
        int temp = UltrasonicLeft.distance(CM);
        temp = temp == -1 ? 50: temp;
        temp1[i] = temp;
        temp = UltrasonicRight.distance(CM);
        temp = temp == -1 ? 50: temp;
        temp2[i] = temp;
    }

    UltraArray[0] = GetMean(temp1);
    UltraArray[1] = GetMean(temp2);
}
```

Figure 5-8 Read ultrasonic data with sampling

# INTERESTING

## 5.1.4 Infrared Sensor

When the machine detects a human, it will return 1 and go on to the next step. If it detects an obstacle, it will return 0 and stops the operation.

```
while(1)
{
    if(ReadXred())
    {
        //myPort.printf("human detect\n");
        humandetect = 1;
    }

    wait(0.5);
}
```

Figure 5-9 Read 5.1.4 Read Infrared Sensor

## 5.1.5 Photoelectric Gate Sensor

A gap is corresponding to 0 and unnotched is corresponding to 1. 1111 is maintenance workers and the remainders are the users.

```
// The card has four pieces to read
// The degrees on the names are not the real
// degrees for servo0, the real degrees have
// been tested and changed on fpga
// Thus we only change the "state" of the FSM
void GetCardInformation()
{
    SelectServo(0);
    servo.Turn_0();
    wait(0.9);
    verifyarray[0] = ReadGate();
    wait(0.1);
    servo.Turn_30();
    wait(0.5);
    verifyarray[1] = ReadGate();
    wait(0.2);
    servo.Turn_60();
    wait(0.5);
    verifyarray[2] = ReadGate();
    wait(0.2);
    servo.Turn_90();
    wait(0.5);
    verifyarray[3] = ReadGate();
    wait(0.2);
}
```

Figure 5-10 Read Photoelectric Gate Sensor

## 5.1.6 Reed Pipe Sensor

If the cover is closed, the value is 1. If the cover is opened, the value is 0. When the user finishes the game, he will close the cover and once the signal is detected, the machine will continue automatically.

```

//Main
int main() {
    myPort.attach(&CommandReceive);
    Magnetss.rise(&GameComplete);
    //TouchUser();
    Initrgbs();
    __disable_irq();

    /* Game logics */
    char a = myPort.getc();
    GameLogicControl();
    /* **** */

```

Figure 5-11 Read 5.1.6 Read Reed Pipe Sensor

## 5.2 Tests and Data

### 5.2.1 Color Sensor

We had discovered that the RGB values change evidently when the distance and the environment altered. So we decided to use control variable method to find the relationship between the environment and the RGB values of color sensors. The distance was not necessary because we would not change the distance during the operation process. Only environment was influential. We choose three typical conditions due to the light intensity without changing the distance.

Indoor		R	G	B
Orange	1	98	70	87
	2	96	71	88
	3	97	73	85
Blue	1	68	57	130
	2	69	58	128
	3	67	53	125
Colourless	1	71	104	80
	2	73	103	79
	3	74	105	77

Table 5-1 RGB values for three colors indoor

Outdoor		R	G	B
Orange	1	110	55	90
	2	111	56	88
	3	113	56	86
Blue	1	61	64	130
	2	59	65	131
	3	61	67	127
Colourless	1	90	100	65
	2	85	103	67
	3	93	99	63

Table 5-2 RGB values for three colors outdoor

# INTERESTING

Dark		R	G	B
Orange	1	130	60	65
	2	129	59	67
	3	133	58	64
Blue	1	83	30	142
	2	81	29	145
	3	84	31	140
Colourless	1	100	100	55
	2	100	102	53
	3	98	101	56

Table 5-3 RGB values for three colors indoor

## 5.2.2 Distance Sensor

Through our nearly hundreds of times of measurements, we conclude some characteristics about distance sensors.

1. It is relatively correct when there is a person within the range.
2. If there is nothing within the range, the return value will be an arbitrary value between 1cm-2cm instead of 0.
3. The maximum value is 25.5cm, which is the upper limit.

## 5.2.3 Ultrasonic Distance Sensors

It has poor detection effect on human body. The return value is -1 when the person is not within the estimation range. The upper limit (maximum value) is 500cm+.

## 5.2.4 Infrared Sensor

When the machine detects a human, it will return 1 and go on to the next step. If it detects an obstacle, it will return 0 and stops the operation.

## 5.2.5 Photoelectric Gate Sensor

A gap is corresponding to 0 and the unnotched is corresponding to 1.

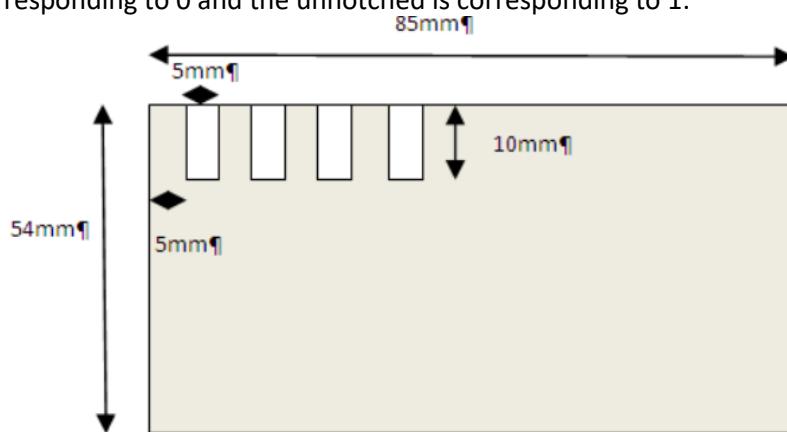


Figure 5-12 The required card

## 5.2.6 Reed Pipe Sensor

If the cover is closed, the value is 1. If the cover is opened, the value is 0.

### 5.3 Circuit Diagrams and Layout

The pin allocation of the sensor system is as following:

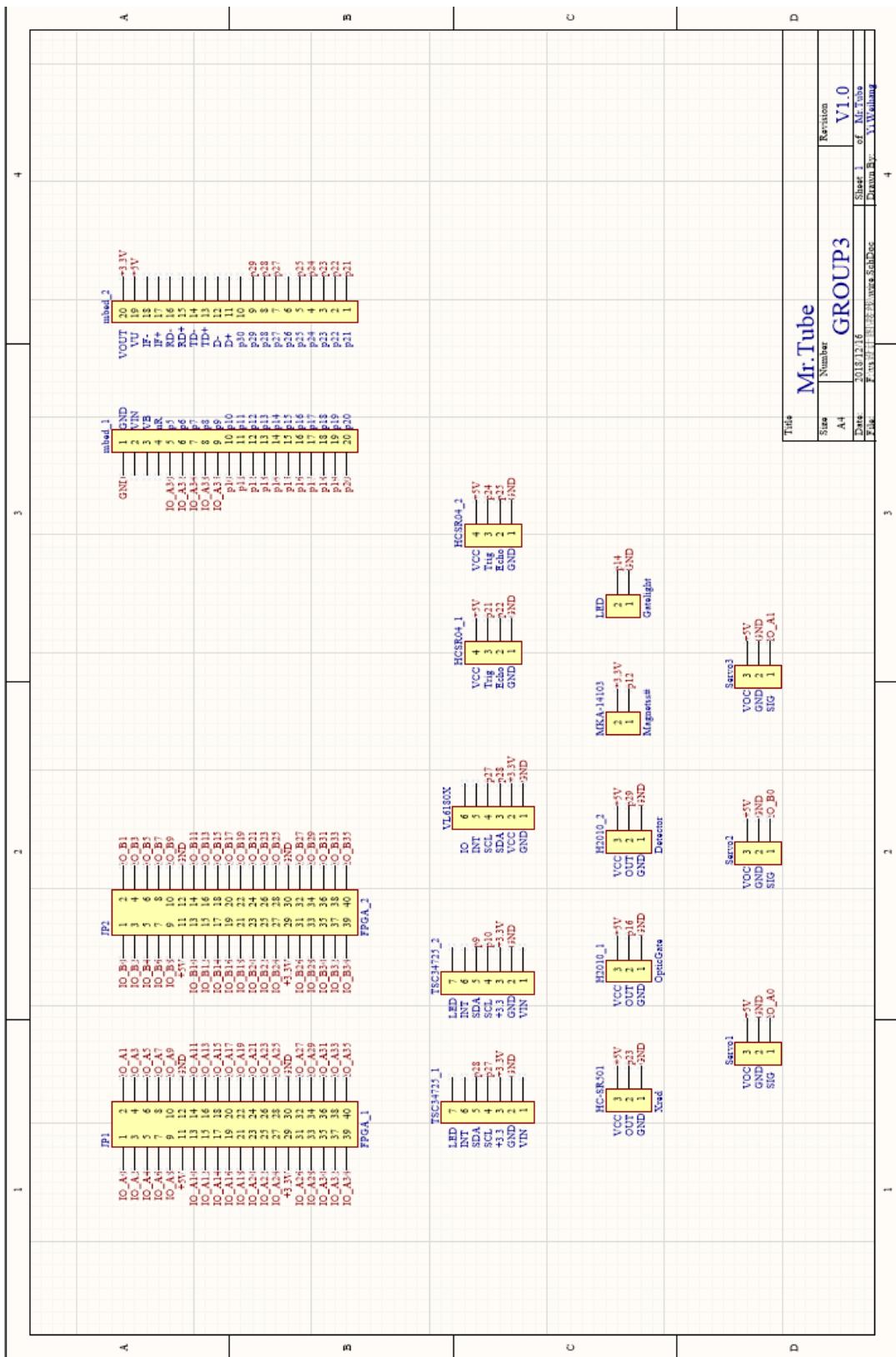


Figure 5-13 The pin allocation of the sensor system

## 6 MARKETING AND PUBLIC RELATION

### 6.1. Overview

The requirement for the product is that it should be an alien robot which can entertain the customers while they are waiting in line. To meet this requirement, we did a detailed research to find out the market demand and trends which are necessary and of vital importance to our product design. Accurate and precise market positioning makes the functions of our product meet the customers' needs with extremely low cost because we have the courage to be better than many designs which look very beautiful and fancy but do litter work in particular situations.

In the later stage of the project, while the product was almost initially manufactured, packaging and propagating were as important as processing it. Only when the product attracts customers at the first sight can its functions and technological innovations be known and acknowledged by more people. Finally, we chose concise and futuristic style as the keynote of our product and we made our posters, propaganda film and website in that way.

### 6.2. Market assessment

#### 6.2.1. Demand analysis

##### (1). The effect of too long queuing time

Queuing is something that everyone is tired of, and the long queuing time will arouse tourists' annoyance, which will greatly affect the tourist experience. Some customers will even stop playing due to their fear of queuing, which greatly reduces the number of repeat visitors and affects the profit of the amusement park.

##### (2). Defects in current queuing system

Although many amusement parks have introduced queueing systems that allow customers to know how much time will be spent for waiting, this prevents uneven distribution of people -- some projects are too crowded, and others are idle.

But for those popular entertainment items, most tourists prefer to wait in line rather than missing out, so many tourists will still suffer from long queues. This is especially true of big, popular amusement parks.

According to statistics, more than 11 million people visited Disneyland in Shanghai during 2016-2017. The average customer queue takes more than 2.5 hours. It can be seen that the current queuing system is not very effective in solving the queuing problem.

Disney once asked visitors what they thought about the current queuing system, with many saying: "I just want to play with that project even with long time for waiting, and you should think about how can make me comfortable when I'm in queue, rather than how long I'm going to stand in line, and that number only makes me more annoyed."

Obviously, the market is in urgent need of a brand-new product, so that tourists can also be entertained in the boring queuing process.

#### 6.2.2. Advantages of physical games

Some people will say: "it's good to play with your phone while waiting in line. Now almost everyone has a smart phone, so it is not necessary to develop new entertainment devices"

Indeed, the mobile phone is definitely the dominant entertainment device in the modern era, but physical games still have a strong irreplaceable role.

First of all, playing with mobile phones for a long time can be harmful to health: medical experiments have shown that playing mobile phones for a long time can be harmful to the head, eyes, neck and skin.

Secondly, in the age of mobile games, interactive physical games can bring customers more novel game experience.

#### 6.2.3. Market saturation analysis

At present, no matter for the domestic market or the foreign market, queuing entertainment equipment is basically in a blank state.

The similar product is only one app developed by Disney, but it is still a mobile game, which has the disadvantages of mobile games and is only available in Disneyland.

As a result, the queue entertainment equipment market saturation is very low, development prospects are promising.

#### 6.2.4. Product features

<i>Features</i>	<i>Benefit</i>
<b>Entrance animation</b>	The game's background is introduced through animation to give players a strong sense of substitution.
<b>Easy to understand</b>	The game is simple and easy to understand, funny and lively, which can effectively alleviate the annoyance of tourists
<b>Stop at any time</b>	The game can be finished at any time, which will not affect tourists for queue.
<b>Physical game</b>	The main part of our product body is a physical game, which enables the tourists to get rid of their mobile phones and brings them a new game experience.

# INTERESTING

## 6.2.5. SWOT



Figure 6-1 SWOT

## 6.3. Poster and Propaganda film

### 6.3.1. Poster

We want our poster to be as concise as possible and do not provide too much information besides title and basic introduction so that it can arouse customers curiosity to know more about it. The poster is based on grey undertone and the sketch of our product is in the middle of it which draws attention first. The name of our product, “Mr. tube”, is in the upper side of the poster as well as the catch-phrase “Need for help” and the time of final presentation meeting is shown in the lower side with the company copyright mentioned in the bottom.

# MR.TUBE



***NEED FOR HELP  
MEET IN 29TH NOV. 2018***

Copyright Interesting™ company made by Group3

Figure 6-2 Poster

# INTERESTING

## 6.3.2. Propaganda film

To introduce and promote our product more efficiently and effectively, the brief propaganda film was made and divided into two parts. First of all, the name of our company will show up slowly when the film began.



Figure 6-3 Film1

Then there will be the face of the robot “Mr.Tube” and it will blink its eyes and express his feelings through the animation to attract the customers.

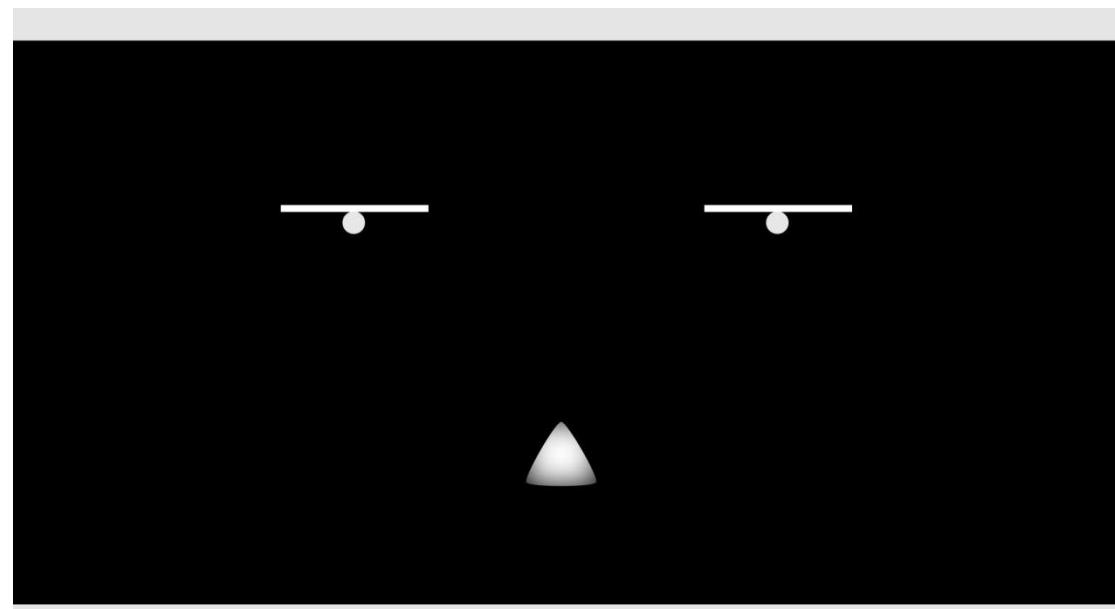


Figure 6-4 Film2

After that, the appearance and simple structure of our product will be introduced as well as showing the rules of the game. The first part of the propaganda film will end with that.



Figure 6-5 Film3

At the second part, we will introduce our team and different responsibilities of each team member.



Figure 6-6 Film4

# INTERESTING

## 6.4. Website

To advertise our company and promote our products, the website was set up by our web engineer and it continued the concise keynote of our tradition. In our website, all the information is listed clearly, and customers can easily find the content they want to read and even reach for more.

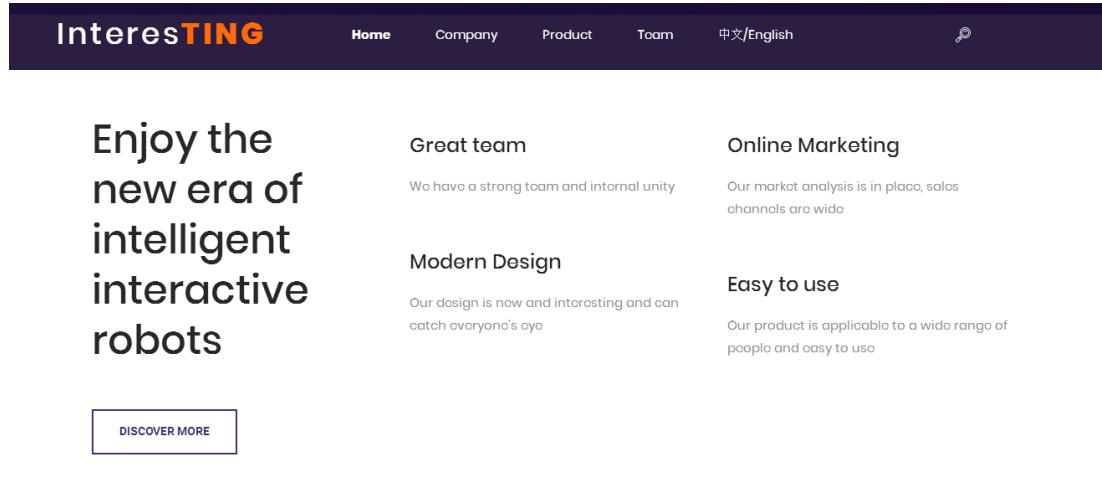


Figure 6-7 Website1

Besides the introduction of our company and team, the most important part of our websites is the detailed advantages of our products. All of them are listed on the “product” page and can be checked and compared by the customers when they are hesitating about ordering or not.

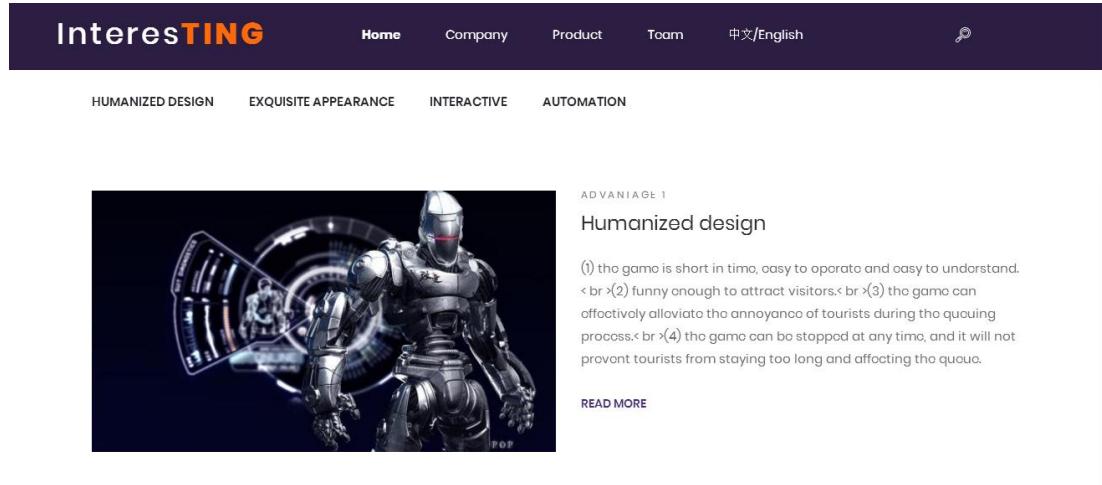


Figure 6-8 Website2

If the customers have any question, they can find the instructions or directly contact with us through the information provided in the website.

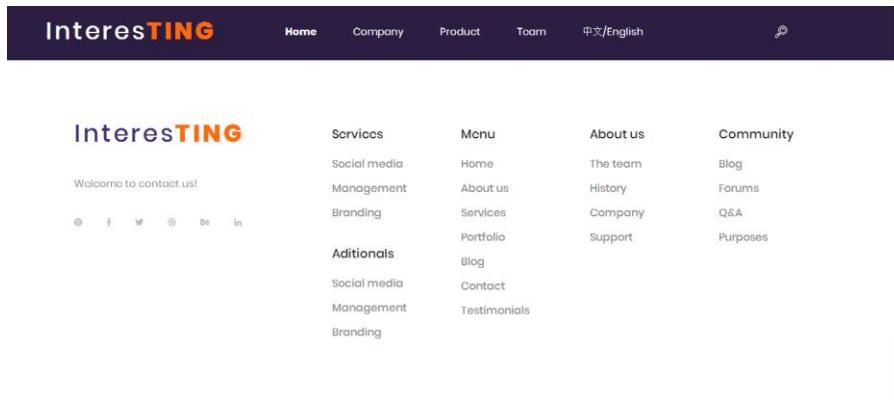


Figure 6-9 Website3

To open wider markets from overseas, we firstly set up the languages of our website to be both English and Chinese so that we can benefit the users of both English-speakers and Chinese-speakers. Other languages will be set up in further consideration.



Figure 6-10 Website4

## 6.5. Exhibition Stand

On November 29th, 2018, an exhibition was held for us to present our product and introduce all the work we had done for the project to all the guests who were invited there. We set up the display stand and decorate it with our poster as well as several lines of codes which intends to attract people by asking them for help. At that day, demonstrations of both operation mode and maintenance mode were shown and many guests played with it by themselves and had a lot of fun.



Figure 6-11 Exhibition Stand

# INTERESTING

## 6.6. Team Member Biography

### MANAGER Weihang Yi

As the project manager, my work is to lead the team and manage the process of the project. I encourage my team members to believe that we will success and make sure that my team are honest to our customers.

### CHIEF SOFTWARE ENGINEER Yifei Jing

As the company's chief software engineer. I am in charge of almost 99% of the difficult tasks about software of the company. What I pursue is the high-efficient communication and the freed ideas and creativity.

### HARDWARE ENGINEER Yuxuan Sun

As the chief mechanical designer and workshop engineer of the company, I was responsible for appearance design and basic function realization of the product together with the project manager. Meanwhile, I was also responsible for the physical production and processing of the product and assisted the secretary in a small part of copywriting related to the product.

### GUI Engineer Ziwei Deng

As the GUI engineer of the company, I am responsible for the design and beautification of software visual interface to improve user experience. I believe that the interface I designed can catch the user's eye and give users a different experience.

### PUBLIC RELATION MANAGER/SECRETARY Xiangxuan Kong

As the company's public relation manager and secretary, besides the basic clerical work, I am also responsible for assisting the project manager in organizing and managing the company's daily operation, maintaining the project progress, controlling the project materials, managing the company finance and coordinating team relations.

### WEBSITE ENGINEER/SECRETARY Ziyao Wang

As the website engineer and secretary of the company, I am responsible for recording each meeting minutes as well as the design, maintenance and operation of the website. I have absolute loyalty to my team leader and I always pay extraordinary enthusiasm to complete my every day work.

## 7. PROJECT MANAGEMENT

### 7.1. Project Plan

The first edition of our project plan was made by the project manager as soon as the goal of the project was set in the first meeting and the responsibilities of each member were clarified. It divided the whole progress of the project into many small tasks and showed the starting and finishing time of them ideally.

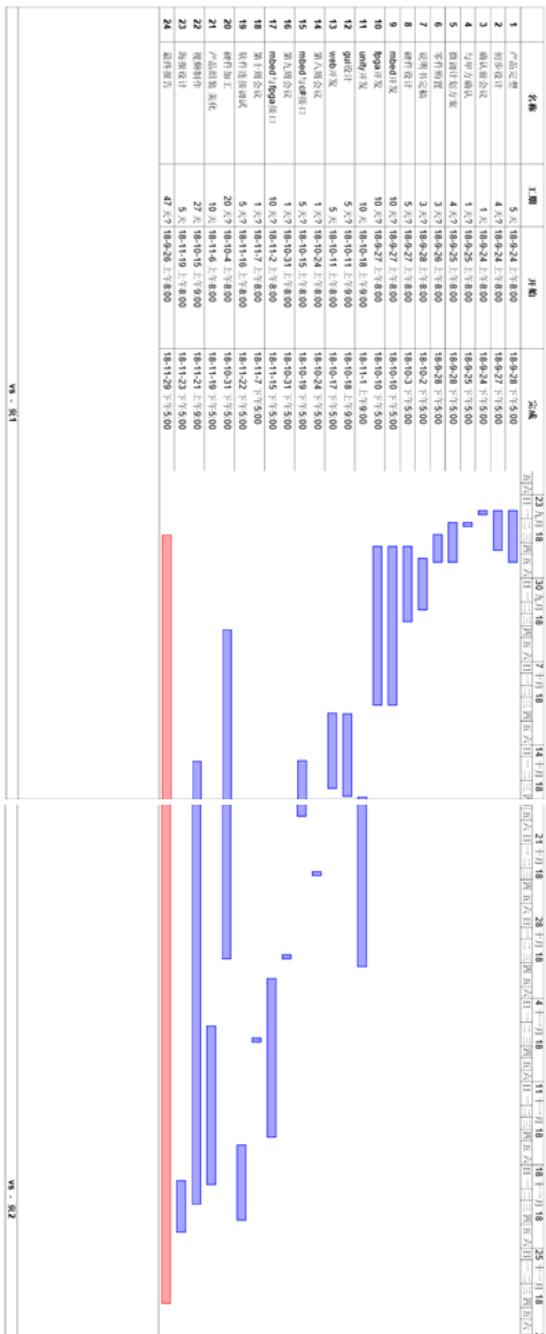


Figure 7-1 The first edition of project plan

# INTERESTING

However, some unexpected emergencies happened, and some regulations changed during the process. The finish time of deciding the overall concepts and main functions of our product were two weeks later than we planned because we had misunderstood the requirement at first so that a lot of time were wasted, and we had to change our minds after several meetings. The time of training for woodworking and the opening time of the work shop were half months later than we plan which left extremely less time for us to process, assemble and decorate.

Fortunately, there was no difficulty that can hold us back. We changed our plan with the reality as soon as possible and finally finished all the tasks on time. The final project plan of our project

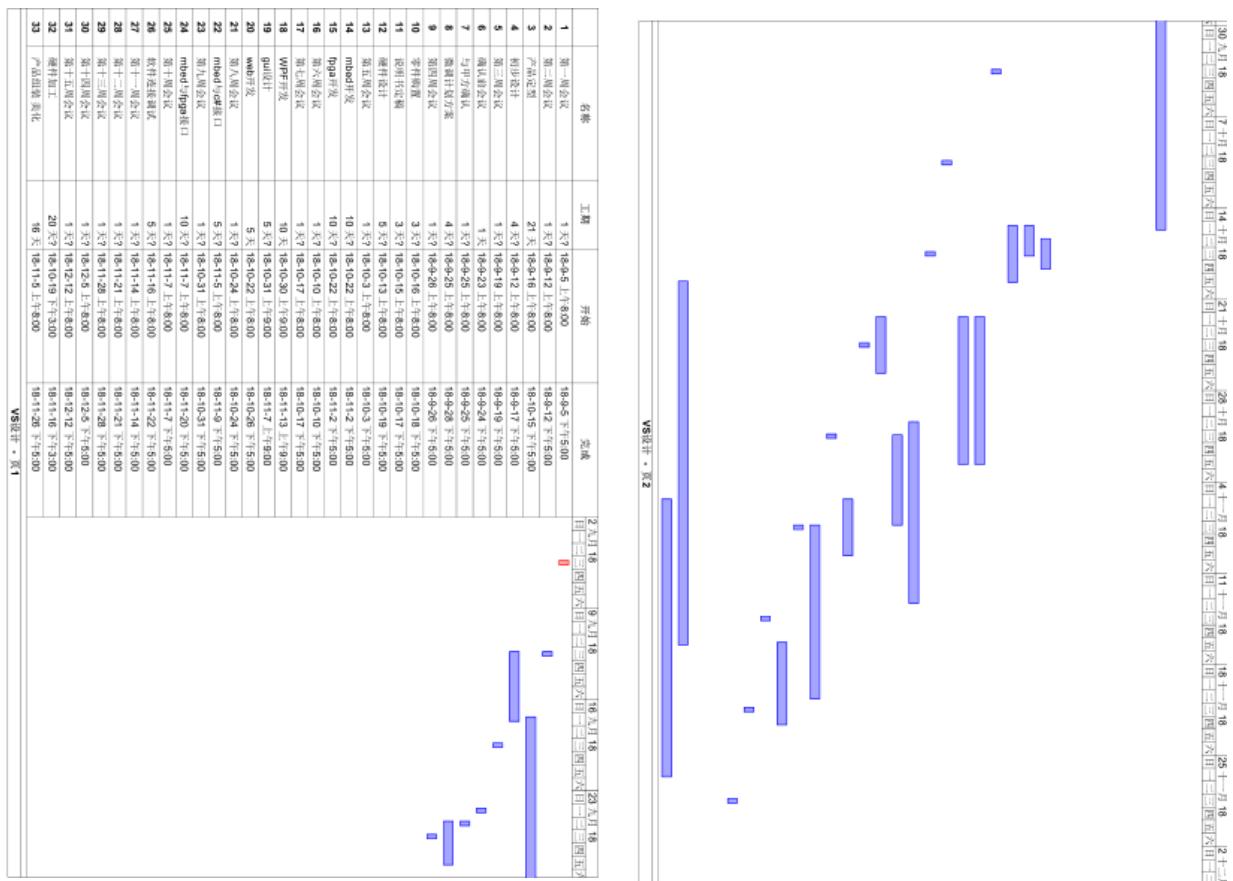


Figure 7-2 The final project plan of project 1

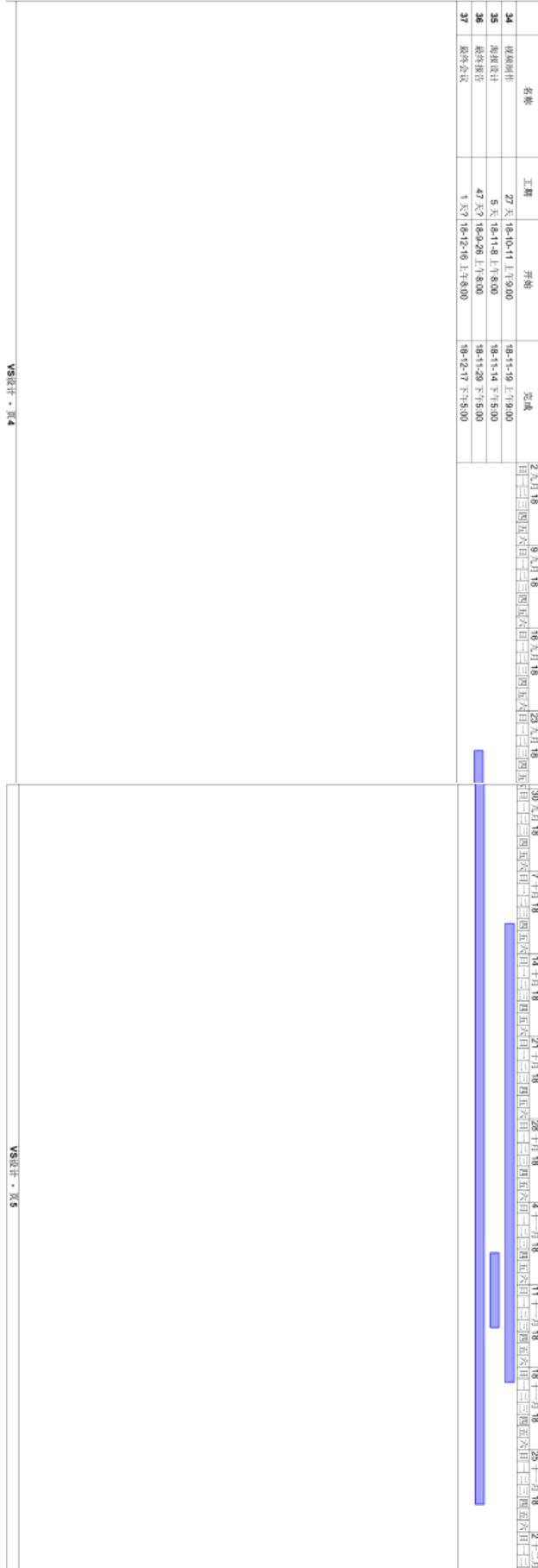


Figure 7-3 The final project plan of project 2

## 7.2. Risk Management

The risk response plan and register was made before processing and manufacturing to make sure that we can handle all the possible risks and had a backup plan to hold our project. The evaluate table will be placed in the appendix.

# INTERESTING

## 7.3. Teamwork

The responsibilities of each team members are firstly divided as shown below.

		Assigned Roles									
Surname	Forename	MD	Sec	EW	DE	SE	PR	ME	WE	WD	
Sun	Yuxuan							X	X		
Wang	Ziyao		X							X	
Kong	Xiangxuan		X				X				
Deng	Ziwei				X	X					
Jing	Yifei				X	X					
Yi	Weihang	X		X							
<b>Managing director</b>								<b>MD</b>			
<b>Company secretary</b>								<b>Sec</b>			
<b>Cheif electronics and wiring engineer</b>								<b>EW</b>			
<b>Chief digital systems engineer</b>								<b>DE</b>			
<b>Software engineer</b>								<b>SE</b>			
<b>Chief public relations organizer.</b>								<b>PR</b>			
<b>Chief mechanical designer.</b>								<b>ME</b>			
<b>Workshop engineer</b>								<b>WE</b>			
<b>Web designer</b>								<b>WD</b>			

However, the responsibility division has been slightly modified during the project because of the unbalance of difficulties of every work. We had to adjust our responsibility division and use the power of our team to ensure the project process. The final tasks of every members are illustrated in table shown below.

		Assigned Roles									
Surname	Forename	MD	Sec	EW	DE	SE	PR	ME	WE	WD	
Sun	Yuxuan							X	X	X	
Wang	Ziyao		X		X					X	
Kong	Xiangxuan		X			X	X				
Deng	Ziwei				X	X	X		X		
Jing	Yifei			X	X	X					
Yi	Weihang	X		X				X	X		
<b>Managing director</b>									<b>MD</b>		
<b>Company secretary</b>									<b>Sec</b>		
<b>Chief electronics and wiring engineer</b>									<b>EW</b>		
<b>Chief digital systems engineer</b>									<b>DE</b>		
<b>Software engineer</b>									<b>SE</b>		
<b>Chief public relations organiser.</b>									<b>PR</b>		
<b>Chief mechanical designer.</b>									<b>ME</b>		
<b>Workshop engineer</b>									<b>WE</b>		
<b>Web designer</b>									<b>WD</b>		

#### 7.4. Meetings

To make sure that every problem happens during the process can be discussed and fixed quickly, timely and efficiently, group meetings were held weekly. The secretary of our company recorded All the valuable details of the meeting and sorted out them into meeting minutes every week. All the minutes will be attached in appendix.

## 7.5. Experience and Lessons

- (1). The requirement needs to be discussed and understood detailed to make sure that the design thought of the project is right and then all the work can be considered and discussed on that basis.
- (2). It is of vital importance in the process of project that all the members should act as a team. Work needs to be divided into small tasks and can be done separately by the person who is in charge, but all of the team members should help each other and make their efforts to do a better job.
- (3). There can be some arguments and disagreements among the team. It is inevitable that team members have different opinions about some decisions. The rules of decision making should be made before starting the project and manager should take his responsibility to learn advices and coordinate relations.
- (4). Communication is very important. We not only need to communicate and cooperate with each other in our team, but also need to keep in touch with the first party who gives the requirement and makes the regulations.
- (5). Even though we have detailed plan and risk response plan, they are never sufficient for the project. We cannot predict every unexpected situation or emergency, all we can do is to just adjust our plan and try our best to make the influence minimum.
- (6). Only when a project is finished and handled on time, it can be evaluated and have the probability to succeed. Never waste your time for any reason because it is the most important influence factor of the project.

## 8. APPENDIX

### 8.1. Meeting Minutes

Meeting Minutes	
<b>Group Name:</b>	Interesting
<b>Group Number:</b>	03
<b>Week:</b>	Week 2
<b>Time:</b>	16:00-18:00pm, 5 <sup>th</sup> September 2018
<b>Place:</b>	Library
<b>Present:</b>	All
<b>Apologies:</b>	None
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>● The function, model and process of the whole project are discussed.</li> <li>● Duties were assigned to each group member.</li> <li>● Selection of team name.</li> <li>● Signing of Team contract.</li> </ul>
<b>Review of previous actions:</b>	None
<b>Main business:</b>	<ul style="list-style-type: none"> <li>● Team introduction and setup began</li> <li>● Contract assigned</li> <li>● Discussion of product related issues</li> </ul>
<b>Action</b>	<ul style="list-style-type: none"> <li>● The model is a robot with a drum.</li> <li>● The product was initially determined to be a drum game machine.</li> <li>● The assignments are as follows:             <ol style="list-style-type: none"> <li>a. Weihang Yi serves as a manager.</li> <li>b. Weihang Yi and Yifei Jing are the chief hardware engineer and the chief software engineer respectively.</li> <li>c. Weihang Yi and Yuxuan Sun are responsible for hardware, including mechanical drawing, 3D printing, material processing and sensor installation.</li> <li>d. Ziwei Deng and Xiangxuan Kong are responsible for the design of GUI.</li> <li>e. Ziyao Wang and Yuxuan Sun are responsible for the design of the company web site.</li> <li>f. Ziyao Wang and Yifei Jing are responsible for programming, including the programme inMBED and FPGA .</li> <li>g. Xiangxuan Kong and Ziyao Wang serve as secretary,Xiangxuan Kong serves as public manager.</li> </ol> </li> </ul>

# INTERESTING

Meeting Minutes			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 3		
<b>Time:</b>	16:00-18:00pm, 12 <sup>th</sup> September 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Abandon the original game of playing drums and change the product to a more interesting and ornamental cliff jumping game.</li> <li>➤ The control flow of the whole system is reanalyzed.</li> </ul>		
<b>Review of previous actions:</b>	<ul style="list-style-type: none"> <li>➤ The mechanical design group has made preliminary design drawing of the whole product by CAD.</li> <li>➤ The web developer created a template for the company's website.</li> <li>➤ The software team learned how to write simple little games with C# on Unity.</li> </ul>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The change of the project</li> <li>➤ Analysis of system control process</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ The project plan is changed as follows: To simulate the scene that a person is about to fall off a cliff, a small square is used to simulate a person, and a board with controllable inclination is used to simulate the cliff. The user controls the inclination of the board through the key stroke, so as to ensure that the person does not fall from the cliff. The longer the duration, the higher the score.</li> <li>➤ System control process analysis is as follows: When the key is pressed, a wave will be generated. The faster the frequency, the higher the frequency. After receiving this wave, MBED will send a numerical value to FPGA to control the rotation Angle of the steering gear according to different frequency, and the rotation Angle of the steering gear will control the inclination of the board.</li> </ul>		

Meeting Minutes			
Group Name:	Interesting	Group Number:	03
<b>Week:</b>	Week 4		
<b>Time:</b>	16:00-18:00pm, 21 <sup>st</sup> September 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Detailed discussion on risk assessment and related handling measures.</li> <li>➤ The advantages of our products are discussed, and the demand analysis and market assessment of our products are carried out</li> </ul>		
<b>Review of previous actions:</b>	<ul style="list-style-type: none"> <li>➤ The mechanism design group made primary design for most of the components of our product.</li> <li>➤ The software team had completed half of the game programming.</li> </ul>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The risk assessment and related handling measures are discussed.</li> <li>➤ Detailed discussion on the advantages of our products, and we also analysis the demands and market of our products.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ We discussed in detail the three parts of risk management: Risk type, treatment and prevention: In particular, the risk has five main types: technological risk, external risk, organism risk, management risk and healthy risk. The procedure to dealt with these risks is as follow: Firstly, we should recognize and response to these risks, then relative teammates should access the probability and effects of the risk occurring. Finally, some measures should be taken to minimize the loss, such as Share the loss with the partner company and buy insurance for our products. Besides, the monitor of the risk is necessary.</li> <li>➤ The demand analysis and market access include following aspects: The effect of too long queuing time, Defects in current queuing system, Advantages of physical games, Market saturation analysis. We have conducted detailed analysis and evaluation on each aspect and completed the market plan.</li> </ul>		

# INTERESTING

Meeting Minutes			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 5		
<b>Time:</b>	16:00-18:00pm, 24 <sup>th</sup> September 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Detailed discussion on Project proposal which will be submitted on the meeting to our client.</li> <li>➤ Preparation and task allocation for the meeting to our client are discussed.</li> </ul>		
<b>Review of previous actions:</b>	<ul style="list-style-type: none"> <li>➤ The mechanism design group made detailed design for part of the components of our product.</li> <li>➤ The software team had completed three quarters of the game programming.</li> </ul>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The project proposal which will be submitted on the meeting to our client are discussed.</li> <li>➤ Detailed discussion on the preparation and task allocation for the meeting to our client, and the matters needing attention are also discussed.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ We discussed in detail the three parts of project proposal: The brief introduction of our product, The structure and function of our product, The introduction to our company.</li> <li>➤ In particular, the introduction of our product has five main parts: The method of design, The project feasibility analysis, The background of our game, The operation guidelines</li> <li>➤ The detailed parts of the structure and function of our product is as follow: The diagram of design, The structure, The special functions.</li> <li>➤ The introduction to our company includes following aspects: The brief introduction to our company, The members and task allocation.</li> </ul>		

Meeting Minutes			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 7		
<b>Time:</b>	16:00-18:00pm, 10 <sup>th</sup> October 2018 17:00-19:00pm, 11 <sup>th</sup> October 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Detailed discussion on new plan of our project.</li> <li>➤ Preparation and mechanical design for new project are discussed.</li> </ul>		
<b>Review of previous actions:</b>	<p>Under the guidance of the teacher, we found many problems in the old project. After carefully reviewing the old plan, we decided to abandon the old plan and adopt a new project which is more in line with the customer's requirements.</p>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ After carefully analyzing the customer requirements, we decided to consider new ideas on the basis of meeting the customer requirements and the automation of the equipment.</li> <li>➤ After the new plan was decided, we began to discuss the mechanical design and programming of the new project.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ The general idea for the new project is that the balls roll off a track, and players need to control the direction of the track by manipulating a joystick above the control box and place the balls of different colors into a catch basket that matches the color.</li> <li>➤ For the new project, we will add the use of color sensor in the game body and add automatic sorting system to improve the automation level of the product. We add a color sensor to the basket to determine whether the ball falling into the basket matches the required color. The falling pellets are recycled back into the orbit using the automatic sorting and recycling system.</li> </ul>		

# INTERESTING

Meeting Minutes			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 8		
<b>Time:</b>	16:00-18:00pm, 18 <sup>th</sup> October 2018 16:00-18:00pm, 19 <sup>th</sup> October 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Detailed discussion on our 3<sup>rd</sup> project.</li> <li>➤ Preparation and mechanical design for the 3<sup>rd</sup> project is discussed.</li> </ul>		
<b>Review of previous actions:</b>	After discussing with the teacher, we found that the previous plan did not consider the robot elements. In order to add the model of robot, we made a new plan again.		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ We found that the robot's modeling is a necessary element. On the basis of robot modeling, we redesigned the new scheme.</li> <li>➤ Mechanical structure and software design are also discussed.</li> </ul>		
<b>Action</b>	<p>The operation process of our project is designed as follow:</p> <ul style="list-style-type: none"> <li>➤ The robot will find and approach users.</li> <li>➤ Voice leads the user to insert the id card, identify the id and open the cabin.</li> <li>➤ The user clicks the button to start the game and the timer starts.</li> <li>➤ The game is set up for the robot to seek help from the user, uses needs to help the robot repair internal pipes. Our project will use different color blocks to represent different shapes of pipes, and users need to put different shapes of pipes in the right place to connect the whole pipe.</li> <li>➤ The user clicks the end button to end the game and the color sensor analyses the results.</li> <li>➤ Close the cabin and sort the reclaimed wood.</li> <li>➤ If successful, the robot will give users water.</li> <li>➤ The robot is once again in search mode.</li> </ul>		

Meeting Minutes			
Group Name:	Interesting	Group Number:	03
<b>Week:</b>	Week 9		
<b>Time:</b>	16:00-18:00pm, 26 <sup>th</sup> October 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Detailed discussion on style of our company's website.</li> <li>➤ Size of parts and materials for the 3<sup>rd</sup> project is discussed.</li> </ul>		
<b>Review of previous actions:</b>	<p>The concrete mechanical design and primary GUI for our 3<sup>rd</sup> project has completed.      Mechanical principle and realization method are also determined.</p>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The type, size and quantity of components needed are discussed in detail.          The selection and purchase are also made together.</li> <li>➤ The content and design style of the company website were discussed.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ We determined the detailed size of each part through mechanical design drawing and project requirements and determined the number of each part according to the different needs of different modules. Finally, the above information was counted as a table.</li> <li>➤ The dimensions of the boards used as machine shells are determined according to the mechanical design drawing. We also discussed how to fix planks and how to process them.</li> <li>➤ After discussion, everyone agreed to make the website and video into a technical style.</li> </ul>		

# INTERESTING

Meeting Minutes			
Group Name:	Interesting	Group Number:	03
<b>Week:</b>	Week 10		
<b>Time:</b>	16:00-18:00pm, 2 <sup>nd</sup> November 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Some details have been re-discussed.</li> <li>➤ We revisited the connection between the components and the specific assembly structure of the gear set.</li> </ul>		
<b>Review of previous actions:</b>	<p>We have purchased all the parts and materials of our product.          The processing methods corresponding to different materials and components are discussed and determined.</p>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The waterproof problem of electronic products and the problem of sensor sensitivity are discussed in detail.</li> <li>➤ The connection between the components and the specific assembly structure of the gear set were discussed.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ First, the frame board is fixed with an iron right Angle, and the internal structure is temporarily fixed with tape or glue, so that it is easy to disassemble when there is a problem with the equipment.</li> <li>➤ The water in the reward system should be completely isolated from the electrical equipment, which can avoid damage caused by leakage problems.</li> <li>➤ The part that can use cardboard as far as possible do not use board, which can reduce the weight of whole product, so that the facilitate power system to drive equipment to move smoothly.</li> </ul>		
	<ul style="list-style-type: none"> <li>➤</li> </ul>		

<b>Meeting Minutes</b>			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 11		
<b>Time:</b>	16:00-18:00pm, 9 <sup>th</sup> November 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Some details of wood processing are discussed.</li> <li>➤ The problem of product assembly is mainly discussed</li> </ul>		
<b>Review of previous actions:</b>	<p>The waterproof problem of the product was solved by further improving the design.</p> <p>We changed some more complex gear group structure into glue, which reduced the difficulty of product realization.</p>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ How to saw boards and punch holes on boards are discussed in detail.</li> <li>➤ Detailed discussion on how the boards are connected, how are the templates fixed to enhance the structure's stability, and how are the components assembled.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ Board cutting: Use a hand saw to divide the board so that the shape of the board can be precisely controlled.</li> <li>➤ Hole: Determine the position of the hole on the board with rulers, and then the hole.</li> <li>➤ Connection: Iron right Angle and screws are used to fix the parts between the boards. Screws or glue are used to connect the parts between the boards.</li> <li>➤ Welding: Use a welding gun to weld electrical motors and wires of electronic devices.</li> </ul>		

# INTERESTING

Meeting Minutes			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 12		
<b>Time:</b>	16:00-18:00pm, 16 <sup>th</sup> November 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Debugging: we first debug each function of the machine and then test whether the whole process of integrating all the functions can be implemented smoothly.</li> <li>➤ Database: we talked specifically about how the rankings are ranked.</li> </ul>		
<b>Review of previous actions:</b>	<p>The problem of wood processing has been solved perfectly.            We optimize our product assembly through a little change to the design of our mechanical structure.</p>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The procedure of debugging is discussed in detail.</li> <li>➤ Detailed discussion on the design of the .data model of our database.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ All our processes which are needed to be debugging are shown below:            Find the user, get close to the user, machine stop, identification, preparation, game start, game in, game out, machine reset, find the user.</li> <li>➤ The algorithm for the calculation of the ranking:            When the data volume is extremely small, may directly all rank. But when the data is large, it makes little sense to rank all the data, and it consumes too much resources. Therefore, we decided to use the following algorithm for ranking: after a score is generated, it is compared with the lowest value in the ranking. If it is less than the lowest value, it will be rejected directly. If it is greater than the lowest score on the chart, the lowest score on the current chart is excluded and the score is inserted to rank.</li> </ul>		

Meeting Minutes			
<b>Group Name:</b>	Interesting	<b>Group Number:</b>	03
<b>Week:</b>	Week 13		
<b>Time:</b>	16:00-18:00pm, 23 <sup>rd</sup> November 2018		
<b>Place:</b>	Library		
<b>Present:</b>	All		
<b>Apologies:</b>	None		
<b>Agenda:</b>	<ul style="list-style-type: none"> <li>➤ Carry out final assembly and wiring of products</li> <li>➤ Test the stability of the product after assembly and improve the layout and structure of product.</li> </ul>		
<b>Review of previous actions:</b>	<ul style="list-style-type: none"> <li>➤ The debugging for the implementation of the machine function has been basically completed</li> <li>➤ We have carried out the testing for the integral of all subfunctions.</li> <li>➤ The creation of the database has been completed and the ranking mechanism is also determined.</li> </ul>		
<b>Main business:</b>	<ul style="list-style-type: none"> <li>➤ The assembly and wiring of products are carried out.</li> <li>➤ Detailed discussion on the test for the stability of the product after assembly and how to improve the layout and structure of product.</li> </ul>		
<b>Action</b>	<ul style="list-style-type: none"> <li>➤ In order to make the wiring arrangement more orderly, we simplify the circuit by reducing the number and distance of the wires of different devices.</li> <li>➤ We have tested the performance of the product many times after assembly and found that the stability of the product still needs to be improved. In order to improve the stability of the product, , we came up with the following method: use hot melt adhesive to fix the wire connection to prevent the disconnection of two interconnected wires. In addition, some loose parts were fixed to reduce the vibration of the product during operation, which effectively improve the stability of the product.</li> </ul>		

# INTERESTING

## 8.2. Risk Management

### Risk Response Plan and Register

<b>Project Name:</b> “Mr.Tube” entertainment robot		<b>Prepared by Team:</b> 03			<b>Date:</b> 9/21/2018	
<b>Course Code:</b> B39VS		<b>Contact Name:</b> Xiangxuan Kong			<b>Customer/End User group:</b> 03	
<b>Year of study/Discipline:</b> 3		<b>Project Manager:</b> Weihang Yi			<b>Project Sponsor:</b> none	
WBS Number	Risk Event	Risk Type	Probability	Impact	Risk Response	Risk Owner
1	The design is incomplete and has some flaws	Technical Risk	20%	Influences the functional completeness of the final product	Discuss with the first party about the design before the project start and make adjustment with every steps of the whole process	Chief digital systems engineer & Software engineer
1	Backward in construction technology	Technical Risk	40%	Result in the discrepancy between outcome and original expectations	Set up a Plan B which is easier to accomplish and change the plan if Plan A meets a choke point and do not have a breakthrough after some time	Chief electronics and wiring engineer & Chief mechanical designer.
1.2	Injury during the process of mechanical operation	Health and Safety Risk	10%	Vacant jobs which can hinder the whole process may occur	Make sure that every member who needs to operate with some injurious-like work has been well guidance in the security training and every work needs to be done with at least person for operating and supervising	Workshop engineer
1	Abruptly dimission and personnel change	Organisational Risk	Less than 5%	Disturb the division of the project	Each job taken by at least two people as one takes the main responsibility and they can help each other to finish the work	Managing director
1	Disharmony among the group in some decision-making occasion	Organisational Risk	15%	Lead to inefficiency and injudicio usness of decision-making process as well as harm to team relationship and sprit.	Reach an agreement and make a contract to stipulate that all the decisions are made through plurality rule before the project start	Managing director

<b>1</b>	Disorder and imbalance of funds management	Financial Risk	10%	Influence the circulation of Capital chain and the supplement of components	Clear and definite that all the funds about this project are managed by the chief secretary only in order to unification	Company secretary
<b>1</b>	Project behind schedule Fall to finish on time	Project Management Risks	40%	Cannot submit and present the finished product in time	Communications among the group members need to be monitored by the manager to make sure that all the small tasks are done in time	Managing director
<b>1</b>	Alteration of regulations made by the first party	Project Management Risks	10%	The whole project needs to be reconsidered, redesigned and reorganised to fit the new regulation	Keep in contact with the first party and make adjustment time to time instead of redesigned the whole project at last	Chief public relations organiser
<b>1.2</b>	Disqualification and shortage of component caused by the supplier	Project Management Risks	50%	Increase the time cost and may lead to the failure of the outcome	Choose the most reliable supplier after prophase investigation meanwhile choose another reliable supplier to be the backup supplier in case of component shortage and other conditions like that of the first supplier	Company secretary
<b>1</b>	Lack of energy, harsh climate and other force majeure	External Risk	Less than 5%	Hinder the process of the project	Select a backup working station and make advance preparation such as emergency power supply if needed	Managing director

Table 8-1 Risk Response Plan and Register

### 8.3. Chart Index

Figure 2-1 C4D model of game sub-system.....	4
Figure 2-2 Dimension drawing of game sub-system .....	5
Figure 2-2-3 C4D model of recovery sub-system .....	5
Figure 2-4 Dimension drawing of recovery sub-system.....	5
Figure 2-5 C4D model of authentication card reader sub-system .....	6
Figure 2-6 Dimension drawing of authentication card reader sub-system.....	6
Figure 2-7 C4D model of reward sub-system .....	7
Figure 2-8 Dimension drawing of reward sub-system .....	7
Figure 2-9 C4D model of power supply sub-system.....	7
Figure 2-10 Dimension drawing of power supply sub-system .....	8
Figure 2-11 C4D model of power sub-system .....	8
Figure 2-12 Dimension drawing of power sub-system.....	8
Figure 2-13 Front View	Figure 2-14 Right View.....
Figure 2-15 Vertical View .....	9
Figure 2-16 Motor from the bottom	Figure 2-17 Color sensors from the top.....
Figure 2-18 Game board.....	11
Figure 2-19 The cover .....	11
Figure 2-20 Game sub-system.....	12
Figure 2-21 Recovery sub-system.....	12
Figure 2-22 ID card slot.....	13
Figure 2-23 Authentication card reader sub-system.....	13
Figure 2-24 Reward Sub-system.....	13
Figure 2-25 Power supply sub-system.....	14
Figure 2-26 Ultrasonic distance sensors and infrared sensor .....	14
Figure 2-27 Power sub-system .....	15
Figure 2-28 Right view	Figure 2-29 Front view.....
Figure 2-30 Vertical view .....	15
Figure 2-31 Track (small)	Figure 2-32 Forward board (small) .....
Figure 2-33 Top baffle's front 1 (small)	Figure 2-34 Isolation board (small).....
Figure 2-35 Top board (small)	Figure 2-36 Top baffle's front 2 (small).....
Figure 2-37 Base board (small)	Figure 2-38 Left board (big) .....
Figure 2-39 Operation panel (small)	Figure 2-40 Rotating board and base (small) .....
Figure 2-41 Middle baffle 1 (small)	Figure 2-42 Middle baffle 2 (small) .....
Figure 2-43 Overhead gauge (small)	Figure 2-44 Right board (big) .....
Figure 2-45 Back board (big)	Figure 2-46 Box (small) .....
Figure 3-1 PWM explain .....	22
figure 3-2 ASM Chart .....	25
Figure 3-3 State Machine Chart.....	25
figure 3-4 RTL Diagram .....	26
Figure 3-5 Pin Planning.....	26
Figure 4-1 original state Logic.....	28
Figure 4-2 Maintenance Mode.....	28
Figure 4-3 User Mode.....	29
Figure 4-4 4.2.2.5 Window and pages .....	33
Figure 4-5 Design pattern --- MVVM .....	33
Figure 4-6 Operation Procedure.....	34
Figure 4-7 Interface .....	35
Figure 4-8 maintenance mode .....	35
Figure 4-9 Operation Procedure.....	38
Figure 4-10 the UI after the language change.....	40

Figure 4-11 User Information Table .....	41
Figure 4-12 Ranking List.....	41
Figure 4-13 Statistics on the number of players in different types of parks .....	42
Figure 4-14 Statistics on the number of players of different ages .....	42
Figure 4-15 The process of customer analysis .....	42
Figure 4-16 The scroll viewer .....	50
Figure 4-17 Rank List in The Game .....	51
Figure 5-1 Read RGB values.....	57
Figure 5-2 Read RGB values with proportions.....	57
Figure 5-3 Define three colors.....	58
Figure 5-4 Read the map before the game .....	58
Figure 5-5 Final version of color sensor.....	58
Figure 5-6 Read distance .....	59
Figure 5-7 Get distance by ultrasonic.....	59
Figure 5-8 Read ultrasonic data with sampling .....	59
Figure 5-9 Read 5.1.4 Read Infrared Sensor.....	60
Figure 5-10 Read Photoelectric Gate Sensor.....	60
Figure 5-11 Read 5.1.6 Read Reed Pipe Sensor.....	61
Figure 5-12 The required card .....	62
Figure 5-13 The pin allocation of the sensor system.....	63
Figure 6-1 SWOT .....	66
Figure 6-2 Poster .....	67
Figure 6-3 Film1.....	68
Figure 6-4 Film2 .....	68
Figure 6-5 Film3 .....	69
Figure 6-6 Film4 .....	69
Figure 6-7 Website1 .....	70
Figure 6-8 Website2 .....	70
Figure 6-9 Website3 .....	71
Figure 6-10 Website4 .....	71
Figure 6-11 Exhibition Stand .....	71
Figure 7-1 The first edition of project plan.....	73
Figure 7-2 The final project plan of project 1 .....	74
Figure 7-3 The final project plan of project 2 .....	75
Table 5-1 RGB values for three colors indoor .....	61
Table 5-2 RGB values for three colors indoor .....	61
Table 5-3 RGB values for three colors indoor.....	62
Table 8-1 Risk Response Plan and Register .....	91

# INTERESTING

## 8.4. About this Document

Session	Authors
Introduction	Yuxuan Sun
Mechanical Design	Yuxuan Sun
Digital System Design	ZiYao Wang Yifei Jing
Software Design	Yifei Jing Ziyao Wang
Sensor System	Yuxuan Sun Weihang Yi
Market and Web	Xiangxuan Kong
Project Plan	Xiangxuan Kong
Editor	Weihang Yi



Z  
T  
R  
E  
S  
T  
Z  
G

Copyright Interesting Company in 2018

Interesting<sup>©</sup> Company<sup>™</sup> 2018