Xi'an Jiaotong-Liverpool University
西交利物浦大學

# Course Work Submission

| | | |
|---|---|---|
| Name | Pai.Peng, Yangzhi.Gao, Xinyue.Wu, Qinxin.Ren | |
| ID Number | 1822600, 1822209, 1715914, 1824098 | |
| Programme | Digital Media Technology | |
| Module Title | Industrial Awareness and Group Project | |
| Module Code | MEC202 | |
| Assignment Title | MEC202_Group 38_Report | |
| Submission Deadline | 2021.6.6 23:59 | |
| Lecturer Responsible | Dr Jie.Sun | |

I certify that:

- I have read and understood the University's definitions of COLLUSION and PLAGIARISM (available in the Student Handbook of Xi'an Jiaotong-Liverpool University).

With reference to these definitions, I certify that:

- I have not colluded with any other student in the preparation and production of this work;
- this document has been written solely by me and in my own words except where I have clearly indicated and acknowledged that I have quoted or used figures from published or unpublished sources (including the web);
- where appropriate, I have provided an honest statement of the contributions made to my work by other people including technical and other support staff.

I understand that unauthorised collusion and the incorporation of material from other works without acknowledgement (plagiarism) are serious disciplinary offences.

Signature …Pai.Peng, Yangzhi.Gao, Xinyue.Wu, Qinxin.Ren …… Date …2021.6.6……

| For Academic Office use: | Date Received | Days Late | Penalty |
|---|---|---|---|
| | | | |

# MEC202 Report

# 1. Introduction

## 1.1 User needs and survey on existing similar industrial products

**Definition**: Educational intelligent toy

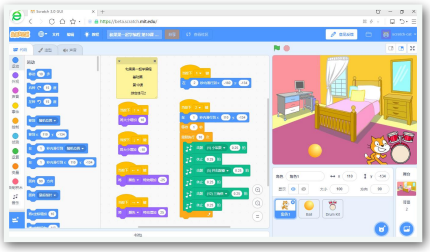
**Target user group**: preschool children

**From the perspective of the user's own needs**: Firstly, in reality, the number and types of toys related to education are not enough, and secondly, products that are both educational and entertaining are very rare. In addition, few toys can balance the user's personal experience with parental participation, and there is not enough interaction between children and their parents. In fact, it can be concluded from more and more news reports and scientific theories that children need adequate education and parental participation during the period of rapid growth. This product called "CUBO" compensates to a certain extent for the three problems that may arise when a product is used as a toy. Through research and discussion, our team also recognized this idea: parents and elementary schools are willing to buy this product, and can achieve its entertainment and educational significance under the prescribed gameplay and rich scenes.

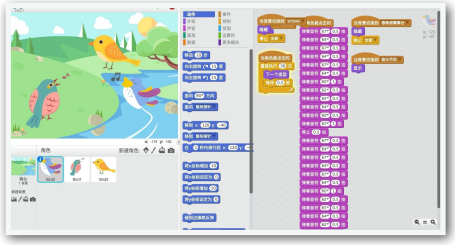
From the perspective of market competition analysis: In addition to the fact that the types and quantities of educational toys are small as mentioned above, our group also found that most of the existing educational toy products lack intelligence and playability, such as toy cars. It can only be controlled simply, or the building blocks are not difficult to interest the modern children who have grown up influenced by science and technology. In addition, in this case, the role of parents is very small. They can only act as guardians or bystanders.

We investigated and analyzed two educational toys or products that account for a large share of the market, namely Scratch and LEGO.

**Scratch**: As a well-known children's programming software, scratch proved that it can successfully let children understand programming and corresponding grammar. But our team believes that this product still needs a computer as a tool, and children are not suitable for staying in front of the computer for a long time, which is not conducive to physical development and vision. In addition, some terms and operations are not easy for preschoolers to understand, and complicated operations will gradually make them lose patience. The operation of these programming languages and computers is too indirect for preschoolers. Our group hopes that a product will enable children to better understand the food in society, such as traffic and pedestrians; better understand nature, and stimulate their curiosity and exploration. At the same time, it is also an educational product that is beneficial to the physical development of preschool children.



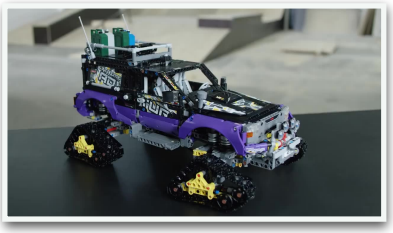
Scratch(1)



Scratch(2)

**LEGO**: As a synonym for building blocks, LEGO has been popular all over the world for more than 100 years. In addition to the simple splicing of building blocks to get a product gameplay, LEGO has also developed programmable building blocks in recent years, which can be considered a true embodiment of scratch. However, whether it is splicing and programming, it is slightly complicated for preschoolers. They not only need to calm down to assemble the finished product with many parts, but also need to check whether the spelling is correct. This is more suitable for children around the third grade. The programmable building blocks are more suitable for older children, around 12 years old. Most preschool children do not have the ability to handle such complex toys. For children, toys need to be more direct and interesting, and parents can also seek high levels of participation.


LEGO(1)



LEGO(2)

**1.2 Summarize a list of specific customer needs and design requirements.**

**Specific customer needs**

To achieve the customer requirements in 1.1 above, we need to refine the requirements for our smart car "CUBO". The composition of the toy requires a Bluetooth device and the smart car itself. A well-built shell is also needed in appearance. The composition of the trolley itself needs to have enough sensors to monitor the activities, trajectories and obstacles of the trolley to enable the trolley to travel smoothly. In terms of the maneuverability of the car, "CUBO" should have enough interaction and interesting operation methods, such as using Bluetooth, magnetic card, remote control, etc. to make the car run. At the same time, customers need to be able to reuse and change the environmental factors at will to maintain the playability of this product. The above are the specific needs of customers for this product.

**Design requirements**

The goal of our team in designing this product is to provide a complete set of screenless programming education solutions for schools and educational institutions, so that children can understand and learn programming knowledge through hands-on physical methods, and give children the necessary abilities that the world needs in the future. The design requirements can have the following five points:

1. Multiple scenes that can be DIY and not too complicated scene layout;
2. Sufficient subjective operation of the child as a user (playability);
3. Parents as participants must have a certain sense of participation and education;
4. The structure of the multifunctional trolley and its acceptable appearance;
5. Sufficient interaction between the car and the user.

To sum up: the characteristics of "CUBO" are: intuitive, interesting and diverse. It can perform well in various scenarios, such as education, entertainment

**1.3 Project planning and workload distribution**

**Project planning**

| | |
|---|---|
| Week 5 | Determine product type and direction. |
| Week 6 | Assembly of “CUBO” |
| Week 7 | The car can receive the information and instructions of the RFID–magnetic card. |
| Week 8 | The decision–making level code is introduced into the relevant library and interacts with other modules. The car completes touch sensing. |
| Week 9 | Assembly and connection of Bluetooth. |
| Week 10 | Problem solving and game scenario planning. |
| Week 11 | Auxiliary sensor assembly and sensing. Code writing of different scenarios corresponding to decision level |

| | |
|---|---|
| Week 12 | Rich functions and some auxiliary controls and pages. Auxiliary sensor assembly and sensing.<br>Shell production, scene assembly and preliminary test |
| Week 13 | Final test, video recording, poster making. Car package. |

**Workload distribution (beginning)**
Xinyue.Wu: Control layer code writing and circuit connection
Qinxin.Ren: Code writing of perception layer and decision layer, Installation of trolley
Pai.Peng: Code writing of perception layer, connection circuit, sensor debugging
Yangzhi.Gao: Decision level code writing, product planning and design

# 2. Design concept comparison, development and screening

## 2.1 Compare design concepts in terms of the specific user requirements.

In designing a product, our team had three different design concepts. The first thing we want to do is a robotic arm. In addition to recognizing and grasping objects, the robotic arm can also have the ability to judge itself after deep learning. Therefore, you can play chess and even have a conversation with the child.
The second idea is to make a smart car. But this car is completely based on magnetic disks, that is, using magnetic disks to make a map. The car reads the information and completes the corresponding operation. This product can be used to teach children how to spell words.
The third type is a smart car that uses multiple media. That is to use multiple devices such as magnets, remote controls, etc. to operate the car. The props of the map can also be not limited to magnetic cards. This can give children more autonomy and fun while parents can also have a sense of participation.

## 2.2 Perform design concept selection and choose one or two concepts for prototyping.

**The model design concept and selection**
Based on the analysis of the competing products of Lego and Scratch, our team identified the toy direction as children's educational toy products that use multi-sensors. Our team tried several products at the beginning of the project. They are a robotic arm, a car based on a magnetic card map, and the last selected multi-control mode (magnetic card, remote control) type car.
Robotic arms are widely used in industrial scenes and are not very safe to use as toys. Secondly, we originally planned to use robotic arms to grab objects and realize chess functions, but this will involve a lot of machine learning, which is not realistic.
The car based on the magnetic card map is the predecessor of the current product. Because the magnetic card can contain a lot of information, the use of magnetic sheets to make a map of the car driving can realize the functions of spelling words and autonomously interpreting the forward direction. But the disadvantage of this idea is that firstly this requires a large number of magnetic cards, and secondly, children do not have enough operability and interaction with toys in the game.
Combining the above two ideas, we finally decided to make a smart car that can be controlled with a remote control, or it can read instructions from a magnetic card to complete steering and forward. At the same time, the magnetic card also contains different information in different scenarios to achieve different functions. This concept allows children to have sufficient operating experience first, and secondly retains the addition of magnetic cards.

**The car as the choice and its design concept**
Based on 1.1 and 1.2 wheel types, our team believes that the design concept of the car is based on the overall design of the interactive system of the smart car and the design of modular functions with strong expansion
The main device of "CUBO": Smart car "CUBO"
Additional equipment: Bluetooth sub-control equipment, RFID
In terms of game application scenarios: cover the basic function package of puzzle education including word reading and writing, danger perception and basic common sense recognition, and

puzzle games including dungeons and track construction. At the same time, it provides additional scenes and supports the creation of new scenes.

**Scenarios**

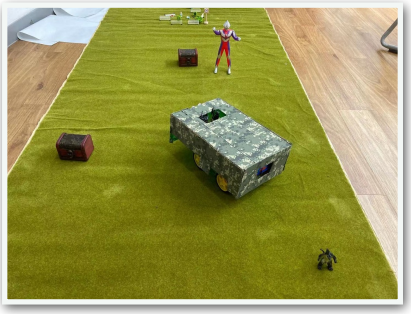
**Educational scene**

Children control "CUBO" to perform operations corresponding to preset traffic signals, such as "straight", "speed" and "stop".

**Entertainment scene: "Castle Road"**

Parents set up the scene and the children control "CUBO" to meet the monster and scan their beacon cards to fight with the monsters.

Educational scene



Entertainment scene

**Selection of main sensors**

Bluetooth interactive system: provide more information sources and interaction possibilities for system perception and execution (e.g. external control device—remote control, external sensing device—Bluetooth traffic light.)

RFID-like geomagnetic marking and interaction design: The RFID magnetic card is used as the main interactive medium, which can carry rich instructions and perceptual information. At the same time, the basic scene can be modified and expanded through simple methods like puzzles and swiping cards.

Multi-scene and perception modular design: with touch, Hall sensor, and ultrasonic sensor control while it realizes the safety guarantee based on children's careless operations and the simple control and perception addition to the creation of multiple new scenes.

LCD display screen and sound device: through video and sound output, increase the feedback that children get in the interaction process, improve the sense of accomplishment and playability

Other sensors: touch sensor, buzzer.

## 3. Prototyping and testing

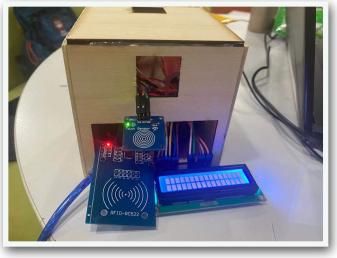
### 3.1 Prototyping methods and physical prototype building process. Computer models and user interface design can be provided if necessary.

**Shell**

Start with the appearance of the car. In order to adapt to the size of the trolley and to be easily modified, the outer shape of the trolley is made of corrugated paper. After cutting, it is glued with hot melt adhesive. Four short wooden strips were inserted in the four corners to maintain the stability of the shell. The two small holes on the top are for inserting the wooden strips through

the holes on the plane of the cart after the shell is put on the cart to make the shell more stable to attach to the cart.
The Bluetooth shell uses six 10cm*10cm wooden boards, one of which is dug with a large hole and a small hole. The large hole is used to expose the LCD and RFID magnetic card sensor sensor, and the small hole is used to expose the touch sensor.

Shell for the car



Shell for the bluetooth

**Code for the decision level**
In the code of the adventure scene, there are a total of four threads. These four threads are used for timing, calculating the changes in blood volume in the fight with monsters, monitoring whether they are within the specified area and accepting documents. When "CUBO" sweeps to the corresponding magnet while driving, the value of its corresponding attribute will also change.
In the road safety scenario, there are also two threads. One thread is for distance measurement to avoid obstacles and alarm. The other thread is used to receive remote control commands.

```
def touch_sensor():
    global touch
    with open(FileName, 'r', encoding='utf8')as fp:
        lists = json.load(fp)
        touch = lists[0].get('val')#return whether it is touched
    return touch


def meet_trace():
    global set_time
    while True:
        try:
            with open(FileName, 'r', encoding='utf8')as fp:
                lists = json.load(fp)
                if(lists[3].get('val')!='mid'):#return whether meet the black area
                    set_time -= 10
                    print("you reach the forbidden area")
                    #redlight()
                    time.sleep(3)
```

```
def beat():
    while True:
        try:
            global count, Monster_hp, ult
            if Monster_hp <= 0:
                count -= 1
                print('you beat the monster')
                #music1
                print("now the count is", count)
                Monster_hp = 1
                ult = 10
                #greenlight()
            if Player_hp < 0:
                event.clear()
                #music2
                #func
        except:
            pass
```

```
def counter(): #设定初始时间 超时判负
    global set_time
    while True:
        set_time -= 1
        time.sleep(1)
```

```
def recv_json():
    while True:
        try:
            global fileMt, Monster_hp, Monster_attack, Player_hp, Player_attack,set_time,touch, mag_G, mag_b, num, ult
            if fileMt != getmtime(FileName):
                fileMt = getmtime(FileName)
                with open(FileName, 'r', encoding='utf8')as fp:
                    lists = json.load(fp)
                    for dic in lists:
                        type = dic.get('type')
                        if type == 'dis':

                            if int(dic.get('val')) <= 5 and int(dic.get('val'))!=dis:
                                dis = int(dic.get('val'))
                                identity = data_G[int(num)].get('identity')

                                if identity == 'monster':
                                    Monster_hp = data_G[num].get('hp')
                                    Monster_attack = data_G[num].get('attack')
                                    print('The hp of monster is',Monster_hp)

                                if identity == 'treasure':
                                    touch = touch_sensor()
                                    if touch:
                                        func = data_G[num].get('func')
                                        if func == 'extra_hp':
                                            Player_hp += data_G[num].get('value')
                                            print('Now the player hp is', Player_hp)


                                        if func == 'extra_time':
                                            set_time += data_G[num].get('value')
                                            print('Now the remain time is', set_time)
```

```
def distance_detector():
    global point
    while True:
        try:
            with open(FileName, 'r', encoding='utf8')as fp:
                lists = json.load(fp)
                distance = lists[5].get('val')
                if distance < 5:
                    point -= 5
                    #yellow light
                    print('Now your distance is', distance)
                time.sleep(3)
        except:
            continue
```

```
def is_trace():
    with open(FileName, 'r', encoding='utf8')as fp:
        lists = json.load(fp)
        return lists[2].get('val') or lists[3].get('val')

def detect_speed():
    with open(FileName, 'r', encoding='utf8')as fp:
        lists = json.load(fp)
        return lists[4].get('val')

def recognise_flag():
    while True:
        try:
            global point, fileMt, stop, straight, trace, speed_limit, detour, point, mag

            if fileMt != getmtime(FileName):
                fileMt = getmtime(FileName)
                with open(FileName, 'r', encoding='utf8')as fp:
                    lists = json.load(fp)
                    for dic in lists:
                        type = dic.get('type')
                        if type == 'mag':
                            if dic.get('val') != mag:
                                mag = dic.get('val')-1
                                identity = data[mag].get('identity')
                                if identity == 'stop':
                                    #car stop
                                    print('you have to stop')
                                if identity == 'straight' and data[mag].get('val') == 1:
                                    trace = is_trace()
                                    if trace:
                                        print('you are not following the trace')
                                        #yellow light, music
                                        point -= 10
                                        trace = False
                                        time.sleep(1)
                                if identity == 'straight_end':
                                    print(11111)
                                    data[mag-1].update({"val": 0})
                                    print('you have finished the straight routine')
                                if identity == 'destination':
                                    justify()

                                #high speed item
```

```
def main():
    t1 = threading.Thread(target = counter,)
    t2 = threading.Thread(target = recv_json,)
    t3 = threading.Thread(target = beat,)
    t4 = threading.Thread(target = meet_trace,)
    t1.start()
    t2.start()
    t3.start()
    t4.start()


if __name__ == '__main__':
    main()
```

**Key sensors and evolution**

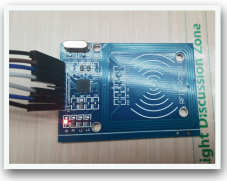
**RFID card**

The magnetic card can perform three functions in this product.

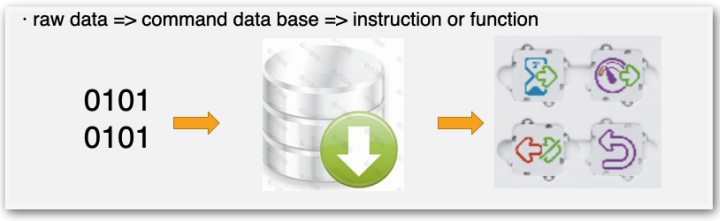
1. Motion control, such as stop the car
2. object representation: For example, detect the object or role named monster or treasure
3. Goal orientation, such as discover the mission to cross the road and the wining condition.

UID and 8-bit data are used in the information transmission of the magnetic card, which can read and write the information to the program. In the process of information transmission, data changes from raw data to command data base, and finally to instruction or function

RFID card

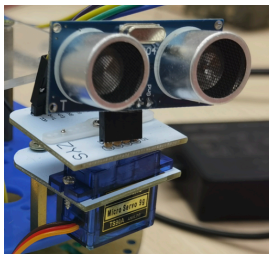


RC-522

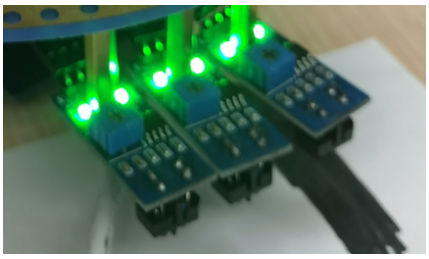



Data transfer process

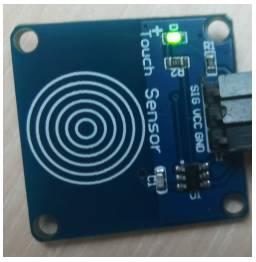
**Other sensors**



ultrasonic sensor
for distance
detection
(Automatic obstacle
avoidance)

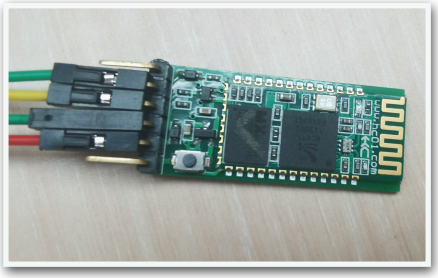


hall sensor with
light sensor
For line tracking

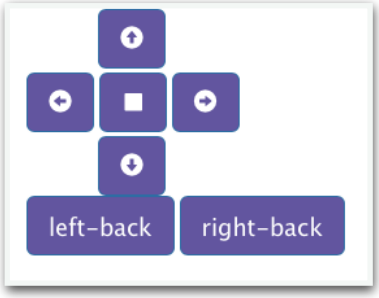



Stress sensor
For touch control
(Emergency
braking)

**Bluetooth and Wi-Fi**
Two functions:
1. Remote device control, such as make the car to stop or make to car to play music
2. Extend the scenario, such as use BT to transport the sensing of traffic light change
Although Bluetooth can realize remote data transmission, the distance is still too short. And Bluetooth only has low bandwidth broadcast. So our team introduced WiFi subnet, WiFi range and high bandwidth transport layer protocol (UDP, TCP)



HC-05 in sub device



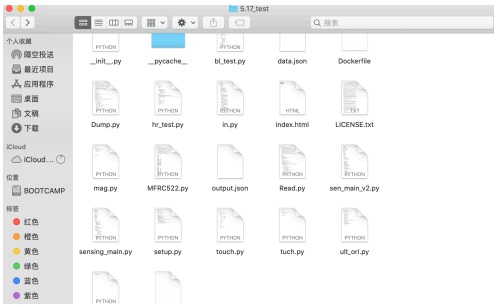

Wifi motion control
interface

**3.2 Testing plan in terms of user needs and design requirements.**
**Block test**
As the code needs to detect every method, the birth of a product also needs to detect every function of it. Before setting up the scene, our team tested the normal function of each sensor environment after it was successfully built according to time. The function of the whole car was tested after the whole car was completed. Finally, a Bluetooth module was added to remotely test the functions of scanning the magnetic card, touch and LCD screen.
In the testing process, it is also divided into two parts of independent testing: decision-making and control. The decision-making part can successfully follow the code logic in the IDE, and outputting some sentences means success at the decision-making level. The control layer alone judges success by the response of the magnetic card or sensor. After the two parts of the code are combined, replace part of the code at the decision-making level to call related methods. If the decision-making level shows that the code runs successfully, and the car can achieve the corresponding movement, it means that the test of a function from sensing to decision-making to control level is successful.

**Overall test**
The overall test means testing all the functions of the entire product in a prepared real scene, including such as the stability of the housing, the trajectory of the car, the speed, the sensitivity and availability of the sensor, the transmission of Bluetooth, the layout and trade-offs of the scene. The scene is a 250cm*100cm turf, divided into front and back sides. There is grass on the front and not on the back.

**Scenario one**
There are road stickers on the reverse side, the black ones are the roads and the white ones are the highways. On the reverse side, parents can place street signs according to their needs, such as speed signs, traffic lights, zebra crossings, and pedestrians. The child needs to drive the car from the designated starting point to the end point, and control the car or observe the behavior of the car according to the road signs and magnetic cards placed on the ground. The specific performance is as follows: when the speed sign appears, the car can drive, and the child needs to use the remote control to control the speed of the car when passing the zebra crossing. When there is a pedestrian, the car needs to move left and right to avoid it. If the avoidance is not timely, the touch sensor can be used to brake the vehicle urgently via Bluetooth remotely. When cornering, the child agrees that the remote control needs to be used to operate the car to turn. And in some special places where children can be educated, such as traffic lights, magnetic sheets will be placed on the ground in advance. When the car passes the magnetic sheet and reads the information that there is a traffic light ahead, it will stop and wait for a certain number of seconds, and then move forward a corresponding distance. In order to ensure that this section of the driving car can not deviate from the road, we also added a tracking function. The car can keep the car in the middle of the road according to the black line on the ground. When the car arrives at the destination, it reads the magnets placed on the destination ground. Then the car will play music to celebrate the child's successful arrival. During the process, the car will maintain ultrasonic ranging. If there is an obstacle in front and the car that is less than a safe distance, the car will flash a yellow light.

**Scenario two**
Scene 2 used the front side of the turf, which is the side with the grass. Parents can place treasure chests and monsters arbitrarily on this large area, which gives parents a sense of participation. The information of each treasure chest and monster is stored in the magnetic card under them. The child manipulates the car through the remote control and WI-FI, opens the treasure chest it needs and defeats the monster within the specified time, and finally reaches the end-"beautiful town". If the child fails to reach the end within the specified time or fails to gain enough combat power by opening the treasure chest, and fails in the duel with the monster, the game will fail. Every time the car reads the information a magnetic card,
The information will be sent to main.py file in the form of json file, and then the corresponding words will be output on the LCD screen.




All code

**3.3 User feedbacks and result analysis.**

We interviewed a children's programming teacher, he said, “As a children's programming teacher, I think such products can be well used in the classroom, enrich the programming scene, make the teaching more simply.”
Some roommates and friends of our group members participated in the experience and evaluation of this product. They experienced two scenes respectively, changed the corresponding prop position, and the car can move according to the set rules.
In summary, interviewees and experiencers think this product is interesting because it has enough sensors, interesting scenes, props and shells, selectable scenes, diverse control and response methods, and quick-reading responses. speed. In addition, "CUBO" can teach children basic traffic safety knowledge in educational settings. Activate the curiosity of children in the adventure scene. But at the same time, if the playability is improved again, having fewer and more useful props and scenes is a point that we can improve in the future.

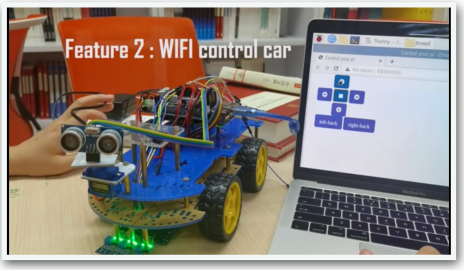
# 4. Conclusion

## 4.1 Conclusion of design process, final design performance and further recommendation.

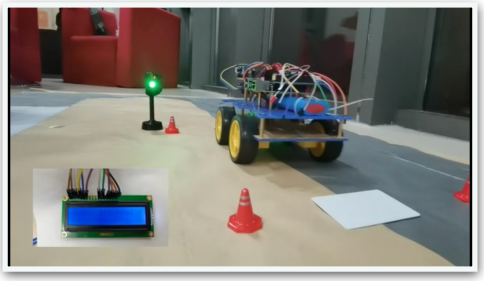
**Design process**

At the beginning of the project, our team held many discussions on the selection of specific products. During the period, through the selection of target users, discussions with teachers, and product innovation discussions, our team finally decided to make a smart car positioned as an Educational intelligent toy, named "CUBO". We believe that the integration of cultural and creative products with the subject of this course is a novel perspective. In the middle of the project, as mentioned in 3.1, our team also encountered a lot of difficulties in the code part. Because we mainly use python as a programming language, in addition to importing sensor-related libraries, we also need to be proficient in using python. As time goes by, the car receives commands and information from the magnetic card from the smart card, and slowly has more successfully tested sensors. The introduction of Bluetooth also allows "CUBO" to be remotely controlled. In the choice of test scenes, our group did not know how to make better use of these sensors to realize the function of a toy. In Week9, a scene about traffic safety was proposed, which is considered a good scene by us. But only one scene of a toy lacks playability. In Week10, the second scene of the castle adventure was confirmed. It gives parents and children a high degree of freedom and can be regarded as a true embodiment of some online adventure games. In the layout of the scene, we also purchased an appropriate amount of props. In terms of the outer shell, the outer shell of the car is made of corrugated paper, which is fixed with a wooden stick on the inside and camouflage stickers on the outside. And we also give the bluetooth device a wooden case. Therefore, the entire toy, including the car, scene and Bluetooth are packaged and then can be considered a product.
To sum up: The car has sufficient and concise interaction with the user, which is entertaining and entertaining. Technically, multiple sensors are used to realize the corresponding functions. The combination of these sensors and the car not only makes the operation easier, but also gives users enough experience and fun in the sense of the scene and the product. Python is mainly used to implement the code of the car function.

**Design performance**





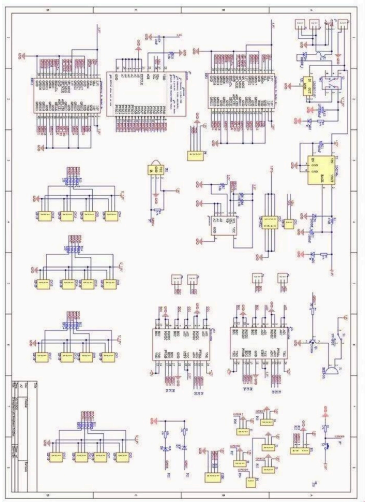


**Further recommendation.**
Obviously, due to limited technology and materials, this toy is not beautiful enough in appearance. In addition, the scenes and usage methods are more cumbersome. But this toy can be considered a very good prototype. First of all, this toy can have more advanced sensors and equipment, such as cameras, better sound playback devices, and a smaller body. More equipment means more functions can be achieved. For example, the camera can allow the car to observe the road and the scene intuitively, and the use of machine learning can allow it to actively identify objects. In addition, "CUBO" can have more application scenarios, for example, when it is not used as a toy, it can be used as a patrol car to patrol at home. During the patrol, you can monitor the children's activities and the safety of the house to avoid some bad things that may happen. The third point is that if "CUBO" wants to be a better product, it needs a system of parts and scenes. Props and lawns can be encapsulated in a box, and some 3D scenes and smarter props can also be added to it. In this way, "CUBO" can give children a more immersive experience and greater autonomy. In addition to these three points, "CUBO" can also have a branch for the transmission of items between family members, air monitoring and alarms, and it can be regarded as a light and efficient "robot" car.

**4.2 Detail individual contribution description.**
Xinyue.Wu: Code for the control part of the car, the wiring and connection of the circuit; the setting of sensors; video.
Qinxin.Ren: most of the code for car decision-making; car testing and debugging; product creativity; car assembly; video
Pai.Peng: Code for the car's perception level and part of the control level; debugging and setting of sensors; video; team leader
Yangzhi.Gao: Design of the product, the scene and the production of the car shell, part of the decision-making layer code; poster; writing of the report.

# 5. Appendices

## Appendix A Engineering Drawings



## Appendix B Components' name and price list

| Name | Price |
|---|---|
| Raspberry pi | ¥367 |
| Main board | ¥370 |
| HC-06 | ¥82 |
| Turf | ¥76 |
| Small materials | About ¥150 |
| IC card | ¥11 |

## Appendix C Individual Contribution Form

| Student Name/ID | Student Name/ID<br><br>Contribution percentage<br><br>(100% in total) | Brief description of individual contribution |
|---|---|---|

| | | |
|---|---|---|
| Pai.Peng | 25% | Code for the car's perception level, sensors and part of the control level; debugging and setting of sensors; video; team leader |
| Yangzhi.Gao | 25% | Design of the product, the scene and the production of the car shell, part of the decision-making layer code; poster; writing of the report. |
| Qinxin.Ren | 25% | most of the code for car decision-making; car testing and debugging; product creativity; car assembly; video |
| Xinyu.Wu | 25% | Code for the control part of the car, the wiring and connection of the circuit; the setting of sensors; video; car assembly |

**Appendix D Group meeting minutes and individual student's logbook**
2-3 times about 10 hours per day. Our group's logbook for some scenes is below: