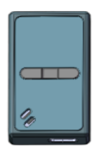
Xi'an Jiaotong-Liverpool University
西交利物浦大学

MEC202

INDUSTRIAL AWARENESS AND GROUP PROJECT

SMART DESKTOP ORNAMENT

Group Number	No.5	
Group Member 1	Siqi Chen	1931101
Group Member 2	Nuo Chen	1931017
Group Member 3	Yuxin Qian	1930038
Group Member 4	Jingyi Wang	1929591
Group Member 5	Wenbo Yu	1929603
Submission Date	24th May 2022	


Catalogue

1. Introduction

At the end of 2019, the covid-19 swept the world, seriously affecting the health of human beings and the global economic development. Due to the repeated outbreaks of the epidemic and it is difficult to be controlled quickly and effectively in a short period of time. In order to protect health and safety of public and prevent a large-scale outbreak of the epidemic, most schools carry out online teaching, and some companies also implement online office. We interviewed students at various studying stages in the form of a questionnaire survey, and learned that some of them can quickly adapt to such special situations and maintain efficient learning efficiency, however, there are still a large number of students who have been greatly affected and have difficulty staying focused on learning. In addition, with the recurrence of the epidemic, the time people at home have been extended irregularly, and many people also feel that home study is very boring, lack of communication between peers, and lose the interactive experience of onsite teaching.

In this context, our group tried to design a smart desktop gadget to help students better adapt to online teaching. The results of the previous investigation indicated that the main problems faced by the students are showed in **Figure 1**.

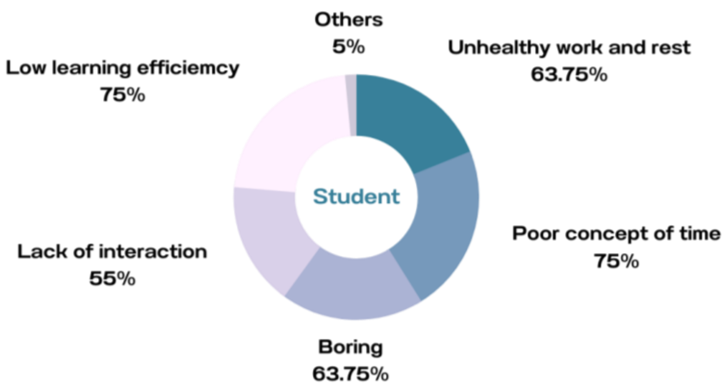


Figure 1 Online Study Difficulty Survey of Students.

Therefore, based on these questions, we studied the existing product prototypes. For example, the mobile app Pomodoro is a software that many students used, which can monitor study time of students and establish online study rooms. However, we hope that students can get away from their mobile phones. Various information and app notifications will affect their concentration in learning, so we try to transplant this function into our products, so that students can get rid of the control of mobile phones. At the same time, there are similar desktop gadgets or smart home robots such as Xiao-Du smart screens, whose functions are very complete at present, and they can realize the interaction of different groups of users, such as storytelling for children, voice commands and so on. Therefore, we hope to produce more innovative ideas by targeting the user group more specifically to students, then propose the idea of online proctoring equipment.

Based on this idea, we conducted a questionnaire survey with 20 teachers who are currently teaching online. The results showed that most teachers also feel that the quality of online teaching is difficult to improve. It is believed that there are loopholes in online invigilation, and there is no way to pay attention to the status of students in all directions to ensure the fairness of the test.

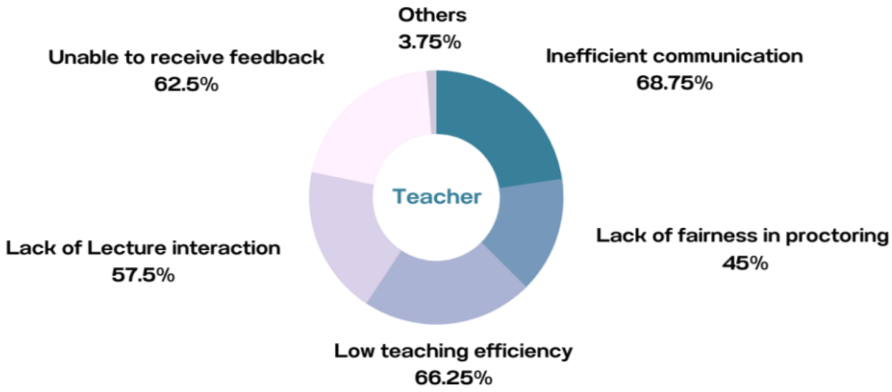


Figure 2 Online Teaching Difficulty Survey of Teachers.

Therefore, we have determined this idea and defined this device as a medium for communication between students with their classmates and teachers which expected to help solve the problem of boring and inefficient online learning, and help teachers pay better attention to learning situation of students. Moreover, through the remote control of the camera, it can reasonably provide better invigilation conditions and improve the fairness of online examinations.

Our plan is that two students will cooperate in the simulation part, two team members will assist in the construction of the physical product, and one student will conduct user research, 3d modeling and visualization. At the same time, we will conduct functional comparison and exploration together, as well as report writing and demonstration video production.

2. Design Concept Comparison, Development and Screening

After constructing the basic product vision, we started brainstorming what features to add, screening through user research, and discussing whether it was feasible to implement. In the functional demand survey, our objects are still students studying at home and teachers who teach online, and we use matrix scale questions to obtain the demand of requirement for various functions. The results are shown in **Figure 3**, which indicated that the research objects have relatively higher demand for the functions of Pomodoro, online study room, online proctoring, and reminder of work and rest time. Therefore, we decided to keep these functions in the product and consider the implement ability. If some functions cannot be realized during simulation, we will try to show them in real objects as much as possible.

Compare the four mainline features identified with existing concepts and discuss how to advance and implement the preset features.

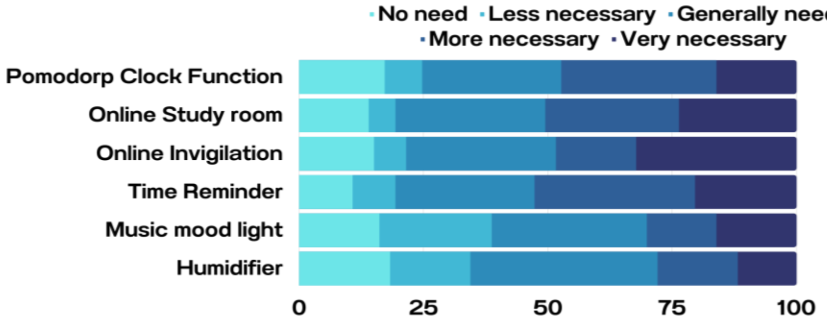


Figure 3 The result of the functional requirement degree.

First of all, we learned that in the existing products, there are already relatively professional eye tracking and recognition functions for online invigilation. However, eye tracking and recognition is too complicated to realize in Arduino board, so we choose a combination of rotatable camera and remote control, which is suitable for more scenarios. It can freely control the angle of the camera according to the needs, help the other party to observe the situation from arbitrary viewing angle, so that it assists the teacher in invigilating the exam. At the same time, it can be used as an auxiliary tool for online study or meetings with friends, because it has a more flexible and convenient camera device than a computer. In this way, you can not only find a better video perspective, but also let the person on the opposite side feel the environment on your side when necessary, thereby reducing the sense of distance between people.

Secondly, the Pomodoro clock function and the work or rest reminder function are all small parts of the normal mode. According to the concept of the Pomodoro app compared earlier, our product hopes to realize the countdown of the learning time, meanwhile, let students get rid of the constraints of mobile phones and improve their concentration in learning. Just use the pressure sensor to set the time and display the countdown on the screen. As for the work and rest reminder function, we regret that the analog part cannot implement this function for it requires a network connection, so we will show this part in the physical prototype.

Finally, we summarize three modes: **invigilation mode, normal mode and learning mode** to include the functions we design. The function description of the product is shown in **Figure 4**.

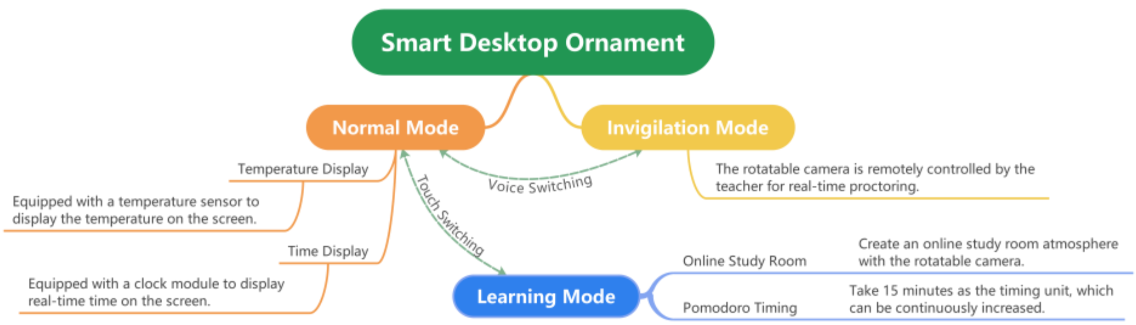


Figure 4 Function branch flow chart.

3. Simulation Prototyping and Testing

Our solution consists of two components: **Desktop Ornament** and **Wearable Remote Controller**. The two components are connected by the Internet. For each component, the hardware design, mechanical design, software design will be introduced, and the simulation and physical verification results will be shown in the corresponding section.

3.1 Desktop Ornament

This section will introduce the fabrication and testing of the desktop ornament prototype at several levels: mechanical design, hardware design, software design, and implementation in different dimensions of simulation and physical objects, where the platform used for simulation is TinkerCAD.

3.1.1 Mechanical Design

Figure 5 is the perspective view and exploded view of the 3D model of the desktop ornament, including all the elements used, of which key components such as sensors have been marked in detail.

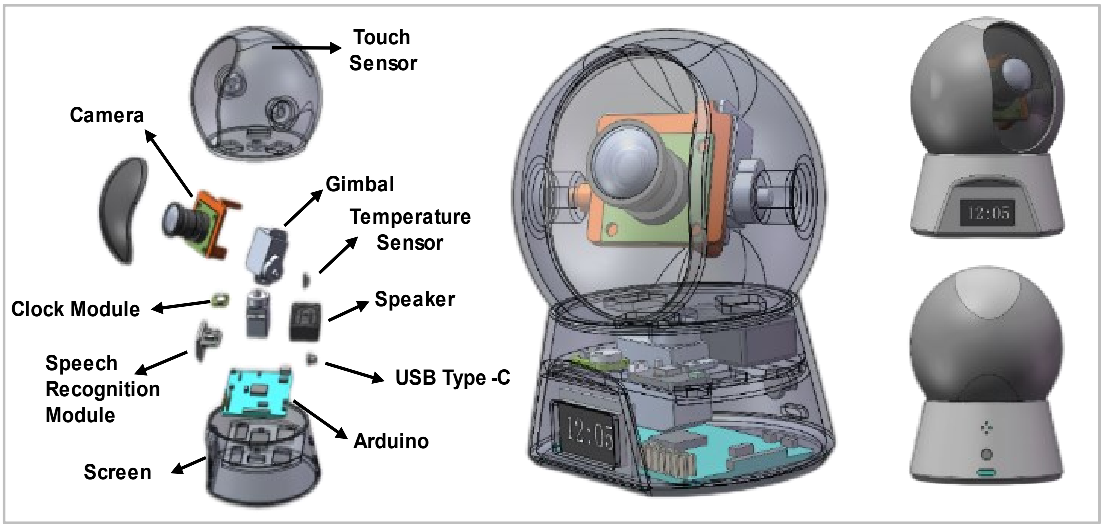


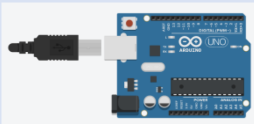
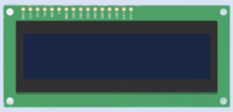
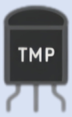
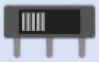
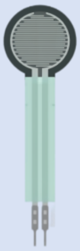
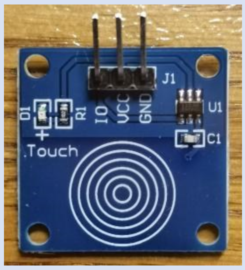
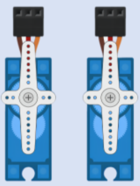
Figure 5 Mechanical design of Desktop Ornament

The above structure picture consists of three main parts. First, the leftmost part of the picture shows an exploded view of the 3D model, where the specific names of the main electrical components and are exhaustively labeled. Secondly, in the center of the picture is a perspective view of the 3D model, depicting the exact location of each component. Then, on the far right is a drawing of the final product model, the main part of which is a desktop monitor camera with a rotatable head and a screen display mounted underneath.

3.1.2 Hardware Design

The hardware part of the Desktop Ornament includes Arduino UNO microcontroller, speech recognition module, two servo, LCD screen, and serval sensors. A comparison of selected components and their images in simulation and physical prototype is shown in **Table 1.**

Table 1 Experimental device summary table

Simulation		Physical prototype	
Device name	**Device image**	**Device name**	**Device image**
Arduino Uno R3		Arduino Uno R3	
LCD 1602		LCD5110	
Temperature Sensor [TMP36]		Temperature Sensor [DS18B20]	
Slide switch		Speech Recognition Module	 
Force Sensor		Capacitive Touch Sensor TTP223	
Micro Servos		Two-axis Gimbal	
		Clock Module DS1302	

It can be seen from the above table that the realization of the same function of the simulation and the physical prototype depends on the electronic components that are not exactly the same. The reason is mainly due to the component limitations of the simulation platform, so the selection of physical devices also takes into account the performance considerations compared to the devices in the simulation. The reasons for choosing the devices listed in the table above are briefly stated next.

First of all, the temperature sensors used in the two parts are different. The TMP36 used in the simulation part requires additional code to calculate the sensing data to obtain the final temperature, while the DS18B20 in the physical part can present the ambient temperature without additional data processing. Moreover, there are differences between the simulation and the physical prototype for the display screen selection. Compared with the simulated LCD1602, the LCD5110 used in the physical prototype has more advantages in the information display. In addition, since speech recognition cannot be completely simulated in the simulation platform, the slide switch is used to replace the speech recognition module in the simulation part to realize the switching between normal mode and invigilation mode. It is worth noting that the actual gimbal is mainly composed of two microcomputer servos, so it is reasonable to use two microcomputer servos to simulate the function of the gimbal in the simulation part. Finally, there is no clock module in the simulation platform, so this functionality is only shown in the physical prototype.

3.1.3 Software Design

In order to maximize development efficiency at the software level, it is necessary to understand and integrate user requirements. Therefore, before the formal development, the following use case diagram was obtained by analyzing and summarizing the requirements of the surveyed population.

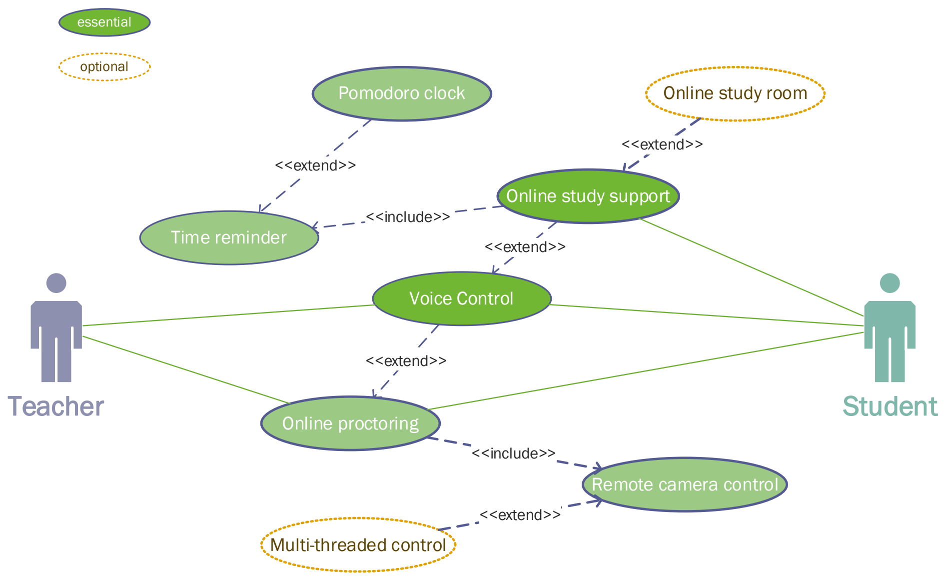


Figure 6 Use case diagram

As can be seen from the figure above, in the online learning scenario, both interviewed groups, students and teachers, made reasonable demands for building a fair, transparent environment for online exams. Besides, many students felt it would be nice to have a desktop ornament to help them improve their study efficiency. In addition, the language control function was favored by many; however, due to the functional limitations of **TinkerCAD**, this module will be covered in the physical part, with a switch as replacement in the simulation part to demonstrate the code logic.

After refining the key requirements, the software model was grouped into 3 components, Normal mode, Learning mode and Invigilation Mode.

Then, a rough topological model was obtained by estimating the complexity and convergence of each important module, as follows, where the **Learning mode** is grouped as a sub-mode of the **Normal mode** while the **Invigilation Mode** is assigned as a separate module.

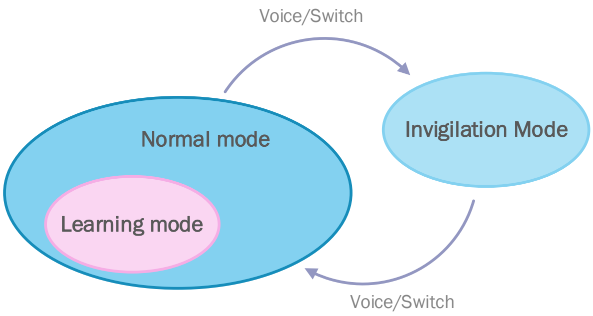


Figure 7 Topological model

After meticulous analysis, several major system states were finally filtered out and modeled using a state machine diagram as the initial model, as shown in the figure below. For clarity, we specify several terms commonly used in simulation.

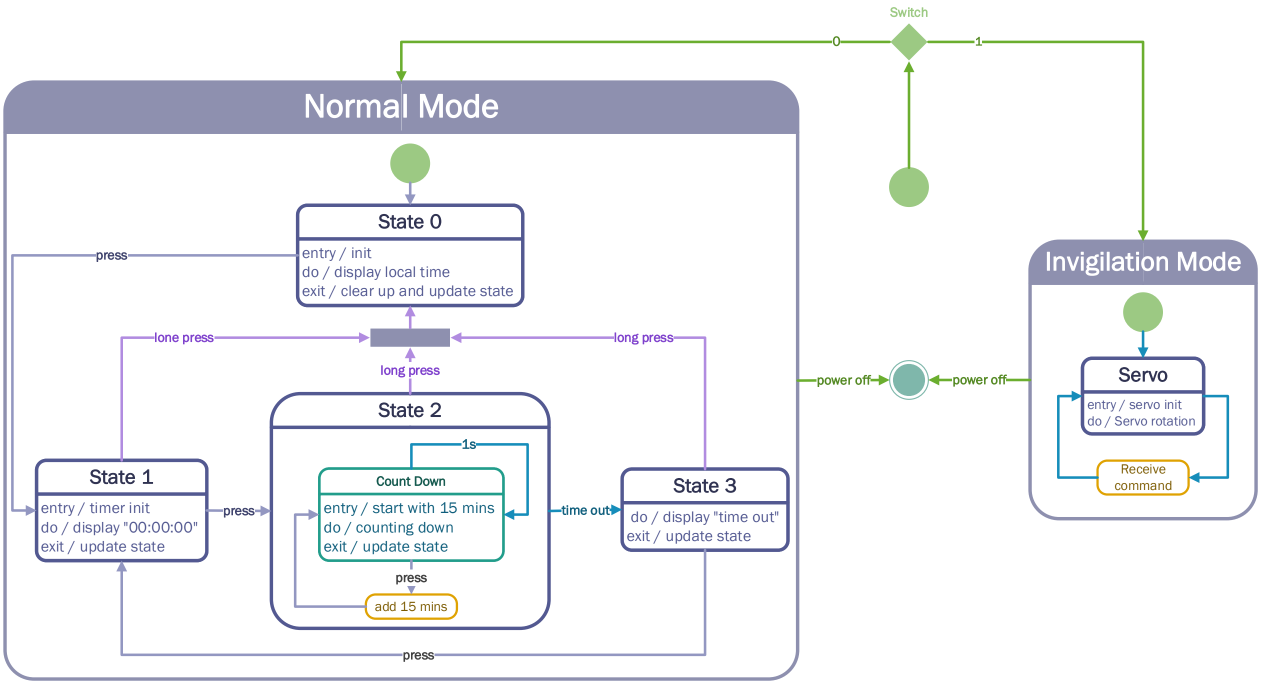


Figure 8 State machine diagram

State: parameter to record the current status of the system, stored with *int* type data.

Command: A general term for all inputs of signals from outside the system, including inputs from various sensors, serial inputs and so on.

Action: A general term for all feedback or actions performed by the system, determined by both **Command** and **State**.

The logical relationship between **State** and **Command** can be found in the following table.

Table 2 State transition table

Present State	Sensor Reading	Action	Next State
0	0	Display temperature	0
	1	State change	1
1	0	Start	2
	1	Exit	0
2	0	Add time	2
	1	Exit	0
3	0	Clear	1
	1	Exit	0

- **State** 0: Default state in **Normal mode**, and the display local time as well as the display temperature are set to **action** limited to this state.
- **State** 1: The initial state after entering the 'tomato clock' mode, during which the display will default to '00:00:00' until a new **command** input is received.
- **State** 2: Countdown state, the display will keep refreshing the remaining time until the timeout, during this period, it can receive the **add time** and **exit** command.
- **State** 3: Timeout state, the display will show the 'timeout' character until a **reset** or **exit** command is received.

Up to this point, the design phase of the software has been basically completed, and the following will introduce the specific function implementation through two class diagrams. As shown in the figure below, the system contains two main classes, *State* and *Tomato*, which cover the methods for implementing mode switching and the Pomodoro in **Learning mode**, respectively.

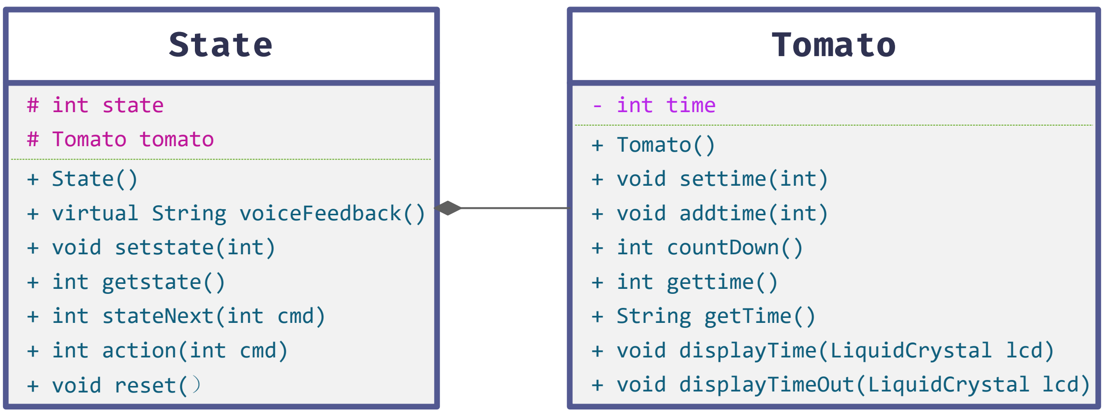


Figure 9 Class diagram

The specific code of these two classes is attached in appendix part.

In addition, to enhance scalability, some interfaces can be reserved in the *State* class (implemented with the *virtual* keyword in C++), for example, in order to link with the voice module (covered in other parts) to provide a voice feedback, you can add the *voiceFeedback()* method to the *State* class, as shown in the code above, to deal with the state information.

To further illustrate, a simple example will be given. In practice, the Pomodoro should alert the user with a "timed out" tone after the timeout, rather than just displaying the "time out" character on the screen as shown in the simulation. Of course, simply using a buzzer to alert is a simple and conventional solution, but this may limit further interaction, so replacing the traditional buzzer with a module with speech recognition is an appropriate and achievable approach, which also coincides with user requirements.

Assuming that the voice module is encapsulated in the *Voice* class, as follows, the interface provided by the*State* class allows it to access the **state** of the system and give the corresponding feedback.

```
class Voice : public State
{
public:
    string voiceFeedback() {
        string feedback = " ";
        if (State::state == 3){
            feedback = "Time out!";
        }
      return feedback;
    }
};
```

The code for the **speech module** is shown in Appendix.

3.1.4 Simulation Verification

The simulation circuit diagram of the basic functions of the desktop ornament is shown in **Figure 10** below, and the names of the electronic components used are labelled. The desired angles of the two servos can be set by sending two integers to serial.

The simulation in TinkerCAD is available at:
https://www.tinkercad.com/things/1IwB74rEleN-mec202group5desktopornament/editel?sharecode=otirStnxbptPCjblfyf_WN0zVwEmkdT5wEPa1-Jicbw

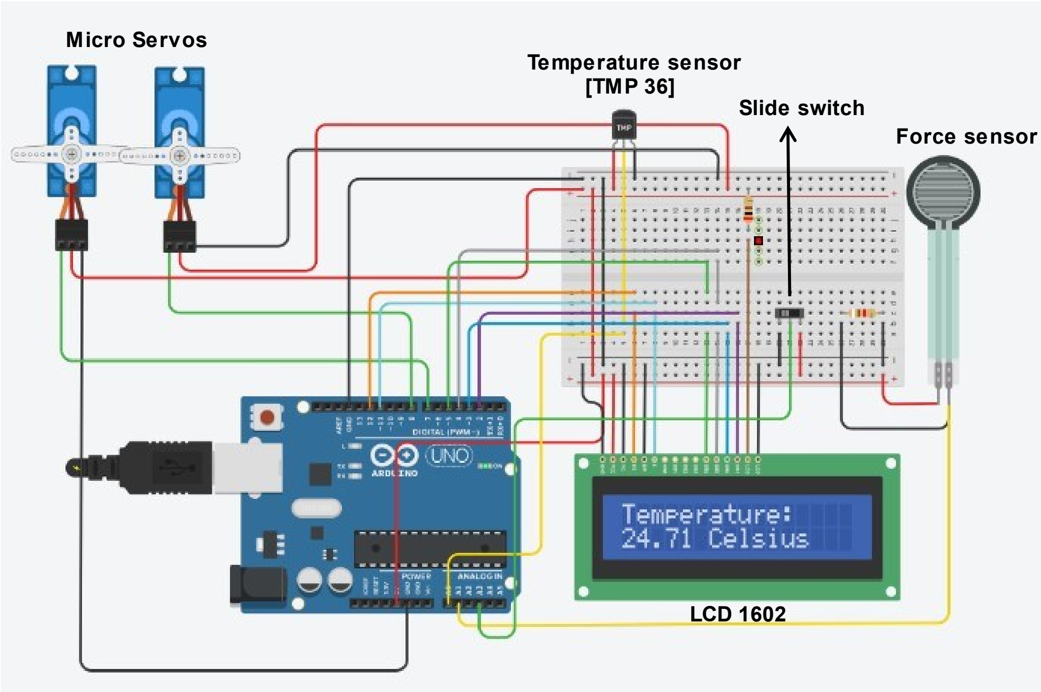


Figure 10 Simulation circuit for Desktop Ornament

1) LCD1602:

Liquid crystal displays are widely used in control display panels, called Liquid Crystal Display, or LCD for short, and are ideal displays for various portable electronic products. LCD1602 is a dot-matrix liquid crystal module specially used to display letters, numbers, symbols, etc.

Table 3 Pin connection table of LCD1602

Pin (LCD1602)	Wiring principle	Pin (Arduino)
VCC	VCC is connected to the positive pole of 5V power supply.	**5V**
GND	GND is the power ground.	**GND**
VO	VO is the LCD contrast adjustment terminal, the contrast is the weakest when the positive power is connected, and the highest when the grounding power is on.	**GND**
RS	RS is for register selection. Data register is selected when the high level is 1 and instruction register is selected when the low level is 0.	**D12**
RW	RW is a read-write signal line, read operation at high level (1), write operation at low level (0).	**GND**
E	The E (or EN) end is the enable pin. It reads information when the level (1) is high and executes instructions when the negative jump changes.	**D11**
DB0~DB7	DB0~DB7 are 8-bit bidirectional data pins. In the connection, only DB4~DB7 are connected to the D5~D2 pins of Arduino to realize data I/O control.	**D5~D2**

LED+ / LED-	Backlight power supply. LED+ is connected to 5V in series with a 200Ω resistor. LED- is connected to GND.	**5V / GND**

2) TMP36:

The temperature sensor generates a varying voltage signal according to the temperature it senses. The output voltage of TMP36 is proportional to the temperature in Celsius. The analog output of TMP36 is connected to A0 pin of Arduino.

3) Micro Servo:

The micro-servo requires a 50Hz PWM signal as the setpoint of angle. The duration in range from 1000μs to 2000μs indicates the desired angle. The signals of two servos are generated by pin 7 and 8 on Arduino.

4) Force Sensor:

The force sensor used in the experiment is a force-sensing resistor (FSR). The resistance of an FSR depends on the pressure that is applied to the sensing area. The higher the pressure applied, the lower the resistance. The resistance range is quite large: from 10MΩ (no pressure) to 200Ω (max pressure). Most FSRs can sense force in 100 g to 10 kg. One resistor terminal is connected to pin A1 with a 9.1k pull-down resistor, while another terminal is connected to Vcc. Hence the force of the sensor can be calculated by measuring the voltage by pin A1.

5) Slide switch:

A slide switch is used to switch a circuit by sliding its switch handle to switch it on or off. The current switch state can be read from pin A3.

3.1.5 Physical Prototype Verification

This section describes the selected electronic components and wiring of the physical prototype. **Figure 11** below shows the physical prototype of the desktop ornament, the named of electronic components used are labelled on the figure.

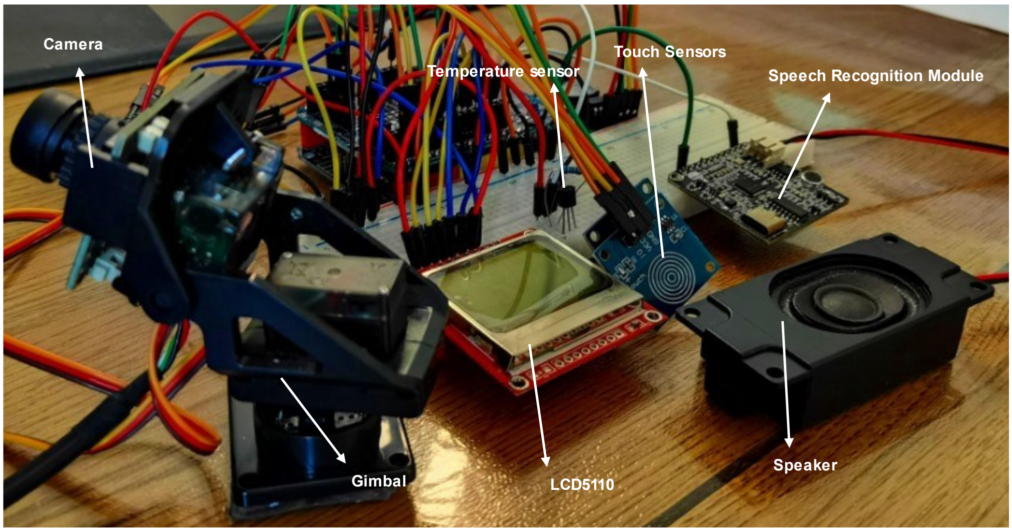


Figure 11 Physical prototype of Desktop Ornament

1) LCD5110:

Nokia 5110 LCD [1] adopts serial communication interface to communicate with the main controller, with 84×48-pixel-resolution, working voltage of 2.7 ~ 3.3V, current of 240 μA under the normal working mode, especially suitable for portable mobile devices powered by battery.

Table 4 Pin connection table of LCD5110

Pin (LCD 5110)	Wiring principle	Pin (Arduino)
LED	Backlight	**3V3**
SCLK	CLK pin	**8**
DN	Data input	**9**
D/C	Data and command switching pin	**10**
RST	Reset pin	**11**
SCE	Chip selection pin	**12**
GND	Common Ground	**GND**
VCC	Power Supply	**5V**

2) Temperature sensor (DS18B20):

DS18B20 is commonly used as digital temperature sensor. It has the characteristics of small volume, strong anti-interference ability and high precision. With a unique single line interface mode, DS18B20 only needs single serial wire to realize the two-way communication between microprocessor and DS18B20. In addition, its temperature measurement range: $-\mathbf{55}\ ℃ \sim +\mathbf{125}\ ℃$. The serial wire is connected to pin 7 on Arduino.

3) Touch Sensor:

The capacitive touch sensor module is based on a special TTP223 touch sensor IC. The module provides an 11 x 10.5mm integrated touch sensing area with a sensor range of approximately 5mm. When this sensor is triggered, the module output will switch from its low level to high level. The output of the touch sensor is not related to pressure, but to charged substances, including human skin. When a finger hits the screen, a small charge will be transferred to the finger to form a circuit, generating a voltage drop at this point of the screen. The two touch sensors are connected to A1 and A2 pins of Arduino.

4) Gimbal:

The two-axis gimbal used in this project is mainly composed of two servos on pitch and yaw axes. The type of used servo is SG90. They can be controlled by PWM. The length of the pulse determines the turning degree of the servo motor.

Table 5 Pin connection table of gimbal

Pin (Servo)	Wiring principle	Pin (Arduino)
Vcc	Power Supply	**5V**
GND	Ground	**GND**
PWM	Data input	**5,6**

5) Speech Recognition Module:

As mentioned before, due to the limitations of the simulation platform, this module is mainly shown in the physical prototype. In this part, the design idea will be introduced.

In terms of speech recognition function, we used the speech recognition module, which has the ability of automatic speech recognition and have its specific image programming software. In order to realize the function of switching state, we connect IO1_1 of speech recognition board with Pin13 of Arduino Uno, and the output from the speech recognition board will control the switching of the using mode.

Table 6 Pin connection table of Clock module

Pin	Wiring principle	Pin (Arduino)
IO1_1	Data output	**13**
GND	Ground	**GND**

6) Clock Module [DS1302]:

DS1302 [2] contains a real-time clock / calendar and 31-byte static RAM, which communicates with MCU through a simple serial interface. The real-time clock / calendar circuit provides information of seconds, minutes, hours, days, weeks, months and years. Clock operation can be determined in 24- or 12-hour format through AM / PM instruction.

Table 7 Pin connection table of Clock module

Pin (DS1302)	Wiring principle	Pin (Arduino)
Vcc	Power Supply	**5V**
GND	Ground	**GND**
CLK	Serial Clock	**2**
DAT	Data output	**3**
RST	Reset	**4**

3.2 Wearable Remote Controller

The Wearable Remote Controller is designed for the invigilator or someone who want to control the Desktop Ornament through the Internet. The remote controller senses the motion of itself using MPU6050 and clones its attitude to the camera of Desktop Ornament. The remote controller can be not only held in hand, but also mounted on the ear. This requires it to be small, light and wireless. **Figure 12** shows the hardware, software and mechanical design of this remote controller.

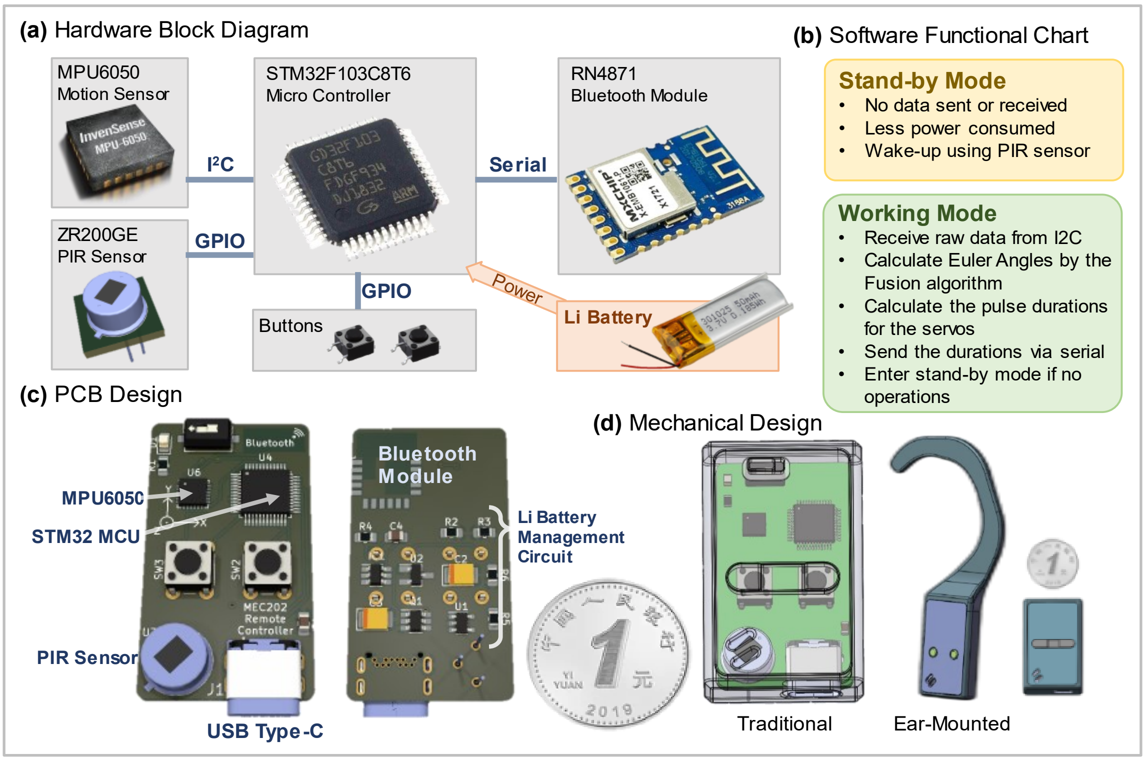


Figure 12 Design for the wearable remote controller, **(a)** hardware block diagram, **(b)** software functional chart, **(c)** PCB design, and **(d)** mechanical design.

3.2.1 Hardware Design

To obtain a small and light device. we selected some sensors, STM32 micro controller and Bluetooth module and integrated them in one PCB board. The names of selected ICs and the used communication prototypes are shown in Figure 12 (a). Note that the sent Bluetooth message is received by the investigator's PC and finally sent to student's PC via Internet. The key press events are also sent to investor's PC. To achieve the wireless, the Li-battery is used, and a charging and protecting circuit is added in our hardware design. The battery can be charged from 5V power supply of USB.

The PCB is designed by KiCAD software, and the schematic is attached in the Appendix part. Figure 12 (c) shows the completed PCB 3D images. For size comparison, a 1-yuan coin is shown in the figure. Due to the lockdown, we did not fabric this PCB.

3.2.2 Software Design

The functional chart for software is shown in Figure 12 (b). The software is usually running in stand-by mode to minimalize the power consummation, and to avoid the miss-operations. In the standing-by mode, the STM32 runs also in low-power mode, while the MPU6050 motion sensor and Bluetooth module are powered off.

If the motion is detected by PIR sensor, then the remote controller is woke-up. It firstly receives the 6-axis motion data (3-axis angular speeds, 3-axis accelerations). Then it calculates the Euler angles (pitch, yaw and roll angles) by using an open-source algorithm [3]. Finally, the software maps the angles to the PWM pulse durations, the integer from 1000 to 2000, and send to the Bluetooth.

3.2.3 Mechanical Design

The mechanical design of the remote controller is shown in Figure 12 (d). The design consists two forms: handheld and ear-mounted. In the real product, there two forms may be easily converted to each other. For PIR sensor, there is a window on the top surface of the structure. Due to the lockdown, we did not 3D print this model.

3.2.4 Simulation Verification

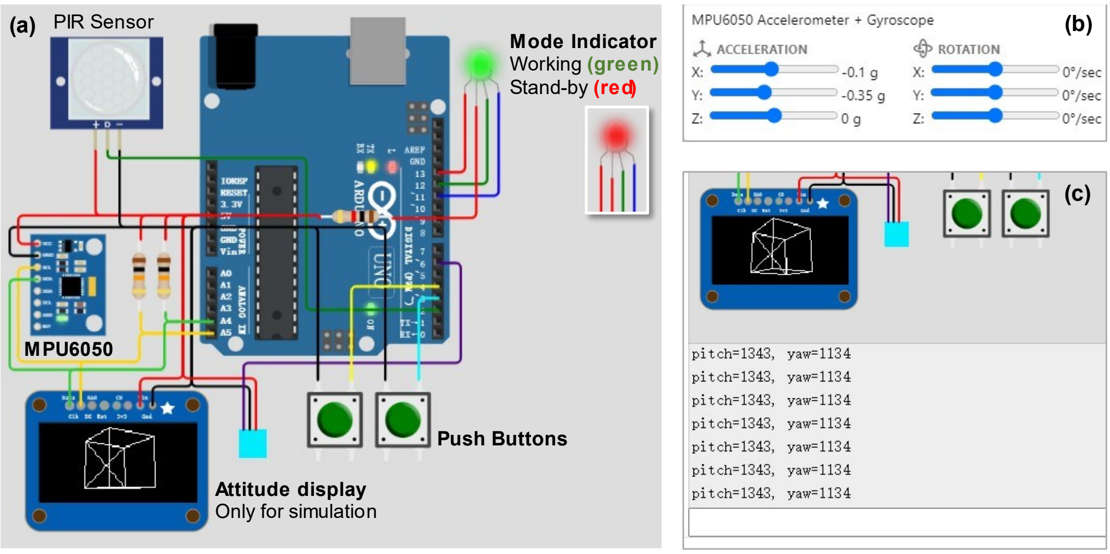


Figure 13 Simulation verification for the wearable remote controller, **(a)** the simulation circuit, **(b)** MPU6050 control panel, **(c)** the serial output message.

The online simulation is completed by using WOWKI. We used Arduino micro-controller instead of STM32 for easier simulation. An additional OLED screen (SSD1306, 128×64) is added in the circuit to display the current attitude of the remote controller, as shown in **Figure 13 (a)**. The control panel of the input of MPU6050 is shown in Figure 13 (b). We can set the values of the 6-axis inputs. A simpler algorithm is running in Arduino to obtain the Euler angles. An additional RGB LED is also added in the simulation to indicate the current status (green for working, red for stand-by). The serial output is shown in Figure 13 (c). This message is expected to be received by investigator's PC and then transmitted to the student's PC via Internet. The range for the two integers in each line is from 1000 to 2000.

The simulation is available at: https://wokwi.com/projects/332089614566163026

3.2.5 Physical Prototype Verification

The designed PCB board is not fabricated due two the lockdown. We chose an existing board, RoboMaster STM32 Development Board, to complete the physical prototype verification. A MPU6500 motion sensor is integrated on this type of board. The micro controller on the board is STM32F427. The wireless Bluetooth communication between the remote controller and PC is instead of wired USB serial. The PIR sensor and stand-by mode are not implemented in this physical prototype. In addition, two python socket scripts are programmed to transmit the data from RC to Desktop Ornament via Internet. In the perspectives of RC and Desktop Ornament, the PCs and the Internet are like a pipe. How the pipe implemented is transparent to the RC and Desktop Ornament. **Figure 14** shows the dataflow and the investigator's perspective.

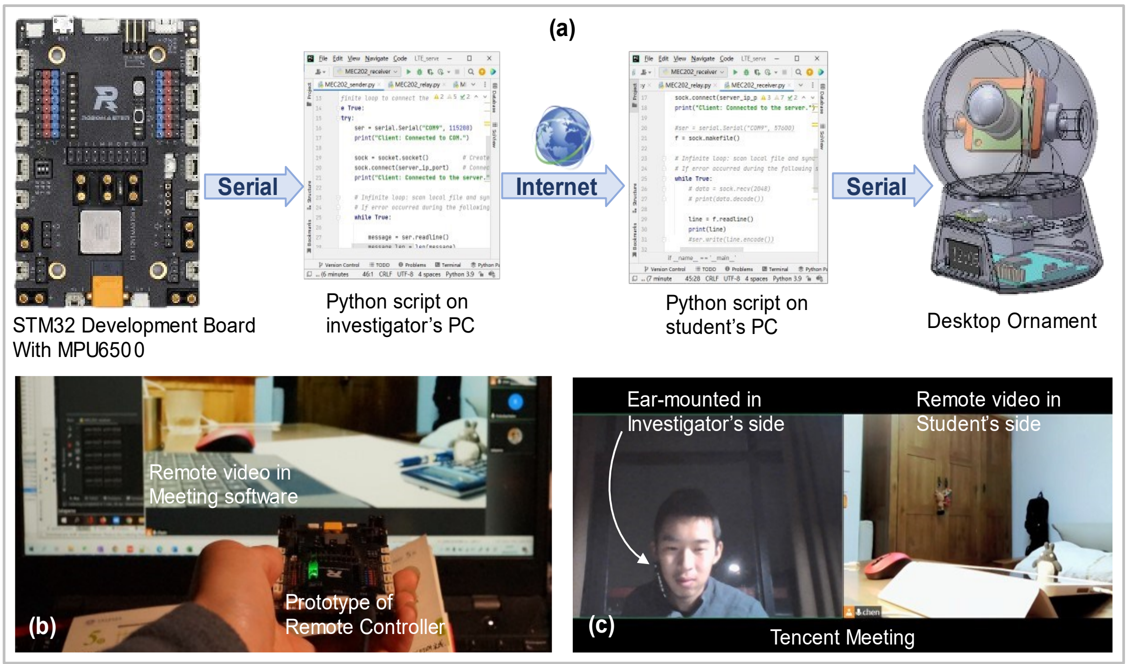


Figure 14 Physical prototype verification for the wearable remote controller, **(a)** the used development board and the dataflow, **(b)** the investigator's perspective, and **(c)** the 3rd person's perspective.

4.Conclusion

4.1 Brief Conclusion

In this group project, we designed an online invigilator. It took us nearly a month from the preliminary idea to turning it into a physical prototype. So far, the basic functions of the online invigilator have been fully realized.

4.2 Reflection

Although the design has certain practicability, there are still many problems worth discussing. The details are as follows:

◆ **Personal privacy**

For personal privacy, we believe that this problem will not exist to some extent after obtaining the consent of the monitored person. In this regard, we plan to improve our design in software part in the future, so that the monitoring person can control the camera only after obtaining the consent of the monitored person.

◆ **Realization of multi device control**

For the problem that one remote controller controls multiple devices, this can be realized, and it is not difficult for us. We only need to modify the software. However, due to financial and time constraints, we only showed one-to-one control in the end. But this does not mean that our products can only achieve one-to-one control.

◆ **Optimization**

After testing the gimbal and camera, it was observed that the video is blurred due to the shake of the gimbal.

This problem is caused by the poor stability of the servo and gimbal structure. We imagine that the gimbal made of plastic can be replaced with more solid and better-quality material, and the used servos may be replaced with the brushless motors under FOC control to obtain a higher stability.

4.3 Detail individual contribution description

◆ **Siqi Chen**

- User survey data analysis
- Poster making
- Sorting out the framework of experimental report
- Experimental report of data analysis and product function introduction
- Product modeling
- Main part of presentation

◆ **Yuxin Qian**

- Sorting out the framework of experimental report
- Construction and debugging of simulation part
- Experimental report of relevant simulation part
- Main part of presentation
- Video clip

◆ **Wenbo Yu**

- Code of simulation part
- Code of main physical part

- Construction and debugging of simulation part
- Function display video of simulation part

◆ **Jingyi Wang**

- Designed the hardware, software and mechanical structure of the remote controller;
- Completed the simulation verification for the remote controller;
- Completed the physical prototype of the remote controller;
- Implemented the python program for the communication between the desktop ornament and the remote controller through the Internet.

◆ **Nuo Chen**

- Construction and commissioning of main physical parts
- Write a report on the main physical part
- Write a summary of the experimental report and some appendices
- Measurement and photographing of relevant component dimensions
- Provide the function display video of the physical part
- Product modeling

References

[1] "LCD5110_Basic - Rinky-Dink Electronics." http://www.rinkydinkelectronics.com/library.php?id=44 (accessed May 24, 2022).

[2] "DS1302 - Rinky-Dink Electronics." http://www.rinkydinkelectronics.com/library.php?id=5 (accessed May 24, 2022).

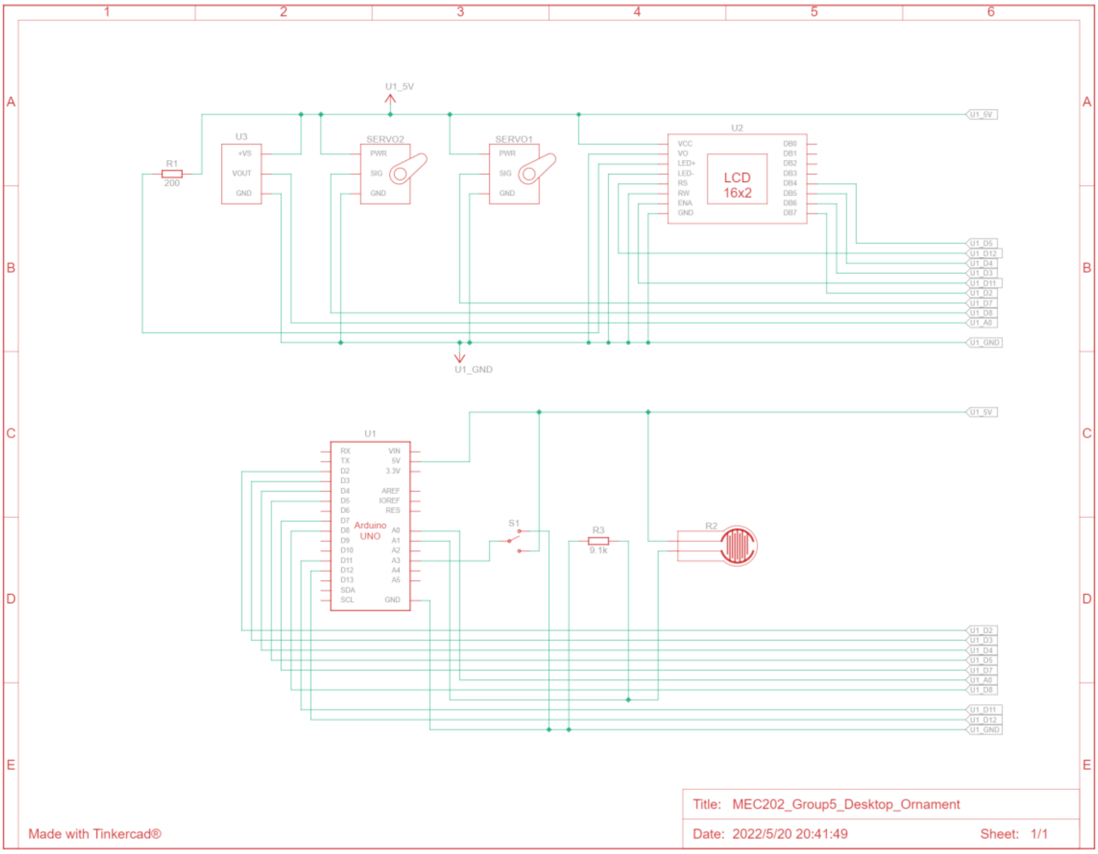
[3] "Open source IMU and AHRS algorithms – x-io Technologies." https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/ (accessed May 24, 2022).

[4] "MEC202_Group_topics.pdf." Accessed: May 24, 2022. [Online]. Available: https://www.learningmall.cn/pluginfile.php/376955/mod_resource/content/2/MEC202_Group_topics.pdf

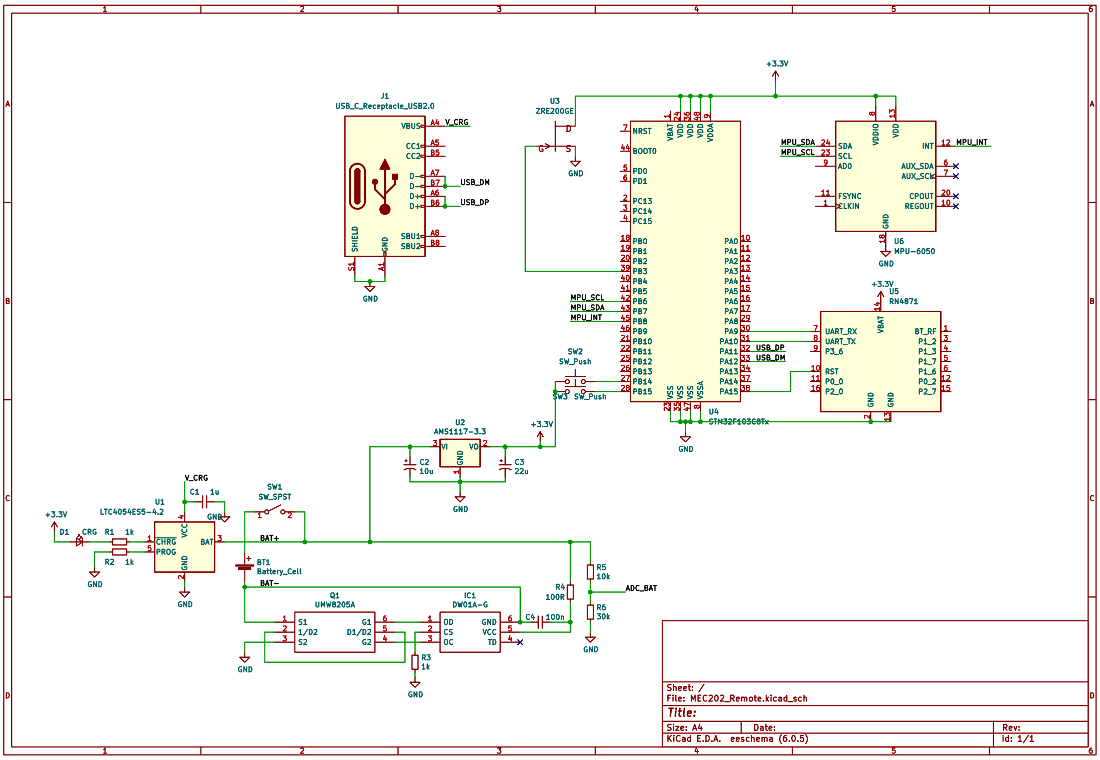
[5] "Group Project Online Platform Tutorial.pdf." Accessed: May 24, 2022. [Online]. Available: https://www.learningmall.cn/pluginfile.php/377104/mod_resource/content/1/Group%20Project%20Online%20Platform%20Tutorial.pdf

Appendix A: Schematic

a. Desktop Ornament



b. Wearable Remote Controller



Appendix B: Hardware BOM

a. Desktop Ornament

Equipment	Price (RMB)
Arduino UNO	128
Speech recognition module	68
Camera (720P 280mm)	58
Thin film pressure sensor	26
Two-axis Gimbal	43
Temperature sensor	7
5110 LCD Screen	14.2
Clock module	15.5
PIR sensor	4
Capacitor touch sensor	6.4
Total	370.1

b. Wearable Remote Controller

Equipment	Price (RMB)
CH32F103C8T6 (STM32)	16.8
Bluetooth Module	36
PIR Sensor ZR200GE	3.2
MPU6050	13
350mAh Li Battery	8
Temperature sensor	7
Resistors, capacitors, buttons, other ICs	10
Total	94

Appendix C: Group Members and contribution

Name	Student ID	Contribution
Jingyi wang	1929591	Hardware, software, simulation, physical of Remote Controller
Wenbo Yu	1929603	Software and simulation of Pomodoro and camera
Nuo Chen	1931017	Physics model and software of main part and 3D Model.
Yuxin Qian	1930038	Simulation of temperature display and video cutting.
Siqi Chen	1931101	User research, user feedback, poster. 3D Model.

Appendix D: Meeting Record

Time	Member	Meeting progress
28/04/2022	ALL	1. Discuss the project theme 2. Make plan to collect relevant information
29/04/2022	ALL	1. Make a questionnaire 2. Make a list of components 3. Preliminary discuss the product functions 4. The tentative idea of the preliminary product model
02/05/2022	ALL	1. Process the questionnaire information and analyze the needs of user groups 2. Formulate the final product function 3. Assign tasks and set the deadline of each task 4. The contents and pages of the report written by each group members were allocated in detail
06/05/2022	ALL	1. Report progress 2. Discuss the problems in the process of completing the project 3. Some modeling problems are solved 4. Discuss the writing logic and content of the report again
08/05/2022	ALL	1. The first draft of the poster is completed 2. Solve some technical problems encountered in the physical part and simulation part 3. New task assignments are made and the time for completion of each task is specified
10/05/2022	ALL	1. Discuss the problem of modeling 2. Discuss the realization of the function of the physical prototype 3. Discuss the function and idea of the simulation part 4. A new round of task assignment has been carried out and a new plan has been formulated
12/05/2022	ALL	1. Discuss video presentation content 2. Discuss each member's next task and completion time
15/05/2022	ALL	1. Discuss the content of the presentation 2. Discuss problems related to video modification 3. Discuss problems related to modeling modification
19/05/2022	ALL	1. Prepare materials for presentation 2. Complete the final draft of the poster
20/05/2022	ALL	1. Final practice of presentation

Appendix E: Partial code

a. Arduino board in Desktop Ornament

C++ code for class **Tomato** and **State**:

```
class Tomato
{
    /*** variable ***/
    private:
        int time;   // remaining time before time out
    public:
    /*** function ***/
        Tomato();
        void settime(int);  // change time
        void addtime(int);  // add time
        int countDown();    // time -= 1
        int gettime();      // return remaining time
        String getTime();   // Returns standard time in H:M:S format
        void displayTime(LiquidCrystal lcd);    // display time on lcd
        void displayTimeOut(LiquidCrystal lcd); // display 'time out' on
lcd
};
class State{
    /*** variable ***/
    private:
        int state;  // save present state
    public:
        Tomato tomato;
    /*** function ***/
        State();                // init state
        virtual String voiceFeedback(){return " ";} // interface
        void setstate(int);     // change state
        int getstate();         // return present state
        int stateNext(int cmd); // obtain next state depends on present
state and command
        int action(int cmd);    // do the action based on present state
and command
        void reset();           // state -> 0 ; tomato clear
};
```

b. Speak Recognition Module

```
#include "asr.h"
#include "setup.h"
#include "myLib/luxiaoban.h"

uint32_t snid;

void ASR_CODE()
{
  if(snid == 29){
   luxiaoban_digital_write(0,0);
```

```
  }
   if(snid == 30){
   luxiaoban_digital_write(0,1);
   Serial.println("1");
  }
  if(snid == 31){
   luxiaoban_digital_write(0,0);
   Serial.println("0");
  }
}

void setup()
{
 Serial.begin((9600));
 luxiaoban_digital_write_all(0);
}
```