ECE 445

SENIOR DESIGN LABORATORY

DESIGN DOCUMENT

Mimicry Stage Lighting Control System

Team #29

RUIQI LI
(ruiqili4@illinois.edu)
HAOZHE CHEN
(haozhe3@illinois.edu)
ZHAOHUA YANG
(zhaohua2@illinois.edu)
ANQI TAN
(anqitan2@illinois.edu)

Sponsor: Timothy Lee

March 24, 2022

Contents

1 Introduction

1.1 Problem and Solution Overview

Stage lighting technology is important in scenarios like theaters, concerts, live houses, and even home parties. Stage lighting system (SLS) [1] is used to control all kinds of lanterns and related equipment to perform artistic lighting to achieve specific effects. Current SLSs are all designed for large stage performance, related protocols are also very mature. However, under some smaller and amateur circumstances, like home parties or discos, a complete SLS may encounter some problems:

1. It's too unwieldy. A complete SLS consists of lanterns, cables, dimmers, and a control desk. Lanterns of permanent SLS should be mounted on the ground or scaffolds. Each component is connected with DMX cables [2]. This whole system is cumbersome for smaller scenarios because we only need a small portable lantern array with a user-friendly controller. Beyond that, the domestic power support system may not be compatible with professional use.

2. It's difficult to operate. An SLS needs professional stage lighting engineers to install and operate. But for home parties or small discos/bars, it's too expensive to hire a well-trained operator. Therefore, we need to reduce the cost of learning for amateur use.

3. It lacks liveness. In real practice, live manipulation and improvisation are unavoidable. DJs and performers may want the lights to move to the rhythm of the live song or just want to guide the movement of the audience's lightsticks. The preprogrammed and rigid lighting sequence cannot meet this requirement, and it's difficult for performers on the stage to operate the control desk directly. So we need a mechanism to allow the performers to control the color and direction of the light at will without paying much attention.

This project aims to build a **mimicry stage lighting control system**. It's an SLS that the light array will follow the direction of the operator's arms and imitate immediately, where the operator needs to hold two small joysticks.

More specifically, an operator (mostly a presenter or DJ) holds a joystick in each hand. The joysticks track the movement of each arm and transmit the direction signal to the central server. The central server is a PC running a control service. This signal is then processed and redirected to the portable projector light arrays to control the actual direction of beams. The direction control is implemented by a portable robot arm (gimbal).

There are two groups (arrays) on the stage, i.e., the left one and the right one. Each hand controls each side of the lighting array. The system can also memorize a period of movement and playback. Besides, the color of the lights can be manually determined or be automatically determined by the rhythm or tempo of the music being played. The color command sequence with timestamps extracted from music is precomputed before the demonstration. These changes of color and movement modes can be controlled by hitting some buttons on the joysticks.

It's worth mentioning that we will not implement the whole system according to DMX protocol [2]. There are three reasons for that: 1) the DMX devices are expensive and we don't have enough budget. 2) we mainly focus on the mimicry part, so the power supply and communication protocols only need to be similar to professional and dominant design. 3) for domestic or small-disco use, professional devices are unnecessary and inconvenient. Therefore, we will implement the communication system wirelessly and the lanterns will be designed using LEDs.

Our project fixes all the problems discussed above. The whole system consists of a group of portable directed lights, two joysticks, and a PC, which is lightweight and mobile. Also, the imitative mode of operation reduces the difficulty of control significantly, the operator only needs to wave arms with the joysticks, or hit some buttons to change the mode.

1.2 Visual Aid

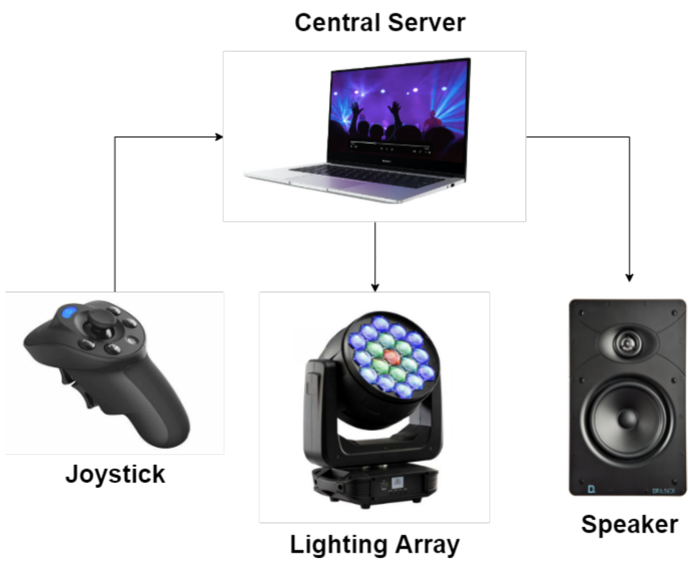

Figure 1: A pictorial representation of project

The overall system consists of four components: a pair of joysticks, an array of lights, a central server, and a speaker.

1. The joystick is a handle that can be held by hand. There are three buttons on the joystick to control different colors and movement modes. The inertial measurement unit in the joystick measures the attitude and direction information. The communication system then transmits this signal to the central server.

2. There are two arrays of lights and one hand for each. Each array consists of multiple lights (2-10 lights). Each light consists of a controllable robot arm (gimbal) and a lantern mounted on it. The gimbal has two-degree of freedom. The attitude control is implemented with the PID algorithm.

3. The central server, in our case, is a PC. It runs a service to maintain communication within a local area network, which consists of the joysticks, the lighting array, and the PC itself. It also runs a service to compute the music analysis algorithm. The color command sequence will be extracted from the song being played and therefore controls the color variance of the lights.

4. The speaker is just for demonstration purposes. The music being played is an important element in our design, so we need a speaker to play it. For simplicity, it's the PC's speaker or a peripheral.

1.3 High-level Requirements List

1. Precision. For a good mimetic stage effect, the orientation of lights should have a small difference from joysticks. Temporally, the delay between the movement of the operator and the response of the light array needs to be less than 100ms. Spatially, the presented direction of projector light (or more specifically, the gimbal) should be approximately parallel to the actual direction of the operator's arm, with less than 10° in any instance.

2. Portability. The system should be friendly for carrying and installation for personal users. Therefore, it should have a reasonably small size, light weight, low energy consumption, and be wireless connected. The program of the central server should be easily installed and executed on a popular operating system, like MS Windows 10/11 and Apple macOS. The minimum physical components configuration of the system (two joysticks and two lights) should be able to contain in a 40L bag and less than 10 kg. To ensure the flexibility of installation, peripherals (joysticks and lights) should be physically isolated and be charged by batteries. Endurance of 60 minutes for each device under continuous usage should be guaranteed.

3. User-friendliness. The system should do the job of a professional stage lighting engineer while letting the user control it with simple commands. Firstly, the user should be able to control the orientation of the light intuitively. The light array should have an alternative waving mode, like imitating the joysticks or randomly moving, and can be switched by simply clicking buttons. The color of light will also be automatically encoded. The normal and stable behavior of the music analysis system takes responsibility for this feature. This requires the complexity of the music analysis algorithm to be low enough. However, since the color command sequence is precomputed, a certain amount of time of computation is acceptable. As a high-level requirement, we need the algorithm to work within a time limit of 1% of the length of the song.

2 Design

2.1 Block Diagram

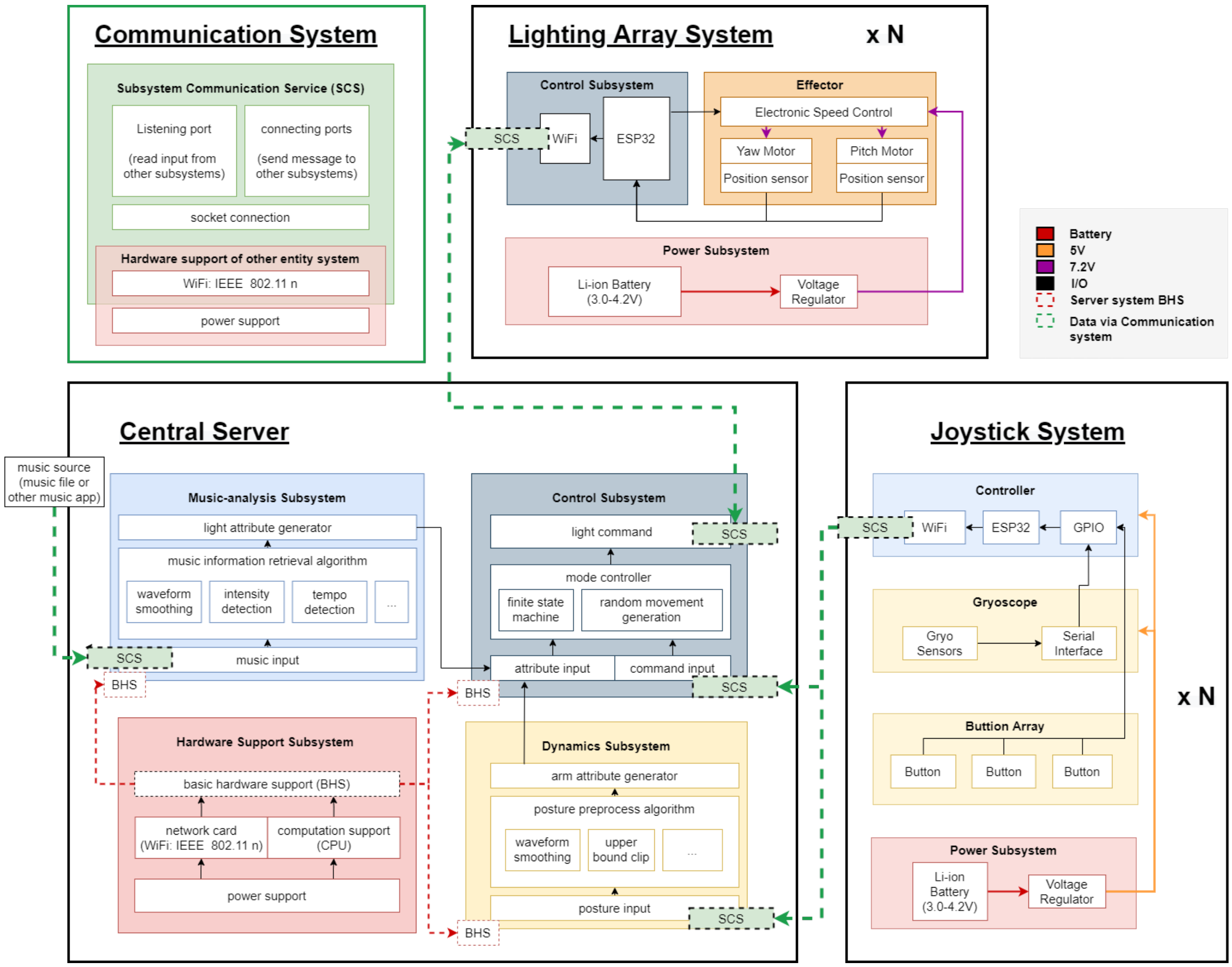

Figure 2: System Block Diagram

2.2 Subsystem Description

2.2.1 Joystick Subsystem

The power system should supply 3.3V with a max current greater than 400mA. The MPU-6050 has a built-in gyro sensor, which can automatically detect the value in pitch, rolling, and yaw axis. After conditioning, the signals are encoded and transmitted to ESP32 via the I2C interface. The button array module includes buttons, which control the working model of the robot arm. For example, if we want to record the action of the arm and repeat that, we just need to press the start button and the recording will start. If we want to stop that process, just press the stop button. After pressing the third button our robot arm will repeat the recording action. The power subsystem starts with the 3.0-4.2V battery, and then goes through the battery management system (BMS), which is a rectifier, and finally a voltage regulator. The power applies to the controller. All signals that go into the controller are encoded and transmitted to the central server.

The definition of all interfaces between blocks are as below:

ESP32 and MPU-6050: MPU-6050 provides output data on an I2C bus. We can use the I2C bus interface of MPU6050 to transfer a 3-axis accelerometer and 3-axis gyroscope values to ESP32. There is a specific dedicated address assigned to each parameter value in the MPU6050 I2C interface. Therefore, we can use a while loop and read from these addresses to get positions from time to time. For the detailed connection between ESP32 and MPU6050, we only need VCC, GND, SCL, SDA, where VCC is for power, GND for ground, SCL for a serial clock, SDA for serial data. The I2C bus is composed of SCL and SDA. The VCC pin is connected with the 3.3V from the ESP32 module to power up. Both the grounds of the two devices are connected in common. The SCL pin of MPU6050 is connected with the default SCL pin of ESP32. Likewise, the SDA pin is connected with the default SDA pin of ESP boards.

ESP32 and computer: When debugging the program that is burned into the ESP32, we use the UART protocol and can monitor the program running in the ESP32.

ESP32 and button array: We only need to PIN to use the button. One is the PIN that can always output high voltage and another PIN that receives input. If we connect these two PINs then if we press the button, input high voltage, otherwise low voltage. The signal from the input PIN can determine whether we need to record the motion or repeat it.

ESP32 and power subsystem: The ESP32 and power subsystem is connected by a transformer to regulate the voltage.

2.2.2 Lighting Array Subsystem

Lighting Array System consists of several independently controlled two-axis robotic arms. Each two-axis manipulator has three parts: control subsystem, manipulator subsystem, and power subsystem.

The control subsystem has an ESP32 chip that receives instructions from the Central Server over WIFI. The YAW and PITCH axis of the manipulator subsystem is controlled by two stepper motors respectively. While receiving the speed control signal from the ESP32, the two stepper motors collect the position signal from the position sensor and transmit it to the ESP32. EPS32 uses PID signal processing technology to control closed-loop feedback based on electrical signals to ensure the accuracy and stability of the manipulator attitude.

The power supply of each manipulator was obtained from Li-ion Battery. The current passes through the Battery Management System (BMS) to prevent backflow and then through the voltage regulator for waveform management.

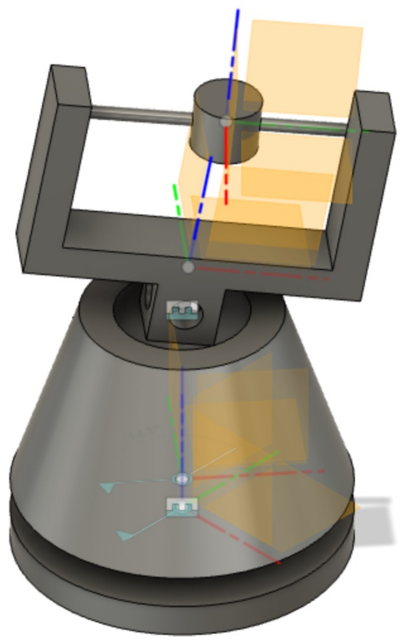

Figure 3: Conceptual sketch of the robot arm

Each of the lighting control systems is controlled by a two-axis robotic arm. The general concept of the two-axis manipulator is shown in Figure 3. At the bottom of the head holder, a brushless DC motor with large torque directly controls the rotation of the Yaw axis to realize the relative rotation of the BASE and the overall gimbal. By cross roller bearing to undertake base and upper gimbal. Through the steering gear to achieve pitch shaft only swing, greatly improve the working space of the manipulator.

Connect the outer ring of the base cross roller bearing, and connect the motor shaft and the inner ring of the cross roller bearing with the BRUShless DC motor. At the same time, the cross-roller bearing inner ring and the "neck shell" are connected to each other through copper columns to form a platform. And connect the extension of the electric slip ring to the neck shell to ensure that the wire will not be wound because of continuous rotation. The only ESP32 chip of the whole manipulator is connected to the base part, so the power supply and control signal of the LED lamp as well as the assistant and control signal of the steering gear need to be wired through the electric slip ring.

2.2.3 Communication Subsystem

The communication subsystem is a set of abstract software that handles all the connections between subsystems. To meet the precision and portability in high requirements, stable and robust communication via a wireless network is essential. The quality of the wireless network and the number of clients (lambs and joysticks) can vary with physical configuration. To cope with the effect caused by executing environment, a set of software is developed to adapt different platforms to guarantee the robustness and stability of the connection, as well as the low latency.

This subsystem is based on the hardware of its host devices, in this project context, the ESP32 development boards, and the PC central server. Specifically, the host device should provide a stable power supply, which supports the normal behavior of the network card, and operating system. The network card should allow a device to connect to LAN with IEEE 802.11n protocol and the operating system should support basic network I/O computation. The wireless environment can be established by an additional WiFi router or let the central server play the role of the router. With those infrastructures, TCP/IP and socket protocol can be implemented and each subsystem can establish its communication service. There will be a listening port in the service that receives all messages sent to it and many other ports can be utilized to connect with other subsystems.

2.2.4 Central Server Subsystem

The central server subsystem, in our case, is a PC. However, this subsystem is just a physical concept. To be more clear, each service running on this central server can be treated as an individual subsystem. Generally, these different services are **music analysis subsystem**, **dynamics subsystem** and **control subsystem**. To reduce redundancy, these subsystems will be discussed individually below.

2.2.5 Music-analysis Subsystem

The music analysis subsystem is one of the services running on the PC (or the central server). The music analyzer takes the music signal as input and generates a color control signal as output. For simplicity, the system precomputes the command sequence before the play is started (that is, the operator first determines which song to play, then the music analyzer generates a complete color and movement command sequence. This sequence is then ready to be transmitted to gimbals to control the color and direction), rather than being a real-time computation.

To be more specific, the music analyzer consists of two parts: analysis and synthesis. The analysis part extracts latent features from the music. The synthesis part takes the features extracted as input and synthesis a corresponding color command sequence with timestamps using a heuristic formula.

Music Feature Analysis

Currently, the analysis part needs to extract 4 features: tempo, beat sequence, and energy sequence. The music signal is sampled at $sr = 22050$ Hz.

Tempo information and beat sequence can be extracted together. This part refers to a well-established algorithm, the beat tracking algorithm by dynamic programming [3]. This algorithm firstly calculates the onset envelope of the input songs. The ”onset strength envelope” is a sequence of time, which indicates at what times it has a large chance to be an actual beat. Daniel uses a crude perceptual model on the Mel spectrogram (spectrogram according to Mel scale [4]) to generate this onset envelope. It’s worth mentioning that the

Mel spectrogram is generated with hop length 512. Therefore, the resulting beat sequence is sampled at sample rate $\frac{sr}{512} = 43.07$ Hz.

Once the onset strength envelope is generated, a global tempo estimate can be made (it's worth mentioning that Daniel's algorithm assumes a constant tempo through the music segment):

$$TPS(\tau) = W(\tau) \sum_t O(t)O(t-\tau) \tag{1}$$

where $TPS(\tau)$ indicates tempo period strength, $O(t)$ is the onset strength envelope, $W(\tau)$ is a Gaussian weighting function on a log-time axis:

$$W(\tau) = \exp\{-\frac{1}{2}(\frac{\log_2 \frac{\tau}{\tau_0}}{\sigma_\tau})^2\} \tag{2}$$

where τ_0 is the center of the tempo periods, and σ_τ controls the width of the weighting curve in octaves. The log scale is intended to transform the time axis into an octave measure. τ is then the global tempo prediction, i.e.,

$$\hat{\tau} = \underset{\tau}{\operatorname{argmax}}\, TPS(\tau) \tag{3}$$

Eventually, we can pick peaks in onset strength approximately consistent with the estimated tempo and generate the predicted beat sequence. An example beat tracking demonstration implemented using `Python` library `Librosa` is shown as Figure 4.

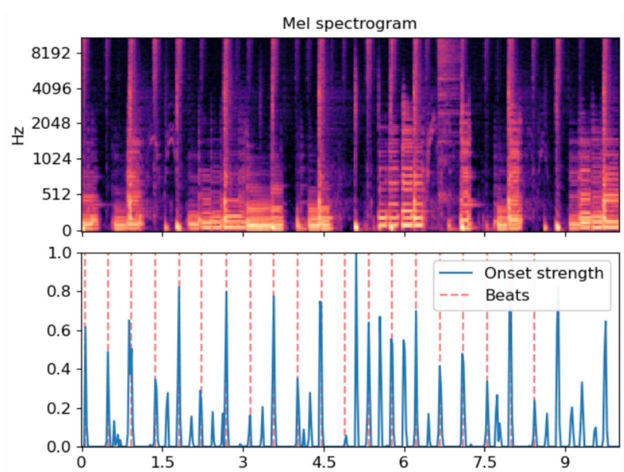

Figure 4: Beating tracking demonstration from `Librosa` [5]

Energy sequence is relatively easy to generate. According to Parseval's Theorem, the total energy of a segment of sequence $x[n]$ is:

$$E = \sum_{n=n_1}^{n_2} |x[n]|^2 \quad (4)$$

Therefore, to compute an energy-time curve, we slice the music signal into small frames, with hop length 512 and frame length 2048 (so the sample rate of energy curve is also $\frac{sr}{512} = 43.07$ Hz). For each frame, we compute the root-mean-square as the corresponding effective energy. The energy curve can be shown as

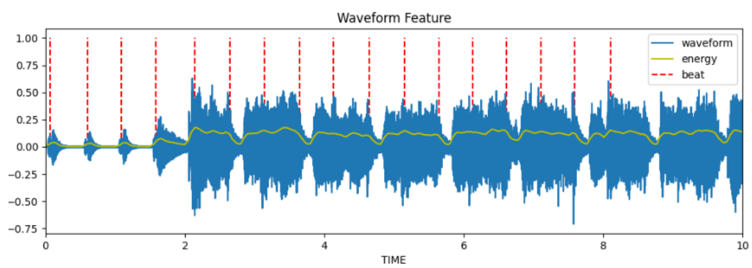

Figure 5: Demonstration of energy curve of a sample segment.

Color Sequence Synthesis

The color sequence synthesis part is a heuristic formula that maps the music feature derived before to color information. A simple approach is HSV color model [6]. HSV model measures color in three metrics: hue, saturation, and value. Hue is measured with angle (0 - 360°). Therefore, a simple heuristic mapping is: the HSV value has a linear relationship with the energy per frame, while beats have a slightly smaller impact on it.

To implement a natural effect of beats, we need to smooth the beat sequence. We convolve a Hamming window with the beat sequence to produce a smoothed beat curve. The Hamming window has a length of 80% of the length of the average beat gap. This can be represented as

$$\tilde{b}[i] = b[i] * h[i] = \sum_{m=-\infty}^{\infty} b[m]h[i-m] \quad (5)$$

where $b[i]$ is the beat sequence, which equals 1 if there is a beat at that time point and 0 if not. $h[n]$ is the Hamming window. Therefore, $\tilde{b}[i]$ is the smoothed beat curve. A demonstration can be shown as Figure 6. We use i as the index here because the sample rate is 512 times the sample rate of the music.

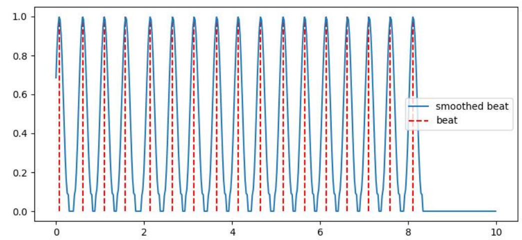

Figure 6: Smoothed beat curve by convolution with Hamming window.

With these features ready, a simple heuristic formula can be designed:

$$C[i] = E[i] + 0.3 \times b[i] \tag{6}$$

$$C[i] := 360 \times \frac{C[i]}{\max(C[j])} \tag{7}$$

Where $C[i]$ represents the color sequence with sample rate $\frac{sr}{512} = 43.07$ Hz. $E[i]$ is the energy curve with the same sample rate. The second formula above normalizes the value into the range of 0 - 360°.

Finally, using the algorithm provided by Smith [6] we can map the HSV hue value to RGB values. A demonstration can be shown as

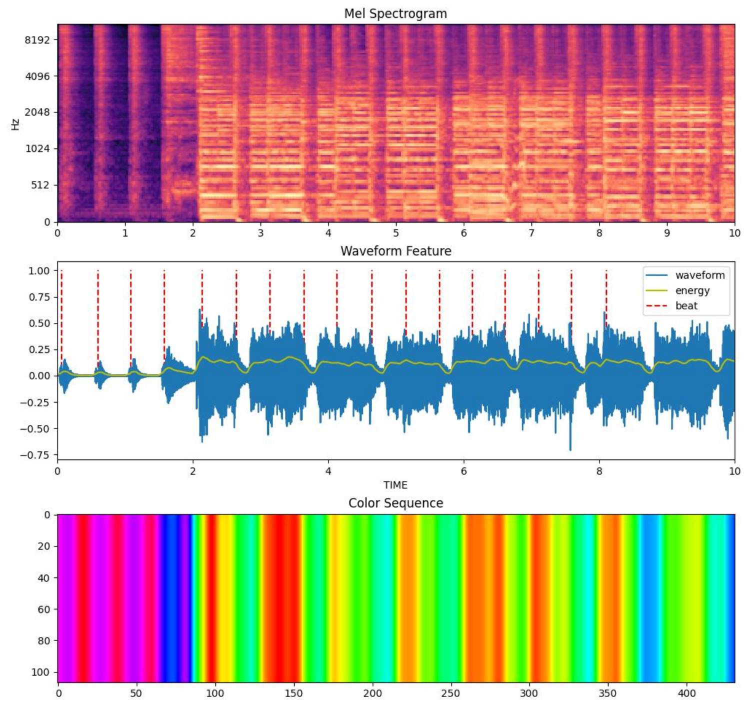

Figure 7: An example of color sequence output

Interface with Control Subsystem

After precomputation, the color sequence will be saved on the disk. The whole data file is represented as an Nx3 matrix: there are N rows of sample points with a sample rate of $hop\_length \times sr$ and 3 columns, which are respectively red, green, and blue values. The RGB value should be rounded to integers since the lanterns only support 256 tiers each. At the beginning of this file is the final sample rate. The control service can easily read the data file and generates a color command sequence.

Software Flowchart

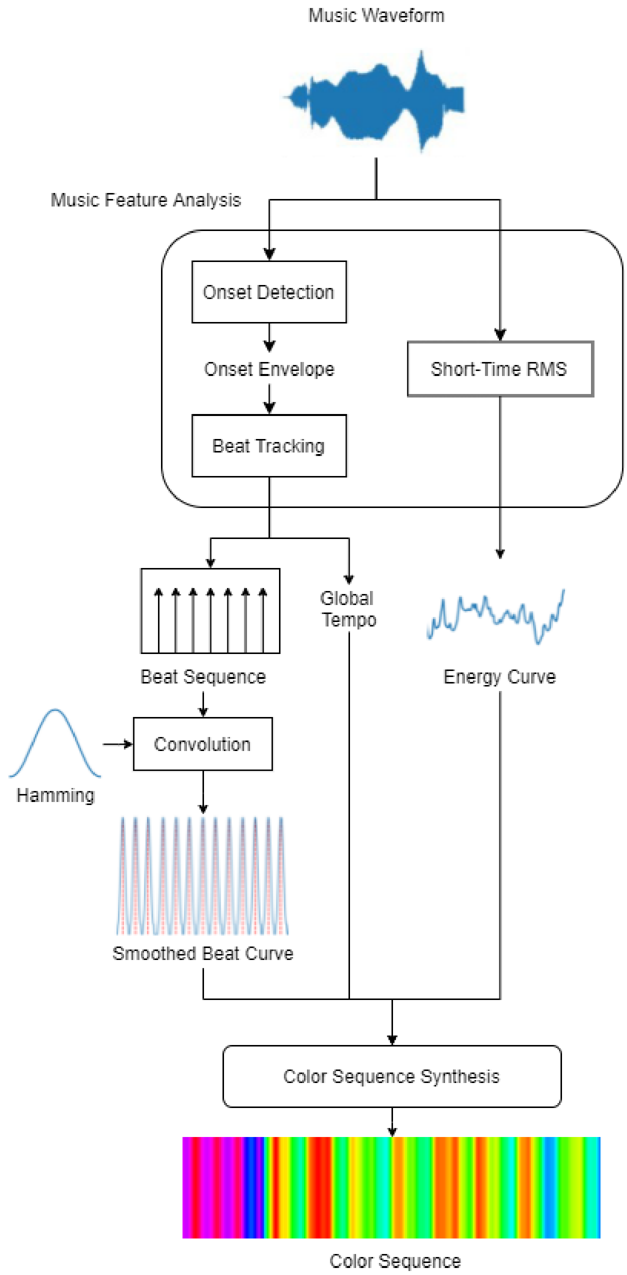

Figure 8: The overall flowchart of the algorithm.

2.2.6 Dynamics Subsystem

The dynamics subsystem is a set of programs that handles the reception, processing, and remapping of posture data to the movement of the arm. The programs need to design efficiently enough to satisfy the precision in our high-level requirements. The precision involves both time and space, which means that our robot arms can react to the posture of the operator with negligible delay and get position data with high accuracy.

This subsystem is a program executing in the central server, which acts as a preprocessor of received movement data. The communication is based on the communication subsystem introduced before. Although the communication subsystem will recover the broken connection as soon as possible, the received posture data and button command would probably lose messages during the failure period and would cause an unpredictable mess. Hence, the system should is to predict and fill missing values using known values and discard values that are far away from the mean. The data sequence should also be smoothed using a popular algorithm like Kalman filter to avoid imitating fierce movement. The domains of the values of the three axes are considered as well. Limited by the physical structure of lights, the servo should not reach some extreme angle like 0 or 359 degrees. Therefore, the upper bound clip is used to cut values that we could not reach and add a buffer effect when motors reach boundaries. Finally and core function of this subsystem is to convert the posture data of joysticks to a movement control command of the gimbal of the light array. Specifically, the data from the joystick system consists of the values from X, Y, Z coordinates and we need to map them into coordinates that the stepping motor can use. Our stepping motor only has 2 degrees of freedom. Ideally, it is just a mapping from a Cartesian coordinate system to a spherical coordinate system. In the end, we should pack what's mentioned above into one function and give this function to the control subsystem. The only thing that the control subsystem need to do is to call this function and get coordinate for stepping motors from time to time.

2.2.7 Control Subsystem

This subsystem plays the role of the synthesizer of the light array control command. It receives the user's button control command from joysticks, the color and intensity attribute from the music analysis subsystem, and the arm angle attribute from the dynamic subsystem. Those attributes along with the music source file are the components of the system control frame. The mode controller in the subsystem acts as a filter and combiner of the control frame. The random generator will provide random movement and color control signals with the same format as that from other subsystems. Two inner finite state machines are also been maintained in the controller and filtering the coming command. Three buttons' command from the joystick will be sent to the control subsystem and separately controls the transition of states as shown in the Figure 9. A single click on the movement mode button will cause successive transitions of movement mode state, in order of imitation, repeat, and random. If the state machine is in imitation mode, the user can record its posture by holding the record button. This period of posture will be played repeatedly in the repeat mode. The color mode can be changed similarly by singly clicking the color mode button. After filtering, the final movement command, color command, and the music will be combined into a control frame and sent to speak or a lighting array. The data flow of this subsystem can be found in Figure 10.

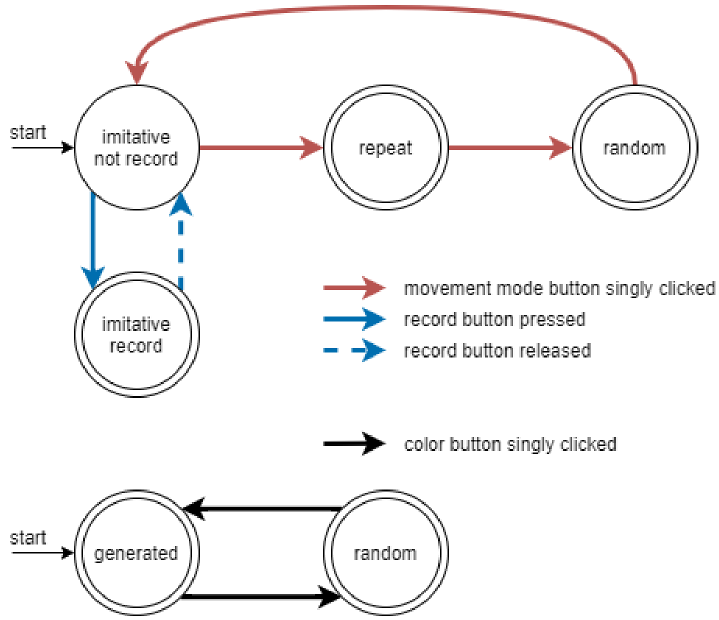

Figure 9: Finite state machines in mode controller

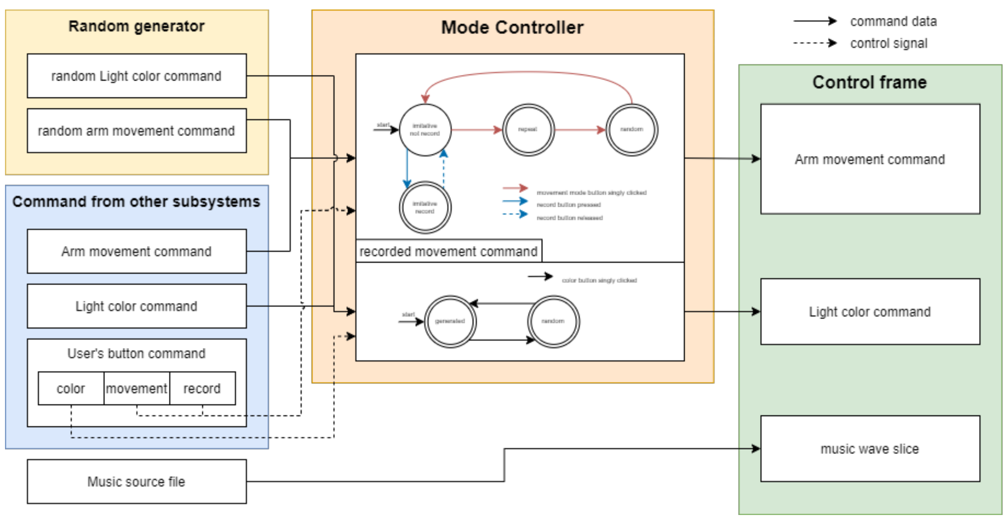

Figure 10: Dataflow chart of control subsystem

2.3 Requirements and Verification

2.3.1 Joystick Subsystem

Requirements	Verification
MPU6050 1) The mechanical frequency of gyroscope is X_Axis: 33kHz, Y_Axis: 30kHz, Z_Axis: 27kHz	1) We need to make sure that when DJ is waving his arm, the robot arm can reach every position of the DJ's position with the minimum delay.
ESP32 1) work at about 3.3V with a minimum current 0.5A.[7] 2) Button should be debounced and when pressed or released. 3) The switch of the ESP32 should be able to immediately control the chip. 4) we should avoid the situation where ESP32 stop working due to the lack of voltage.	1) A multimeter will be used to check if the voltage and current satisfy the requirements. Use an oscilloscope to check if the voltage signal is steady. 2) Voltage measured from the pin connected to the button behaves as a digital active-low signal when buttons are pressed or released (pressed=LOW, released=HIGH). Using oscilloscope to measure the bounce time. If a person press a button and quickly release, the bounce time is less than 10ms, the debounce is successful. 3) In practice, if we need to quickly reset ESP32, and there is capacitance in the peripheral, the voltage might not go to 0 immediately after we cut off the switch. Hence, if we start again the chip will not be thoroughly reset. we should add an extra discharge circuit to accelerate the discharging of the capacitance. If we try 100 times and the reset button works every time, the switch works. 4) we need a power supply supervisor, to lower the CHIP_PU if detect that the voltage is less than 2.3V. Then we just see if ESP32 will shut down when working.

2.3.2 Lighting Array Subsystem

Requirements	Verification
Servo for pitch shaft on the gimbal	**Dsservo 360° continuous rotation servo**
1) High control progress and good linearity can achieve a fast response. 2) Given a PWM pulse width, the SERVP output Angle can be locked, reducing the delay for arm movement following. 3) It has a large working Angle. In addition to mechanical foot collision, it can restore human arm movements as much as possible.	1) Using the digital servo rather than the analog servo, the minimum control Angle is below 0.9°. 2) The stability of the servo is high. The internal servo control board of the servo is controlled by MCU. 3) Choose servo with motion Angle over 360 °.
Bearing with bearing force at the bottom	**crossed roller bearing**
1) It can absorb more than 300kN vertical pressure while maintaining smooth rotation.	1) Because the roller is cross arranged, so only 1 set of cross roller shaft rings can bear the load in all directions, compared with the traditional model, the rigidity is improved 3 to 4 times.
2) The inner and outer rings can be fixed with the bottom and head respectively in a relatively simple way, while saving the installation space.	2) Cross roller bearing, because the divided inner ring or outer ring, after loading the roller and spacer, is fixed with the cross roller shaft ring to prevent mutual separation, so the operation is simple when installing the cross roller shaft ring.
3) High speed and excellent rotation accuracy, full softening.	3) Because the inner ring or outer ring of the cross roller bearing is two split structures, the bearing clearance can be adjusted, even if the preload is applied, it can also obtain high-precision rotating motion.

Requirements	**Verification**
Linearly varying LED lamp array 1) Intelligent back connection protection, power back connection will not damage IC. 2) The control circuit and RGB chip are integrated in a 5050 package of components to form a complete external control pixel point. Built-in signal shaping circuit, any pixel received, after waveform shaping output, to ensure that the line waveform distortion does not cloud accumulation. 3) The three primary colors of each pixel point can achieve a 256-level brightness display and the scanning frequency is not less than 400Hz/S. 4) Data can be received and decoded through a single signal line. 5) Can directly connect to DMX512 lighting control protocol.	**LED ws2812 rgb development board** 1) Built-in power-on reset and power-off reset circuit to protect the circuit. 2) Naked eye observation of light waveform changes following the change of music rhythm. 3) The optional LED array refresh frequency is 30 frames/s, and the number of cascades is not less than 1024 points. 4) The transmission speed of the selected data line can reach 800KBps, which can be monitored through the data transmission port of the computer. 5) RS-485 bus interface is required.

Requirements	Verification
Motor for Yaw shaft at the bottom of the gimbal	**RoboMaster 6623 gimbal motor**
1) Since the robot arm needs attitude calibration and PID closed-loop control to improve the accuracy of the following attitude, the motor needs built-in sensors to carry out position closed-loop control.	1) The motor has a built-in position sensor, which CAN obtain rotor position, rotor speed, motor temperature, and other information through the CAN bus. Position feedback accuracy 13bit. Robomaster motors are equipped with Hall sensors that provide feedback on motor speed, position, and other information for closed-loop control.
2) To keep up with the east speed of the human arm, the maximum speed needs to be greater than 120rpm.	2) The motor is A brushless DC motor with A torque constant of 0.38nm /A. The speed constant is 25RPM/V. Angle control accuracy is ±0.05. The voltage requirement is 24V, which can be powered by batteries and the idling speed is 600RPM. Blocking torque 2N/m.
3)The Yaw axis, as the chassis bearing axis, has no synchronous belt and gear in the current design and carries the force directly through bearings and motors. Because the total amount of head is 0.5kg-2.5kg at present. Therefore, the motor rotating shaft can undertake more than 50kN force in the shear direction.	3) RoboMaster 6623 Gimbal Motor has been purchased, which can weigh more than 80KN on D-rod.
4) The application places of this product include home parties, small disco bars, etc. Personnel interaction is complicated, and the head will be manually fluctuated or reversed. The motor selected for this product should be able to rotate when power is off, and there is no limit on the direction of rotation when power is supplied.	4) brushless dc motor without the use of traditional mechanical structure has a brush motor brush, but through the electronic commutator to realize reversing, compared the performance of traditional motor has some advantages, generally using a brushless dc motor requires matching electrically adjustable, by changing the electrically adjustable output current size and direction, and rotational speed of the motor can be changed. When the power is off, it can be rotated freely without causing damage to the motor.
5) The volume of the whole lighting system can be controlled by the small overall volume of the motor and electric modulation.	5) The integrated design of the motor and electric modulation integrates the high-precision absolute encoder, which greatly improves the accuracy of position feedback and torque output, and reduces the space occupancy rate and circuit complexity.

2.3.3 Communication Subsystem

Requirements	**Verification**
WiFi router: Wireless environment 1) The router should support IEEE 802.11n protocol 2) The wireless signal should be strong enough for maintaining stable communication. The typical demonstration environment can be simplified as an 8m*8m*3m room containing all devices and the WiFi router. The router should ensure the bottom threshold of WiFi signal strength of all devices is higher than -70dBm in RSSI (Received Signal Strength Indication)	1) My PC is used to establish the wireless network. The network card: `Intel Wi-Fi 6 AX200` support IEEE 802.11n IEEE WLAN standard and can be used as router as mentioned in [8] 2) The ESP32 development is used for signal strength detection. 1. Burn the Micropython image [9] to the ESP32 development board via Thonny. 2. Code the program with network package [10] in MicroPython for recording WiFi signal strength and alarm if the connection break or WiFi signal strength is less than the threshold. 3. Put the ESP32 and the router into an 8m*8m*3m room. Start the WiFi router and battery-charged ESP32 development board. Connect the ESP32 with the router via WiFi. 4. Hold the ESP32 board, start the detect program and walk around the room. The speed should be slower than 0.5m/s to imitate the fixed client in the usage scenario. The board's movement trace should cover most of the room space. 5. Analyze the signal strength record to see whether signal strength reaches a point below the threshold.

Requirements	Verification
PC central server 1) The central server should provide a network card which supports IEEE 802.11n protocol 2) The operating system of central server should support network and parallel programming.	1) My PC is used to establish the wireless network. The network card: `Intel Wi-Fi 6 AX200` support IEEE 802.11n IEEE WLAN standard and can be used as router as mentioned in [8] 2) All central server programs will be developed and tested on my PC (with win11). All of the programs will be coded with Python (version $\geq$ 3.7) and the final version of programs will be tested on popular operating systems, like Win10/11 and macOS Monterey to ensure universality.
ESP32 development board 1) The development board should provide hardware that supports the IEEE 802.11n protocol. 2) The development board should support network I/O programming.	1) According to [11], the ESP32 Wi-Fi Radio and Baseband support the 802.11 b/g/n IEEE WLAN standard. 2) The MicroPython is chosen for embedded development on ESP32 and the default networking module of it allows us to develop WiFi and socket programs on ESP32 [10].

Requirements	Verification
Communication stability 1) The service on a central server should about to maintain at least 32 connections concurrently and handle them in parallel. 2) The delay between two devices should be limited to 30ms. 3) The connection failure can be detected in 100ms and new connection will be automatically established for repairing.	1) Threading programming is utilized in central server programs to make parallel execution possible. A front-end application will be developed to monitor the running threading pool. 2) The socket communication will be established on TCP protocol. The Network Time Protocol (NTP) method in the TCP protocol takes responsibility for time synchronization. The delay can also be manually tested. For example, let devices attach a local timestamp to messages and calculate the difference between the local timestamp and received timestamp. 3) There will be a daemon program of communication subsystem that will always try to connect to the configured wireless network. The robustness can be simply verified by taking the device out of the room then back or turning off the router for a while to see the alarm signal on the devices.

2.3.4 Music-analysis Subsystem

Requirements	**Verification**
Music Feature Analysis Here are the parameters of each feature extraction algorithm. Those not mentioned remain default. 1) Beat tracking algorithm. We use `librosa.beat.beat_track()` to generate initial beat sequence. This function returns the beat sequence and the estimated global tempo. 1. Firstly generate onset strength envelope using `onset_strength()` in `librosa`, which is one of the input of `beat_track()`. 2. `sr` is the sample rate of the music, which is default to be 22050 Hz. 3. `hop_length = 512`. This value can vary from 256 to 1024. 2) Energy curve. We compute the root-mean-square curve using `rms()` function in `librosa`. 1. `frame_length = 2048`. This value can vary from 1024 to 4096. 2. `hop_length = 512`. This value is better to be the same as the one in beat tracking. 3) Smoothed beat curve. We use `numpy.convolve` to achieve convolution between the original beat sequence and the Hamming window. 1. The hamming window is implemented as `numpy.hamming(0.8 * beat_gap)`. 2. `beat_gap` is the average interval between each beat: `beat_gap = 60 / tempo * sr / hop_length`. 3. the `mode` of convolution should be "same" in order to keep the same shape.	1) While implementing, explicitly set those arguments with variables. The estimated global tempo should be within a reasonable range: 40 - 208. One can directly listen to the music to see if the estimated tempo is reasonable. 2) The energy curve is one of the measurements of how "rousing" the music is. So it should accord with the overall amplitude of the music. If the song gradually gets louder, the energy level should increase correspondingly. 3) The output shape of the convolution should be the same as the input beat sequence. One can also easily check the smoothness of the curve.

Requirements	Verification
Color Sequence Synthesis 1) The parameters of the heuristic color sequence synthesis formula are all shown in the description part. It's worth mentioning that this formula is highly personalized, the current one is still experimental.	1) The output HSV hue value should range from 0 to 360°. 2) A user may install `Anaconda` and relative packages on a PC with Windows 10 and install the environment requirements. Run the `main.py` file we provide and check if an output file is generated. The first line of the file should be the sample rate, followed by an Nx3 matrix, which is the color sequence generated.
Development Environment 1) We use `Python` as development language. We recommend `Anaconda` as the package and environment manager. The environment is constructed on a PC with Microsoft Windows 10 operating system. 2) To specify the environment and version of libraries, we use: 1. `python==3.7`. 2. `librosa==1.18.5`. 3. `numpy==1.18.5`. 4. `ffmpeg`. We recommend installing this using `conda-forge`.	1) The OS type of PC and the configuration of `Anaconda` is easy to verify. If you configure the environment variable for `Anaconda`, you can easily launch environment by typing `conda activate` in `Anaconda prompt CMD` or `power shell`. 2) Firstly activate the corresponding `Anaconda` environment in the `Anaconda prompt`. Then type `conda list` to generate all the packages installed and check if the requirements are met.

2.4 Tolerance Analysis

1. The wireless communication generates inherent delay between each component. The computational center and control system also produces potential delay. These factors jointly contribute to a significant overall delay. However, a certain amount of delay is tolerable, since this whole system is used in scenarios where the human audience is the subject. Therefore, as long as the delay is not noticed by a human observer, it's not important. For example, with the music playing, a 500 ms delay is tolerable.

2. Due to the existence of the base and its lack of freedom, the reachable angle of the two-axis mechanical arm of this product is smaller than that of the real human arm. The acceptable result is that the workspace of the manipulator meets the stage/room lighting coverage.

3. At the same time, because of the singularity of the mechanical arm, the mechanical arm may be stuck after reaching a certain angle and unable to move, thus leading to product damage. An acceptable result is to avoid the attitude of the manipulator en-

tering the singularity by setting up a specific mechanical protection structure. While some workspace is sacrificed, structural components are protected to the maximum.

4. The transmission of a signal through WIFI between modules could be energy-consuming because the information of button and action is real-time, which means the query is highly frequent. We suggest that the whole system could sustain for an hour without damaging the elements or causing danger since it is about the time for DJ to perform one show.

3 Cost and Schedule

3.1 Labor

Name	Hourly Rate	Hours	Total	Total*2.5
Ruiqi Li	¥180	300	¥54000	¥135000
Haozhe Chen	¥180	300	¥54000	¥135000
Zhaohua Yang	¥180	300	¥54000	¥135000
Anqi Tan	¥180	300	¥54000	¥135000
Total				¥540000

Table 5: Labor

3.1.1 Parts

Description	Quantity	Manufacturer	Vendor	Cost/Unit	Total Cost
ESP32-WROOM-32	4	espressif	Taobao	¥21.5	¥86
thicken bearing	4		Taobao	¥22.5	¥90
servo	2	Dsservo	Taobao	¥130.8	¥261.6
robomaster6623 gimbal motor	2	DJI	Taobao	¥898	¥1796
Total					¥2,233.6

Table 6: Parts

3.1.2 Grand Total

Section	Total
Labor	¥540000
Parts	¥2233.6
Grand Total	¥542233.6

Table 7: Grand Total

3.2 Schedule

Week	Task	Responsibility
03/14/22	**Project Proposals Due (Mon.)**	All
	Development board decision and purchase	Zhaohua Yang
	Prepare design review	Haozhe Chen
03/21/22	Preliminary analysis of music signals	Ruiqi Li
	Design Document Due (Thur.)	All
	Research and select all hardware modules	Anqi Tan
	Design Reviews (Fri.)	All
03/28/22	Purchase hardware & all parts, rough 3D modeling	Anqi Tan
	Teamwork Evaluation I (Thur.)	All
	Study datasheet for sensor, microcontroller, bluetooth modules	Haozhe Chen
04/04/22	Program microcontroller & control button	Zhaohua Yang
	Overall 3D modeling and preparation for 3D printing	Anqi Tan
	Assemble and test communication module	Haozhe Chen
04/11/22	**Individual Progress Report (Mon.)**	All
	Complete the printing, machining and assembly of non-standard parts	Anqi Tan
04/18/22	Test the lighting system and the movement of gimbal	Anqi Tan
	Test the joystick system and the connection to central server	Zhaohua Yang

Week	Task	Responsibility
04/25/22	Black-box testing on individual components	Ruiqi Li
	Overall control system code debugging	Haozhe Chen
05/02/22	**Prepare Mock Demo**	Haozhe Chen
	Confirm the achievement of the overall function	Ruiqi Li
	Finally debug the remaining problems	Haozhe Chen
05/09/22	**Mock Demo**	All
	Prepare Demonstration	Zhaohua Yang
	Prepare Prsentation	Anqi Tan
05/16/22	**Finalize Demonstration**	All
	Prepare Final report	Ruiqi Li
	Finalize Presentation (Fri.)	All
05/23/22	**Final Report due (Tue.)**	All
	Teamwork Evaluation II due Tue.	All
	Lab checkout and finalize final paper	Anqi Tan

4 Ethics and Safety

4.1 Ethics

4.1.1 Ethics of our product

Our group guarantees that the robot will not use strong light that can harm people's eyes by proper controlling of light energy and divergence.

We will reinforce the gimbal and try our best to avoid potential safety hazards caused by the falling of the light during use.

We will do our best to remind our users to avoid using lights late at night, to avoid disturbing others' sleep.

4.1.2 Ethics of research and development

Our group guarantees that the code of this product does not involve any plagiarism

Our group guarantees that this product modeling will not copy any existing modeling solutions

Our group states that this product will not use any solution of existing products on the market.

4.1.3 Ethics sources

This project is bound and inspired by the Code of Ethics published by professional societies, such as IEEE and ACM.

Members of our group will urge each other to comply with the IEEE Code of Ethics[12] and strive to ensure that the Code is followed without retaliation against individuals who report violations.

1. To hold fundamental the security, wellbeing, and welfare of the open, to endeavor to comply with the moral plan and feasible improvement hones, to ensure the security of others, and to reveal instantly components that might imperil the open or the environment;

2. To move forward the understanding by people and society of the capabilities and societal suggestions of routine and developing innovations, counting shrewdly systems;

3. To maintain a strategic distance from genuine or seen clashes of interest whenever conceivable, and to reveal them to influenced parties when they do exist;

4. To dodge illegal conduct in proficient exercises, and to dismiss bribery in all its forms;

5. To look for, acknowledge, and offer genuine feedback of specialized work, to recognize and adjust blunders, to be fair and reasonable in expressing claims or gauges based on accessible information, and to credit appropriately the commitments of others;

6. To preserve and progress our specialized competence and to attempt innovative errands for others as it were on the off chance that qualified by preparing or encounter, or after full revelation of relevant limitations;

7. To treat all people decently and with regard, and to not lock in in segregation based on characteristics such as race, religion, sexual orientation, incapacity, age, national beginning, sexual introduction, sexual orientation personality, or sexual orientation expression;

8. To not lock-in in badgering of any kind, counting sexual badgering or bullying behavior;

9. To maintain a strategic distance from harming others, their property, notoriety, or work by wrong or noxious activities, rumors, or any other verbal or physical abuses;

10. To back colleagues and co-workers in taking after this code of morals, to endeavor to guarantee the code is maintained, and to not strike back against people announcing an infringement.

4.2 Safety

Production safety - our team guarantees that we will have more than two people in the laboratory in the process of production and processing. At the same time, the students participating in the experiment have completed the online safety training of the laboratory and received certificates.[13]

Electricity safety - Our team fully read, understand, and follow the guidelines for the safe use of batteries before using them. There must be at least one person to be around the powered circuit.[14]

Mechanical structure safety - Our team choose the safest shell shape and mechanical structure. Besides, we only choose to work where there is no sharp or dangerous object.

Material processing safety - Our team pays attention to the safety of production and use when using sheet metal materials and other materials that are easy to cause damage to the human body.

references

[1] J. Primrose, *Stage lighting system*, [Online], https://www.theatrecrafts.com/pages/home/topics/teaching/lighting/system/ Accessed March 23, 2022.

[2] USITT, *Dmx protocol basics*, [Online], http://www.dmx-512.com/dmx-protocol/dmx-protocol-basics/ Accessed March 23, 2022.

[3] D. P. Ellis, "Beat tracking by dynamic programming," *Journal of New Music Research*, vol. 36, no. 1, pp. 51–60, 2007.

[4] Wikipedia, *Mel_scale*, [Online], https://en.wikipedia.org/wiki/Mel_scale Accessed March 23, 2022.

[5] LDT, *Librosa.beat.beat*$_{t}rack$, [Online], https://librosa.org/doc/latest/generated/librosa.beat.beat_track.html#librosa.beat.beat_track Accessed March 23, 2022.

[6] A. R. Smith, "Color gamut transform pairs," in *SIGGRAPH '78*, 1978, pp. 12–19.

[7] espressif, *Esp32-wroom-32 datasheet*, [Online], https://www.espressif.com/zh-hans/support/documents/technical-documents Accessed March 23, 2022.

[8] Intel-Corporation, *Intel® wi-fi 6 ax200 cyclone peak 2 product brief*, [Online], https://www.intel.com/content/dam/www/public/us/en/documents/product-briefs/wi-fi-6-ax200-module-brief.pdf Accessed March 24, 2022.

[9] MicroPython, *Micropython esp32 firmware*, [Online], https://micropython.org/download/esp32/.

[10] P. S. Damien P. George and contributors, *Quick reference for the esp32*, [Online], http://docs.micropython.org/en/latest/esp32/quickref.html Accessed March 23, 2022.

[11] espressif, *Esp32 datasheet*, [Online], https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf Accessed March 23, 2022.

[12] IEEE. ""IEEE Code of Ethics"." (2016), [Online]. Available: https://www.ieee.org/about/corporate/governance/p7-8.html (visited on 02/08/2020).

[13] OSHA. ""Occupational Safety and Health Administration"." (2016), [Online]. Available: https://www.osha.gov/ (visited on 02/08/2020).

[14] FCC. ""Federal Communications Commissions"." (2016), [Online]. Available: https://www.fcc.gov/ (visited on 03/23/2022).